Received August 30, 2021, accepted November 1, 2021, date of publication November 8, 2021, date of current version November 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125713

An Effective Bicubic Convolution Interpolation-Based Iterative Luma Optimization for Enhancing Quality in Chroma Subsampling

KUO-LIANG CHUNG, (Senior Member, IEEE), CHIH-YUAN HUANG, AND CHEN-WEI KAO
Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10672, Taiwan

Corresponding author: Kuo-Liang Chung (klchung01@gmail.com)

This work was supported by the Ministry of Science and Technology, Taiwan, under Contract MOST-108-2221-E-011-077-MY3 and Contract MOST-110-2221-E-011-088-MY3.

ABSTRACT Traditionally, prior to compressing an RGB full-color image, for each converted 2×2 CbCr block B^{CbCr}, chroma subsampling only downsamples B^{CbCr}, but without changing the luma block B^{Y} at all. In the current research, a special linear interpolation-based, namely the COPY-based, chroma subsampling-first luma modification (CSFLM) study has attempted to change the luma block for enhancing the quality of the reconstructed RGB full-color image. In this paper, a fast and effective nonlinear interpolation, namely the bicubic convolution interpolation (BCI), based iterative luma modification method for CSFLM is proposed. In our iterative method, a BCI-based distortion function and its convex property proof are first provided. Next, based on the proposed convex distortion function, a pseudoinverse technique is applied to obtain the initial luma modification solution, and then an iterative method is proposed to improve the initial luma modification solution. Based on five testing image datasets, namely the IMAX, Kodak, SCI (screen content images), CI (classical images), and Video datasets, the thorough experimental results have demonstrated that on the newly released Versatile Video Coding (VVC) platform VTM-12.0, our iterative luma modification method achieves substantial quality, execution-time, and quality-bitrate tradeoff improvements when compared with the existing state-of-the-art methods.

INDEX TERMS Chroma subsampling-first luma modification (CSFLM), convex distortion function, iterative luma optimization, quality-bitrate tradeoff, quality enhancement, RGB full-color image, versatile video coding (VVC).

I. INTRODUCTION

Prior to compression, the input RGB full-color image I^{RGB} is first converted to a YCbCr image I^{YCbCr} using the BT.601-5 transformation [8]:

$$\begin{bmatrix} Y_i \\ Cb_i \\ Cr_i \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{1}$$

where (Y_i, Cb_i, Cr_i), $1 \le i \le 4$, denotes the ith YCbCr triple-value of each 2×2 YCbCr block B^{YCbCr}, and (R_i, G_i, B_i) denotes the ith triple-value of the collocated 2×2 ground truth RGB full-color block B^{RGB}.

Conventionally, chroma subsampling only downsamples each 2×2 CbCr block B^{CbCr}, but without changing the collocated luma block B^{Y} at all. There are two chroma subsampling formats, namely 4:2:2 and 4:2:0. 4:2:0 downsamples the (Cb, Cr)-pair for each B^{CbCr}; 4:2:2 downsamples the (Cb, Cr)-pair for each row of B^{CbCr}.

In the chroma subsampling-first luma modification (CSFLM) scheme [4], as depicted in Fig. 1, after performing the chroma subsampling method, such as 4:2:0(A), 4:2:0(L), 4:2:0(R), 4:2:0(DIRECT), or Anchor [12], on B^{CbCr}, it attempts to modify each luma value in B^{Y} for better enhancing the quality of the reconstructed RGB full-color image.

A. RELATED CHROMA SUBSAMPLING WORKS

In this subsection, we briefly introduce the above-mentioned five widely used chroma subsampling methods.

The associate editor coordinating the review of this manuscript and approving it for publication was Eduardo Rosa-Molinar.

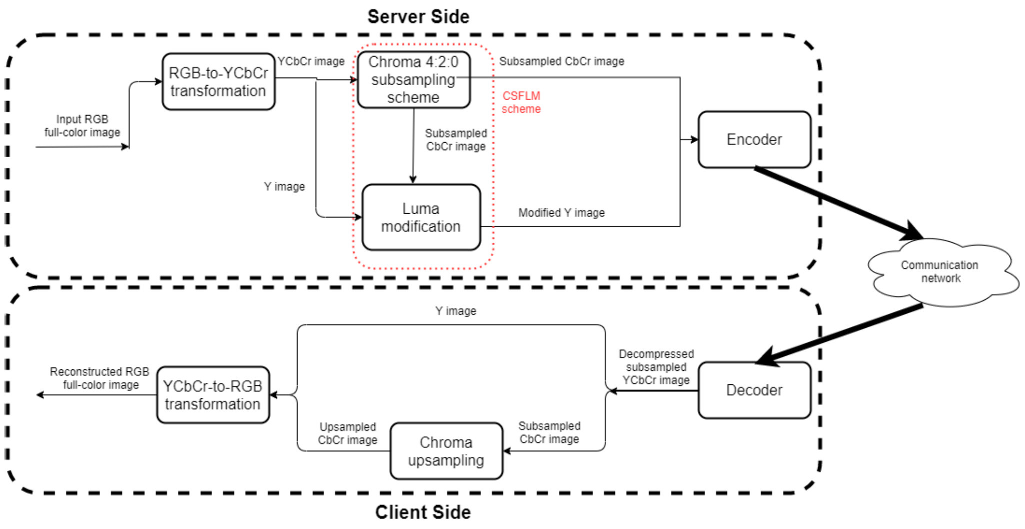


FIGURE 1. **The CSFLM scheme in the coding system.**

4:2:0(A) calculates the subsampled (Cb, Cr)-pair by averaging the four chroma pairs of B^{CbCr}. 4:2:0(L) and 4:2:0(R) calculate their subsampled (Cb, Cr)-pairs by averaging the chroma pairs in the left and right columns of B^{CbCr}, respectively. 4:2:0(DIRECT), which is abbreviated as 4:2:0(D), takes the top-left chroma pair of B^{CbCr} as the subsampled (Cb, Cr)-pair. Throughout this paper, the term "(Cb, Cr)-pair" and the term "chroma pair" denote the same thing.

The Anchor method [12] calculates the subsampled chroma pair by first performs a 3-tap filter [1, 6, 1]/8 at the leftmost location of each row of B^{CbCr}, and then performs a 3-tap filter $([0, 4, 4]/8)^T$ at the top-left entry of B^{CbCr} which has been updated by the first step.

B. RELATED CSFLM WORKS

At the server side of Fig. 1, as a preprocessing step of the CSFLM scheme [4], chroma subsampling is first performed on B^{CbCr} to obtain the subsampled (Cb, Cr)-pair, namely (Cb_s, Cr_s).

Next, a COPY-based upsampling process is applied to duplicate the chroma pair (Cb_s, Cr_s) four times for constructing an estimated 2×2 CbCr block $B^{est,CbCr}$, where each entry (Cb_i^{est}, Cr_i^{est}), $1 \leq i \leq 4$, equals (Cb_s, Cr_s). Here, the COPY-based upsampling method is exactly the nearest neighbor (NN) based upsampling method provided by the Versatile Video Coding (VVC) standard [18] and its predecessor High Efficiency Video Coding (HEVC) standard [6].

Using the ith estimated (Cb, Cr)-pair of $B^{est,CbCr}$, namely (Cb_i^{est}, Cr_i^{est}), $1 \leq i \leq 4$, and the luma modification parameter, namely Y_i', the ith RGB full-color pixel-distortion function $PD(Y_i')$ is delivered to measure the sum of square errors (SSE) between the ith ground truth RGB full-color pixel (R_i, G_i, B_i) and the ith estimated RGB full-color pixel $(R_i^{est}, G_i^{est}, B_i^{est})$. $(R_i^{est}, G_i^{est}, B_i^{est})$ can be obtained by replacing Cb_i and Cr_i in Eq. (2) with Cb_s and Cr_s, respectively:

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.391 & -0.813 \\ 1.164 & 2.018 & 0 \end{bmatrix} \begin{bmatrix} Y_i - 16 \\ Cb_i - 128 \\ Cr_i - 128 \end{bmatrix} \quad (2)$$

Setting the estimated ith RGB full-color pixel $(R_i^{est}, G_i^{est}, B_i^{est})$ to equal the ground truth ith full-color pixel (R_i, G_i, B_i), it yields an equation for each color $\in \{R, G, B\}$, resulting in the following three equations:

$$\begin{aligned} Y_i'^R &= \frac{[R_i - 1.596(Cr_s - 128)]}{1.164} + 16 \\ Y_i'^G &= \frac{[G_i + 0.391(Cb_s - 128) + 0.813(Cr_s - 128)]}{1.164} + 16 \\ Y_i'^B &= \frac{[B_i - 2.018(Cb_s - 128)]}{1.164} + 16 \end{aligned} \quad (3)$$

where $Y_i'^R$, $Y_i'^G$, and $Y_i'^B$ denote the exact luma modification solutions for the first, second, third equations, respectively. However, it is intractable to determine the unique luma modification solution of Y_i' that satisfies the three equations in Eq. (3) simultaneously. Therefore, in the luma modification (LM) method for CSFLM [4], the best luma modification solution of Y_i' is determined by examining all luma values in the search interval $[Low_i, High_i]$ with $Low_i = \lfloor min(Y_i'^R, Y_i'^R, Y_i'^R) \rfloor$ and $High_i = \lceil max(Y_i'^R, Y_i'^R, Y_i'^R) \rceil$, where $\lceil . \rceil$ and $\lfloor . \rfloor$ denote the ceiling and floor operations, respectively, such that the minimal COPY-based pixel-distortion value is achieved.

Based on the IMAX, Kodak, and SCI (screen content images) datasets, the experimental data indicated that the average number of the luma values examined in the search interval is 4.66, and the execution time overhead required in the LM method is only 6.4% relative to that required to complete the 4:2:0(A) chroma subsampling and the encoding process in HEVC. Experimental results demonstrated the quality enhancement and quality-bitrate tradeoff merits of the LM method in the CSFLM scheme.

Although the search interval is short for the pixel in the smooth area, it may be long for the pixel in the textural area. Therefore, applying a differentiation technique on the search interval [4], Lin *et al.* [11] proposed a luma optimization (LO) method to reduce the number of luma values examined in the search interval. However, due to the differentiation technique used, the luma values examined in [11] are real values, so it still needs to take a floor operation and a ceiling operation on each examined luma value, and then a better luma modification solution is selected. With competitive quality performance, the LO method is faster than the LM method. Zhu *et al.* [21] deployed the LM idea [4] in their DCT (discrete cosine transform) based color cross-space distortion minimization-based image compression method to compensate for the residual between the original image and the reconstructed image, where the residual is caused by quantization and dequantization in JPEG. In [20], Zhu *et al.* proposed a cholesky decomposition approach to compress the color image by combining the chroma subsampling and luma modification.

As depicted in Fig. 1, after performing the luma modification for the whole luma image at the server side, the subsampled CbCr image and the modified luma image are fed into the encoder for compression. After transmitting the compressed subsampled chroma image and the luma image to the receiver via the communication network, at the client side, the decompressed subsampled CbCr image is upsampled, and then by Eq. (2), the upsampled YCbCr image is converted into a reconstructed RGB full-color image.

C. CONTRIBUTIONS

Relative to the current two methods [4], [11], the three contributions of the proposed nonlinear interpolation-based, namely the bicubic convolution interpolation (BCI)-based, iterative luma modification method for I^{RGB} are clarified as follows.

1) THE FIRST CONTRIBUTION

Differing from the COPY-based pixel-distortion function [4], [11], we propose a new and more effective BCI-based pixel-distortion function. Further, the convex property of the proposed distortion function, which serves as the base of the initial luma modification solution, is proved.

2) THE SECOND CONTRIBUTION

Differing from the search ways used in the LM and LO methods, in this paper, we model the luma modification problem for CSFLM as a BCI-based overdetermined system, and then we apply a pseudoinverse technique to obtain the initial luma modification solution. Furthermore, an iterative luma modification method is proposed to improve the initial luma modification solution, achieving substantial quality, execution time, and quality-bitrate tradeoff improvements.

3) THE THIRD CONTRIBUTION

Based on the Kodak [10], IMAX [7], SCI (screen content images) [14], CI (classical images) [20], [21], and Video [17] datasets, the thorough experimental results have demonstrated that on the newly released Versatile Video Coding (VVC) platform VTM-12.0 [18], our BCI-based iterative luma modification method achieves substantial quality, execution time, and quality-bitrate tradeoff improvements when compared with the LM method [4] and the LO method [11]. The considered quality metrics include the CPSNR (color peak signal-to-noise ratio), SSIM (structural similarity index measure) [19], and visual effect. The quality-bitrate tradeoff metric used is the BD-rate (Bjøntegaard delta bitrate) [1].

The rest of this paper is organized as follows. In Section II, the proposed BCI-based pixel-distortion function is presented, and then we prove that the proposed pixel-distortion function is a convex function. In Section III, an iterative luma modification method is proposed. In Section IV, the thorough experimental results are illustrated to justify the substantial quality, execution time, and quality-bitrate tradeoff improvements of our method. In Section V, some concluding remarks are made.

II. THE PROPOSED NEW BCI-BASED PIXEL-DISTORTION FUNCTION

In the CSFLM scheme of Fig. 1, the subsampled (Cb, Cr)-pair of each B^{CbCr} is known in advance and it is denoted by (Cb_s, Cr_s). In this section, we first propose a novel and more effective BCI-based pixel-distortion function to measure the sum of squared errors (SSE) between the ground truth triple-value (R_i, G_i, B_i) and the estimated analogue $(R_i^{est}, G_i^{est}, B_i^{est})$, $1 \leq i \leq 4$. Next, we prove that the proposed pixel-distortion function is a convex function which serves as the base of the initial luma modification solution in our iterative method.

A. THE PROPOSED PIXEL-DISTORTION FUNCTION

Before estimating the triple-value $(R_i^{est}, G_i^{est}, B_i^{est})$, $1 \leq i \leq 4$, we first estimate each entry of $B^{est,CbCr}$, namely (Cb_i^{est}, Cr_i^{est}).

1) THE ESTIMATION OF (Cb_i^{est}, Cr_i^{est})

Without the loss of generality, we only describe how to apply our BCI-based estimation method to estimate Cb_1^{est} using the subsampled Cb value of B^{Cb}, namely Cb_s, and the subsampled Cb values of the neighboring 2×2 Cb blocks.

As depicted in Fig. 2(a), each 2×2 Cb block B^{Cb} is viewed as a 1×1 macro pixel marked in blue. As the origin of the xy-coordinate system, the location of the subsampled Cb value of B^{Cb}, namely Cb_s, is set to (0, 0). Therefore, Cb_1^{est} marked in red is located at $(-0.25, 0.25)$.

Let $d_{i,j}^h$ and $d_{i,j}^v$ denote the horizontal and vertical distances between the reference subsampled Cb point, namely $Cb(i, j)$, and the point Cb_1^{est} in the x-coordinate and y-coordinate, respectively. As depicted in Fig. 2(b), for Cb(−1, 1),

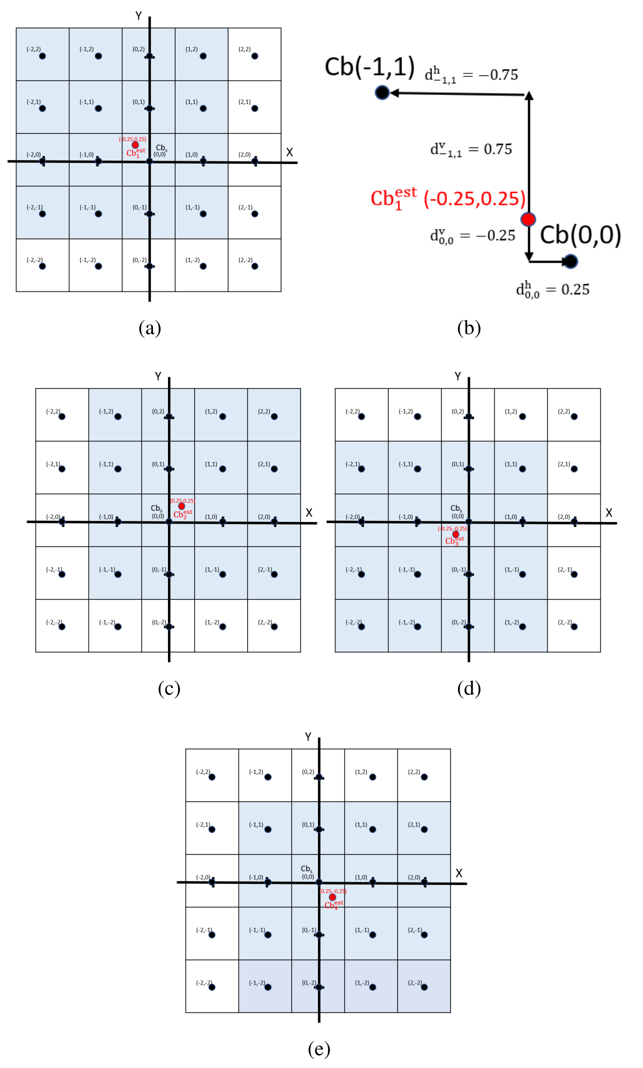


FIGURE 2. The sixteen reference subsampled Cb values for estimating Cb_i^{est} for $1 \leq i \leq 4$. (a) The sixteen reference subsampled Cb values in blue for Cb_1^{est}. (b) The values of $d^h_{-1,1}$, $d^v_{-1,1}$, $d^h_{0,0}$, and $d^v_{0,0}$. (c) The 16 reference subsampled Cb values in blue for estimating Cb_2^{est}. (d) The 16 reference subsampled Cb values in blue for estimating Cb_3^{est}. (e) The 16 reference subsampled Cb values in blue for estimating Cb_4^{est}.

the values of $d^h_{-1,1}$ and $d^v_{-1,1}$ are -0.75 (=−1 + 0.25) and 0.75 (=1 − 0.25), respectively. Similarly, for Cb(0, 0) (=Cb_s), it yields $d^h_{0,0}$ = 0.25 (=0 + 0.25) and $d^v_{0,0}$ = −0.25 (=0 − 0.25).

After defining $d^h_{i,j}$ and $d^v_{i,j}$, we deploy the bicubic convolution interpolation [9] in our BCI-based estimation method for estimating Cb_1^{est}. The weight assigned to each reference subsampled Cb point, namely $Cb(i, j)$, is given by

$$\bar{W}(i,j) = W(d^h_{i,j})W(d^v_{i,j}) \tag{4}$$

with

$$W(d) = \begin{cases} (p+2)|d|^3 - (p+3)|d|^2 + 1 & \text{for } |d| \leq 1 \\ p|d|^3 - 5p|d|^2 + 8p|d| - 4p & \text{for } 1<|d|<2 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

where in our experiment, the best choice of the parameter p in Eq. (5) is −0.5. Due to the constraint: $|d| < 2$, for estimating Cb_1^{est}, as depicted in Fig. 2(a), only sixteen reference subsampled Cb values are considered.

Therefore, Cb_1^{est} can be estimated by

$$\begin{aligned} Cb_1^{est} &= \bar{W}(0.25, -0.25)Cb(0,0) \\ &\quad + \sum_{i=-2}^{1} \sum_{\substack{j=-1 \\ (i,j)\neq(0,0)}}^{2} \bar{W}(i,j)Cb(i,j) \\ &= \bar{W}(0.25)\bar{W}(-0.25)Cb(0,0) + Cb_1^{const} \\ &= 0.752Cb_s + Cb_1^{const} \end{aligned} \tag{6}$$

where $\bar{W}(0.25) = \bar{W}(-0.25) = 0.8672$. All values of $\bar{W}(i, j)$ for $-2 \leq i \leq 1$ and $-1 \leq j \leq 2$ have been calculated in advance and they have been stored in a lookup table.

2) THE PROPOSED PIXEL-DISTORTION FUNCTION

Before presenting our new pixel-distortion function, we first estimate the estimated ith RGB triple-value (R_i^{est}, G_i^{est}, B_i^{est}), $1 \leq i \leq 4$. Based on the parameter Y_i' and the two estimated chroma values, namely Cb_i^{est} and Cr_i^{est}, by Eq. (2), the estimated ith RGB triple-value, (R_i^{est}, G_i^{est}, B_i^{est}), can be obtained by the following YCbCr-to-RGB color conversion:

$$\begin{bmatrix} R_i^{est} \\ G_i^{est} \\ B_i^{est} \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.391 & -0.813 \\ 1.164 & 2.018 & 0 \end{bmatrix} \begin{bmatrix} Y_i' - 16 \\ Cb_i^{est} - 128 \\ Cr_i^{est} - 128 \end{bmatrix} \tag{7}$$

Using the SSE metric, the proposed RGB full-color pixel-distortion between the ground truth pixel (R_i, G_i, B_i), $1 \leq i \leq 4$, and the estimated pixel (R_i^{est}, G_i^{est}, B_i^{est}) is expressed as

$$\begin{aligned} PD(Y_i') &= [(R_i - R_i^{est})^2 + (G_i - G_i^{est})^2 + (B_i - B_i^{est})^2] \\ &= \left\{R_i - [1.164(Y_i' - 16) + 1.596(Cr_i^{est} - 128)]\right\}^2 \\ &\quad + \left\{G_i - [1.164(Y_i' - 16) - 0.391(Cb_i^{est} - 128) \right. \\ &\quad \left. - 0.813(Cr_i^{est} - 128)]\right\}^2 \\ &\quad + \left\{B_i - [1.164(Y_i' - 16) + 2.018(Cb_i^{est} - 128)]\right\}^2 \\ &= \left\{Const_1 - [1.164(Y_i')]\right\}^2 \\ &\quad + \left\{Const_2 - [1.164(Y_i')]\right\}^2 \\ &\quad + \left\{Const_3 - [1.164(Y_i')]\right\}^2 \end{aligned} \tag{8}$$

where $Const_1 = R_i + 1.164 * 16 + 1.596(Cr_i^{est} - 128)$, $Const_2 = G_i + 1.164*16 - 0.391(Cb_i^{est} - 128) - 0.813(Cr_i^{est} - 128)$, and $Const_3 = B_i + 1.164*16 + 2.018(Cb_i^{est} - 128)$. From Eq. (8), we know that the proposed pixel-distortion function $PD(Y_i')$ is a quadratic function with the parameter Y_i'.

In what follows, we prove that $PD(Y_i')$ is a convex function, and this convex property serves as the base of the initial luma modification solution in our iterative luma modification method.

B. CONVEX PROPERTY PROOF OF THE PROPOSED PIXEL-DISTORTION FUNCTION

According to the convex function definition [2], if a function f satisfies $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$ for $0 \leq \theta \leq 1$, where the two points, x and y, are taken from the function f, then f is called a convex function.

We now prove that our new pixel-distortion function $PD(Y'_i)$, $1 \leq i \leq 4$, in Eq. (8) is a convex function because it satisfies the condition: "$PD(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \leq \theta PD(Y'_{i_1}) + (1-\theta)PD(Y'_{i_2})$," where any two points, namely Y'_{i_1} and Y'_{i_2}, are taken from $PD(Y'_i)$. By Eq. (8), the left hand side in the inequality of the above-mentioned condition is expressed as

$$\begin{aligned} &PD(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\quad = \left\{Const_1 - [1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2})]\right\}^2 \\ &\qquad + \left\{Const_2 - [1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2})]\right\}^2 \\ &\qquad + \left\{Const_3 - [1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2})]\right\}^2 \end{aligned} \tag{9}$$

where the three constant terms, namely $Const_1$, $Const_2$, and $Const_3$, have been defined in Eq. (8). Expanding the right hand side of Eq. (9), it yields

$$\begin{aligned} &PD(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\quad = \{Const_1^2 - 2.328(Const_1)(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\qquad + (1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}))^2\} \\ &\qquad + \{Const_2^2 - 2.328(Const_2)(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\qquad + (1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}))^2\} \\ &\qquad + \{Const_3^2 - 2.328(Const_3)(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\qquad + (1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}))^2\} \end{aligned} \tag{10}$$

By the same argument, the right hand side in the inequality of the above-mentioned convex function condition is expressed as

$$\begin{aligned} &\theta PD(Y'_{i_1}) + (1-\theta)PD(Y'_{i_2}) \\ &\quad = \theta\left\{Const_1 - [1.164(Y'_{i_1})]\right\}^2 \\ &\qquad + (1-\theta)\left\{Const_1 - [1.164(Y'_{i_2})]\right\}^2 \\ &\qquad + \theta\left\{Const_2 - [1.164(Y'_{i_1})]\right\}^2 \\ &\qquad + (1-\theta)\left\{Const_2 - [1.164(Y'_{i_2})]\right\}^2 \\ &\qquad + \theta\left\{Const_3 - [1.164(Y'_{i_1}]\right\}^2 \\ &\qquad + (1-\theta)\left\{Const_3 - [1.164(Y'_{i_2})]\right\}^2 \\ &\quad = \{Const_1^2 - 2.328Const_1(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\qquad + \theta(1.164Y'_{i_1})^2 + (1-\theta)(1.164Y'_{i_2})^2\} \\ &\qquad + \{Const_2^2 - 2.328Const_2(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\qquad + \theta(1.164Y'_{i_1})^2 + (1-\theta)(1.164Y'_{i_2})^2\} \\ &\qquad + \{Const_3^2 - 2.328Const_3(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &\qquad + \theta(1.164Y'_{i_1})^2 + (1-\theta)(1.164Y'_{i_2})^2\} \end{aligned} \tag{11}$$

Subtracting Eq. (10) from Eq. (11) yields

$$\begin{aligned} &\theta PD(Y'_{i_1}) + (1-\theta)PD(Y'_{i_2}) - PD(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \\ &= \{\theta(1.164Y'_{i_1})^2 + (1-\theta)(1.164Y'_{i_2})^2 \\ &\quad - (1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}))^2\} \\ &\quad + \{\theta(1.164Y'_{i_1})^2 + (1-\theta)(1.164Y'_{i_2})^2 \\ &\quad - (1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}))^2\} \\ &\quad + \{\theta(1.164Y'_{i_1})^2 + (1-\theta)(1.164Y'_{i_2})^2 \\ &\quad - (1.164(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}))^2\} \\ &= 3(1.164)^2\{(\theta-\theta^2){Y'_{i_1}}^2 + (\theta-\theta^2){Y'_{i_2}}^2 - 2(\theta-\theta^2)Y'_{i_1}Y'_{i_2}\} \\ &= 3(1.164)^2(\theta-\theta^2)\{Y'_{i_1} - Y'_{i_2}\}^2 \geq 0 \end{aligned} \tag{12}$$

Eq. (12) implies that the convex function condition "$PD(\theta Y'_{i_1} + (1-\theta)Y'_{i_2}) \leq \theta PD(Y'_{i_1}) + (1-\theta)PD(Y'_{i_2})$" holds to our new proposed pixel-distortion function $PD(Y'_i)$ in Eq. (8). We thus have the following result.

Proposition 1: Our new pixel-distortion function $PD(Y'_i)$ in Eq. (8) is a convex function.

III. THE PROPOSED FAST AND EFFECTIVE ITERATIVE LUMA MODIFICATION METHOD

We first transform the luma modification problem in CSFLM to an overdetermined system. Next, we apply the pseudo-inverse technique to determine the initial integer luma modification solution. By Proposition 1, we propose a fast and effective iterative luma modification method to improve the initial integer luma modification solution.

A. DETERMINING THE INITIAL INTEGER LUMA MODIFICATION SOLUTION

Ideally, we hope to determine the luma modification solution of Y'_i such that the resultant triple-value $(R_i^{est}, G_i^{est}, B_i^{est})$ in the left side of Eq. (7) could be equal to the original RGB triple-value (R_i, G_i, B_i). Accordingly, we consider the three equations:

$$\begin{aligned} R_i &= R_i^{est} \\ &= 1.164(Y'_i - 16) + 1.596(Cr_i^{est} - 128) \\ G_i &= G_i^{est} \\ &= 1.164(Y'_i - 16) - 0.391(Cb_i^{est} - 128) \\ &\quad - 0.813(Cr_i^{est} - 128) \\ B_i &= B_i^{est} \\ &= 1.164(Y'_i - 16) + 2.018(Cb_i^{est} - 128) \end{aligned} \tag{13}$$

In the overdetermined system of Eq. (13), there are three equations, but there is only one parameter Y'_i. To solve Y'_i in Eq. (13), we first move the constant terms in the right hand side of each equality in Eq. (13) to the left hand side. Let the three constant terms be denoted by a 3×1 vector, namely $(\bar{R}_i, \bar{G}_i, \bar{B}_i)^t = (R_i - 1.596(Cr_i^{est}) + 222.912,$

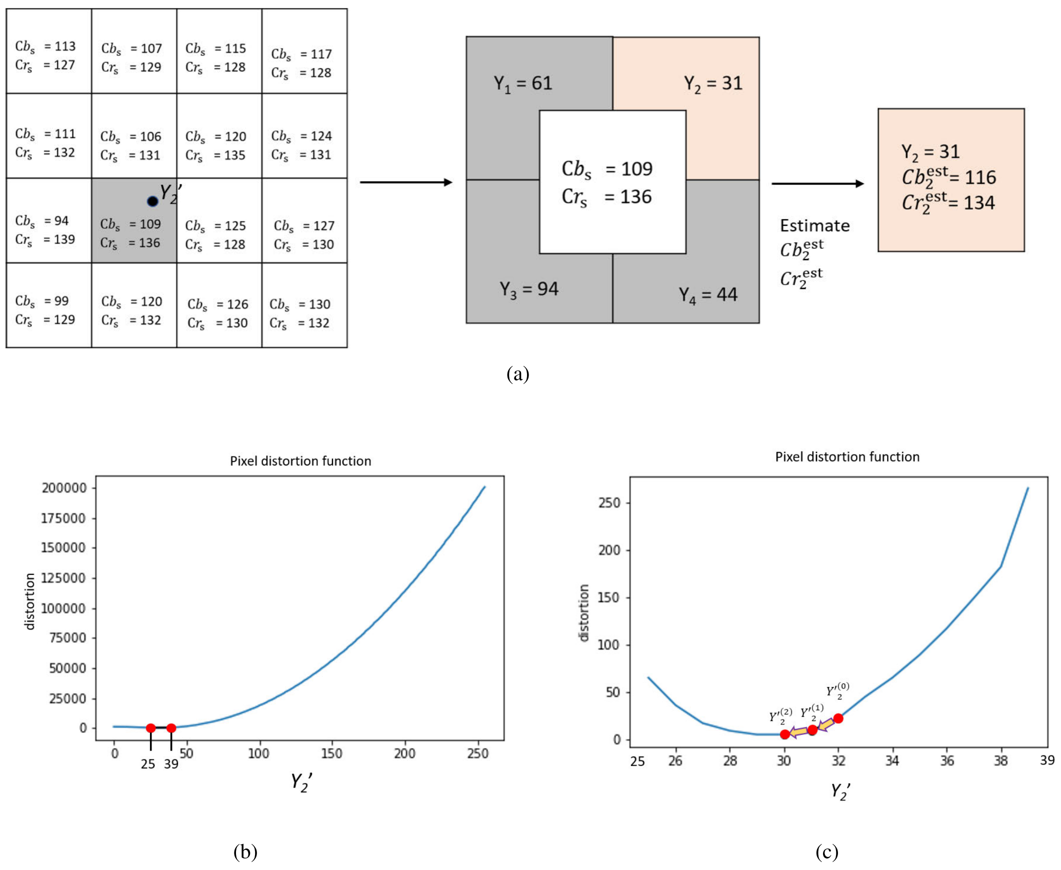


FIGURE 3. The sketch of the proposed iterative luma modification method. (a) An example for determining the luma modification solution for Y_2'. (b) The plot of the convex pixel-distortion function $PD(Y_2')$ for Fig. 3(a) in the real domain. (c) The initial integer luma modification solution $Y_2'^{(0)}$ the refined solution $Y_2'^{(1)}$, and the refined solution $Y_2'^{(2)}$.

$G_i + 0.391(Cb_i^{est}) + 0.831(Cb_i^{est}) - 135.488, B_i - 2.018(Cb_i^{est}) + 276.928)^t$. Eq. (13) is thus expressed as

$$\begin{bmatrix}\bar{R}_i\\ \bar{G}_i\\ \bar{B}_i\end{bmatrix} = \begin{bmatrix} R_i - 1.596(Cr_i^{est}) + 222.912\\ G_i + 0.391(Cb_i^{est}) + 0.831(Cb_i^{est}) - 135.488\\ B_i - 2.018(Cb_i^{est}) + 276.928\end{bmatrix} = \begin{bmatrix}1.164\\1.164\\1.164\end{bmatrix}[Y_i'] \tag{14}$$

Let $b = (\bar{R}_i, \bar{G}_i, \bar{B}_i)^t$ and $A = (1.164, 1.164, 1.164)^t$. Eq. (14) is expressed as

$$AY_i' = b \tag{15}$$

We apply the pseudo-inverse technique to solve Y_i' in Eq. (15). Multiplying both sides of Eq. (15) by A^t, it yields $A^tAY_i' = A^tb$, and then multiplying $(A^tA)^{-1}$ to both sides of $A^tAY_i' = A^tb$, it yields

$$Y_i' = (A^tA)^{-1}A^tb \tag{16}$$

Putting $(A^tA)^{-1} = \frac{1}{3*(1.164)^2}$ and $A^t = (1.164, 1.164, 1.164)$ back to Eq. (16), the initial real luma modification solution for Y_i' is given by

$$\begin{aligned} Y_i' &= \frac{1}{3*(1.164)^2}(1.164, 1.164, 1.164)b\\ &= (0.286, 0.286, 0.286)b\\ &= 0.286(R_i + G_i + B_i) - 0.224Cr_i^{est} - 0.466Cb_i^{est}\\ &\quad + 104.34 \end{aligned} \tag{17}$$

where the values of R_i, G_i, and B_i are known; the values of Cb_i^{est} and Cr_i^{est} have been calculated using Eq. (6) or its analogue. Considering the practical integer domain, we take the ceiling and flooring operations on the initial real luma modification solution in Eq. (17) to obtain two possible integer luma modification solutions, namely $L = \lfloor(0.286, 0.286, 0.286)b\rfloor$ and $H = \lceil(0.286, 0.286, 0.286)b\rceil$, where $H = L + 1$. Next, we select the best one, L or H, with the minimal pixel-distortion value as the initial integer luma modification solution, namely $Y_i'^{(0)}$.

B. THE PROPOSED ITERATIVE LUMA MODIFICATION METHOD

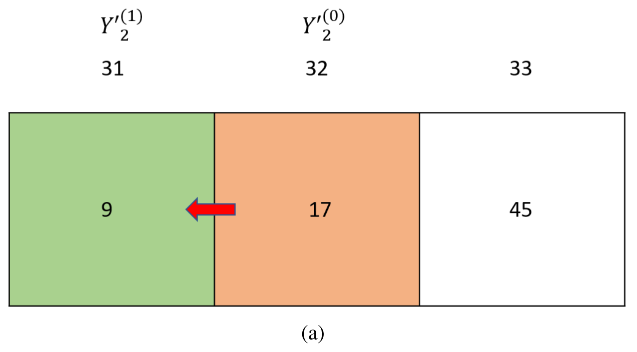
For easy readability, we take a practical example to sketch the room to improve the initial integer luma modification solution using our iterative method. Given a practical example in Fig. 3(a), the 16 reference subsampled chroma pairs are shown in the 4×4 macro block, where the subsampled (Cb, Cr)-pair of the current CbCr block B^{CbCr} is denoted by (Cb_s, Cr_s) $(= (109, 136))$, and the considered luma modification parameter Y'_2 is marked by a black bullet.

Using an analogue of Eq. (6) to estimate Cb_2^{est} and Cr_2^{est}, it yields $Cb_2^{est} = 116$ and $Cr_2^{est} = 134$. It is known that $R_2 = 26$, $G_2 = 15$, and $B_2 = 2$. Furthermore, by Eq. (8), the pixel-distortion function $PD(Y'_2)$ is expressed as

$$\begin{aligned} PD(Y'_2) &= (R_2 - [1.164Y'_2 + 1.596Cr_2^{est} - 222.912])^2 \\ &\quad + (G_2 - [1.164Y'_2 - 0.391Cb_2^{est} - 0.813Cr_2^{est} \\ &\quad + 135.488])^2 + (B_2 - [1.164Y'_2 + 2.018Cb_2^{est} \\ &\quad - 276.928])^2 \\ &= (35.048 - 1.164Y'_2)^2 + (33.81 - 1.164Y'_2)^2 \\ &\quad + (44.84 - 1.164Y'_2)^2 \end{aligned} \tag{18}$$

In the real domain, the plot of the convex function $PD(Y'_2)$ in Eq. (18) is depicted in Fig. 3(b).

Considering the interval [25, 39] for Y'_2 in Fig. 3(b), we cut off the corresponding convex curve segment, and then the enlarged discrete plot of the curve segment is depicted in Fig. 3(c). In Fig. 3(c), the path from the point $Y_2'^{(0)}$ to $Y_2'^{(2)}$ indicates the room to improve the initial integer luma modification solution, where the iteration number used in our iterative method is 2.



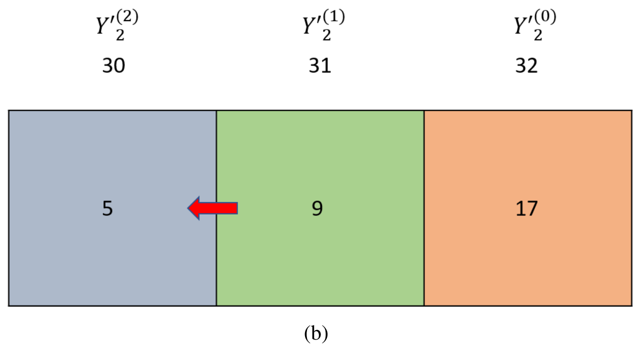


FIGURE 4. Two luma modification solution-refinement iterations. (a) First solution-refinement iteration. (b) Second solution-refinement iteration.

To refine the initial integer luma modification solution $Y'^{(0)}_2$, for easy exposition, we assume $Y_2'^{(0)} = \lfloor Y_2'^{(0)} \rfloor = L$. In the example of Fig. 3(c), the initial integer luma modification solution $Y_2'^{(0)}$ equals 32 and the pixel-distortion value $PD(32)$ equals 17 which is marked in orange in Fig. 4(a). Due to the assumption: $Y_2'^{(0)} = \lfloor Y_2'^{(0)} \rfloor = L$, we further consider the left neighboring luma modification solution 31 $(=Y_2'^{(0)} - 1 = 32 - 1)$ as the refined luma modification solution candidate. Because the value of PD(31) equals 9, as marked in green in Fig. 4(a), the luma modification solution is refined from $Y_2'^{(0)}$ (=32) to $Y_2'^{(1)}$ (=31), and the pixel-distortion reduction is 8 (=17 − 9).

By the same argument, in the second iteration, i.e. $k = 2$, as depicted in Fig. 4(b), the luma modification solution is refined from 31 to 30 and the pixel-distortion reduction is 4 (=9 − 5). The above iterative luma modification solution-refinement process continues to the left until no improvement is achievable. It is notable that if $Y_2'^{(0)} = \lceil Y_2'^{(0)} \rceil = H$, the above iterative luma modification solution-refinement process continues to the right until no improvement is achievable.

For $Y_2'^{(0)} = \lfloor Y_2'^{(0)} \rfloor = L$, the proposed BCI-based iterative luma modification method, namely Algorithm 1, is listed below.

Algorithm 1 The Proposed BCI-Based Iterative Luma Modification Method

```
Input: The values of $R_i$, $G_i$, $B_i$, $Cr_i^{est}$, and $Cb_i^{est}$.
Output: Integer luma modification solution for $Y'_i$.
Step 1: Calculate the initial integer luma modified solution $Y_i'^{(0)}$, and then by Eq. (8), we calculate the value of $PD(Y_i'^{(0)})$. Perform $k := 0$.
Step 2: Perform $Y_i'^{(k+1)} := Y_i'^{(k)} - 1$. Then, we calculate the value of $PD(Y_i'^{(k+1)})$.
Step 3:
If $PD(Y_i'^{(k+1)}) < PD(Y_i'^{(k)})$
then
    Perform $k := k + 1$. Go to Step 2.
end If
Step 4: Return $Y_i'^{(k)}$ as the output.
```

IV. EXPERIMENTAL RESULTS

Based on the Kodak dataset with 24 images, the IMAX dataset with 18 images, the Video dataset with 200 images, the CI dataset with 8 images, and the SCI dataset with 20 images, the quality, execution time, and quality-bitrate trade-off improvements of our iterative luma modification method are demonstrated when compared with the two state-of-the-art methods [4], [11]. For convenience, the five considered chroma subsampling methods are denoted by the set C_s = {4:2:0(A), 4:2:0(L), 4:2:0(R), 4:2:0(D), Anchor}; the three considered chroma upsampling methods are denoted by the set C_u = {NN, 4-tap, INTER_CUBIC} where the nearest neighbor (NN) method and the 4-tap method are supported by the VVC platform [18]. The INTER_CUBIC method is supported by the OpenCV library [3], [13].

TABLE 1. The CPSNR and SSIM performance of all combinations in C_s × LM [4] × C_u and C_s × LO [11] × C_u.

	C_u	IMAX		Kodak		Video		SCI		CI		Average	
		CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM
	NN	38.1194	0.9641	45.0211	0.9860	47.6667	0.9942	35.8452	0.9790	36.2574	0.9500	40.5819	0.9746
4:2:0(A)-LM [4]	$INTER_{CUBIC}$	39.6489	0.9713	46.8118	0.9895	49.3028	0.9955	36.4501	0.9784	36.9967	0.9522	**41.8420**	**0.9773**
	4-tap	37.4926	0.9604	44.6007	0.9854	46.8674	0.9933	35.0180	0.9732	36.0228	0.9442	40.0003	0.9713
	NN	36.7039	0.9551	43.7895	0.9828	46.9454	0.9935	34.3499	0.9765	35.4355	0.9387	39.4448	0.9693
4:2:0(L)-LM [4]	$INTER_{CUBIC}$	38.9419	0.9683	46.5898	0.9893	49.0638	0.9953	35.6432	0.9769	36.3954	0.9461	**41.3268**	**0.9751**
	4-tap	38.3768	0.9646	45.3839	0.9867	47.1781	0.9935	35.3208	0.9742	36.0228	0.9442	40.4564	0.9726
	NN	36.7633	0.9562	44.0202	0.9835	47.2639	0.9940	34.3037	0.9763	35.0002	0.9406	39.4702	0.9701
4:2:0(R)-LM [4]	$INTER_{CUBIC}$	38.9530	0.9682	46.1844	0.9880	49.0768	0.9953	35.5923	0.9770	35.7129	0.9456	**41.1046**	**0.9769**
	4-tap	34.9809	0.9422	42.3654	0.9789	45.6424	0.9921	32.8261	0.9680	33.7620	0.9289	37.9153	0.9620
	NN	35.4224	0.9438	42.5095	0.9782	45.5569	0.9924	33.3114	0.9744	33.9834	0.9213	38.1567	0.9620
4:2:0(D)-LM [4]	$INTER_{CUBIC}$	38.0933	0.9630	45.7774	0.9876	48.4868	0.9950	34.8764	0.9755	35.2268	0.9349	**40.4921**	**0.9712**
	4-tap	36.8629	0.9546	43.9487	0.9830	46.0017	0.9927	34.1756	0.9718	34.5874	0.9305	39.1152	0.9665
	NN	36.7812	0.9561	43.3353	0.9837	45.4866	0.9932	34.6956	0.9769	35.5101	0.9434	39.1617	0.9718
Anchor-LM [4]	$INTER_{CUBIC}$	38.8579	0.9682	45.4535	0.9893	46.8522	0.9949	35.9855	0.9781	36.5444	0.9501	**40.7387**	**0.9761**
	4-tap	37.9781	0.9627	43.5308	0.9860	44.5171	0.9931	35.6155	0.9769	35.9267	0.9477	39.5136	0.9732
	NN	37.8851	0.9649	45.0311	0.9858	47.6452	0.9941	35.4721	0.9786	36.1993	0.9500	40.4471	0.9746
4:2:0(A)-LO [11]	$INTER_{CUBIC}$	39.6625	0.9717	46.8961	0.9894	49.4231	0.9956	36.2207	0.9782	37.0124	0.9519	**41.8429**	**0.9782**
	4-tap	37.4127	0.9608	44.6065	0.9853	46.8958	0.9934	34.7904	0.9729	35.8020	0.9453	39.9014	0.9715
	NN	36.3679	0.9556	43.7672	0.9825	46.9053	0.9934	33.9107	0.9757	35.2237	0.9383	39.2356	0.9691
4:2:0(L)-LO [11]	$INTER_{CUBIC}$	39.0558	0.9686	46.6996	0.9893	49.1661	0.9953	35.4779	0.9768	36.4224	0.9458	**41.3643**	**0.9751**
	4-tap	38.3663	0.9650	45.4180	0.9866	47.2106	0.9936	35.1173	0.9740	36.0409	0.9439	40.4306	0.9484
	NN	36.3714	0.9557	43.8385	0.9825	47.0411	0.9936	33.8282	0.9754	34.8637	0.9387	39.1885	0.9691
4:2:0(R)-LO [11]	$INTER_{CUBIC}$	39.0868	0.9688	46.2931	0.9880	49.2305	0.9954	35.4767	0.9769	35.7391	0.9454	**41.1652**	**0.9749**
	4-tap	34.8045	0.9428	42.3460	0.9787	45.6571	0.9922	32.5312	0.9673	33.7292	0.9285	37.8136	0.9619
	NN	35.0646	0.9440	42.4623	0.9778	45.5196	0.9924	32.8513	0.9733	33.7263	0.9202	37.9248	0.9615
4:2:0(D)-LO [11]	$INTER_{CUBIC}$	38.2703	0.9635	45.8925	0.9876	48.6341	0.9952	34.7609	0.9753	35.3049	0.9344	**40.5725**	**0.9712**
	4-tap	36.7633	0.9550	43.9356	0.9829	46.0239	0.9928	33.9305	0.9714	34.5335	0.9297	39.0373	0.9663
	NN	36.4637	0.9568	43.3330	0.9835	45.4626	0.9931	34.3489	0.9764	35.2618	0.9430	38.974	0.9705
Anchor-LO [11]	$INTER_{CUBIC}$	38.9364	0.9686	45.5608	0.9894	46.9742	0.9952	35.8504	0.9779	36.6480	0.9500	**40.7939**	**0.9762**
	4-tap	37.9377	0.9632	43.5649	0.9861	44.5793	0.9934	35.3399	0.9765	36.0163	0.9477	39.4865	0.9733

All the concerned experiments are implemented on a computer with an Intel Core i7-9700 CPU 3.0 GHz and 32 GB RAM. The operating system is the Microsoft Windows 10 64-bit operating system. The program development environment is Visual C++ 2017. The VVC reference software platform used for compression is VTM-12.0. The execution code of our iterative luma modification method can be accessed from the website [5].

A. QUALITY AND EXECUTION TIME IMPROVEMENTS OF OUR LUMA MODIFICATION METHOD

When setting QP (quantization parameter) to zero, we take the two quality metrics, namely CPSNR and SSIM, to demonstrate the quality enhancement merit of our iterative luma modification method, abbreviated as 'Ours'.

CPSNR is used to evaluate the average quality of the reconstructed RGB full-color images for one dataset with N images, and it is defined by

$$\mathrm{CPSNR} = \frac{1}{N}\sum_{n=1}^{N} 10\log_{10}\frac{255^2}{CMSE} \tag{19}$$

with $CMSE = \frac{1}{3WH}\sum_{p\in P}\sum_{c\in\{R,G,B\}}[I_{n,c}^{RGB}(p) - I_{n,c}^{RGB}(p)]^2$ in which P $= \{(x, y)|1 \le x \le H, 1 \le y \le W\}$ denotes the set of pixel coordinates in one $W \times H$ image. $I_{n,c}^{RGB}(p)$ and $I_{n,c}^{RGB}(p)$ denote the c-color value of the pixel at position p in the nth original RGB full-color image and the reconstructed analogue, respectively. The CPSNR value equals the mean of the five CPSNR values for the five datasets.

SSIM [19] is used to measure the joint preservation effects of luminance, contrast, and structure similarity between the original image and the reconstructed one. For I^{RGB}, the SSIM value is measured by the mean of the three SSIM values for the R, G, and B color planes.

1) QUALITY ENHANCEMENT MERIT

To demonstrate the quality merit of our luma modification method "Ours", in Table 1, we first demonstrate the CPSNR and SSIM performance of all 15 combinations in $C_s \times LM \times C_u$, where "LM" denotes the LM method in [4]. In the three combinations, namely $c_s \times LM \times C_u$, the combination "c_s-LM-$INTER_{CUBIC}$" always has the best CPSNR and SSIM performance, as shown in boldface. We thus select the five combinations "c_s-LM-$INTER_{CUBIC}$" as the comparative combinations. Similarly, for the 15 combinations in $C_s \times LO \times C_u$, where "LO" denotes the LO method in [11], we select five combinations "C_s×LO-$INTER_{CUBIC}$" as the comparative combinations.

For comparison fairness, considering each c_s in C_s, we compare our combination "c_s-Ours-$INTER_{CUBIC}$" with the two comparative combinations, namely c_s-LM-$INTER_{CUBIC}$ and c_s-LO-$INTER_{CUBIC}$, to justify the quality superiority of our luma modification method "Ours" over the LM method [4] and the LO method [11]. Based on the five datasets, for QP = 0, Table 2 indicates that the CPSNR gains of our five combinations over the corresponding five comparative combinations, namely C_s×-LM-$INTER_{CUBIC}$, are 0.4644 dB, 0.8239 dB, 0.9274 dB, 1.1261 dB, and 0.6441 dB, respectively. On average, the CPSNR gain of our luma modification method "Ours" over the LM method [4] is 0.7972 dB. In the same table, we observe that the average CPSNR gain of our luma modification method over the LO method [11] is 0.7502 dB.

Table 2 also indicates that the average SSIM gain of our luma modification method over the LM method and the LO method are 0.0023 and 0.0025, respectively.

2) EXECUTION TIME MERIT

Based on the five datasets, for one image, the average execution time required in the LM method [4] is 0.2049 seconds. The average execution time required in the LO method [11] is 0.1931 seconds. The average execution time required in our

TABLE 2. For QP = 0, the quality enhancement merit of our LUMA modification method "ours" over the LM method [4] and LO [11] methods.

Combination	IMAX		Kodak		Video		SCI		CI		Average			
	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	SSIM	CPSNR	CPSNR gain	SSIM	SSIM gain
4:2:0(A)-LM-$INTER_{CUBIC}$ [4]	39.6489	0.9713	46.8118	0.9895	49.3028	0.9955	36.4501	0.9784	36.9967	0.9522	41.8420		0.9773	
4:2:0(A)-LO-$INTER_{CUBIC}$ [11]	39.6625	0.9717	46.8961	0.9894	49.4231	0.9956	36.2207	0.9782	37.0124	0.9519	41.8429	0.0009	0.9782	0.0009
4:2:0(A)-Ours-$INTER_{CUBIC}$	40.4102	0.9748	47.2875	0.9904	49.6412	0.9957	36.8314	0.9801	37.3612	0.9539	**42.3064**	**0.4644**	**0.9790**	**0.0017**
4:2:0(L)-LM-$INTER_{CUBIC}$ [4]	38.9419	0.9683	46.5898	0.9893	49.0638	0.9953	35.6432	0.9769	36.3954	0.9461	41.3268		0.9751	
4:2:0(L)-LO-$INTER_{CUBIC}$ [11]	39.0558	0.9686	46.6996	0.9893	49.1661	0.9953	35.4779	0.9768	36.4224	0.9458	41.3643	0.0375	0.9751	0
4:2:0(L)-Ours-$INTER_{CUBIC}$	40.2603	0.9739	47.5182	0.9909	49.4964	0.9955	36.4198	0.9795	37.0588	0.9496	**42.1507**	**0.8239**	**0.9780**	**0.0029**
4:2:0(R)-LM-$INTER_{CUBIC}$ [4]	38.9530	0.9682	46.1844	0.9880	49.0768	0.9953	35.5923	0.9770	35.7129	0.9456	41.1046		0.9769	
4:2:0(R)-LO-$INTER_{CUBIC}$ [11]	39.0868	0.9688	46.2931	0.9880	49.2305	0.9954	35.4767	0.9769	35.7391	0.9454	41.1652	0.0606	0.9749	-0.0020
4:2:0(R)-Ours-$INTER_{CUBIC}$	40.4469	0.9751	47.0800	0.9899	49.9190	0.9960	36.4254	0.9798	36.2887	0.9507	**42.0320**	**0.9274**	**0.9784**	**0.0015**
4:2:0(D)-LM-$INTER_{CUBIC}$ [4]	38.0933	0.9630	45.7774	0.9876	48.4868	0.9950	34.8764	0.9755	35.2268	0.9349	40.4921		0.9712	
4:2:0(D)-LO-$INTER_{CUBIC}$ [11]	38.2703	0.9635	45.8925	0.9876	48.6341	0.9952	34.7609	0.9753	35.3049	0.9344	40.5725	0.0804	0.9712	0
4:2:0(D)-Ours-$INTER_{CUBIC}$	39.8181	0.9675	47.0219	0.9931	49.3001	0.9953	35.9001	0.9778	36.0507	0.9381	**41.6182**	**1.1261**	**0.9744**	**0.0032**
Anchor-LM-$INTER_{CUBIC}$ [4]	38.8579	0.9682	45.4535	0.9893	46.8522	0.9949	35.9855	0.9781	36.5444	0.9501	40.7387		0.9761	
Anchor-LO-$INTER_{CUBIC}$ [11]	38.9364	0.9686	45.5608	0.9894	46.9742	0.9952	35.8504	0.9779	36.6480	0.9500	40.7939	0.0552	0.9762	0.0001
Anchor-Ours-$INTER_{CUBIC}$	39.9582	0.9731	46.0670	0.9907	47.1227	0.9952	36.5722	0.9798	37.1938	0.9531	**41.3828**	**0.6441**	**0.9784**	**0.0023**

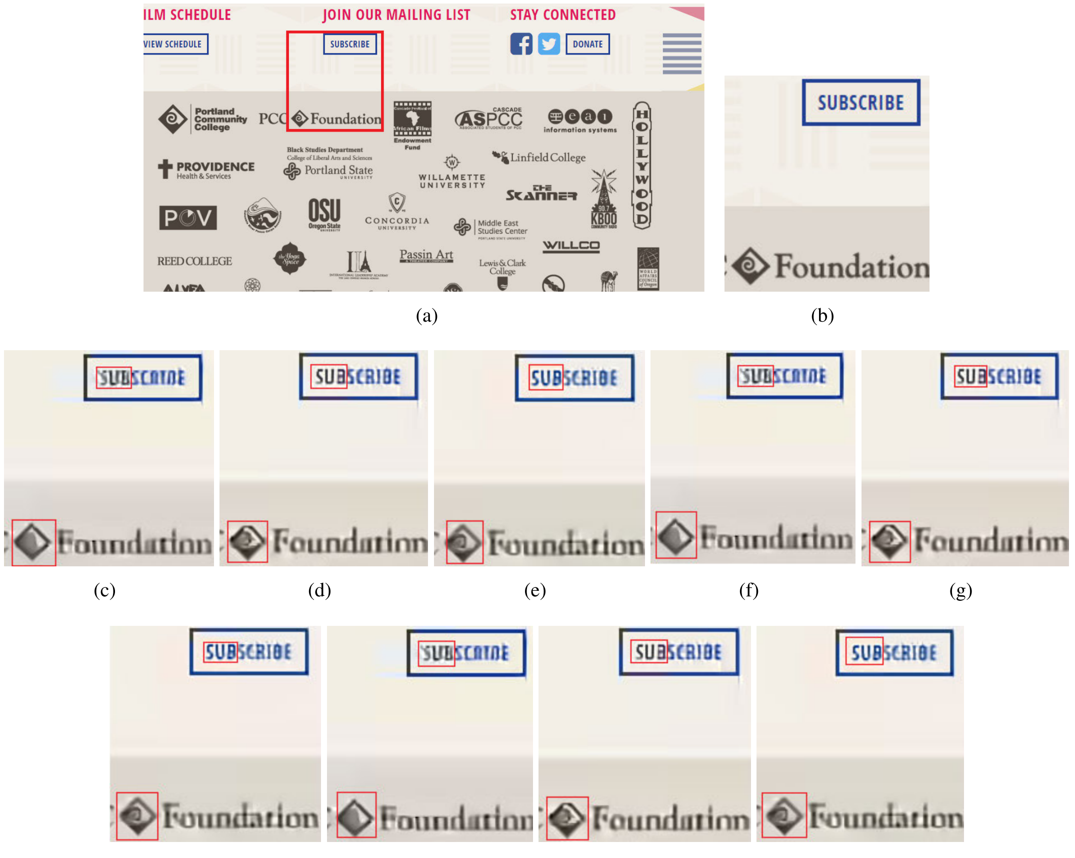


FIGURE 5. The first visual effect merit example of our luma modification method. (a) The 5th ground truth SCI image. (b) The magnified subimage of (a). (c) 4:2:0(A)-LM-NN [4]. (d) 4:2:0(A)-LO-NN [11]. (e) 4:2:0(A)-Ours-NN. (f) 4:2:0(A)-LM-(4-tap). (g) 4:2:0(A)-LO-(4-tap) [11]. (h) 4:2:0(A)-Ours-(4-tap). (i) 4:2:0(A)-LM-*INTER*$_{CUBIC}$ [4]. (j) 4:2:0(A)-LO-*INTER*$_{CUBIC}$ [11]. (k) 4:2:0(A)-Ours-*INTER*$_{CUBIC}$.

luma modification method is 0.1564 seconds. It indicates the execution time improvement merit of our method over the LM and LO methods.

B. VISUAL EFFECT AND BD-RATE MERITS OF OUR LUMA MODIFICATION METHOD

In this subsection, the visual effect and the quality-bitrate tradeoff merits of our luma modification method are illustrated.

1) THE VISUAL EFFECT MERIT

We first take the 5th ground truth SCI image in Fig. 5(a) to demonstrate the visual effect merit of the reconstructed RGB full-color image using our three combinations, namely 4:2:0(A)-Ours$\times C_u$, relative to the six comparative combinations, namely 4:2:0(A)-LM$\times C_u$ [4] and 4:2:0(A)-LO$\times C_u$ [11].

The magnified subimage in Fig. 5(b) is decoupled from the color region containing words "VIEW SCHEDULE" in

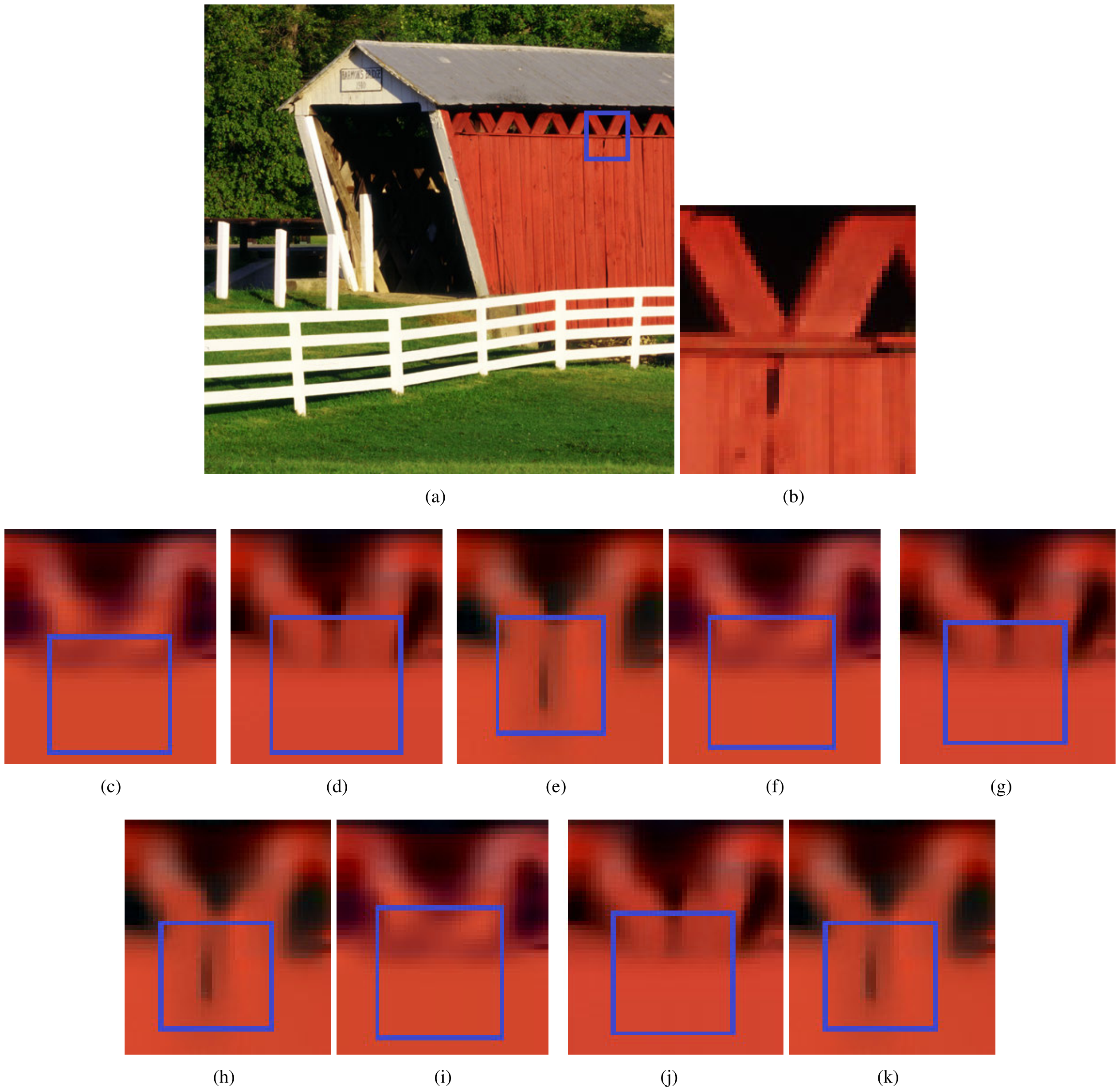


FIGURE 6. The second visual effect merit example of our luma modification method. (a) The 18th ground truth IMAX image. (b) The magnified subimage of (a). (c) 4:2:0(A)-LM-NN [4]. (d) 4:2:0(A)-LO-NN [11]. (e) 4:2:0(A)-Ours-NN. (f) 4:2:0(A)-LM-(4-tap). (g) 4:2:0(A)-LO-(4-tap) [11]. (h) 4:2:0(A)-Ours-(4-tap). (i) 4:2:0(A)-LM-*INTER*$_{CUBIC}$ [4]. (j) 4:2:0(A)-LO-*INTER*$_{CUBIC}$ [11]. (k) 4:2:0(A)-Ours-*INTER*$_{CUBIC}$.

Fig. 5(a). For QP = 48, after performing the 9 considered combinations on Fig. 5(b), the 9 reconstructed magnified subimages for Fig. 5(b) are shown in Figs. 5(c)-(k). From Figs. 5(c)-(k), we observe that our combination "4:2:0(A)-ours-c_u", $c_u \in C_u$, always has a better visual effect than "4:2:0(A)-LM-c_u" [4] and "4:2:0(A)-LO-c_u" [11].

Next, we take the 18th ground truth IMAX image in Fig. 6(a) to demonstrate the visual effect merit of the reconstructed RGB full-color image using our three combinations, 4:2:0(A)-Ours$\times C_u$, relative to the above-mentioned six comparative combinations. The magnified subimage in Fig. 6(b) is decoupled from a wood wall region in Fig. 6(a). For QP = 48, after performing the 9 considered combinations on Fig. 6(b), the 9 reconstructed magnified subimages for Fig. 6(b) are shown in Figs. 6(c)-(k). From Figs. 6(c)-(k), we observe that our combination "4:2:0(A)-ours-c_u"', $c_u \in C_u$, still has a better visual effect than "4:2:0(A)-LM-c_u" and "4:2:0(A)-LO-c_u", also indicating the visual effect merit of our luma modification method.

2) THE BD-RATE MERIT

The quality-bitrate tradeoff metric "BD-rate" [1] indicates the average bitrate reduction percentage under the same quality circumstance. When setting four QP intervals, namely

TABLE 3. Average BD-rate (%) merit of our LUMA modification method relative to the LM method [4] and the LO method [11].

BASELINE		QP [4, 20]	QP [12, 24]	QP [20, 32]	QP [28, 40]
4:2:0(A)-LM-NN	4:2:0(A)-LM-$INTER_{CUBIC}$	-28.2011	-19.4648	-10.0312	-4.9524
	4:2:0(A)-LO-$INTER_{CUBIC}$	-26.3424	-18.5149	-9.8938	-4.6914
	4:2:0(A)-Ours-$INTER_{CUBIC}$	**-35.2960**	**-24.0350**	**-12.0350**	**-5.7430**
4:2:0(L)-LM-NN	4:2:0(L)-LM-$INTER_{CUBIC}$	-51.3053	-33.2737	-16.2604	-8.0706
	4:2:0(L)-LO-$INTER_{CUBIC}$	-50.4927	-32.7949	-16.7172	-8.1473
	4:2:0(L)-Ours-$INTER_{CUBIC}$	**-58.9572**	**-42.6885**	**-20.6550**	**-10.4733**
4:2:0(R)-LM-NN	4:2:0(R)-LM-$INTER_{CUBIC}$	-46.9029	-30.8102	-14.2375	-6.3351
	4:2:0(R)-LO-$INTER_{CUBIC}$	-46.5727	-30.7609	-15.1569	-7.0160
	4:2:0(R)-Ours-$INTER_{CUBIC}$	**-54.9912**	**-39.4583**	**-17.7729**	**-8.0296**
4:2:0(D)-LM-NN	4:2:0(D)-LM-$INTER_{CUBIC}$	-65.2660	-46.0250	-20.5181	-9.5217
	4:2:0(D)-LO-$INTER_{CUBIC}$	-66.1487	-46.6353	-22.1845	-10.9033
	4:2:0(D)-Ours-$INTER_{CUBIC}$	**-70.3740**	**-58.5790**	**-27.1803**	**-13.5370**
Anchor-LM-NN	Anchor-LM-$INTER_{CUBIC}$	-48.2139	-31.4733	-15.2844	-7.5494
	Anchor-LO-$INTER_{CUBIC}$	-48.2022	-31.8881	-16.1559	-7.8624
	Anchor-Ours-$INTER_{CUBIC}$	**-56.2545**	**-39.7256**	**-19.3293**	**-9.5309**

[4, 20], [12, 24], [20, 32], and [28, 40], we adopt the BD-rate metric to demonstrate the quality-bitrate tradeoff merit of our luma modification method.

We take the five combinations in C_s×LM-NN [4] as the five baselines. Then, based on the four QP intervals, the BD-rates of our five combinations, namely C_s×Ours-$INTER_{CUBIC}$ and the ten comparative combinations, namely C_s×LM-$INTER_{CUBIC}$ [4] and C_s×LO-$INTER_{CUBIC}$ [11], are used to show the quality-bitrate tradeoff merit of our luma modification method "Ours". From Table 3, we observe that for each $c_s \in C_s$, our combination "c_s-Ours-$INTER_{CUBIC}$" has better BD-rate performance in boldface among the three considered combinations, indicating the BD-rate merit of our luma modification.

V. CONCLUSION

For I^{RGB}, we have presented the proposed BCI-based iterative luma modification method for the CSFLM scheme. First, we propose a new and more effective BCI-based pixel-distortion function to measure the SSE between the ground truth RGB full-color pixel and the estimated one at the server side. Next, we prove that the proposed pixel-distortion function is a convex function. Then, we transform the luma modification problem in CSFLM to an overdetermined system, and we apply the pseudo-inverse technique to obtain the initial luma modification solution. Finally, based on the convex property of the proposed pixel-distortion function, a BCI-based iterative luma modification method is proposed to better enhance the quality of the reconstructed RGB full-color image. Based on the five datasets, the comprehensive experimental results have demonstrated the quality enhancement and the quality-bitrate tradeoff merits of our luma modification method relative to the state-of-the-art methods [4], [11].

The future work is to deploy our BCI-based luma modification method in Zhu *et al.'s* DCT quantization and color cross-space distortion minimization-based image compression method [21], and deploy our method in the cholesky decomposition, chroma subsampling, and luma modification-based color image compression method [20] for achieving better quality performance of the reconstructed images.

ACKNOWLEDGMENT

The authors appreciate the valuable comments of the four anonymous reviewers and the proofreading help of Ms. C. Harrington to improve the manuscript.

REFERENCES

[1] G. Bjøntegaard, *Calculation Average PSNR Difference Between RD-Curves*, document ITU-T SG16 Q.6, VCEG-M33, 2001.
[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. 2004, p. 67.
[3] G. Bradski and A. Kaehler, *Learning OpenCV: Comput. Vision With the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, 2008.
[4] 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.
[5] *Execution Code of the Our BCI-Based Iterative Luma Modifi-Cation Method*. [Online]. Available: http://140.118.175.164/CSFLM/CSFLM.zip
[6] *HM-16.18*. Accessed: Jan. 2018. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.18/
[7] *IMAX True Color Image Collection*. Accessed: Aug. 2014. [Online]. Available: https://www4.comp.polyu.edu.hk/ cslzhang/CDM_Dataset.htm
[8] *Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-screen 16:9 Aspect Ratios*, document Rec. ITU-R BT-601-5, ITU, 2011.
[9] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.
[10] *Kodak True Color Image Collection*. Accessed: Aug. 2014. [Online]. Available: https://www.math.purdue.edu/~lucier/PHOTO_CD/BMP_IMAGES/
[11] T.-L. Lin, B.-H. Liu, and K.-H. Jiang, "An efficient algorithm for luminance optimization in chroma downsampling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3719–3724, Sep. 2021.
[12] A. Luthra, E. François, and W. Husak, *Call for Evidence (CfE) for HDR and WCG Video Coding*, Standard ISO/IEC JTC1/SC29/WG11 (MPEG), Geneva, Switzerland, Feb. 2015.
[13] *Open CV Codes*. Accessed: Nov. 2018. [Online]. Available: https://sourceforge.net/projects/opencvlibrary/?fbclid=IwAR2vXid_ZecTmGKWiCuOe2XuaCKdAq4-Aes7FfeEVIpDM-BwOJAgg6TbI-k
[14] *SCI Image Database*. Accessed: Jan. 10, 2016. [Online]. Available: http://140.118.175.164/SCI
[15] *Spatial Scalability Filters*, document ISO/IEC JTC1/SC29/WG11 ITU-T SG 16 Q.6, Jul. 2005.
[16] M. Tachi, "Image processing device, image processing method, and program pertaining to image correction," U.S. Patent 8 314 863, Nov. 20, 2012.
[17] *The Video Dataset*. Accessed: Apr. 2020. [Online]. Available: ftp://140.118.175.164/CFASS/
[18] *VTM-11.0*. Accessed: 2021. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet//VVCSoftware_VTM
[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[20] S. Zhu, C. Cui, R. Xiong, Y. Guo, and B. Zeng, "Efficient chroma sub-sampling and Luma modification for color image compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1559–1563, May 2019.

[21] 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.

KUO-LIANG CHUNG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the National Taiwan University, Taipei, Taiwan, in 1982, 1984, and 1990, respectively. He has been a Chair Professor of the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, since 2009. His research interests include image processing, video compression, and deep learning. He was a recipient of the Distinguished Research Award, for the period 2004–2007 and 2019–2022, and the Distinguished Research Project Award, for the period 2009–2012, from the Ministry of Science and Technology of Taiwan. He has also been an Associate Editor of the *Journal of Visual Communication and Image Representation*, since 2011.

CHIH-YUAN HUANG received the B.S. degree in computer science and engineering from the National Changhua University of Education, Changhua, Taiwan, in 2019. He is currently pursuing the M.S. degree in computer science and information engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include image processing and video compression.

CHEN-WEI KAO received the B.S. degree in computer science and engineering from the National Taiwan Ocean University, Keelung, Taiwan, in 2019, and the M.S. degree in computer science and information engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2021. His research interests include image processing and video compression.