IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 10, 2022, accepted February 22, 2022, date of publication February 24, 2022, date of current version March 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3154487

A Cubic Convolution Interpolation-Based Chroma Subsampling Method for Bayer and RGBW CFA Raw Images

KUO-LIANG CHUNG, (Senior Member, IEEE), TSZ-CHING LEUNG, TZU-YI LIU, AND YA-CHI TSENG

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)

ABSTRACT The Bayer and RGBW color filter array (CFA) raw images, denoted by I^{Bayer} and I^{RGB}, respectively, have been widely used in consumer markets. In the demosaicking-first compression scheme, chroma subsampling is necessary prior to compressing I^{RGB} and I^{RGBW}. Several linear interpolation-based chroma subsampling methods have been developed for I^{Bayer}, but no nonlinear interpolation-based chroma subsampling methods have targeted the above two CFA image types simultaneously. In this paper, we first propose a nonlinear interpolation-based, namely the cubic convolution interpolation-based (CCI-based), 2×2 block-distortion function for each 2×2 CFA block B^{CbCr}. Next, using the Cauchy-Schwarz inequality, we prove that the proposed block-distortion function is a convex function in the real domain, which further serves as the base of the initial subsampled chroma solution. Then, a CCI-based iterative method is proposed to improve the initial subsampled chroma solution. The results of comprehensive experimental tests using Bayer and RGBW CFA images created from the three RGB full-color datasets, namely IMAX, SCI (screen content images), and CI (classical images), demonstrate that on the Versatile Video Coding (VVC) platform VTM-11.0, the proposed method achieves substantial quality enhancement and quality-bitrate tradeoff merits of the reconstructed Bayer and RGBW CFA images compared with existing chroma subsampling methods.

INDEX TERMS Bayer color filter array (CFA) image, chroma subsampling, convex block-distortion function, demosaicking, quality-bitrate tradeoff, quality enhancement, RGBW CFA image.

I. INTRODUCTION

To save hardware cost, most color digital cameras employ a single-sensor technology with a Bayer color filter array (CFA) to capture real-world scenes. In addition, professional photographers and designers tend to prefer to work with raw Bayer CFA images directly because they can maximize control over the post-processing [22], [32]. At the same time, compressing raw Bayer CFA images also has the potential to improve compression performance relative to working with RGB full-color images.

Accordingly, several compression-first schemes were developed to compress raw Bayer CFA images directly, and these schemes include the geometric rotation approach [23], the integer-reversible spectral–spatial color domain approach [25], and the structure conversion approach [7], [10]. However, the intrinsic difficulty in converting CFA structures to the working domain limit the compression performance. Therefore, a demosaicking-first compression scheme [18] was proposed for compressing Bayer CFA images such that the quality of the reconstructed Bayer CFA images has better performance.

In this study, not only the Bayer CFA image I^{Bayer}, we also consider the RGBW CFA image I^{RGBW}. For the former, the four 2×2 Bayer CFA patterns [1], as depicted in Figs. 1(a)-(d), have been widely used in modern color digital cameras. Each pixel in I^{Bayer} has only one R (red), G (green), or B (blue) color value. I^{Bayer} contains 25% R, 50% G, and 25% B color values. However, in the low illumination condition, the thermal noise side effect of I^{Bayer} results in low quality of the reconstructed RGB full-color image [16]. To receive more luminance, three RGBW CFA

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

patterns, namely the RGBW-Sony CFA pattern [33], the RGBW-Kodak CFA pattern [5], and Yamagami *et al.*'s CFA pattern [37], were proposed and are depicted in Figs. 2(a)-(c), respectively. Each pixel in I^{RGBW} has only one R, G, B, or W (white) color value. I^{RGBW} contains 12.5% R, 25% G, 12.5% B, and 50% W color values.

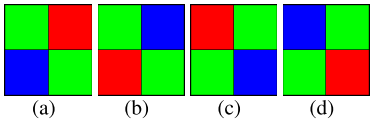

FIGURE 1. Four 2 × 2 Bayer CFA patterns. (a) $pat_1 = <G_1, R_2, B_3, G_4>$. (b) $pat_2 = <G_1, B_2, R_3, G_4>$. (c) $pat_3 = <R_1, G_2, G_3, B_4>$. (d) $pat_4 = <B_1, G_2, G_3, R_4>$.

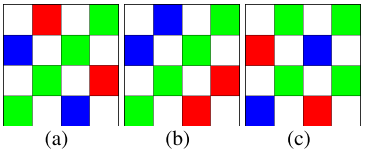

FIGURE 2. Three 4 × 4 RGBW CFA patterns. (a) The RGBW-Sony pattern [33]. (b) The RGBW-Kodak pattern [5]. (c) Yamagami *et al.*'s pattern [37].

In the demosaicking-first chroma subsampling scheme, as depicted in Fig. 3, both I^{Bayer} and I^{RGBW} should be first demosaicked to RGB full-color images at the server side, and then the demosaicked RGB full-color images are converted to YCbCr images which serve as the base of chroma subsampling. To demosaic I^{Bayer}, the methods in [15], [26], [39], [41] can be used; to demosaic I^{RGBW}, the methods in [6], [29] can be used. Here, we take Kiku *et al.*'s code [15] to demosaic I^{Bayer} and take Condat's code [6] to demosaic I^{RGBW}. After that, the demosaicked RGB full-color image I^{RGB} is converted to a YCbCr image I^{YCbCr} by using the BT.709-5 color conversion [13]:

$$\begin{bmatrix} Y_i \\ C_{b_i} \\ C_{r_i} \end{bmatrix} = \begin{bmatrix} 0.183 & 0.614 & 0.062 \\ -0.101 & -0.338 & 0.439 \\ 0.439 & -0.399 & -0.040 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

Then, for each 2 × 2 CbCr block B^{CbCr} in I^{CbCr}, one 4:2:0 subsampling method is used to obtain the subsampled (Cb, Cr)-pair of B^{CbCr}. Accordingly, the subsampled CbCr image $I^{sub,CbCr}$ and the luma image I^{Y} are fed into the encoder for compression. The lack of correlation between I^{Y} and $I^{sub,CbCr}$ greatly contributes to the effectiveness of the image compression. After encoding $I^{sub,CbCr}$ and I^{Y}, the encoded bitstream is transmitted to the decoder via the internet.

At the client side in Fig. 3, the decoded subsampled CbCr image $I^{sub,CbCr}$ is upsampled by the selected chroma upsampling process. Based on the 2 × 2 Bayer CFA pattern, the decompressed CbCr image and luma image can be directly transformed to the reconstructed Bayer CFA image $I^{rec,Bayer}$

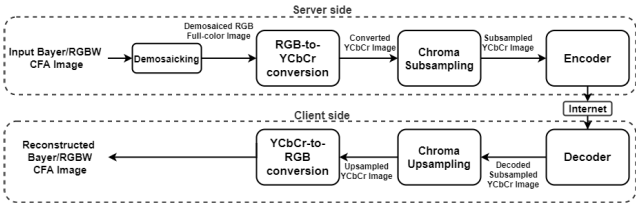

FIGURE 3. The demosaicking-first chroma subsampling scheme in the coding system.

using the following YCbCr-to-RGB conversion:

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.793 \\ 1.164 & -0.213 & -0.534 \\ 1.164 & 2.115 & 0 \end{bmatrix} \begin{bmatrix} Y_i - 16 \\ C_{b_i} - 128 \\ C_{r_i} - 128 \end{bmatrix} \quad (2)$$

Based on the 4 × 4 RGBW CFA pattern, the R, G, and B color pixels in the reconstructed RGBW CFA image $I^{rec,RGBW}$ can be obtained directly using Eq. (2), while the W color pixels in $I^{rec,RGBW}$ can be obtained by averaging the reconstructed R, G, and B color values at the same location.

A. RELATED CHROMA SUBSAMPLING WORKS

Chroma subsampling has a long history. Without considering the CFA patterns, the commonly used eight chroma subsampling methods include the six traditional methods, namely 4:2:0(DIRECT), 4:2:0(A), 4:2:0(L), 4:2:0(R), 4:2:0(MPEG-B) [31], and Anchor [21], and the two classical methods, namely the IDID (interpolation-dependent image downsampling) method [43] and the JCDU (joint chroma downsampling and upsampling) method [36]. Considering the Bayer CFA patterns, the four state-of-the-art chroma subsampling works for I^{Bayer} include the methods in [3], [4], [18], [19]. To the best of our knowledge, in the literature, no chroma subsampling methods targeting only I^{RGBW} have ever been proposed.

1) EIGHT CHROMA SUBSAMPLING METHODS WITHOUT CONSIDERING THE CFA PATTERNS

4:2:0(DIRECT) selects the top-left chroma pair of B^{CbCr} as the subsampled (Cb, Cr)-pair. 4:2:0(A) averages the four chroma pairs of B^{CbCr} as the subsampled chroma pair. 4:2:0(L) and 4:2:0(R) obtain their chroma pairs by averaging the chroma pairs in the left and right columns of B^{CbCr}, respectively. Here, 'chroma pair' and '(Cb, Cr)-pair' denote the same thing. 4:2:0(MPEG-B) determines the subsampled chroma pair of B^{CbCr} by performing the 13-tap filter $[2, 0, -4, -3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64$ on the top-left location of B^{CbCr}. The Anchor method [21] consists of two steps. The first step performs a 3-tap filter $[1, 6, 1]/8$ at the leftmost location of each row of B^{CbCr}; the second step performs a 3-tap filter $([0, 4, 4]/8)^T$ at the top-left location of B^{CbCr}, which has been updated by the first step.

Zhang *et al.* [43] proposed an IDID method that prefers the chroma upsampling process NEDI (new edge-directed interpolation) [17] used at the client side. To improve the

IDID method, considering the palette mode used for screen content images [20], Wang *et al.* [36] proposed a JCDU method. The JCDU method prefers the chroma upsampling process BICU (bicubic interpolation) [27], [28] used at the client side.

2) FOUR BAYER CFA PATTERN-BASED CHROMA SUBSAMPLING METHODS

Different from the eight chroma subsampling methods mentioned above Chen *et al.* [3] proposed a direct mapping (DM) method when the input image is I^{Bayer} based on the following observation: in Eq. (2), the R value is dominated by the Y and Cr values, and the B value is dominated by the Y and Cb values. Therefore, the subsampled chroma pair of B^{CbCr} equals (Cb_3, Cr_2). Lin *et al.* [18] proposed a COPY-based 2×2 Bayer CFA block-distortion function in which the operator COPY is a special linear interpolation used to copy the subsampled chroma-pair parameter of B^{CbCr}, denoted by (Cb_s, Cr_s), as the four estimated chroma pairs of B^{CbCr}. Then, based on the COPY-based block-distortion function, they applied the differentiation (DI) technique to determine the solution of (Cb_s, Cr_s) as the subsampled chroma pair of B^{CbCr}. The DI method prefers the chroma upsampling process 'COPY' used at the client side. In fact, the upsampling process 'COPY' equals the process 'NN (nearest neighbor)' supported by the VVC (Versatile Video Coding) platform VTM-11.0 [34].

To improve the DI method, Chung *et al.* [4] proposed a COPY- and gradient-descent (GD)-based chroma subsampling method. In the GD method, the block-distortion function is still based on the COPY-based approach, but they utilized the value verification technique to prove that their block-distortion function is a convex function, leading to the design of the GD- and BILI-based (bilinear interpolation-based) iteration method to improve the DI method. The GD method prefers the chroma upsampling process BILI [27], [28] used at the client side. Based on the DI method [18] but considering the demosaicked RGB full-color block-distortion, Lin *et al.* [19] proposed a 'modified 4:2:0(A)' chroma subsampling method that selects the best case among the four subsampled chroma pairs of B^{CbCr} by considering the ceiling operation-based 4:2:0(A) and the floor operation-based 4:2:0(A). The 'modified 4:2:0(A)' method prefers the chroma upsampling process TN (three neighboring reference pixel-based method) [40] used at the client side.

B. MOTIVATION

To the best of our knowledge, no chroma subsampling methods targeting I^{Bayer} can also be used for I^{RGBW}, and no chroma subsampling methods targeting only I^{RGBW} have ever been proposed. In addition, for I^{Bayer}, the block-distortion function used in the state-of-the-art chroma subsampling methods [4], [18], [19] is based on a special linear interpolation approach, i.e., the COPY-based approach, limiting the quality performance.

The above reasons motivated us to design a new and more effective nonlinear interpolation-based method, namely the cubic convolution interpolation-based (CCI-based) chroma subsampling method, to handle chroma subsampling for I^{Bayer} and I^{RGBW} simultaneously, achieving substantial quality enhancement and quality-bitrate tradeoff improvements of the reconstructed Bayer and RGBW CFA images.

C. CONTRIBUTIONS

The four major contributions of the proposed new CCI-based chroma subsampling method for I^{Bayer} and I^{RGBW} related to the previously published papers [3], [4], [18], [19] for only I^{Bayer} are clarified as follows.

(1) In the previously published paper [3], based on the observation from the YCbCr-to-RGB color transformation, the subsampled chroma pair of B^{CbCr} equals (Cb_3, Cr_2). Instead of the COPY-based block-distortion functions used for only I^{Bayer} [4], [18], [19], in this paper, a new and more effective nonlinear interpolation-based, namely the cubic convolution interpolation-based (CCI-based), block-distortion function is proposed for I^{Bayer} and I^{RGBW} simultaneously.

(2) In [4], using the value verification approach for only the Bayer CFA pattern in Fig. 1(a), i.e., pat_1, the determinant of the Hessian matrix of the COPY-based block-distortion function is 1.5577, proving the convex property of their block-distortion function. In this paper, using the Cauchy-Schwarz inequality, we provide solid proof that our CCI-based block-distortion function is a convex function for all ten considered Bayer and RGBW CFA patterns, i.e., pat_i for $1 \leq i \leq 10$. In particular, for the Bayer CFA pattern 'pat_1', the determinant of the Hessian matrix of our CCI-based block-distortion function is 22.0337 versus 1.5577 [4].

(3) Using the convex property of our CCI-based block-distortion function, we derive a formula to determine the initial subsampled chroma solution of B^{CbCr}. Furthermore, we propose an iterative method to refine the initial subsampled chroma solution for I^{Bayer} and I^{RGBW} simultaneously. In particular, for I^{Bayer}, the experimental results have demonstrated that when compared with the two combinations, DI-COPY [18] and GD-BILI [4], the PSNR (peak signal-to-noise ratio) gains of our combination are 2.4841 dB and 2.0531 dB, respectively, when setting QP (quantization parameter) to zero.

(4) Based on the ground truth Bayer and RGBW CFA images which are created from the three RGB full-color datasets, namely IMAX [12], SCI (screen content images) [30], and CI (classical images) [44], the comprehensive experimental results show that on VTM-11.0 [34], our method achieves the best PSNR, SSIM (structural similarity index measure) [35], FSIM (feature similarity index measure) [42], visual effect, and BD-PSNR (Bjøntegaard delta PSNR difference) [2] performance of the reconstructed Bayer and RGBW CFA images when compared with the six traditional methods and the six state-of-the-art methods [3], [4], [18], [19], [36], [43].

The remainder of this paper is organized as follows. In Section II, the proposed CCI-based block-distortion function is presented. In Section III, we prove the convex property of the proposed block-distortion function, and then the proposed iterative chroma subsampling method is presented. In Section IV, the experimental results are provided to justify the quality and quality-bitrate tradeoff merits of our method. In Section V, some concluding remarks are offered.

II. THE PROPOSED CCI-BASED BLOCK-DISTORTION FUNCTION

Different from the COPY-based block-distortion functions used in the previous works [4], [18], [19] only for I^{Bayer}, we propose a new and more effective CCI-based 2×2 block-distortion function for I^{Bayer} and I^{RGBW} simultaneously.

Because the four Bayer CFA patterns in Fig. 1, denoted by pat_i for $1 \le i \le 4$, are all 2×2 blocks, to handle the 2×2 Bayer and 4×4 RGBW CFA blocks simultaneously, the three 4×4 RGBW CFA patterns in Fig. 2 are partitioned into six basic 2×2 RGBW CFA patterns, namely pat_i for $5 \le i \le 10$, in Figs. 4(a)-(f). According to the row-major scanning order, the three 4×4 RGBW CFA patterns in Fig. 2 are expressed as the following three ordered sets:

$$\begin{aligned} \text{RGBW-Sony CFA pattern} &= <pat_5, pat_6, pat_6, pat_5> \\ \text{RGBW-Kodak CFA pattern} &= <pat_7, pat_6, pat_6, pat_8> \\ \text{Yamagami } et\ al.\text{'s CFA pattern} &= <pat_9, pat_{10}, pat_{10}, pat_9> \end{aligned} \tag{3}$$

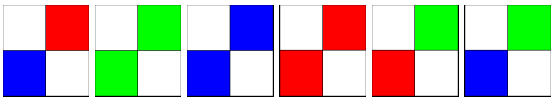

FIGURE 4. The partitioned six basic 2 × 2 RGBW CFA patterns for Fig. 2(a)-(c). (a) $pat_5 = <W_1, R_2, B_3, W_4>$. (b) $pat_6 = <W_1, G_2, G_3, W_4>$. (c) $pat_7 = <W_1, B_2, B_3, W_4>$. (d)$pat_8 = <W_1, R_2, R_3, W_4>$. (e) $pat_9 = <W_1, G_2, R_3, W_4>$. (f) $pat_{10} = <W_1, G_2, B_3, W_4>$.

A. THE CCI-BASED METHOD TO ESTIMATE THE FOUR CHROMA PAIRS OF B^{CbCr}

Before presenting the proposed CCI-based block-distortion function, we first propose a more effective CCI-based estimation method to estimate the four chroma pairs of B^{CbCr} compared with the COPY-based estimation used in [4], [18], [19].

For simplicity, we only describe how to estimate the four Cb entries of B^{Cb}. The estimation of the four Cr entries of B^{Cr} can be easily followed. For easy exposition, only the top-left Cb component of B^{Cb}, namely Cb_1^{est} in Fig. 5, is estimated, and the other three estimated Cb components, namely Cb_2^{est}, Cb_3^{est}, and Cb_4^{est}, can be obtained using the same method. Let each 2×2 Cb block in Fig. 5 be viewed as a 1×1 macropixel. Initially, we put the origin of the x-y coordinate system at the location of Cb_1^{est}. Therefore, Cb_1^{est} is located at (0, 0), and the locations of the subsampled Cb parameter Cb_s and the neighboring reference subsampled Cb values are illustrated in Fig. 5, where the subsampled Cb parameter Cb_s (= $Cb_{2,2}$) is located at (0.25, -0.25), $Cb_{1,2}$ is located at (0.25, 0.75), and so on.

We deploy the cubic convolution interpolation, which was originally proposed by Keys [14], in the estimation of Cb_1^{est}. The value of the weight function assigned to each reference point $Cb_{i,j}$, as depicted by one black bullet with yellow background in Fig. 5, is dependent on the vertical distance $d_{i,j}^v$ and the horizontal distance $d_{i,j}^h$ relative to the location of Cb_1^{est}. As mentioned before, Cb_1^{est} is located at (0, 0). For example, for the reference point $Cb_{2,2}$ (= Cb_s) in Fig. 5, $d_{2,2}^v = 0.25$ (= 0.25 - 0), and $d_{2,2}^h = -0.25$ (= −0.25 - 0). Let d denote the vertical distance $d_{i,j}^v$ or the horizontal distance $d_{i,j}^h$. The weight function W(d) is expressed as the following cubic function:

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

Due to the constraint $|d|<2$, as depicted in Fig. 5, only 16 reference points in the 4×4 sliding window are needed. Among the 16 reference subsampled Cb values, the former 10 reference subsampled Cb values are obtained in advance by using the proposed chroma subsampling method, which will be formally presented in Section III. Except for the subsampled Cb parameter Cb_s, the five future subsampled Cb values can be obtained by using one available subsampling method, e.g., 4:2:0(A). The value of the weight function $\hat{W}(i,j)$ assigned to $Cb_{i,j}$ is thus expressed as

$$\hat{W}(i,j) = W(d_{i,j}^v) W(d_{i,j}^h) \tag{5}$$

Consequently, the estimation of Cb_1^{est} is expressed as

$$\begin{aligned} Cb_1^{est} &= \sum_{i=0}^{4} \sum_{j=0}^{4} \hat{W}(i,j) Cb_{i,j} \\ &= \hat{W}(2,2) Cb_{2,2} + \sum_{i=0}^{4} \sum_{\substack{j=0 \\ (i,j) \ne (2,2)}}^{4} \hat{W}(i,j) Cb_{i,j} \\ &= \hat{W}(2,2) Cb_s + Cb_1^{const} \end{aligned} \tag{6}$$

In Eq. (6), the estimation of Cb_1^{est} is expressed as the sum of 16 products.

Furthermore, as depicted in Fig. 6(a), we first put the origin of the x-y coordinate system at the location of Cb_2^{est}, and then we move the 4×4 sliding window one pixel to the right to estimate Cb_2^{est}. By the same argument, we move the 4×4 sliding window one pixel downward to estimate Cb_4^{est}, and then we move the 4×4 sliding window one pixel to the left to estimate Cb_3^{est}. It is notable that the above cubic convolution interpolation-based computation method for estimating the four entries of B^{CbCr} can explain why our method is called the cubic convolution interpolation-based, i.e., the CCI-based, method.

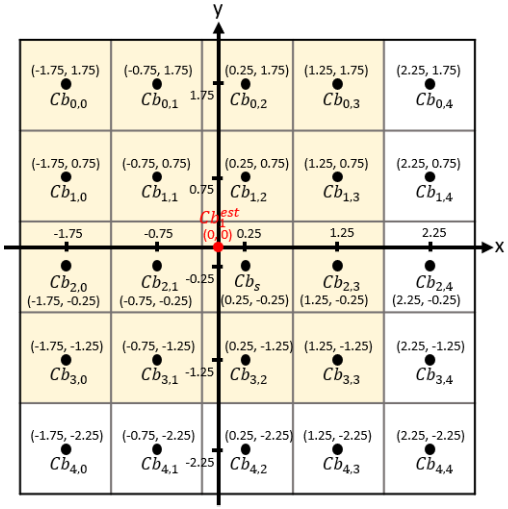

FIGURE 5. The coordinate system used in our CCI-based method to estimate Cb_1^{est} of B^{Cb} within the 4 × 4 sliding window marked in yellow.

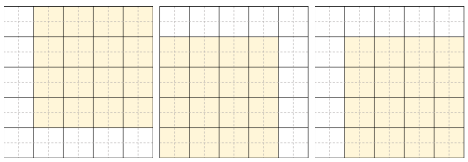

FIGURE 6. The 16 reference subsampled Cb points referred to by our CCI-based chroma estimation method. (a) For Cb_2^{est}. (b) For Cb_3^{est}. (c) For Cb_4^{est}.

B. THE PROPOSED CCI-BASED CFA BLOCK-DISTORTION FUNCTION

After presenting the proposed CCI-based estimation method for estimating the four chroma pairs of B^{CbCr}, the estimated 2×2 CbCr block is denoted by $B^{est,CbCr}$. Based on $B^{est,CbCr}$, the collocated 2×2 luma block B^{Y}, and the 2×2 CFA pattern pat_i, which corresponds to the input ground truth 2×2 CFA raw block B^{pat_i}, by Eq. (2), the estimated 2×2 CFA block B^{est,pat_i} can be obtained at the server side.

We now present the proposed CCI-based 2×2 CFA block-distortion function to measure the sum of squared errors (SSE) between each 2×2 ground truth CFA block B^{pat_i} and the estimated 2×2 CFA block B^{est,pat_i} for $1\le i\le 10$. For easy exposition, we take the 2×2 Bayer CFA pattern pat_1 in Fig. 1(a) as the representative for I^{Bayer}. Accordingly, our block-distortion between B^{pat_1} $(=(G_1,R_2,B_3,G_4))$ and B^{est,pat_1} $(=(G_1^{est},R_2^{est},B_3^{est},G_4^{est}))$ is expressed as

$$\begin{aligned} D^{pat_1} &= ||(G_1,R_2,B_3,G_4)\text{-}(G_1^{est},R_2^{est},B_3^{est},G_4^{est})||^2 \\ &= (G_1-G_1^{est})^2+(R_2-R_2^{est})^2 \\ &\quad +(B_3-B_3^{est})^2+(G_4-G_4^{est})^2 \\ &= \sum_{k\in\{1,4\}}(G_k-G_k^{est})^2+(R_2-R_2^{est})^2+(B_3-B_3^{est})^2 \end{aligned} \tag{7}$$

where $||.||$ denotes a 2-norm operation.

Using the YCbCr-to-RGB transformation in Eq. (2), our block-distortion function in Eq. (7) can be rewritten as

$$\begin{aligned} &D^{pat_1}(Cb_s,Cr_s) \\ &= \sum_{k\in\{1,4\}}[(1.164(Y_k-16)-0.213(Cb_k-128) \\ &\quad -0.534(Cr_k-128)) \\ &\quad -(1.164(Y_k-16)-0.213(Cb_k^{est}-128) \\ &\quad -0.534(Cr_k^{est}-128))]^2 \\ &\quad +[(1.164(Y_2-16)+1.793(Cr_2-128)) \\ &\quad -(1.164(Y_2-16)+1.793(Cr_2^{est}-128))]^2 \\ &\quad +[(1.164(Y_3-16)+2.115(Cb_3-128)) \\ &\quad -(1.164(Y_3-16)+2.115(Cb_3^{est}-128))]^2 \\ &= \sum_{k\in\{1,4\}}[-0.213(Cb_k-Cb_k^{est})-0.534(Cr_k-Cr_k^{est})]^2 \\ &\quad +[1.793(Cr_2-Cr_2^{est})]^2+[2.115(Cb_3-Cb_3^{est})]^2 \\ &= \sum_{k=1}^{4}[\alpha_k(Cb_k-Cb_k^{est})+\beta_k(Cr_k-Cr_k^{est})]^2 \\ &= \sum_{k=1}^{4}[\alpha_k(Cb_k-(\hat{W}(2,2)Cb_s+Cb_k^{const})) \\ &\quad +\beta_k(Cr_k-(\hat{W}(2,2)Cr_s+Cr_k^{const}))]^2 \end{aligned} \tag{8}$$

with

$$\alpha_k=\begin{cases}-0.213, & \text{for k = 1, 4 (w.r.t. G color)}\\ 0, & \text{for k = 2 (w.r.t. R color)}\\ 2.115 & \text{for k = 3 (w.r.t. B color)}\end{cases}$$

$$\beta_k=\begin{cases}-0.534, & \text{for k = 1, 4 (w.r.t. G color)}\\ 1.793, & \text{for k = 2 (w.r.t. R color)}\\ 0 & \text{for k = 3 (w.r.t. B color)}\end{cases} \tag{9}$$

Furthermore, we take the 2×2 RGBW CFA pattern pat_6 as the representative for I^{RGBW}. Similarly, our block-distortion function to measure the SSE between B^{pat_6} $(=(W_1,G_2,G_3,W_4))$ and B^{est,pat_6} $(=(W_1^{est},G_2^{est},G_3^{est},W_4^{est}))$ is expressed as

$$\begin{aligned} D^{pat_6} &= ||(W_1,G_2,G_3,W_4)\text{-}(W_1^{est},G_2^{est},G_3^{est},W_4^{est})||^2 \\ &= \sum_{k\in\{1,4\}}(W_k-W_k^{est})^2+\sum_{k\in\{2,3\}}(G_k-G_k^{est})^2 \\ &= \sum_{k\in\{1,4\}}[W_k-(R_k^{est}+G_k^{est}+B_k^{est})/3]^2 \\ &\quad +\sum_{k\in\{2,3\}}(G_k-G_k^{est})^2 \end{aligned} \tag{10}$$

Using the YCbCr-to-RGB transformation in Eq. (2), Eq. (10) can be rewritten as

$$\begin{aligned} &D^{pat_6}(Cb_s,Cr_s) \\ &= \sum_{k\in\{1,4\}}[(1.164(Y_k-16)+1.902(Cb_k-128)/3 \\ &\quad +1.259(Cr_k-128)/3) \end{aligned}$$

$$
\begin{aligned}
&-(1.164(Y_k-16)+1.902(Cb_k^{est}-128)/3\\
&+1.259(Cr_k^{est}-128)/3)]^2\\
&+\sum_{k\in\{2,3\}}[(1.164(Y_k-16)-0.213(Cb_k-128)\\
&-0.534(Cr_k-128))\\
&-(1.164(Y_k-16)-0.213(Cb_k^{est}-128)\\
&-0.534(Cr_k^{est}-128))]^2\\
=&\sum_{k\in\{1,4\}}[1.902(Cb_k-Cb_k^{est})/3+1.259(Cr_k-Cr_k^{est})/3]^2\\
&+\sum_{k\in\{2,3\}}[-0.213(Cb_k-Cb_k^{est})-0.534(Cr_k-Cr_k^{est})]^2\\
=&\sum_{k=1}^{4}[\alpha_k(Cb_k-Cb_k^{est})+\beta_k(Cr_k-Cr_k^{est})]^2\\
=&\sum_{k=1}^{4}[\alpha_k(Cb_k-(\hat{W}(2,2)Cb_s+Cb_k^{const}))\\
&+\beta_k(Cr_k-(\hat{W}(2,2)Cr_s+Cr_k^{const}))]^2 \qquad (11)
\end{aligned}
$$

with

$$
\alpha_k=\begin{cases}1.902/3, & \text{for k = 1, 4 (w.r.t. W color)}\\ -0.213, & \text{for k = 2, 3 (w.r.t. G color)}\end{cases}
$$

$$
\beta_k=\begin{cases}1.259/3, & \text{for k = 1, 4 (w.r.t. W color)}\\ -0.534, & \text{for k = 2, 3 (w.r.t. G color)}\end{cases} \qquad (12)
$$

From the explicit form of the block-distortion $D^{pat_1}(Cb_s, Cr_s)$ in Eqs. (8)-(9) for I^{Bayer}, we deduce that the values of α_k and β_k, $1 \le k \le 4$, are dependent on the color of the kth entry of B^{pat_1}. Similarly, from the explicit form of the block-distortion $D^{pat_6}(Cb_s, Cr_s)$ in Eqs. (11)-(12), we deduce that the values of α_k and β_k, $1 \le k \le 4$, are also dependent on the color of the kth entry of B^{pat_6}. Let $pat_{i,k}$ denote the kth entry of pat_i, $1 \le k \le 4$ and $1 \le i \le 10$. For example, for $pat_1 = <pat_{1,1}, pat_{1,2}, pat_{1,3}, pat_{1,4}>$, we have $pat_{1,1}$ = G, $pat_{1,2}$ = R, $pat_{1,3}$ = B, and $pat_{1,4}$ = G. Table 1 lists the values of $\alpha_{pat_{i,k}}$ and $\beta_{pat_{i,k}}$ for $1 \le k \le 4$ and $1 \le i \le 10$.

Consequently, our general 2 × 2 CFA block-distortion function, which is used to measure the SSE between B^{pat_i} and B^{est,pat_i}, $1 \le i \le 10$, is expressed as

$$
\begin{aligned}
&D^{pat_i}(Cb_s, Cr_s)\\
&=\sum_{k=1}^{4}[\alpha_{pat_{i,k}}(Cb_k-(\hat{W}(2,2)Cb_s+Cb_k^{const}))\\
&+\beta_{pat_{i,k}}(Cr_k-(\hat{W}(2,2)Cr_s+Cr_k^{const}))]^2 \qquad (13)
\end{aligned}
$$

TABLE 1. The values of $\alpha_{pat_{i,k}}$ and $\beta_{pat_{i,k}}$, $1 \le i \le 10$ and $1 \le k \le 4$.

$pat_{i,k}$	R	G	B	W
$\alpha_{pat_{i,k}}$	0	-0.213	2.115	1.902/3
$\beta_{pat_{i,k}}$	1.793	-0.534	0	1.259/3

III. THE PROPOSED CCI-BASED ITERATIVE CHROMA SUBSAMPLING METHOD

For the considered Bayer and RGBW CFA image types, we first provide solid proof to show that our 2 × 2 CFA block-distortion function in Eq. (13) is a convex function in the real domain, which further serves as the base of the initial subsampled chroma solution of B^{CbCr}. Next, a formula is derived to calculate the initial subsampled chroma solution. Furthermore, we propose an iterative method to refine the initial subsampled chroma solution.

A. CONVEX PROPERTY PROOF OF THE PROPOSED 2 × 2 CFA BLOCK-DISTORTION FUNCTION

Let $det(H(D^{pat_i}(Cb_s, Cr_s)))$ denote the determinant of the Hessian matrix of $D^{pat_i}(Cb_s, Cr_s)$ in Eq. (13). According to the convex function definition [24], if $det(H(D^{pat_i}(Cb_s, Cr_s)))$ is positive, the proposed block-distortion function $D^{pat_i}(Cb_s, Cr_s)$ is a convex function for I^{Bayer} and I^{RGBW}. The Hessian matrix of $D^{pat_i}(Cb_s, Cr_s)$ is expressed as

$$
H(D^{pat_i}(Cb_s, Cr_s))=\begin{bmatrix}\frac{\partial^2 D^{pat_i}}{\partial Cb_s^2} & \frac{\partial^2 D^{pat_i}}{\partial Cb_s \partial Cr_s}\\ \frac{\partial^2 D^{pat_i}}{\partial Cr_s \partial Cb_s} & \frac{\partial^2 D^{pat_i}}{\partial Cr_s^2}\end{bmatrix} \qquad (14)
$$

for $1 \le i \le 10$ with

$$
\begin{aligned}
\frac{\partial^2 D^{pat_i}}{\partial Cb_s^2}&=2\hat{W}(2,2)^2\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\\
\frac{\partial^2 D^{pat_i}}{\partial Cr_s^2}&=2\hat{W}(2,2)^2\sum_{k=1}^{4}\beta_{pat_{i,k}}^2\\
\frac{\partial^2 D^{pat_i}}{\partial Cb_s \partial Cr_s}&=\frac{\partial^2 D^{pat_i}}{\partial Cr_s \partial Cb_s}=2\hat{W}(2,2)^2\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}} \qquad (15)
\end{aligned}
$$

Then, the determinant of $H(D^{pat_i}(Cb_s, Cr_s))$ is expressed as

$$
\begin{aligned}
&det(H(D^{pat_i}))\\
&=2\hat{W}(2,2)^2\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\times 2\hat{W}(2,2)^2\sum_{k=1}^{4}\beta_{pat_{i,k}}^2\\
&-2\hat{W}(2,2)^2\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}}\\
&\times 2\hat{W}(2,2)^2\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}}\\
&=4\hat{W}(2,2)^4[\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\sum_{k=1}^{4}\beta_{pat_{i,k}}^2-(\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}})^2] \qquad (16)
\end{aligned}
$$

Let $v_1 = <\alpha_{pat_{i,1}}, \alpha_{pat_{i,2}}, \alpha_{pat_{i,3}}, \alpha_{pat_{i,4}}>$ and $v_2 = <\beta_{pat_{i,1}}, \beta_{pat_{i,2}}, \beta_{pat_{i,3}}, \beta_{pat_{i,4}}>$ be two 1 × 4 vectors. Using the Cauchy-Schwarz inequality [11], we know $||v_1||^2||v_2||^2 \ge <v_1, v_2>^2$, where $<.>$ denotes the inner product operation.

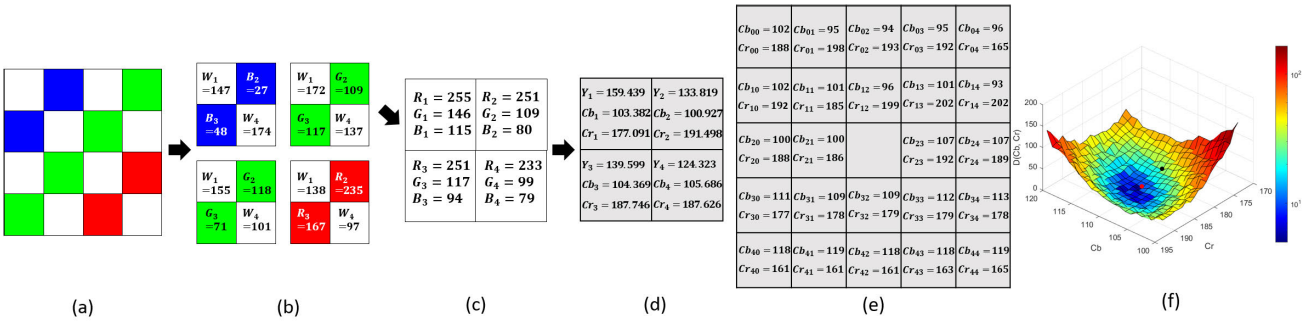

FIGURE 7. The idea in the proposed iterative chroma subsampling method. (a) One 4 × 4 RGBW CFA block example. (b) The four partitioned blocks with CFA patterns pat_7, pat_6, pat_6, and pat_8. (c) The top-right demosaicked 2 × 2 RGBW CFA block of Fig. 7(b). (d) The converted YCbCr block of Fig. 7(c). (e) The neighboring reference subsampled (Cb, Cr)-pairs. (f) The grid plot of the block-distortion function $D^{pat_6}(Cb_s, Cr_s)$ in the integer domain, where the black circle denotes the initial subsampled chroma solution and the red circle denotes the desirable subsampled chroma solution.

It thus yields

$$\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\sum_{k=1}^{4}\beta_{pat_{i,k}}^2 \geq (\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}})^2 \tag{17}$$

From Table 1, we know that the vector v_1 ($= <\alpha_{pat_{i,1}}, \alpha_{pat_{i,2}}, \alpha_{pat_{i,3}}, \alpha_{pat_{i,4}}>$) and the vector v_2 ($= <\beta_{pat_{i,1}}, \beta_{pat_{i,2}}, \beta_{pat_{i,3}}, \beta_{pat_{i,4}}>$) are not parallel with each other. On the other hand, the condition '$||v_1||^2||v_2||^2 \neq <v_1, v_2>^2$' is always true. Therefore, Eq. (17) is simplified as the following strict inequality:

$$\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\sum_{k=1}^{4}\beta_{pat_{i,k}}^2 > (\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}})^2 \tag{18}$$

Putting the strict inequality in Eq. (18) back on the right-hand side of Eq. (16) yields $det(H(D^{pat_i})) > 0$ for $1 \leq i \leq 10$, implying that $H(D^{pat_i})$ is positive definite. For pat_i, $1 \leq i \leq 10$, we thus prove that the proposed general 2 × 2 CFA block-distortion function in Eq. (13) is a convex function.

Proposition 1: The proposed CCI-based 2 × 2 *CFA block-distortion function in Eq. (13) for* pat_i, $1 \leq i \leq 10$, *is a convex function.*

Compared with the COPY-based block-distortion function [4], [18], [19] used only for the 2 × 2 Bayer CFA pattern pat_1 in Fig. 1(a), the proposed CCI-based block-distortion function for pat_i, $1 \leq i \leq 10$, is more effective and general. It is notable that for pat_1 in Fig. 1(a), the determinant of the Hessian matrix of our CCI-based block-distortion function is 22.0337, while the determinant of the Hessian matrix of the previous COPY-based block-distortion function is 1.5577.

B. DETERMINING THE INITIAL SUBSAMPLED CHROMA SOLUTION OF B^{CbCr}

Proposition 1 implies that in the real domain, a critical point exists in the convex block-distortion function in Eq. (13) at which the convex block-distortion function has a minimal value. However, practically, we consider the nonnegative integer domain for chroma subsampling. Therefore, the critical point of Eq. (13) is used as the initial subsampled chroma solution in our iterative method.

To determine the critical point of Eq. (13), we take the first derivative on Eq. (13) with the subsampled chroma parameters Cb_s and Cr_s, respectively. Setting the two derivatives to zero yields

$$\begin{aligned}\frac{\partial D^{pat_i}(Cb_s, Cr_s)}{\partial Cb_s} &= 0\\ \frac{\partial D^{pat_i}(Cb_s, Cr_s)}{\partial Cr_s} &= 0.\end{aligned} \tag{19}$$

Furthermore, the explicit form of Eq. (19) is expressed as

$$\begin{aligned}&\sum_{k=1}^{4}\{2[\alpha_{pat_{i,k}}(Cb_k - \hat{W}(2,2)Cb_s - Cb_k^{const})\\ &\quad +\beta_{pat_{i,k}}(Cr_k - \hat{W}(2,2)Cr_s - Cr_k^{const})]\\ &\quad \times(-\hat{W}(2,2)\beta_{pat_{i,k}})\} = 0.\\ &\sum_{k=1}^{4}\{2[\alpha_{pat_{i,k}}(Cb_k - \hat{W}(2,2)Cb_s - Cb_k^{const})\\ &\quad +\beta_{pat_{i,k}}(Cr_k - \hat{W}(2,2)Cr_s - Cr_k^{const})]\\ &\quad \times(-\hat{W}(2,2)\alpha_{pat_{i,k}})\} = 0\end{aligned} \tag{20}$$

Solving Eq. (20), the solution of (Cb_s, Cr_s) is expressed as $(Cb_s^{(0),pat_i}, Cr_s^{(0),pat_i})$ in Eq. (21), as shown at the bottom of the next page, and is used as our initial subsampled chroma solution of B^{CbCr} for pat_i, $1 \leq i \leq 10$. It is notable that for an RGBW CFA image I^{RGBW} with the RGBW-Sony CFA pattern ($= < pat_5, pat_6, pat_6, pat_5 >$), the two initial subsampled chroma solutions of B^{CbCr} for pat_5 and pat_6 can be obtained by calling Eq. (21) with $i = 5$ and $i = 6$, respectively.

C. THE PROPOSED ITERATIVE CHROMA SUBSAMPLING METHOD

We take a 4 × 4 RGBW-Kodak CFA block example in Fig. 7(a) to sketch the idea of the proposed iterative chroma subsampling method. The four 2 × 2 partitioned CFA blocks of Fig. 7(a) are pat_7, pat_6, pat_6, and pat_8.

For simplicity, we only take the top-right 2×2 RGBW CFA block of Fig. 7(b) to sketch the idea of our iterative method. The demosaicked 2×2 RGB full-color block of the top-right 2×2 RGBW CFA block of Fig. 7(b) is demonstrated in Fig. 7(c). By Eq. (1), the converted 2×2 YCbCr block of Fig. 7(c) is demonstrated in Fig. 7(d). According to the neighboring reference subsampled chroma pairs in Fig. 7(e), the grid plot of the block-distortion function $D^{pat_6}(Cb_s, Cr_s)$ is depicted in Fig. 7(f). By Eq. (21), the initial subsampled chroma solution of B^{CbCr} is depicted by a black circle in Fig. 7(f). Clearly, there is room for improvement from the initial subsampled chroma solution to the desirable subsampled chroma solution, as depicted by the red circle.

To refine the initial subsampled chroma solution, we first consider the eight neighboring distance-1 reference points of the initial solution, namely $(Cb_s^{(0),pat_i}+m, Cr_s^{(0),pat_i}+n)$-pairs for $(m,n) \in \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,1), (1,-1), (1,0), (1,1)\}$, which are marked in yellow in Fig. 8(a). Then, we select the best neighboring point with the smallest block-distortion.

If the block-distortion of one neighboring distance-1 reference point is smaller than that obtained in the previous iteration, it replaces the old subsampled chroma solution, and we go to the next solution-refinement process. Otherwise, to avoid getting stuck in a local minimum, we try an uphill-move step in the current iteration by considering the 16 neighboring distance-2 reference points, which are marked in blue in Fig. 8(b). We repeat the above solution-refinement process until it reaches the termination condition. For this case in Fig. 8, the current subsampled (Cb, Cr)-pair solution (111, 181) with block-distortion 11 is replaced by a better subsampled (Cb, Cr)-pair solution (112, 183) with block-distortion 9.

In our experience, the uphill-move step in the proposed iterative method can achieve average PSNR gains of 0.04 dB, 0.4 dB, and 0.2 dB for one Bayer CFA image, RGBW-Kodak CFA image, and RGBW-Sony CFA image, respectively. Empirically, for each B^{CbCr}, the average numbers of distance-1 iterations required in the proposed method for the Bayer, RGBW-Sony, and RGBW-Kodak CFA patterns are 1.57, 1.69, and 1.78, respectively. The average numbers of distance-2 iterations required in the proposed method for the three considered CFA patterns are 1.03, 1.1, and 1.15.

The proposed CCI-based iterative chroma subsampling method for each 2×2 CbCr block B^{CbCr} with the CFA pattern

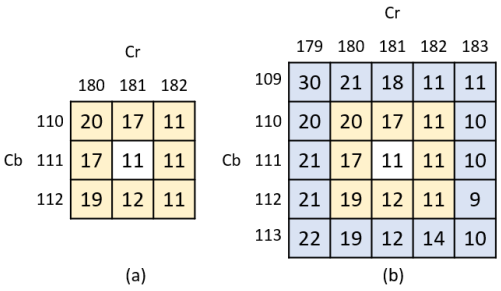

FIGURE 8. Two snapshots of the proposed iterative chroma subsampling method. (a) Eight neighboring distance-1 reference points marked in yellow in the current iteration. (b) Sixteen neighboring distance-2 reference points marked in blue used in the uphill-move step.

pat_i for $1 \le i \le 10$, denoted by B^{CbCr,pat_i}, is listed in Algorithm 1.

The execution code of our combination 'Proposed-CCI' can be accessed from the website [9], where 'Proposed' denotes our iterative chroma subsampling method used at the server side and 'CCI' denotes the chroma upsampling process used at the client side. It is notable that after receiving the decompressed subsampled CbCr image, we first construct an initial upsampled CbCr image. Each time, we consider sixteen subsampled chroma pairs within a 4×4 window, which are marked by the 16 black bullets in Fig. 6, and then the chroma upsampling process 'CCI', which is quite similar to the CCI-based method to estimate the four chroma pairs of B^{CbCr} described in Subsection II.A, fills the four missing chroma pixels one by one. The upsampling process 'CCI' continues the filling operations until all missing chroma pixels are constructed.

IV. EXPERIMENTAL RESULTS

Based on the ground truth Bayer and RGBW CFA images which are created from the three RGB full-color datasets, IMAX [12], SCI [30], and CI [44], the quality enhancement and quality-bitrate tradeoff improvement merits of our chroma subsampling method for I^{Bayer} and I^{RGBW} are demonstrated. The execution time comparison is also reported.

The proposed method and all comparative methods are implemented on a computer with an Intel Core i7-8700 CPU 3.2 GHz and 24 GB RAM. The operating system is the Microsoft Windows 10 64-bit operating system.

$$
\begin{aligned}
Cb_s^{(0),pat_i} &= \frac{\left(\sum_{k=1}^{4}\beta_{pat_{i,k}}^2\right)\left[\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2(Cb_k - Cb_k^{const}) + \alpha_{pat_{i,k}}\beta_{pat_{i,k}}(Cr_k - Cr_k^{const})\right] - \left(\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}}\right)\left[\sum_{k=1}^{4}\beta_{pat_{i,k}}^2(Cr_k - Cr_k^{const}) + \alpha_{pat_{i,k}}\beta_{pat_{i,k}}(Cb_k - Cb_k^{const})\right]}{\hat{W}(2,2)\left[\left(\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\right)\times\left(\sum_{k=1}^{4}\beta_{pat_{i,k}}^2\right) - \left(\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}}\right)^2\right]} \\
Cr_s^{(0),pat_i} &= \frac{\left(\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\right)\left[\sum_{k=1}^{4}\beta_{pat_{i,k}}^2(Cr_k - Cr_k^{const}) + \alpha_{pat_{i,k}}\beta_{pat_{i,k}}(Cb_k - Cb_k^{const})\right] - \left(\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}}\right)\left[\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2(Cb_k - Cb_k^{const}) + \alpha_{pat_{i,k}}\beta_{pat_{i,k}}(Cr_k - Cr_k^{const})\right]}{\hat{W}(2,2)\left[\left(\sum_{k=1}^{4}\alpha_{pat_{i,k}}^2\right)\times\left(\sum_{k=1}^{4}\beta_{pat_{i,k}}^2\right) - \left(\sum_{k=1}^{4}\alpha_{pat_{i,k}}\beta_{pat_{i,k}}\right)^2\right]}
\end{aligned}
\tag{21}
$$

Algorithm 1 The Proposed CCI-Based Iterative Chroma Subsampling Method

Input: A 2×2 CbCr block B^{CbCr,pat_i}.
Output: The subsampled (Cb, Cr)-pair of B^{CbCr,pat_i}.
Step 1: We perform $m \leftarrow 0$, where the index m denotes the iterative number; initially, we set $m = 0$. By Eq. (21), we calculate the initial subsampled chroma pair of B^{CbCr,pat_i}, denoted by $(Cb_s^{(m)}, Cr_s^{(m)})$. Then, by Eq. (13), we calculate the block-distortion $D^{pat_i}(Cb_s^{(m)}, Cr_s^{(m)})$.
Step 2: We calculate the eight block-distortion values of the neighboring distance-1 reference points of $(Cb_s^{(m)}, Cr_s^{(m)})$. Among the eight neighboring points, we select the neighboring (Cb, Cr)-pair with the smallest block-distortion as a candidate subsampled chroma pair of B^{CbCr,pat_i}, denoted by $(Cb_s^{(m+1)}, Cr_s^{(m+1)})$.
Step 3: If $D^{pat_i}(Cb_s^{(m+1)}, Cr_s^{(m+1)}) < D^{pat_i}(Cb_s^{(m)}, Cr_s^{(m)})$, we perform $m \leftarrow m + 1$ and go to step 2; otherwise, we go to step 4.
Step 4: We calculate the 16 block-distortion values of the neighboring distance-2 reference points of $(Cb_s^{(m)}, Cr_s^{(m)})$. Among the 16 neighboring points, we select the neighboring (Cb, Cr)-pair with the smallest block-distortion as a candidate subsampled chroma pair of B^{CbCr,pat_i}, which is denoted by $(Cb_s^{(m+1)}, Cr_s^{(m+1)})$.
Step 5: If $D^{pat_i}(Cb_s^{(m+1)}, Cr_s^{(m+1)}) < D^{pat_i}(Cb_s^{(m)}, Cr_s^{(m)})$, we perform $m \leftarrow m + 1$ and go to step 2; otherwise, we stop the algorithm and return $(Cb_s^{(m)}, Cr_s^{(m)})$ as the final subsampled (Cb, Cr)-pair of B^{CbCr,pat_i}.

The program development environment is Visual C++ 2019. The VVC reference software platform used for compression is VTM-11.0.

A. QUALITY ENHANCEMENT MERIT OF OUR METHOD AND EXECUTION TIME COMPARISON

When setting QP to zero, we take the three quality metrics, namely PSNR, SSIM [35], and FSIM [42], to report the quality enhancement merit of our method for I^{Bayer} and I^{RGBW} relative to the existing methods. PSNR is used to evaluate the average quality of one reconstructed CFA image $I^{rec,CFA}$ and it is defined by

$$\text{PSNR} = \frac{1}{N}\sum_{n=1}^{N} 10\log_{10}\frac{255^2}{MSE} \tag{22}$$

where N denotes the number of the CFA images in one dataset; MSE (mean square error) equals $\frac{1}{XY}\sum_{i=1}^{X}\sum_{j=1}^{Y}(I^{CFA} - I^{rec,CFA})^2$ where I^{CFA} denotes a ground truth CFA image and XY denotes the image size. We first calculate the PSNR value of each dataset and then calculate the average PSNR value of the three PSNR values of the three considered datasets.

The quality metric SSIM is expressed as the product of the luminance mean similarity, the contrast similarity, and the structure similarity between the ground truth CFA image and the reconstructed CFA image. The quality metric FSIM utilizes the phase consistency and gradient magnitude to weight the local quality maps to obtain a quality score of the reconstructed image.

1) QUALITY ENHANCEMENT MERIT FOR I^{Bayer}

For simplicity, we take the commercial Bayer CFA pattern in Fig. 1(a) as the representative to demonstrate the quality enhancement merit of the reconstructed Bayer CFA images using our iterative chroma subsampling method, namely 'Proposed'. For fairness, our two combinations, namely 'Proposed-BICU' and 'Proposed-CCI', are included in the comparative combinations. The two combinations 'Proposed-BICU' and 'Proposed-CCI' are implemented for the Bayer, RGBW-Sony, and RGBW-Kodak CFA images, respectively, and in total, there are six variants whose execution codes can be accessed from the website [9].

In Table 3, the individual results of IMAX, SCI, and CI are tabulated in the parentheses (), [], and {}, respectively. Table 3 indicates that our combination 'Proposed-CCI' has the highest PSNR, SSIM, and FSIM in boldface among the 18 considered combinations. In particular, relative to the five combinations, 4:2:0(A)-BICU, 4:2:0(MPEG)-BICU, DM-BICU [3], DI-COPY [18], and GD-BILI [4], the PSNR gains of our combination 'Proposed-CCI' are 7.6386 (= 47.2409 - 39.6023) dB, 11.0689 dB, 8.3057 dB, 2.4841 dB, and 2.0531 dB, respectively. It is notable that another of our combination 'Proposed-BICU' also outperforms the 16 comparative combinations. For SSIM and FSIM, our combination 'Proposed-CCI' still has positive gains when compared with the concerned combinations.

2) QUALITY ENHANCEMENT MERIT FOR I^{RGBW}

We take the two commercial RGBW CFA patterns, namely the RGBW-Sony CFA pattern in Fig. 2(a) and the RGBW-Kodak CFA pattern in Fig. 2(b), as representatives to show the quality enhancement merit of our combination 'Proposed-CCI'. Table 4 indicates that our combination 'Proposed-CCI' has the highest PSNR, SSIM, and FSIM in boldface among the considered 11 combinations. Excluding the combination 'Proposed-BICU', the overall average PSNR gain of our combination 'Proposed-CCI' over the other nine comparative combinations is 4.033 dB, achieving a significant quality improvement.

For I^{RGBW} with the RGBW-Sony CFA pattern in Fig. 2(a), as tabulated in the parentheses () in Table 4, the PSNR gains of our combination 'Proposed-CCI' over the four combinations, 4:2:0(A)-BICU, 4:2:0(MPEG)-BICU, IDID-NEDI, and JCDU-BICU, are 3.8925 (= 48.9298 - 45.0373) dB, 5.9621 dB, 4.6263 dB, and 3.8406 dB, respectively, indicating significant quality enhancement effects of our combination. For I^{RGBW} with the RGBW-Kodak CFA pattern, as tabulated in the parentheses [] in Table 4, the PSNR gains of our combination 'Proposed-CCI' over the above four comparative combinations are 1.5503 (= 46.8499 - 45.2996) dB,

3.6223 dB, 2.3390 dB, and 1.5129 dB, respectively, also indicating significant quality enhancement effects of our combination.

3) COMPUTATIONAL TIME ANALYSIS AND EXECUTION TIME COMPARISON

We first analyze the computational complexities required in the proposed iterative chroma subsampling method for the 2×2 Bayer, RGBW-Sony, and RGBW-Kodak CFA patterns. Next, we analyze the computational complexities required in the proposed method and the four state-of-the-art chroma subsampling methods, namely DM [3], DI [18], GD [4], and modified 4:2:0(A) [19], for the 2×2 Bayer CFA pattern. Finally, the execution times are compared.

a: COMPUTATIONAL TIME ANALYSIS OF THE PROPOSED METHOD FOR THE 2×2 BAYER, RGBW-SONY, AND RGBW-KODAK CFA PATTERNS

Because in Eq. (5), the values of $\hat{W}(i,j)$, $0 \leq i,j \leq 4$ and $(i,j) \neq (2,2)$ can be calculated in advance, Eq. (6) takes 29 operations to calculate the value of Cb_k^{const} for $1 \leq k \leq 4$ and 29 operations to calculate the value of Cr_k^{const} for $1 \leq k \leq 4$. To calculate the initial subsampled chroma solution $(Cb_s^{(0),pat_i}, Cr_s^{(0),pat_i})$, Eq. (21) takes 286 $(= 29 \times 4 + 29 \times 4 + 54)$ operations. Putting the initial subsampled chroma solution $(Cb_s^{(0),pat_i}, Cr_s^{(0),pat_i})$ into Eq. (13), 43 operations are required to calculate the distortion. However, in the subsequent iterations, the calculations of the two difference values, namely $(Cb_k - Cb_k^{const})$ and $(Cr_k - Cr_k^{const})$, can be waived because both difference values are not changed in each iteration. Therefore, in each distance-1 (or distance-2) iteration, only 35 operations are required to calculate the distortion for one new subsampled (Cb, Cr)-pair trial. Overall, the next distance-1 iteration takes 292 $(= 8 + 4 + 35 \times 8)$ operations to determine the best refined solution. Similarly, the next distance-2 iteration takes 580 $(= 16 + 4 + 35 \times 16)$ operations to determine the best refined solution.

As mentioned before, for each 2×2 CbCr block B^{CbCr}, the average numbers of distance-1 iterations required in the proposed method for the Bayer, RGBW-Sony, and RGBW-Kodak CFA patterns are 1.57, 1.69, and 1.78, respectively. The average numbers of distance-2 iterations required in the proposed method for the Bayer, RGBW-Sony, and RGBW-Kodak CFA patterns are 1.03, 1.1, and 1.15, respectively.

Therefore, for the Bayer, RGBW-Sony, and RGBW-Kodak CFA patterns, the time complexities required in our method are bounded by 1384.84 $(= 286 + 43 + 292 \times 1.57 + 580 \times 1.03)$, 1460.48 $(= 286 + 43 + 292 \times 1.69 + 580 \times 1.1)$, and 1515.76 $(= 286 + 43 + 292 \times 1.78 + 580 \times 1.15)$ operations, respectively.

b: COMPUTATIONAL TIME ANALYSIS OF THE FOUR STATE-OF-THE-ART METHODS FOR THE 2×2 BAYER CFA PATTERN

It can be easily verified that the DM method [3] needs 2 memory access operations to obtain the subsampled chroma pair, namely (Cb_3, Cr_2). The DI method [18] needs 38 operations (see Eqs. (12)-(13) in [18]) to determine the subsampled chroma pair of B^{CbCr}.

The GD method [4] first takes 38 operations to determine the initial subsampled chroma solution using Eqs. (12)-(13) in [18]. Next, it calculates the distortion of the initial subsampled chroma solution. Furthermore, GD considers the distance-1 iterations. Because our CCI-based block-distortion function in Eq. (13) is more complicated than that in GD and our method must consider distance-1 and distance-2 iterations, the computational complexity required in GD is less than ours.

For the 2×2 Bayer CFA pattern, let the computational complexity required in DM [3], DI [18], GD [4], and our method be denoted by T_{DM}, T_{DI}, T_{GD}, and $T_{Proposed}$, respectively. From the above analysis, we tabulate the time complexities of the four methods in Table 2, where #(memory accesses) denotes the number of memory accesses and #(BILI-based D-1 iterations), #(CCI-based D-1 iterations), and #(CCI-based D-2 iterations) denote the number of BILI-based distance-1 iterations, the number of CCI-based distance-1 iterations, and the number of CCI-based distance-2 iterations, respectively. From Table 2, we thus have $T_{DM} < T_{DI} < T_{GD} < T_{Proposed}$. The actual execution time cost required in the considered four methods, as tabulated in the last column of Table 3, justifies the computational complexity order of the four methods.

c: EXECUTION TIME COMPARISON

For I^{Bayer} and I^{RGBW}, the average execution time (in seconds) required in each considered combination is tabulated in the last columns of Tables 3 and 4, respectively. For each combination, the execution time requirement equals the sum of the execution time required in chroma subsampling and that required in chroma upsampling. In particular, when compared with the two combinations 'IDID-NEDI' [43] and 'JCDU-BICU' [36], our combination 'Proposed-CCI' takes much less time and has higher average PSNR, SSIM, and FSIM. Although our combination takes more time than the other combinations, our combination achieves much better quality enhancement effects.

B. BD-PSNR MERIT AND VISUAL EFFECT MERITS OF OUR METHOD

When setting four QP intervals, namely [4, 16], [12, 24], [20, 32], and [28, 40], we adopt the BD-PSNR metric [2] to demonstrate the quality-bitrate tradeoff merit of our combination 'Proposed-CCI' relative to the considered combinations. To calculate the BD-PSNR value of one combination, we take the combination '4:2:0(A)-BICU' as the basis of comparison. For one considered combination and the comparative combination, BD-PSNR indicates the average PSNR difference under the same bitrate. In addition, the visual effects of our combination are compared with those of some existing combinations.

TABLE 2. Operations required in the three comparative methods and the proposed method for the 2 × 2 Bayer CFA pattern B^{Bayer}.

	#(Memory Accesses)	Initial COPY-based block-distortion	Initial CCI-based block-distortion	#(BILI-based D-1 iterations)	#(CCI-based D-1 iterations)	#(CCI-based D-2 iterations)
DM [3]	2	-	-	-	-	-
DI [18]	-	1 (= 38 operations)	-	-	-	-
GD [4]	-	1 (= 38 operations)	-	3.23 (= 3.23 * 46 operations)	-	-
Proposed	-	-	1 (= 286 operations)	-	1.57 (= 1.57 * 292 operations)	1.03 (= 1.03 * 580 operations)

TABLE 3. The PSNR, SSIM, and FSIM comparison (QP = 0) for reconstructed Bayer CFA images.

I^{Bayer}	PSNR	PSNR Gain	SSIM	SSIM Gain	FSIM	FSIM Gain	Time (s)
4:2:0(A)-BICU	39.6023 (42.2395) [35.9074] {40.6601}	7.6386 (7.2922) [7.9631] [7.6604]	0.9960 (0.9966) [0.9944] {0.9970}	0.0030 (0.0025) [0.0042] {0.0023}	0.9969 (0.9985) [0.9937] {0.9984}	0.0023 (0.0011) [0.0048] {0.0012}	0.0338 (0.0254) [0.0452] {0.0309}
modified 4:2:0(A)-TN [19]	39.8849 (42.3923) [36.2783] {40.9841}	7.3560 (7.1394) [7.5922] {7.3364}	0.9962 (0.9967) [0.9947] {0.9972}	0.0028 (0.0024) [0.0039] {0.0021}	0.9969 (0.9984) [0.9938] {0.9985}	0.0023 (0.0012) [0.0047] {0.0011}	0.0377 (0.0288) [0.0509] {0.0334}
modified 4:2:0(A)-BICU [19]	39.6028 (42.2400) [35.9075] {40.6609}	7.6381 (7.2917) [7.9630] {7.6596}	0.9960 (0.9966) [0.9944] {0.9970}	0.0030 (0.0025) [0.0042] {0.0023}	0.9969 (0.9985) [0.9937] {0.9984}	0.0023 (0.0011) [0.0048] {0.0012}	0.0356 (0.0271) [0.0477] {0.0320}
4:2:0(L)-BICU	37.9644 (40.0947) [34.5083] {39.2902}	9.2765 (9.4370) [9.3622] {9.0303}	0.9946 (0.9951) [0.9930] {0.9958}	0.0044 (0.0040) [0.0056] {0.0035}	0.9963 (0.9980) [0.9930] {0.9979}	0.0029 (0.0016) [0.0055] {0.0017}	0.0322 (0.0246) [0.0435] {0.0284}
4:2:0(R)-BICU	37.1383 (39.7552) [34.0693] {37.5904}	10.1026 (9.7765) [9.8012] {10.7301}	0.9939 (0.9945) [0.9923] {0.9949}	0.0051 (0.0046) [0.0063] {0.0044}	0.9958 (0.9976) [0.9923] {0.9975}	0.0034 (0.0020) [0.0062] {0.0021}	0.0328 (0.0247) [0.0450] {0.0286}
4:2:0(DIRECT)-BICU	36.0713 (38.0702) [32.7891] {37.3546}	11.1696 (11.4615) [11.0814] {10.9659}	0.9920 (0.9925) [0.9896] {0.9940}	0.0070 (0.0066) [0.0090] {0.0053}	0.9949 (0.9970) [0.9904] {0.9972}	0.0043 (0.0026) [0.0081] {0.0024}	0.0320 (0.0245) [0.0429] {0.0286}
4:2:0(MPEG-B)-BICU	36.1720 (38.3259) [33.1020] {37.0880}	11.0689 (11.2058) [10.7685] {11.2325}	0.9922 (0.9926) [0.9900] {0.9940}	0.0068 (0.0065) [0.0086] {0.0053}	0.9951 (0.9970) [0.9910] {0.9973}	0.0041 (0.0026) [0.0075] {0.0023}	0.0361 (0.0251) [0.0449] {0.0383}
Anchor-(4-tap) [21]	38.3094 (40.1431) [35.0406] {39.7445}	8.9315 (9.3886) [8.8299] {8.5760}	0.9873 (0.9927) [0.9743] {0.9950}	0.0117 (0.0064) [0.0243] {0.0043}	0.9968 (0.9979) [0.9944] {0.9982}	0.0024 (0.0017) [0.0041] {0.0014}	0.0378 (0.0284) [0.0515] {0.0336}
Anchor-BICU [21]	37.7955 (39.7407) [34.4788] {39.1670}	9.4454 (9.7910) [9.3917] {9.1535}	0.9899 (0.9938) [0.9808] {0.9952}	0.0091 (0.0053) [0.0178] {0.0041}	0.9965 (0.9979) [0.9936] {0.9980}	0.0027 (0.0017) [0.0049] {0.0016}	0.0330 (0.0252) [0.0445] {0.0293}
DM-BICU [3]	38.9352 (40.1907) [36.5137] {40.1013}	8.3057 (9.3410) [7.3568] {8.2192}	0.9957 (0.9952) [0.9954] {0.9965}	0.0033 (0.0039) [0.0032] {0.0028}	0.9965 (0.9977) [0.9939] {0.9979}	0.0027 (0.0019) [0.0046] {0.0017}	0.0312 (0.0242) [0.0415] {0.0279}
IDID-NEDI [43]	39.0233 (40.6937) [35.4904] {40.8859}	8.2176 (8.8380) [8.3801] {7.4346}	0.9954 (0.9954) [0.9939] {0.9970}	0.0036 (0.0037) [0.0047] {0.0023}	0.9967 (0.9980) [0.9938] {0.9984}	0.0025 (0.0016) [0.0047] {0.0012}	17.6221 (12.7773) [25.4922] {14.5969}
JCDU-BICU [36]	39.9216 (41.4642) [36.4203] {41.8803}	7.3193 (8.0675) [7.4502] {6.4402}	0.9958 (0.9960) [0.9940] {0.9974}	0.0032 (0.0031) [0.0046] {0.0019}	0.9972 (0.9984) [0.9943] {0.9988}	0.0020 (0.0012) [0.0042] {0.0008}	0.2247 (0.1658) [0.3192] {0.1891}
DI-COPY [18]	44.7568 (46.5123) [42.0190] {45.7391}	2.4841 (3.0194) [1.8515] {2.5814}	0.9986 (0.9985) [0.9985] {0.9989}	0.0004 (0.0006) [0.0001] {0.0004}	0.9986 (0.9990) [0.9978] {0.9991}	0.0006 (0.0006) [0.0007] {0.0005}	0.0316 (0.0243) [0.0423] {0.0281}
DI-BICU [18]	39.3547 (40.6465) [36.8883] {40.5294}	7.8862 (8.8852) [6.9822] {7.7911}	0.9960 (0.9956) [0.9957] {0.9968}	0.0030 (0.0035) [0.0029] {0.0025}	0.9965 (0.9978) [0.9939] {0.9979}	0.0027 (0.0018) [0.0046] {0.0017}	0.0316 (0.0241) [0.0427] {0.0281}
GD-BILI [4]	45.1878 (47.2586) [41.6577] {46.6470}	2.0531 (2.2731) [2.2128] {1.6735}	0.9987 (0.9988) [0.9981] {0.9991}	0.0003 (0.0003) [0.0005] {0.0002}	0.9989 (0.9995) [0.9978] {0.9995}	0.0003 (0.0001) [0.0007] {0.0001}	0.0741 (0.0542) [0.1040] {0.0641}
GD-BICU [4]	39.6827 (40.8730) [37.3457] {40.8295}	7.5582 (8.6587) [6.5248] {7.4910}	0.9959 (0.9956) [0.9954] {0.9966}	0.0031 (0.0035) [0.0032] {0.0027}	0.9976 (0.9984) [0.9957] {0.9986}	0.0016 (0.0012) [0.0028] {0.0010}	0.0660 (0.0486) [0.0918] {0.0575}
Proposed-BICU	46.1968 (48.1200) [43.0772] {47.3931}	1.0441 (1.4117) [0.7933] {0.9274}	0.9987 (0.9988) [0.9983] {0.9991}	0.0003 (0.0003) [0.0003] {0.0002}	0.9990 (0.9995) [0.9980] {0.9995}	0.0002 (0.0001) [0.0005] {0.0001}	0.0950 (0.0669) [0.1407] {0.0775}
Proposed-CCI	**47.2409** (**49.5317**) [**43.8705**] {**48.3205**}	- - - -	**0.9990** (**0.9991**) [**0.9986**] {**0.9993**}	- - - -	**0.9992** (**0.9996**) [**0.9985**] {**0.9996**}	- - - -	0.1235 (0.0870) [0.1814] {0.1020}

TABLE 4. The PSNR, SSIM, and FSIM comparison (QP = 0) for reconstructed RGBW CFA images.

I^{RGBW}	PSNR	PSNR Gain	SSIM	SSIM Gain	FSIM	FSIM Gain	Time (s)
4:2:0(A)-BICU	45.1685 (45.0373) [45.2996]	2.7214 (3.8925) [1.5503]	0.9967 (0.9964) [0.9969]	0.0016 (0.0020) [0.0012]	0.9982 (0.9980) [0.9984]	0.0012 (0.0014) [0.0010]	0.0088 (0.0042) [0.0134]
4:2:0(L)-BICU	44.0848 (44.0488) [44.1207]	3.8051 (4.8810) [2.7292]	0.9964 (0.9962) [0.9966]	0.0019 (0.0022) [0.0015]	0.9979 (0.9978) [0.9980]	0.0015 (0.0016) [0.0014]	0.0072 (0.0027) [0.0117]
4:2:0(R)-BICU	43.9323 (43.7687) [44.0959]	3.9576 (5.1611) [2.7540]	0.9963 (0.9960) [0.9966]	0.0020 (0.0024) [0.0015]	0.9979 (0.9976) [0.9981]	0.0015 (0.0018) [0.0013]	0.0077 (0.0029) [0.0125]
4:2:0(DIRECT)-BICU	43.0126 (42.9487) [43.0765]	4.8773 (5.9811) [3.7734]	0.9957 (0.9956) [0.9958]	0.0026 (0.0028) [0.0023]	0.9973 (0.9972) [0.9973]	0.0021 (0.0022) [0.0021]	0.0071 (0.0028) [0.0113]
4:2:0(MPEG-B)-BICU	43.0977 (42.9677) [43.2276]	4.7922 (5.9621) [3.6223]	0.9958 (0.9955) [0.9960]	0.0025 (0.0029) [0.0021]	0.9973 (0.9971) [0.9974]	0.0021 (0.0023) [0.0020]	0.0089 (0.0046) [0.0132]
Anchor-(4-tap) [21]	42.5580 (42.4645) [42.6515]	5.3319 (6.4653) [4.1984]	0.9896 (0.9894) [0.9898]	0.0087 (0.0090) [0.0083]	0.9982 (0.9981) [0.9983]	0.0012 (0.0013) [0.0011]	0.0146 (0.0105) [0.0187]
Anchor-BICU [21]	43.2377 (43.1874) [43.2880]	4.6522 (5.7424) [3.5619]	0.9927 (0.9925) [0.9928]	0.0056 (0.0059) [0.0053]	0.9979 (0.9978) [0.9980]	0.0015 (0.0016) [0.0014]	0.0093 (0.0048) [0.0137]
IDID-NEDI [43]	44.4072 (44.3035) [44.5109]	3.4827 (4.6263) [2.3390]	0.9963 (0.9961) [0.9965]	0.0020 (0.0023) [0.0016]	0.9979 (0.9978) [0.9980]	0.0015 (0.0016) [0.0014]	17.4315 (17.1970) [17.6659]
JCDU-BICU [36]	45.2131 (45.0892) [45.3370]	2.6768 (3.8406) [1.5129]	0.9966 (0.9963) [0.9968]	0.0017 (0.0021) [0.0013]	0.9981 (0.9980) [0.9982]	0.0013 (0.0014) [0.0012]	0.2033 (0.1993) [0.2072]
Proposed-BICU	47.3608 (48.2638) [46.4577]	0.5291 (0.6660) [0.3922]	0.9981 (0.9982) [0.9979]	0.0002 (0.0002) [0.0002]	0.9994 (0.9994) [0.9994]	0.0000 (0.0000) [0.0000]	0.1645 (0.1635) [0.1655]
Proposed-CCI	**47.8899** (**48.9298**) [**46.8499**]	- - -	**0.9983** (**0.9984**) [**0.9981**]	- - -	**0.9994** (**0.9994**) [**0.9994**]	- - -	0.1716 (0.1695) [0.1737]

TABLE 5. Average BD-PSNR results (dB) among the considered combinations for I^{Bayer}.

Combination \ QP	[4, 16]	[12, 24]	[20, 32]	[28, 40]
modified 4:2:0(A)-TN [19]	0.2055	0.1344	0.0674	0.0336
modified 4:2:0(A)-BICU [19]	0.0018	0.0029	0.0013	0.0036
4:2:0(L)-BICU	-1.2935	-0.7836	-0.3741	-0.1368
4:2:0(R)-BICU	-2.0982	-1.4831	-0.9243	-0.4990
4:2:0(DIRECT)-BICU	-3.0037	-2.0519	-1.1494	-0.5252
4:2:0(MPEG-B)-BICU	-2.8910	-1.9810	-1.1115	-0.5039
Anchor-(4-tap) [21]	-0.9750	-0.5541	-0.2395	-0.0922
Anchor-BICU [21]	-1.4257	-0.8773	-0.4298	-0.1774
DM-BICU [3]	-0.4356	-0.0634	0.1213	0.1025
IDID-NEDI [43]	-0.3987	-0.1288	0.0015	0.0301
JCDU-BICU [36]	0.3115	0.3101	0.2125	0.1175
DI-COPY [18]	4.0154	2.6517	1.3843	0.5464
DI-BICU [18]	-0.0809	0.1671	0.2395	0.1505
GD-BILI [4]	4.2997	2.8830	1.5211	0.6256
GD-BICU [4]	0.2027	0.5956	0.5301	0.3045
Proposed-BICU	4.9649	3.1582	1.5708	0.6375
Proposed-CCI	**5.5032**	**3.3977**	**1.6868**	**0.6827**

1) VISUAL EFFECT MERIT

We take the 4th ground truth RGBW-Sony CFA image created from IMAX as the testing image, as depicted in Fig. 9(a), to demonstrate the visual effect merit of the reconstructed RGBW-Sony CFA image using our combination 'Proposed-CCI' relative to the nine comparative combinations. The magnified subimage in Fig. 9(b) is decoupled from the region containing the leaves in Fig. 9(a). For QP = 40, after performing the ten considered combinations on Fig. 9(a), the ten reconstructed magnified subimages for Fig. 9(b) are shown in Figs. 9(c)-(l), and we observe that our combination 'Proposed-CCI' has the best visual and texture-preserving effects of the ten combinations.

2) BD-PSNR MERIT

For I^{Bayer} with the Bayer CFA pattern in Fig. 1(a), we demonstrate the BD-PSNR merit of the reconstructed Bayer CFA images using our combination 'Proposed-CCI' relative to the existing combinations. In Table 5, we observe that our combination 'Proposed-CCI' has the best BD-PSNR performance in boldface among the 17 combinations. In particular, our combination has better BD-PSNR performance than the current work 'GD-BILI'.

Considering the RGBW-Sony CFA pattern in Fig. 2(a) and the RGBW-Kodak CFA pattern in Fig. 2(b), we demonstrate the BD-PSNR merit of the reconstructed RGBW CFA images using our combination 'Proposed-CCI' relative to the existing combinations. Here, the BD-PSNR results for I^{RGBW} with the RGBW-Kodak CFA pattern are listed in the parentheses in Table 6. Table 6 indicates that for the three QP intervals, namely [4, 16], [12, 24], and [20, 32], our combination 'Proposed-CCI' has the best BD-PSNR performance in boldface among the considered combinations. However, for the QP interval [28, 40], our combination 'Proposed-BICU' has

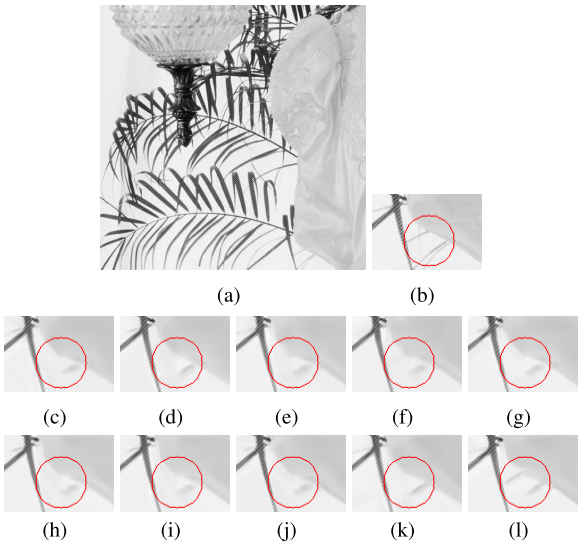

FIGURE 9. The visual effects of the reconstructed RGBW-Sony CFA image using our combination 'Proposed-CCI'. (a) The 4th ground truth RGBW-Sony CFA image created from IMAX. (b)The magnified subimage of (a). (c) 4:2:0(A)-BICU. (d) 4:2:0(L)-BICU. (e) 4:2:0(R)-BICU. (f)4:2:0(D)-BICU. (g) 4:2:0(MPEG-B)-BICU. (h) Anchor-(4-tap). (i) Anchor-BICU. (j) IDID-NEDI. (k) JCDU-BICU. (l) Proposed-CCI.

the best BD-PSNR performance, as shown in boldface, while our combination 'Proposed-CCI' is ranked second.

TABLE 6. Average BD-PSNR results (dB) among the considered combinations for I^{RGBW}.

Combination \ QP	[4, 16]	[12, 24]	[20, 32]	[28, 40]
4:2:0(L)-BICU	-0.6133 (-0.7324)	-0.2411 (-0.2924)	-0.0708 (-0.0842)	-0.0155 (-0.0102)
4:2:0(R)-BICU	-0.8325 (-0.7514)	-0.3929 (-0.3126)	-0.154 (-0.1095)	-0.0504 (-0.0432)
4:2:0(DIRECT)-BICU	-1.3853 (-1.4730)	-0.6601 (-0.6830)	-0.2425 (-0.2337)	-0.0733 (-0.0664)
4:2:0(MPEG-B)-BICU	-1.3498 (-1.3255)	-0.6168 (-0.5623)	-0.2147 (-0.1698)	-0.0631 (-0.0503)
Anchor-(4-tap) [21]	-1.7605 (-1.7892)	-0.8639 (-0.8528)	-0.3558 (-0.3352)	-0.1516 (-0.1404)
Anchor-BICU [21]	-1.2093 (-1.3120)	-0.5467 (-0.5881)	-0.2025 (-0.2137)	-0.0787 (-0.0817)
IDID-NEDI [43]	-0.3731 (-0.3786)	-0.0641 (-0.0381)	0.0254 (0.0308)	0.0217 (0.0301)
JCDU-BICU [36]	0.0809 (0.0855)	0.1093 (0.1042)	0.0752 (0.0633)	0.0405 (0.0309)
Proposed-BICU	1.7116 (0.3570)	1.0553 (0.3415)	0.5291 (0.2161)	**0.2159** (**0.1041**)
Proposed-CCI	**1.9496** (**0.5033**)	**1.1234** (**0.3794**)	**0.5399** (**0.2216**)	0.2098 (0.0984)

V. CONCLUSION

We have presented our new and effective CCI-based iterative chroma subsampling method for Bayer and RGBW CFA images. All three 4 × 4 RGBW CFA patterns in Fig. 2 are first partitioned into six 2 × 2 RGBW CFA patterns. Next, for I^{Bayer} and I^{RGBW}, a CCI-based block-distortion function is proposed. Then, we provide proof to show that the proposed CCI-based block-distortion function is a convex function that serves as the base of the initial subsampled chroma solution. Furthermore, a closed form is derived to obtain the initial subsampled chroma solution. To refine the initial subsampled chroma solution, an iterative chroma subsampling method is proposed. Finally, comprehensive experimental results demonstrate that on VTM-11.0, the proposed CCI-based chroma subsampling method achieves substantial quality enhancement and BD-PSNR improvement merits of the reconstructed Bayer and RGBW CFA images compared with existing methods. Future work will first deploy the abovementioned chroma subsampling works in the encrypted image for secret sharing [8], [38]. Second, the design of a deep learning-based chroma subsampling method will be investigated in future work.

ACKNOWLEDGMENT

The authors appreciate the proofreading help of the American Journal Experts Company and C. Harrington to improve the manuscript.

REFERENCES

[1] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.
[2] G. Bjøntegaard, *Calculation of Average PSNR Difference Between RDcurves*, document VECG-M33, Austin, TX, USA, Apr. 2001.
[3] H. Chen, M. Sun, and E. Steinbach, "Compression of Bayer-pattern video sequences using adjusted chroma subsampling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 12, pp. 1891–1896, Dec. 2009.
[4] K.-L. Chung, Y.-L. Lee, and W.-C. Chien, "Effective gradient descent-based chroma subsampling method for Bayer CFA images in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3281–3290, Nov. 2019.
[5] J. T. Compton and J. F. Hamilton, "Image sensor with improved light sensitivity," U.S. Patent 8 139 130 B2, Mar. 20, 2012.
[6] L. Condat, "A generic variational approach for demosaicking from an arbitrary color filter array," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 1625–1628.
[7] C. Doutre, P. Nasiopoulos, and K. N. Plataniotis, "H.264-based compression of Bayer pattern video sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 725–734, Jun. 2008.
[8] A. A. A. El-Latif, X. Yan, L. Li, N. Wang, J.-L. Peng, and X. Niu, "A new meaningful secret sharing scheme based on random grids, error diffusion and chaotic encryption," *Opt. Laser Technol.*, vol. 54, pp. 389–400, Dec. 2013.
[9] *Execution Code of Our Method*. Accessed: Jul. 2, 2021. [Online]. Available: http://140.118.175.164/CCI-based/CCI-based.rar
[10] F. Gastaldi, C. C. Koh, M. Carli, A. Neri, and S. K. Mitra, "Compression of videos captured via Bayer patterned color filter arrays," in *Proc. 13th Eur. Signal Process. Conf.*, vol. 54, Sep. 2005, pp. 1–4.
[11] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1952.
[12] (2014). *IMAX Dataset*. [Online]. Available: http://www.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm
[13] *Parameter Values for the HDTV Standards for Production and International Programme Exchange*, International Telecommunications Union, Standard ITU-R BT-709-5, 2015.
[14] 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.
[15] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Residual interpolation for color image demosaicking," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2304–2308.
[16] S. H. Lee, P. Oh, and M. G. Kang, "Three dimensional colorization based image/video reconstruction from white-dominant RGBW pattern images," *Digit. Signal Process.*, vol. 93, pp. 87–101, Oct. 2019.
[17] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
[18] 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. 1722–1733, Sep. 2016.

[19] T.-L. Lin, Y.-C. Yu, K.-H. Jiang, C.-F. Liang, and P.-S. Liaw, "Novel chroma sampling methods for CFA video compression in AVC, HEVC and VVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3167–3180, Sep. 2020.
[20] Y. Lu, S. Li, and H. Shen, "Virtualized screen: A third element for cloud-mobile convergence," *IEEE Multimedia Mag.*, vol. 18, no. 2, pp. 4–11, Feb. 2011.
[21] 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.
[22] Y. Lee, K. Hirakawa, and T. Q. Nguyen, "Camera-aware multi-resolution analysis for raw image sensor data compression," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2806–2817, Jun. 2018.
[23] S.-Y. Lee and A. Ortega, "A novel approach of image compression in digital cameras with a Bayer color filter array," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2001, pp. 482–485.
[24] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus With Applications in Statistics and Econometrics*. New York, NY, USA: Wiley, 1988, p. 141.
[25] H. S. Malvar and G. J. Sullivan, "Progressive-to-lossless compression of color-filter-array images using macropixel spectral-spatial transformation," in *Proc. IEEE Conf. Data Compress. (DCC)*, Apr. 2012, pp. 3–12.
[26] Z. Ni, K.-K. Ma, H. Zeng, and B. Zhong, "Color image demosaicing using progressive collaborative representation," *IEEE Trans. Image Process.*, vol. 29, pp. 4952–4964, 2020.
[27] *Open CV Book*. Accessed: Sep. 24, 2008. [Online]. Available: https://books.google.com.tw/books?hl=zh-TW&lr=&id=seAgiOfu2EIC&oi=fnd&pg=PR3&dq=opencv+library&ots=hUL28niGSa&sig=2f-MT4LGYlFPnyfkVd6xLIo27VM&redir_esc=y#v=snippet&q=bilinear&f=false
[28] *Open CV Codes*. Accessed: Aug. 29, 2018. [Online]. Available: https://sourceforge.net/projects/opencvlibrary/?fbclid=IwAR2vXid_ZecTmGKWiCuOe2XuaCKdAq4-Aes7FfeEVIpDM-BwOJAgg6TbI-k
[29] K. E. Paul and V. S. Saraswathibai, "Maximum accurate medical image demosaicing using WRGB based Newton Gregory interpolation method," *Measurement*, vol. 135, pp. 935–942, Mar. 2019.
[30] (2016). *SCI Dataset*. [Online]. Available: http://140.118.175.164/SCI
[31] *Spatial Scalability Filters*, document ISO/IEC JTC1/SC29/WG11 ITU-T SG 16 Q.6, Jul. 2005.
[32] T. Suzuki, "Wavelet-based spectral–spatial transforms for CFA-sampled raw camera image compression," *IEEE Trans. Image Process.*, vol. 29, pp. 433–444, 2020.
[33] M. Tachi, "Image processing device, image processing method, and program pertaining to image correction," U.S. Patent 8 314 863, Nov. 20, 2012.
[34] (2021). *VTM-11.0*. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/releases
[35] 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.
[36] S. Wang, K. Gu, S. Ma, and W. Gao, "Joint chroma downsampling and upsampling for screen content image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1595–1609, Sep. 2016.
[37] T. Yamagami, T. Sasaki, and A. Suga, "Image signal processing apparatus having a color filter with offset luminance filter elements," U.S. Patent 5 323 233, Jun. 21, 1994.
[38] X. Yan, S. Wang, A. A. A. El-Latif, and X. Niu, "Visual secret sharing based on random grids with abilities of AND and XOR lossless recovery," *Multimedia Tools Appl.*, vol. 74, no. 9, pp. 3231–3252, May 2015.
[39] W. Ye and K. Ma, "Color image demosaicing using iterative residual interpolation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5879–5891, Dec. 2015.
[40] Y.-C. Yu, J.-W. Jhang, X. Wei, H.-W. Tseng, Y. Wen, Z. Liu, T.-L. Lin, S.-L. Chen, Y.-S. Chiou, and H.-Y. Lee, "Chroma upsampling for YCbCr 420 videos," in *Proc. IEEE Int. Conf. Consum. Electron.*, Jun. 2017, pp. 163–164.
[41] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, 2011, Art. no. 023016.
[42] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.
[43] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao, "Interpolation-dependent image downsampling," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3291–3296, Nov. 2011.
[44] 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.

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 with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, since 2009. His research interests include machine learning, image processing, and video compression. He was a recipient of the Distinguished Research Award (2004–2007 and 2019–2022) and the Distinguished Research Project Award (2009–2012) from the Ministry of Science and Technology of Taiwan. He has been an Associate Editor of the *Journal of Visual Communication and Image Representation*, since 2011.

TSZ-CHING LEUNG received the B.S. degree in computer science and engineering from the National Ilan University, Ilan, 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 July 2021. Her research interests include image processing and video compression.

TZU-YI LIU 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 engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in July 2021. His research interests include image processing and video compression.

YA-CHI TSENG received the B.S. degree in computer science and engineering from Yuan Ze University, Taoyuan, Taiwan, in 2020. She is currently pursuing the M.S. degree in computer science and engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan. Her research interests include image processing and video compression.