Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-12743-0

An effective bilinear interpolation-based iterative chroma subsampling method for color images

Kuo-Liang Chung[1] · **Szu-Ni Chen**[1]

Received: 29 January 2021 / Revised: 14 June 2021 / Accepted: 21 February 2022

Abstract
Prior to encoding a color image, such as the RGB full-color image I^{RGB}, the Bayer color filter array (CFA) image I^{Bayer}, or the digital time delay integration CFA image I^{DTDI}, performing chroma subsampling on the converted chroma image is a necessary step. Previously, several chroma subsampling methods were developed for I^{RGB}, I^{Bayer}, and I^{DTDI} independently. In this paper, we propose an effective bilinear interpolation-based iterative chroma subsampling method for the considered three image types simultaneously, achieving better reconstructed images. Based on the considered three types of images collected from the Kodak, IMAX, and SCI (screen content images), the comprehensive experimental results demonstrated that under the versatile video coding (VVC) platform, our chroma subsampling method achieves the best quality and quality-bitrate tradeoff of the reconstructed color images when compared with the existing chroma subsampling methods.

Keywords Bayer color filter array (CFA) image · Chroma subsampling · Convex block-distortion function · Digital time delay integration CFA image · Quality-bitrate tradeoff · Quality enhancement · RGB full-color image

1 Introduction

In this chroma subsampling study, we consider three image types: the RGB full-color image I^{RGB}, the Bayer color filter array (CFA) image I^{Bayer}, and the digital time delay integration (DTDI) CFA image I^{DTDI}. The four Bayer CFA patterns [3, 19] in Fig. 1 have been widely used in modern color digital cameras, and each pixel in I^{Bayer} has only one R (red), G (green), or B (blue) color value. The two DTDI CFA patterns [5] in Fig. 2 have been widely used in industrial high-speed line scan cameras, and each DTDI CFA pixel has two color values, namely (G, B)-color value or (G, R)-color value.

In our chroma subsampling scheme, prior to compression for I^{Bayer} and I^{DTDI}, as depicted in Fig. 3, both I^{Bayer} and I^{DTDI} should be demosaicked to RGB full-color images

✉ Kuo-Liang Chung
klchung01@gmail.com

[1] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei, 10672, ROC, Taiwan

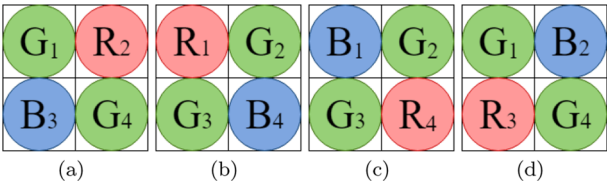

Fig. 1 Four Bayer CFA patterns. (a) $[G_1, R_2, B_3, G_4]$. (b) $[R_1, G_2, G_3, B_4]$. (c) $[B_1, G_2, G_3, R_4]$. (d) $[G_1, B_2, R_3, G_4]$

at the server side. To demosaick I^{Bayer} to an RGB full-color image, several demosaicking methods [13, 15, 20, 23, 27] can be used. Due to the available code, the method in [13] is used to demosaick I^{Bayer}. For I^{DTDI}, it is known that a DTDI image contains 50% R, 100% G, and 50% B color values. Using the modified version of the method [13], we first apply the G color-guided linear regression approach [13] to reconstruct the missing R color pixels, and then apply the same approach to reconstruct the missing B color pixels. The execution code of the modified version of the demosaicking method [13] for I^{DTDI} can be accessed from the website[1]. Then, following BT.601-5 [11], the RGB full-color image is transformed to a YCbCr image I^{YCbCr} using the following RGB-to-YCbCr conversion:

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

where for $1 \leq i \leq 4$, (R_i, G_i, B_i) and (Y_i, Cb_i, Cr_i) denote the triple-values of the ith RGB pixel and the converted YCbCr pixel, respectively, in each 2×2 RGB block and the collocated 2×2 YCbCr block.

Our chroma subsampling scheme focuses on 4:2:0 format which determines one subsampled (Cb, Cr)-pair of each 2×2 CbCr block B^{CbCr}, but our discussion is also applicable to 4:2:2 format. After encoding the subsampled YCbCr image, the encoded bit-stream is transmitted to the decoder via the internet. At the client side in Fig. 3, the decoded subsampled CbCr image is upsampled. When the input image is I^{RGB}, the upsampled YCbCr image is transformed to the reconstructed RGB full-color image using the following YCbCr-to-RGB conversion:

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

When the input image is I^{Bayer} (or I^{DTDI}), by (2), the upsampled YCbCr image is directly transformed to the reconstructed Bayer (or DTDI) CFA image based on the corresponding CFA pattern in Fig. 1 (or Fig. 2). Without the loss of generality, we take Figs. 1(a) and 2(a) as the representatives of the Bayer CFA pattern and the DTDI CFA pattern, respectively.

1.1 Related works

Chroma subsampling has a long history, and previously, several chroma subsampling works were developed for I^{RGB}, I^{Bayer}, and I^{DTDI} separately. We first introduce the related works for I^{RGB}, and then we introduce the related works for I^{Bayer}. Finally, the related works for I^{DTDI} are introduced.

[1]Execution code of the modified version of the demosaicking method [13] for I^{DTDI}, URL: ftp://140.118.175.164/DTDIdemosaicing.

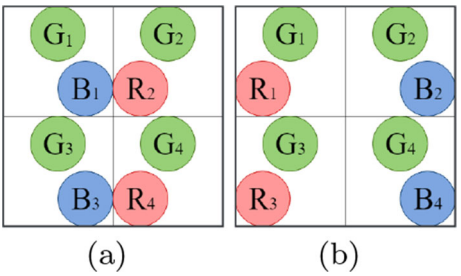

Fig. 2 Two DTDI CFA patterns. (a) $[(G_1, B_1), (G_2, R_2), (G_3, B_3), (G_4, R_4)]$. (b) $[(G_1, R_1), (G_2, B_2), (G_3, R_3), (G_4, B_4)]$

1.1.1 Related works for I^{RGB}

For I^{RGB}, the five traditional chroma subsampling methods, namely 4:2:0(A), 4:2:0(L), 4:2:0(R), 4:2:0(DIRECT), and 4:2:0(MPEG-B) [21] are first introduced. For simplicity, 4:2:0(DIRECT) is abbreviated as 4:2:0(D). Then, we introduce the two state-of-the-art works: the IDID (interpolation-dependent image downsampling) method [30] and the MCIM (major color and index map-based) method [24]. Throughout this paper, the two terms 'chroma pair' and '(Cb, Cr)-pair' denote the same thing.

4:2:0(A) determines the subsampled (Cb, Cr)-pair of B^{CbCr} by averaging the four (Cb, Cr)-pairs of B^{CbCr}. 4:2:0(L) and 4:2:0(R) determine the subsampled (Cb, Cr)-pairs by averaging the two chroma pairs in the left column and right column of B^{CbCr}, respectively. 4:2:0(D) determines the subsampled chroma pair by selecting the top-left (Cb, Cr)-entry of B^{CbCr}. 4:2:0(MPEG-B) determines the subsampled chroma pair by performing the 13-tap filter with mask [2, 0, -4, -3, 5, 19, 26, 19, 5, -3, -4, 0, 2]/64 on the top-left location of B^{CbCr}.

Using the new edge-directed interpolation (NEDI) [14] which improved the edge-directed interpolation (EDI) method [1], Zhang et al. [30] proposed an IDID chroma subsampling method. Their combination is called 'IDID-NEDI' in which NEDI is used as the chroma upsampling process at the client side. Considering the palette mode used for screen content images [18], Wang et al. [24] proposed an MCIM method to improve the IDID method, and their best combination is called 'MCIM-BICU' in which BICU denotes the bicubic interpolation used to upsample the subsampled chroma image at the client side.

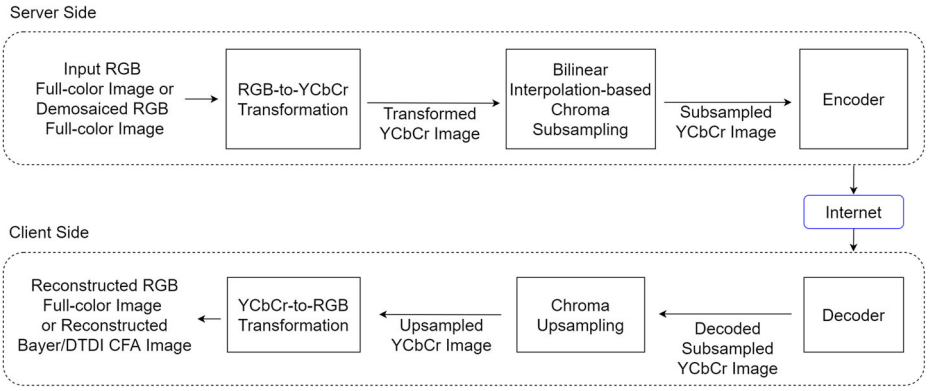

Fig. 3 The proposed chroma subsampling scheme for I^{RGB}, I^{Bayer}, and I^{DTDI} in the coding system

1.1.2 Related works for I^{Bayer}

Differing from the previous chroma subsampling methods for I^{RGB}, when the input image is I^{Bayer}, Chen et al. [6] observed that in (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 (Cb, Cr)-pair of each B^{CbCr} is set to (Cb_3, Cr_2). However, their method does not benefit the quality of the reconstructed G component. To overcome this disadvantage, Lin et al. [16] proposed a COPY-based 2×2 Bayer CFA block-distortion function, in which the operator 'COPY' is used to copy the subsampled chroma parameter-pair of B^{CbCr}, denoted by (Cb_s, Cr_s), for estimating the four chroma pairs of B^{CbCr}. Then, based on the block-distortion function, they applied the differentiation (DI) technique to determine the subsampled (Cb, Cr)-pair of B^{CbCr}. Their combination is called 'DI-COPY' in which 'COPY' is still used as the chroma upsampling process at the client side.

To improve the DI method, Chung et al. [8] proposed a COPY- and gradient-descent (GD)-based chroma subsampling method. In the GD method [8], the 2×2 Bayer CFA block-distortion function is based on the COPY-based chroma upsampling process that was used in the DI method. Their combination is called 'GD-BILI' in which BILI denotes the bilinear interpolation used to upsample the subsampled chroma image at the client side. Based on the COPY-based DI method [16], but considering the demosaiced RGB full-color block-distortion, Lin et al. [17] proposed a modified 4:2:0(A) chroma subsampling method which selects the best case among the four subsampled (Cb, Cr)-pairs of B^{CbCr} by considering the ceiling operation-based 4:2:0(A) and the floor operation-based 4:2:0(A). At the client side, they improved the previous upsampling process [26] by considering the distance between each estimated chroma value and its three neighboring (TN) pixels. Their combination is called 'modified 4:2:0(A)-TN'.

1.1.3 Related works for I^{DTDI}

For each 2×2 DTDI block B^{DTDI} in I^{DTDI} with the DTDI CFA pattern in Fig. 2(a), each pixel in the left column of B^{DTDI} contains one (G, B)-pair, but each pixel in the right column of B^{DTDI} contains one (G, R)-pair. Because (2) indicates that the B color is dominated by the Y and Cb components, and the R color is dominated by the Y and Cr components, in Chung et al.'s color domination (CD) method [7], the subsampled Cb component and the subsampled Cr component of B^{CbCr} are determined by performing 4:2:0(L) and 4:2:0(R) on B^{CbCr}, respectively. Their combination is called 'DI-CA' in which at the client side, the chroma-averaging (CA) based upsampling process is used to reconstruct each missing chroma pair by averaging the neighboring reference chroma pairs which have the same (G, R)-pair or (G, B)-pair as that of the missing chroma pair.

1.2 Motivation

Because the previous chroma subsampling works were developed for I^{RGB}, I^{Bayer}, and I^{DTDI} separately, their approaches are quite different. To the best of our knowledge, in the literature, no other chroma subsampling methods targeted for I^{RGB} can also be used for I^{Bayer} and I^{DTDI} using the same demosaicking method. It motivated us to design a novel and more effective method to handle the chroma subsampling problem for the considered three image types simultaneously.

1.3 Contributions

The major contributions of the proposed effective chroma subsampling method for I^{RGB}, I^{Bayer}, and I^{DTDI} are clarified as follows.

Differing from the COPY-based block-distortion function used in [8, 16, 17] for only I^{Bayer}, we first propose a BILI-based block-distortion function for I^{RGB}, I^{Bayer}, and I^{DTDI} simultaneously. Next, we prove that the proposed BILI-based block-distortion function is a convex function for any image type. Then, we derive a BILI-based formula to determine the initial subsampled (Cb, Cr)-pair for each 2×2 CbCr block B^{CbCr}. Furthermore, we propose an iterative method to refine the initial subsampled chroma pair of B^{CbCr} for any image type, achieving better quality enhancement and quality-bitrate tradeoff, such as the BD-PSNR (Bjøntegaard delta peak signal-to-noise ratio) [4], merits of the reconstructed color images relative to the existing methods.

Based on the considered three kinds of testing color images collected from the Kodak [10], IMAX [29], and SCI (screen content images) [22] datasets, the comprehensive experimental results have justified that under the versatile video coding (VVC) platform VTM-8.0 [12], our chroma subsampling method achieves the best quality and quality-bitrate tradeoff performance of the reconstructed color images when compared with the traditional methods and the existing methods [24, 30], [7, 8, 16, 17].

The remainder of this paper is organized as follows. In Section 2, our BILI-based block-distortion function is proposed. In Section 3, we prove the convex property of the proposed block-distortion function, and then the proposed iterative chroma subsampling method is presented. In Section 4, the experimental results are demonstrated to justify the quality and quality-bitrate tradeoff merits of our chroma subsampling method. In Section 5, some concluding remarks are addressed.

2 The proposed BILI-based block-distortion function

Differing from the COPY-based block-distortion function used in the previous works only just for I^{Bayer}, we propose a new and more effective BILI-based block-distortion function for I^{RGB}, I^{Bayer}, and I^{DTDI} simultaneously.

For easy exposition, we just consider the Cb component of each 2×2 Cb block B^{Cb}. As depicted in Fig. 4, let the subsampled Cb parameter of B^{Cb} be denoted as Cb_s which is located at (1, 0). Let the top-left, top-right, bottom-left, and bottom-right estimated Cb

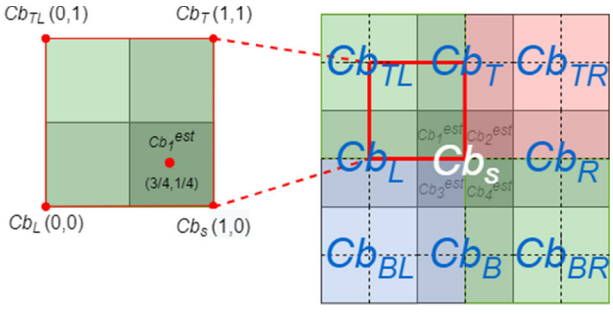

Fig. 4 The depiction of the proposed BILI-based chroma estimation method for estimating Cb_1^{est} at the server side

entries of B^{Cb} be denoted by Cb_1^{est}, Cb_2^{est}, Cb_3^{est}, and Cb_4^{est}, respectively. To estimate Cb_1^{est} located at (3/4, 1/4), we refer to the parameter Cb_s and the three neighboring subsampled Cb values of the corresponding 2×2 Cb blocks, namely Cb_T located at (1, 1), Cb_{TL} located at (0, 1), and Cb_L located at (0, 0), which have been obtained in advance by using the proposed chroma subsampling method which will be presented in Section 3.

Applying BILI on the parameter Cb_s and the three known reference Cb values, Cb_T, Cb_{TL}, and Cb_L in Fig. 4, Cb_1^{est} is estimated as

$$\begin{aligned} Cb_1^{est} &= (\frac{3}{4})(1-\frac{1}{4})Cb_s + (1-\frac{3}{4})(\frac{1}{4})Cb_{TL} \\ &\quad + (\frac{3}{4})(\frac{1}{4})Cb_T + (1-\frac{3}{4})(1-\frac{1}{4})Cb_L \\ &= \frac{9}{16}Cb_s + \frac{1}{16}Cb_{TL} + \frac{3}{16}Cb_T + \frac{3}{16}Cb_L \end{aligned} \tag{3}$$

By the same argument, to estimate Cb_2^{est}, we refer to the parameter Cb_s, the two known subsampled Cb values, Cb_T and Cb_{TR} in Fig. 4, and one future subsampled Cb value, Cb_R in Fig. 4, which can be obtained by using the 4:2:0(A) method for I^{RGB} and I^{DTDI}, but can be obtained by using the DI method [16] for I^{Bayer}. The estimation of Cb_3^{est} and Cb_4^{est} can be similarly followed. Generally, the four estimated Cb entries of B^{Cb}, Cb_1^{est}, Cb_2^{est}, Cb_3^{est}, and Cb_4^{est}, are expressed as the following linear function with parameter Cb_s:

$$Cb_i^{est} = \frac{9}{16}Cb_s + \bar{Cb}_i \tag{4}$$

for $1 \le i \le 4$ with

$$\begin{aligned} \bar{Cb}_1 &= \frac{1}{16}Cb_{TL} + \frac{3}{16}Cb_T + \frac{3}{16}Cb_L \\ \bar{Cb}_2 &= \frac{1}{16}Cb_{TR} + \frac{3}{16}Cb_T + \frac{3}{16}Cb_R \\ \bar{Cb}_3 &= \frac{1}{16}Cb_{BL} + \frac{3}{16}Cb_B + \frac{3}{16}Cb_L \\ \bar{Cb}_4 &= \frac{1}{16}Cb_{BR} + \frac{3}{16}Cb_B + \frac{3}{16}Cb_R \end{aligned} \tag{5}$$

In the same way, the four estimated Cr entries of B^{Cr} can be expressed as a linear function with parameter Cr_s.

After presenting the proposed BILI-based estimation method for estimating the four chroma pairs of B^{Cr}, 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 input image type $t \in \{RGB, Bayer, DTDI\}$, by (2), we can obtain the estimated 2×2 t block $B^{est,t}$ at the server side. For example, when the input image type t equals 'RGB', based on $B^{est,CbCr}$ and B^Y, by (2), the estimated 2×2 RGB block $B^{est,RGB}$ can be obtained.

We now present the proposed BILI-based block-distortion function to measure the sum of squared errors (SSE) between each 2×2 ground truth t block B^t and the estimated 2×2

t block $B^{est,t}$ for $t \in \{RGB, Bayer, DTDI\}$. Let S_i^t, $1 \le i \le 4$, denote the color set of the ith pixel of the 2×2 t block. It thus yields

$$\begin{aligned} B^{RGB} &= \{S_1^{RGB}, S_2^{RGB}, S_3^{RGB}, S_4^{RGB}\} \\ &= \{(R_1, G_1, B_1), (R_2, G_2, B_2), (R_3, G_3, B_3), (R_4, G_4, B_4)\} \\ B^{Bayer} &= \{S_1^{Bayer}, S_2^{Bayer}, S_3^{Bayer}, S_4^{Bayer}\} = \{G_1, R_2, B_3, G_4\} \quad \text{for Fig. 1(a)} \\ B^{DTDI} &= \{S_1^{DTDI}, S_2^{DTDI}, S_3^{DTDI}, S_4^{DTDI}\} \\ &= \{(G_1, B_1), (G_2, R_2), (G_3, B_3), (G_4, R_4)\} \quad \text{for Fig. 2(a)} \end{aligned} \tag{6}$$

Using the SSE criterion, for the input 2×2 ground truth block B^t, $t \in \{RGB, Bayer, DTDI\}$, our BILI-based block-distortion function is expressed as

$$D^t = \sum_{i=1}^{4} ||S_i^t - S_i^{est,t}||^2 \tag{7}$$

where $\|.\|$ denotes a 2-norm operation, $S_i^{est,t}$ denotes the color set of the ith pixel in the estimated 2×2 t block $B^{est,t}$. For $t =$ 'RGB', by (7), the distortion D^{RGB} equals $\sum_{i=1}^{4}[(R_i - R_i^{est})^2 + (G_i - G_i^{est})^2 + (B_i - B_i^{est})^2]$, where $(R_i^{est}, G_i^{est}, B_i^{est})$, $1 \le i \le 4$, denotes the triple-value of the ith RGB pixel in the estimated 2×2 RGB block $B^{est,RGB}$. Similarly, for $t =$ '$Bayer$', the block-distortion D^{Bayer} equals $[(G_1 - G_1^{est})^2 + (R_2 - R_2^{est})^2 + (B_3 - B_3^{est})^2 + (G_4 - G_4^{est})^2]$.

Using (1), our BILI-based block-distortion function in (7) can be rewritten as

$$\begin{aligned} &D^t(Cb_s, Cr_s) \\ &= \sum_{i=1}^{4} \sum_{c \in S_i^t} [(1.164Y_i + a_c Cb_i + b_c Cr_i) - (1.164Y_i + a_c Cb_i^{est} + b_c Cr_i^{est})]^2 \\ &= \sum_{i=1}^{4} \sum_{c \in S_i^t} [a_c(Cb_i - Cb_i^{est}) + b_c(Cr_i - Cr_i^{est})]^2 \\ &= \sum_{i=1}^{4} \sum_{c \in S_i^t} [a_c(Cb_i - (\frac{9}{16}Cb_s + \bar{Cb}_i)) + b_c(Cr_i - (\frac{9}{16}Cr_s + \bar{Cr}_i))]^2 \end{aligned} \tag{8}$$

where for $1 \le i \le 4$, the values of a_c and b_c are defined by

$$a_c = \begin{cases} 0 & \text{for } c = R_i \\ -0.391 & \text{for } c = G_i \\ 2.018 & \text{for } c = B_i \end{cases}$$

$$b_c = \begin{cases} 1.596 & \text{for } c = R_i \\ -0.813 & \text{for } c = G_i \\ 0 & \text{for } c = B_i \end{cases} \tag{9}$$

3 The proposed BILI-based iterative chroma subsampling method

For the considered three image types, we first prove the convex property of the proposed BILI-based block-distortion function in (8), and then a BILI-based formula is derived to calculate the initial subsampled chroma solution of B^{CbCr}. Finally, we propose an iterative chroma subsampling method to refine the initial subsampled chroma solution.

3.1 Convex property proof of the proposed block-distortion function

According to the convex function definition [9], if the determinant of the Hessian matrix of $D^t(Cb_s, Cr_s)$ in (8), denoted by $det(H(D^t(Cb_s, Cr_s)))$, is positive, the proposed block-distortion function $D^t(Cb_s, Cr_s)$ is a convex function for $t \in \{RGB, Bayer, DTDI\}$. The Hessian matrix of $D^t(Cb_s, Cr_s)$ is expressed as

$$H(D^t(Cb_s, Cr_s)) = \begin{bmatrix} \frac{\partial^2 D^t}{\partial Cb_s^2} & \frac{\partial^2 D^t}{\partial Cb_s \partial Cr_s} \\ \frac{\partial^2 D^t}{\partial Cr_s \partial Cb_s} & \frac{\partial^2 D^t}{\partial Cr_s^2} \end{bmatrix} \tag{10}$$

with

$$\begin{aligned} \frac{\partial^2 D^t}{\partial Cb_s^2} &= 2(\frac{9}{16})^2 \sum_{i=1}^{4} \sum_{c \in S_i^t} a_c^2 \\ \frac{\partial^2 D^t}{\partial Cr_s^2} &= 2(\frac{9}{16})^2 \sum_{i=1}^{4} \sum_{c \in S_i^t} b_c^2 \\ \frac{\partial^2 D^t}{\partial Cb_s \partial Cr_s} &= \frac{\partial^2 D^t}{\partial Cr_s \partial Cb_s} = 2(\frac{9}{16})^2 \sum_{i=1}^{4} \sum_{c \in S_i^t} a_c b_c \end{aligned} \tag{11}$$

In detail, the determinant of $H(D^t(Cb_s, Cr_s))$ is expressed as

$$\begin{aligned} &det(H(D^t)) \\ &= 4(\frac{9}{16})^4 (\sum_{i=1}^{4} \sum_{c \in S_i^t} a_c^2 \sum_{i=1}^{4} \sum_{c \in S_i^t} b_c^2 - \sum_{i=1}^{4} \sum_{c \in S_i^t} a_c b_c \sum_{i=1}^{4} \sum_{c \in S_i^t} a_c b_c). \end{aligned} \tag{12}$$

Let $\boldsymbol{v}_1^t = (a_{S_1^t}, a_{S_2^t}, a_{S_3^t}, a_{S_4^t})$ and $\boldsymbol{v}_2^t = (b_{S_1^t}, b_{S_2^t}, b_{S_3^t}, b_{S_4^t})$ be two vectors for $t \in \{RGB, Bayer, DTDI\}$. For example, for $t =$ 'RGB', we have

$$\begin{aligned} \boldsymbol{v}_1^{RGB} &= (a_{R_1}, a_{G_1}, a_{B_1}, a_{R_2}, a_{G_2}, a_{B_2}, a_{R_3}, a_{G_3}, a_{B_3}, a_{R_4}, a_{G_4}, a_{B_4}) \\ &= (0, -0.391, 2.018, 0, -0.391, 2.018, 0, -0.391, 2.018, 0, -0.391, 2.018) \\ \boldsymbol{v}_2^{RGB} &= (b_{R_1}, b_{G_1}, b_{B_1}, b_{R_2}, b_{G_2}, b_{B_2}, b_{R_3}, b_{G_3}, b_{B_3}, b_{R_4}, b_{G_4}, b_{B_4}) \\ &= (1.596, -0.813, 0, 1.596, -0.813, 0, 1.596, -0.813, 0, 1.596, -0.813, 0) \end{aligned} \tag{13}$$

Similarly, for $t =$ '$Bayer$', we have $\boldsymbol{v}_1^{Bayer} = (a_{G_1}, a_{R_2}, a_{B_3}, a_{G_4}) = (-0.391, 0, 2.018, -0.391)$ and $\boldsymbol{v}_2^{Bayer} = (b_{G_1}, b_{R_2}, b_{B_3}, b_{G_4}) = (-0.813, 1.596, 0, -0.813)$. Generally, for $t \in \{RGB, Bayer, DTDI\}$, $\boldsymbol{v}_1^t$ is not parallel to $\boldsymbol{v}_2^t$ and vice versa.

Table 1 The values of the determinant of $H(D^t(Cbs, Crs))$ in (12) for $t \in \{RGB, Bayer, DTDI\}$

	$t = 'RGB'$	$t = 'Bayer'$	$t = 'DTDI'$
$det(H(D^t(Cbs, Crs)))$	86.2040	6.6216	26.4863

From the equality '$\|\boldsymbol{v}_1^t\|\|\boldsymbol{v}_2^t\| cos\theta = <\boldsymbol{v}_1^t, \boldsymbol{v}_2^t>$', it yields the inequality: $\|\boldsymbol{v}_1^t\|^2\|\boldsymbol{v}_2^t\|^2 \geq <\boldsymbol{v}_1^t, \boldsymbol{v}_2^t>^2$. Using this inequality and the unparallel property between $\boldsymbol{v}_1^t$ and $\boldsymbol{v}_2^t$, it yields

$$\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c^2 \sum_{i=1}^{4}\sum_{c\in S_i^t} b_c^2 > (\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c b_c)^2 \tag{14}$$

Using the inequality in (14), (12) indicates that the determinant of $H(D^t(Cbs, Crs))$, namely $det(H(D^t))$, is positive, implying that the proposed BILI-based block-distortion function $D^t(Cbs, Crs)$ in (8) is a convex function for $t \in \{RGB, Bayer, DTDI\}$. Table 1 tabulates the positive values of $det(H(D^t))$ for $t \in \{RGB, Bayer, DTDI\}$, and the readers are suggested to refer to the related calculations in the website[2]. We thus have the following result.

Proposition 1 *For $t \in \{RGB, Bayer, DTDI\}$, the proposed BILI-based block-distortion function $D^t(Cb_s, Cr_s)$ in (8) is a convex function.*

3.2 Determining the initial subsampled chroma solution

Using the convex property in the proposed block-distortion function $D^t(Cbs, Crs)$ in (8), as described in Proposition 1, want to find the critical point of $D^t(Cbs, Crs)$ such that the function $D^t(Cbs, Crs)$ at the critical point has the minimal value in the real domain. Then, the determined critical point will be used as the initial subsampled (Cb, Cr)-pair of each 2×2 CbCr block B^{CbCr} in the practical integer domain.

By taking the first derivative on (8) with respect to Cb_s and Cr_s, respectively, and then setting the two derivatives to zero, it yields

$$\begin{aligned} \frac{\partial D^t(Cb_s, Cr_s)}{\partial Cr_s} &= 0 \\ \frac{\partial D^t(Cb_s, Cr_s)}{\partial Cb_s} &= 0. \end{aligned} \tag{15}$$

After solving (15) in the real domain, the critical point of our BILI-based block-distortion function $D^t(Cbs, Crs)$ is denoted by $(Cb_s^{(0),t}, Cr_s^{(0),t})$, as listed in (16). In the practical integer domain, we take one 2×2 Bayer CFA block example in Fig. 5(a) to point out the merits of our BILI-based initial solution $(Cb_s^{(0),Bayer}, Cr_s^{(0),Bayer})$ relative to the previous COPY-based initial solution used in the previous works. Note that besides the Bayer CFA image type, our initial solution in (16) is also applicable to the RGB and DTDI image types.

[2]Determinant values of Hessian matrix of the block-distortion in (8), URL: ftp://140.118.175.164/adaptive/detHD.pdf.

Algorithm 1 Our BILI-based Iterative Chroma Subsampling Method.

Input: 2×2 t block B^t, $t \in \{RGB, Bayer, DTDI\}$.
Output: Subsampled (Cb, Cr)-pair of B^{CbCr}, $(Cb_s^{(k),t}, Cr_s^{(k),t})$.
Step 1: By (16), we take $(Cb_s^{(0),t}, Cr_s^{(0),t})$ as the initial chroma subsampling solution of the current 2×2 CbCr block B^{CbCr}. By (8), we calculate the t color block-distortion, denoted by $D^t(Cb_s^{(0),t}, Cr_s^{(0),t})$. Set $k = 0$.
Step 2: Under the integer domain $[0, 255] \times [0, 255]$, we calculate all the eight neighboring t block-distortion values of $(Cb_s^{(k),t}, Cr_s^{(k),t})$, namely $D^t(Cb_s^{(k),t} + m, Cr_s^{(k),t} + n)$ for $(m, n) \in \{(0, 1), (0, -1), (1, 0), (-1, 0), (1, 1), (1, -1), (-1, 1), (-1, -1)\}$. Among the eight block-distortion values, we select the subsampled (Cb, Cr)-pair with the minimal block-distortion value as the candidate subsampled (Cb, Cr)-pair of B^{CbCr}, namely $(Cb_s^{(k+1),t}, Cr_s^{(k+1),t})$.
Step 3: If $D^t(Cb_s^{(k+1),t}, Cr_s^{(k+1),t}) \geq D^t(Cb_s^{(k),t}, Cr_s^{(k),t})$, we stop the algorithm and report $(Cb_s^{(k),t}, Cr_s^{(k),t})$ as the finally subsampled (Cb, Cr)-pair of B^{CbCr}; otherwise, we perform $k := k + 1$ and go to Step 2.

$$Cb_s^{(0),t} = \frac{(\sum_{i=1}^{4}\sum_{c\in S_i^t} b_c^2)\cdot[\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c^2(\tilde{Cb}_i - Cb_i) + a_c b_c(\tilde{Cr}_i - Cr_i)] - (\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c b_c)\cdot[\sum_{i=1}^{4}\sum_{c\in S_i^t} b_c^2(\tilde{Cr}_i - Cr_i) + a_c b_c(\tilde{Cb}_i - Cb_i)]}{\frac{9}{16}[(\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c b_c)^2 - (\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c^2)\cdot(\sum_{i=1}^{4}\sum_{c\in S_i^t} b_c^2)]}$$

$$Cr_s^{(0),t} = \frac{(\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c^2)\cdot[\sum_{i=1}^{4}\sum_{c\in S_i^t} b_c^2(\tilde{Cr}_i - Cr_i) + a_c b_c(\tilde{Cb}_i - Cb_i)] - (\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c b_c)\cdot[\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c^2(\tilde{Cb}_i - Cb_i) + a_c b_c(\tilde{Cr}_i - Cr_i)]}{\frac{9}{16}[(\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c b_c)^2 - (\sum_{i=1}^{4}\sum_{c\in S_i^t} a_c^2)\cdot(\sum_{i=1}^{4}\sum_{c\in S_i^t} b_c^2)]} \quad (16)$$

For Fig. 5(a), its demosaiced 2×2 RGB full-color block B^{RGB} and the converted YCbCr block B^{YCbCr} are depicted in Fig. 5(b) and (c), respectively. The eight neighboring (Cb, Cr)-pairs referred by the current 2×2 CbCr block B^{CbCr} are shown in Fig. 5(d). In the practical integer domain, Fig. 5(e) depicts the convex-like grid plot of the previous COPY-based Bayer CFA block-distortion function, and the red point in Fig. 5(e) denotes the COPY-based initial solution. Figure 5(f) depicts the convex-like grid plot of our BILI-based Bayer CFA block-distortion function, and the red triangle in Fig. 5(f) denotes our BILI-based initial solution. Comparing Fig. 5(e) and (f), we have two observations: (1) our initial solution is more close to the optimal solution marked by a yellow triangle in Fig. 5(f), leading to a lower computation cost merit and (2) our optimal solution has less SSE, leading to a better quality enhancement effect.

4 Experimental results

On the newly released VVC reference software platform VTM-8.0, based on the Kodak, IMAX, and SCI datasets, the thorough experimental results demonstrated the quality enhancement and BD-PSNR improvement merits of our BILI-based iterative chroma subsampling method for I^{RGB}, I^{Bayer}, and I^{DTDI} relative to the traditional methods and the state-of-the-art chroma subsampling methods [7, 8, 16, 17, 24, 30]. The execution time comparison among the considered methods is also made.

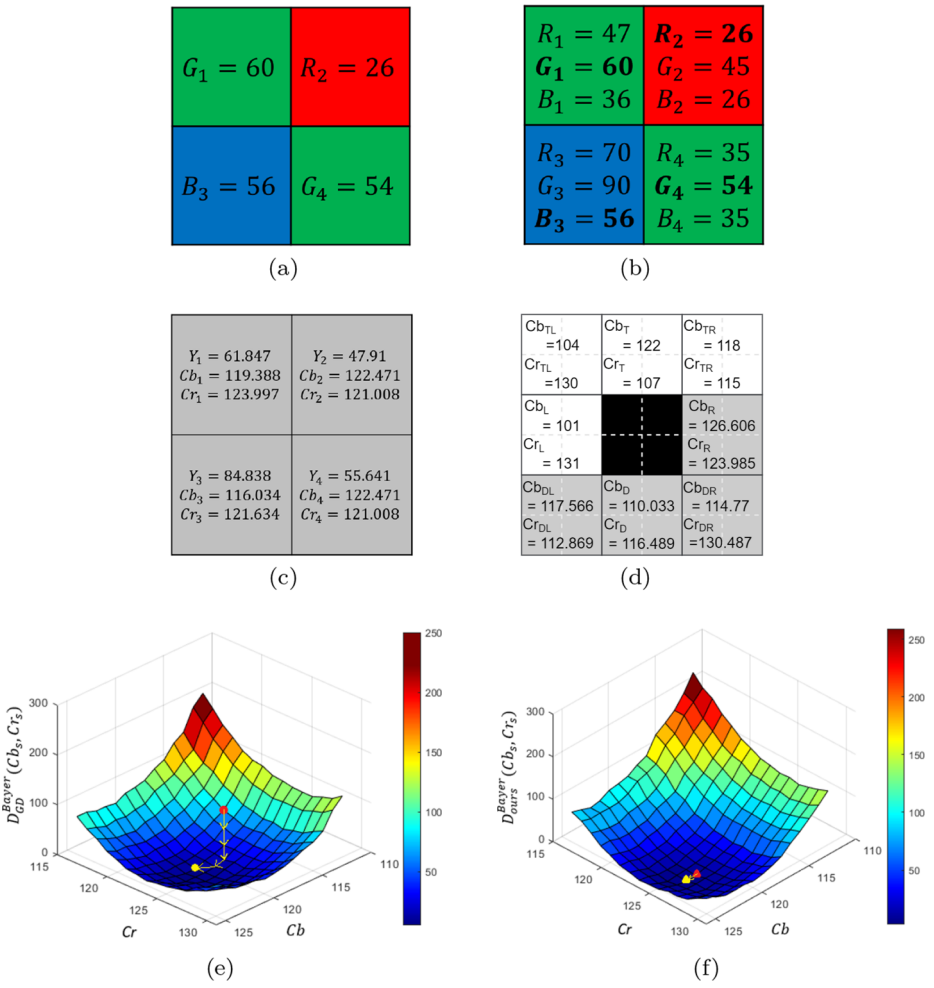

Fig. 5 For $t =$ '$Bayer$', an example to show the time and quality merits of our BILI-based block-distortion function relative to the previous COPY-based block-distortion function. (a) The input 2×2 Bayer CFA block B^{Bayer}. (b) The demosaiced RGB full-color block B^{RGB}. (c) The converted YCbCr block B^{YCbCr}. (d) The eight neighboring subsampled (Cb, Cr)-pairs referred by the current 2×2 CbCr block B^{CbCr}. (e) The convex-like grid plot by the previous COPY-based block-distortion function for Fig. 5(a). (f) The convex-like grid plot by our BILI-based block-distortion function for Fig. 5(a)

All the considered 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. The program development environment is Visual C++ 2019.

4.1 Quality enhancement merit of our method and time comparison

When setting QP to zero, the four quality metrics, namely PSNR, CPSNR (color peak signal-to-noise ratio), SSIM [25], and FSIM [28], are used to show the quality enhancement merit of our chroma subsampling method relative to the existing methods for I^{RGB}, I^{Bayer},

and I^{DTDI}. In these experiments for QP = 0, the related results are computed by passing the compression process and decompression process.

4.1.1 For I^{RGB}

The CPSNR metric of the reconstructed RGB full-color image is defined by

$$\text{CPSNR} = \frac{1}{N}\sum_{n=1}^{N} 10\log_{10}\frac{255^2}{CMSE} \tag{17}$$

with $CMSE = \frac{1}{3XY}\sum_{p\in P}\sum_{c\in\{R,G,B\}}[I_{n,c}(p) - I^{rec}_{n,c}(p)]^2$ in which the testing image is of size 3×X×Y where '3' denotes the number of color spaces. $I_{n,c}(p)$ and $I^{rec}_{n,c}(p)$ denote the c-color pixel-values at position p in the nth input RGB full-color image and the reconstructed analogue, respectively. N denotes the number of testing images in the dataset. Here, N equals 24, 18, and 26 for the Kodak dataset, the IMAX dataset, and the SCI dataset, respectively. We first calculate the CPSNR value of each dataset, and then the average CPSNR value of the three CPSNR values of three datasets is taken as the resultant CPSNR value.

For convenience, let the proposed iterative chroma subsampling method be denoted by 'ours'. Table 2 indicates that our combination 'ours-BILI', where 'BILI' is used as the chroma upsampling process at the client side, has the highest average CPSNR (in boldface) among the considered combinations, indicating the best quality enhancement effect of our combination for I^{RGB}. In Table 2, the individual results of IMAX, Kodak, and SCI are tabulated in the parentheses (), [ ], and {}, respectively.

The average execution time (in seconds) required for each image by the considered chroma subsampling method is tabulated in the last column of Table 2. Although our chroma subsampling method takes more time than the five traditional methods, our method achieves a much better quality enhancement effect. When compared with IDID [30] and MCIM [24], our method takes much less time and has higher average CPSNR. On the other hand, not only our method has a better quality enhancement effect, but our method is also much faster than the comparative methods, IDID and MCIM.

SSIM [25] is used to measure the product of the luminance, contrast, and structure similarity preserving effect between the original image and the reconstructed image. For I^{RGB}, the SSIM value is measured by the mean of the three SSIM values for the R, G, and B color planes. Table 2 demonstrates that our combination 'ours-BILI' has the highest average SSIM among the considered combinations.

FSIM [28] is another image quality assessment with high consistency with the subjective evaluation. FSIM first utilizes the primary feature "phase congruency (PC)" which is contrast invariant and the minor feature "gradient magnitude" to obtain the local quality map, and then FSIM utilizes PC as a weighting function to obtain a quality score. Table 2 also indicates that our combination has the highest FSIM for the IMAX and Kodak datasets among the considered combinations, but the FSIM value of our combination is little lower than 4:2:0(A)-BICU for the SCI dataset.

4.1.2 For I^{Bayer}

Based on the testing Bayer CFA images collected from the three datasets, the PSNR, SSIM, and FSIM values are used to compare the quality of the reconstructed Bayer CFA images. Table 3 indicates that our combination 'ours-BILI' has the highest average PSNR,

Table 2 The CPSNR, SSIM, FSIM, and time comparison among the considered combinations for I^{RGB}

I^{RGB}	CPSNR (dB)	CPSNR Gain (dB)	SSIM	SSIM Gain	FSIM	FSIM Gain	Time (s)
4:2:0(A)-BILI	(38.2951)	(2.1305)	(0.9645)	(0.0109)	(0.999685)	(0.000174)	(0.0011)
	[45.8896]	[1.7567]	[0.9871]	[0.0035]	[0.999981]	[0.000005]	[0.0369]
	{31.7219}	{1.6623}	{0.974}	{0.0056}	{0.998701}	{-0.00028}	{0.004}
	38.6355	1.8499	0.9752	0.0067	0.999456	-0.000035	0.0140
4:2:0(L)-BILI	(38.731)	(1.6946)	(0.9672)	(0.0082)	(0.99968)	(0.000179)	(0.0008)
	[46.9503]	[0.696]	[0.9895]	[0.0011]	[0.999981]	[0.000005]	[0.0188]
	{31.7602}	{1.624}	{0.9745}	{0.0051}	{0.997757}	{0.00066}	{0.0019}
	39.1472	1.3382	0.9771	0.0048	0.999139	0.000282	0.0072
4:2:0(R)-BILI	(38.7328)	(1.6928)	(0.9671)	(0.0083)	(0.99968)	(0.000179)	(0.0006)
	[45.4934]	[2.1529]	[0.986]	[0.0046]	[0.99998]	[0.000006]	[0.021]
	{31.7801}	{1.6041}	{0.9748}	{0.0048}	{0.9978}	{0.000617}	{0.0022}
	38.6688	1.8166	0.9760	0.0059	0.999153	0.000268	0.0079
4:2:0(D)-BILI	(38.8903)	(1.5353)	(0.968)	(0.0074)	(0.999653)	(0.000206)	(0.0007)
	[47.3983]	[0.248]	[0.99]	[0.0006]	[0.999978]	[0.000008]	[0.019]
	{31.6288}	{1.7554}	{0.9756}	{0.004}	{0.997219}	{0.001198}	{0.0019}
	39.3058	1.1796	0.9779	0.0040	0.998950	0.000471	0.0072
4:2:0(MPEG-B)-BILI	(38.5787)	(1.8469)	(0.9658)	(0.0096)	(0.999644)	(0.000215)	(0.0021)
	[46.7852]	[0.8611]	[0.9885]	[0.0021]	[0.999982]	[0.000004]	[0.0696]
	{31.5463}	{1.8379}	{0.9742}	{0.0054}	{0.998263}	{0.000154}	{0.008}
	38.9701	1.5153	0.9762	0.0057	0.999296	0.000125	0.0266
IDID-NEDI [30]	(42.9659)	(0.8914)	(0.9683)	(0.0071)	(0.999635)	(0.000224)	(7.5032)
	[47.623]	[0.0233]	[0.9906]	[0]	[0.999984]	[0.000002]	[185.509]
	{32.3443}	{1.0399}	{0.9742}	{0.0054}	{0.998393}	{0.000024}	{21.3944}
	39.5510	0.9344	0.9777	0.0042	0.999344	0.000077	71.4689
4:2:0(A)-BICU	(39.7294)	(0.6962)	(0.9714)	(0.004)	(0.999838)	(0.000021)	(0.001)
	[46.9147]	[0.7316]	[0.989]	[0.0016]	[0.999984]	[0.000002]	[0.0343]
	{32.4623}	{0.9219}	{0.9754}	{0.0042}	{0.9989}	{-0.000483}	{0.0038}
	39.7021	0.7833	0.9786	0.0033	0.999574	-0.000153	0.0130
4:2:0(L)-BICU	(39.5839)	(0.8417)	(0.9708)	(0.0046)	(0.999757)	(0.000102)	(0.0005)
	[47.4031]	[0.2432]	[0.9901]	[0.0005]	[0.999983]	[0.000003]	[0.0209]
	{31.9891}	{1.3951}	{0.9743}	{0.0053}	{0.997657}	{0.00076}	{0.0023}
	39.6587	0.8267	0.9784	0.0035	0.999132	0.000289	0.0079
4:2:0(R)-BICU	(39.5842)	(0.8414)	(0.9707)	(0.0047)	(0.999757)	(0.000102)	(0.0007)
	[46.2148]	[1.4315]	[0.9873]	[0.0033]	[0.999982]	[0.000004]	[0.0219]
	{32.0039}	{1.3803}	{0.9746}	{0.005}	{0.997697}	{0.00072}	{0.0025}
	39.2676	1.2178	0.9775	0.0044	0.999145	0.000276	0.0084
4:2:0(D)-BICU	(39.2455)	(1.1801)	(0.9688)	(0.0066)	(0.999673)	(0.000186)	(0.0006)
	[47.1661]	[0.4802]	[0.9896]	[0.001]	[0.999978]	[0.000008]	[0.0199]

Table 2 (continued)

I^{RGB}	CPSNR (dB)	CPSNR Gain (dB)	SSIM	SSIM Gain	FSIM	FSIM Gain	Time (s)
	{31.5865}	{1.7977}	{0.9738}	{0.0058}	{0.997008}	{0.001409}	{0.0022}
	39.3327	1.1527	0.9774	0.0045	0.998886	0.000535	0.0076
4:2:0(MPEG-B)-BICU	(39.125)	(1.3006)	(0.968)	(0.0074)	(0.999683)	(0.000176)	(0.002)
	[46.9216]	[0.7247]	[0.9887]	[0.0019]	[0.999983]	[0.000003]	[0.0783]
	{31.748}	{1.6362}	{0.9734}	{0.0062}	{0.998239}	{0.000178}	{0.0077}
	39.2649	1.2205	0.9767	0.0052	0.999302	0.000119	0.0293
MCIM-BICU [24]	(39.1827)	(1.2429)	(0.9707)	(0.0047)	(0.999686)	(0.000173)	(0.0649)
	[48.1996]	[-0.5533]	[0.9911]	[-0.0005]	[0.999985]	[0.000001]	[1.7914]
	{32.7887}	{0.5955}	{0.9732}	{0.0064}	{0.998742}	{-0.000335}	{0.1962}
	40.0570	0.4284	0.9783	0.0036	0.999471	-0.000050	0.6842
ours-BILI	(**40.4256**)		(**0.9754**)		(**0.999859**)		(0.0294)
	[47.6463]		[0.9906]		**[0.999986]**		[0.7525]
	{**33.3842**}		{**0.9796**}		{0.998417}		{0.1137}
	40.4854		**0.9819**		0.999421		0.2984

SSIM, and FSIM in boldface among the considered combinations. In Table 3, the individual results of IMAX, Kodak, and SCI are still tabulated in the parentheses (), [ ], and {}, respectively. In Particular, the PSNR gain of our combination is 1.27 dB over the currently published work 'GD-BILI' [8]. Besides the quality enhancement merit, our chroma subsampling method is also faster than the GD method. Although our method takes more time than 4:2:0(A) and DI [16], our method has significant quality improvement relative to the two methods.

As mentioned before, the COPY-based block-distortion function used in the modified 4:2:0(A) method [17] is based on the demosaiced RGB full-color block-distortion criterion, but not on the Bayer CFA block-distortion criterion. Therefore, their combination 'modified 4:2:0(A)-TN' mainly benefits the quality of the reconstructed RGB full-color image, but not for the reconstructed Bayer CFA image.

4.1.3 For I^{DTDI}

When setting $t = `DTDI'$, the proposed chroma subsampling method can well handle chroma subsampling for I^{DTDI}. Based on the same three datasets, Table 4 indicates that our combination 'ours-BILI' has the highest average CPSNR among the considered combinations. The average CPSNR gains of our combination are 2.288 dB and 0.8128 dB over 4:2:0(A)-BILI and CD-CA [7], respectively. Table 4 also demonstrates that our combination has the highest average SSIM, but the FSIM value of our combination is little lower than 4:2:0(A)-BILI. For each image, the average execution time requirement (in seconds) of each considered chroma subsampling method is tabulated in Table 4. Although our chroma subsampling method takes more time than 4:2:0(A) and the CD method, our method has a significant quality enhancement effect.

Table 3 The PSNR, SSIM, FSIM, and time comparison among the considered combinations for I^{Bayer}

I^{Bayer}	4:2:0(A)-BILI	DI-COPY [16]	GD-BILI [8]	ours-BILI
PSNR (dB)	(40.5732)	(44.0523)	(45.8896)	**(47.4698)**
	[47.4013]	[49.2148]	[51.7971]	**[52.6374]**
	{33.0005}	{37.9361}	{38.096}	{**39.4854**}
	40.3250	43.7344	45.2609	**46.5309**
PSNR Gain (dB)	(6.8966)	(3.4175)	(1.5802)	
	[5.2361]	[3.4226]	[0.8403]	
	{6.4849}	{1.5493}	{1.3894}	
	6.2059	2.7965	1.2700	
SSIM	(0.9954)	(0.9978)	(0.9985)	**(0.9988)**
	[0.9966]	[0.9981]	[0.9988]	**[0.999]**
	{0.9905}	{0.9964}	{0.9958}	{**0.9964**}
	0.9942	0.9974	0.9977	**0.9981**
SSIM Gain	(0.0034)	(0.001)	(0.0003)	
	[0.0024]	[0.0009]	[0.0002]	
	{0.0059}	{0}	{0.0006}	
	0.0039	0.0007	0.0004	
FSIM	(0.998158)	(0.997942)	(0.99919)	**(0.999303)**
	[0.999978]	[0.999897]	[0.999985]	**[0.999987]**
	{0.997324}	{0.997783}	{0.998511}	{**0.998679**}
	0.998487	0.998541	0.999229	**0.999323**
FSIM Gain	(0.001145)	(0.001361)	(0.000113)	
	[0.000009]	[0.000090]	[0.000002]	
	{0.001355}	{0.000896}	{0.000168}	
	0.000836	0.000782	0.000094	
Time (s)	(0.001)	(0.0011)	(0.022)	(0.0176)
	[0.0386]	[0.0352]	[0.472]	[0.3767]
	{0.0035}	{0.0033}	{0.0746}	{0.0532}
	0.0144	0.0132	0.1895	0.1492

4.2 Quality-bitrate tradeoff and visual effect merits of our method

When setting QP =0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, and 51, the visual effect merit of our method is first illustrated, and then we demonstrate the quality-bitrate tradeoff merit of our combination 'ours-BILI' over the other combinations. The quality-bitrate tradeoff of each combination is first represented by a RD-curve where the X-axis denotes the average bitrate required for one image and the Y-axis denotes the average CPSNR or the average PSNR of the reconstructed color images. Next, we demonstrate the average BD-PSNR merit of our combination 'ours-BILI' over the other combinations relative to the baseline combination '4:2:0(A)-BILI'.

Table 4 The CPSNR, SSIM, FSIM, and time comparison among the considered combinations for I^{DTDI}

I^{DTDI}	4:2:0(A)-BILI	CD-CA [7]	ours-BILI
CPSNR (dB)	(41.0412)	(42.9797)	(**43.747**)
	[48.4785]	[49.1936]	[**50.0855**]
	{34.1616}	{35.9337}	{**36.7127**}
	41.2271	42.7023	**43.5151**
CPSNR Gain (dB)	(2.7058)	(0.7673)	
	[1.607]	[0.8919]	
	{2.5511}	{0.779}	
	2.2880	0.8128	
SSIM	(0.9891)	(0.9927)	(**0.9942**)
	[0.9975]	[0.9979]	[**0.9984**]
	[0.9893]	[0.9919]	[**0.992**]
	0.9920	0.9942	**0.9949**
SSIM Gain	(0.0051)	(0.0015)	
	[0.0009]	[0.0005]	
	{0.0027}	{0.0001}	
	0.0029	0.0007	
FSIM	(0.999322)	(0.999238)	(**0.999524**)
	[0.999983]	[0.999980]	[**0.999988**]
	{0.998385}	{0.997601}	{0.997968}
	0.999230	0.998940	0.999160
FSIM Gain	(0.000202)	(0.000286)	
	[0.000005]	[0.000008]	
	{-0.00042}	{0.000367}	
	-0.000070	0.000220	
Time (s)	(0.001)	(0.0006)	(0.024)
	[0.0302]	[0.0185]	[0.5652]
	{0.0037}	{0.0021}	{0.0795}
	0.0116	0.0071	0.2229

We take the 22nd ground truth SCI image in Fig. 6(a) as the testing image to demonstrate the visual effect merit of the reconstructed RGB full-color image using our combination "ours-BILI" relative to the seven comparative combinations. The magnified subimage in Fig. 6(b) is decoupled from the grey color region containing two words "one slide" on the inside of the bottom red rectangle in Fig. 6(a). For QP = 51, after performing the eight considered combinations on Fig. 6(a), the eight reconstructed magnified subimages for Fig. 6(b) are shown in Figs. 6(c)-(j), and we observe that our combination "ours-BILI" has the best visual effect in the eight combinations.

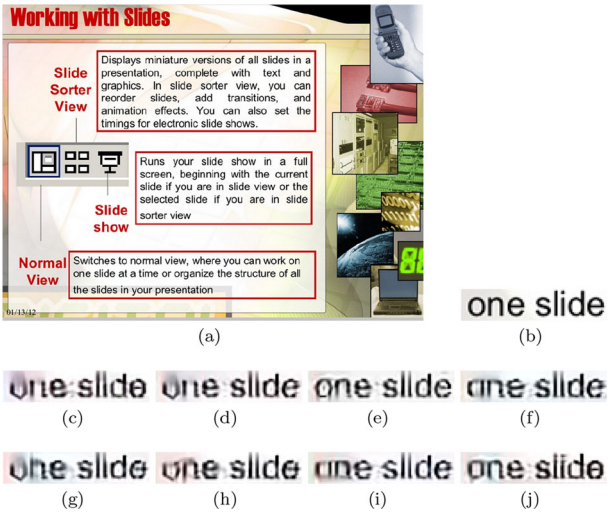

Fig. 6 The visual effect merit of our method. (a) The 22nd ground truth SCI image. (b) The magnified subimage of (a). (c) 4:2:0(A)-BILI. (d) 4:2:0(L)-BILI. (e) 4:2:0(R)-BILI. (f) 4:2:0(D)-BILI. (g) 4:2:0(MPEG-B)-BILI. (h) IDID-NEDI. (i) MCIM-BICU. (j) ours-BILI

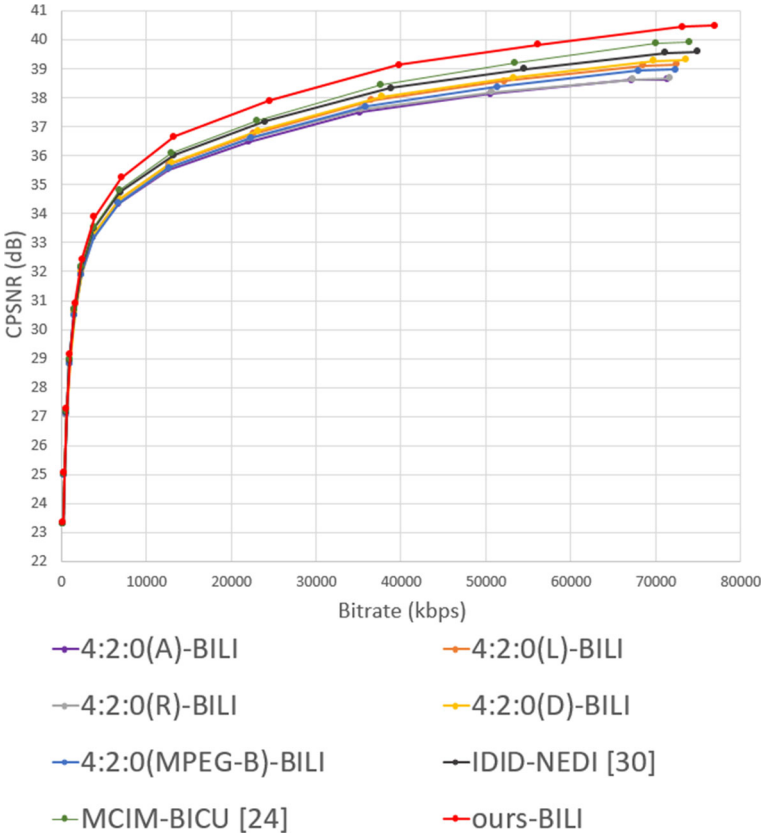

Fig. 7 The quality-bitrate tradeoff merit of our combination for I^{RGB}

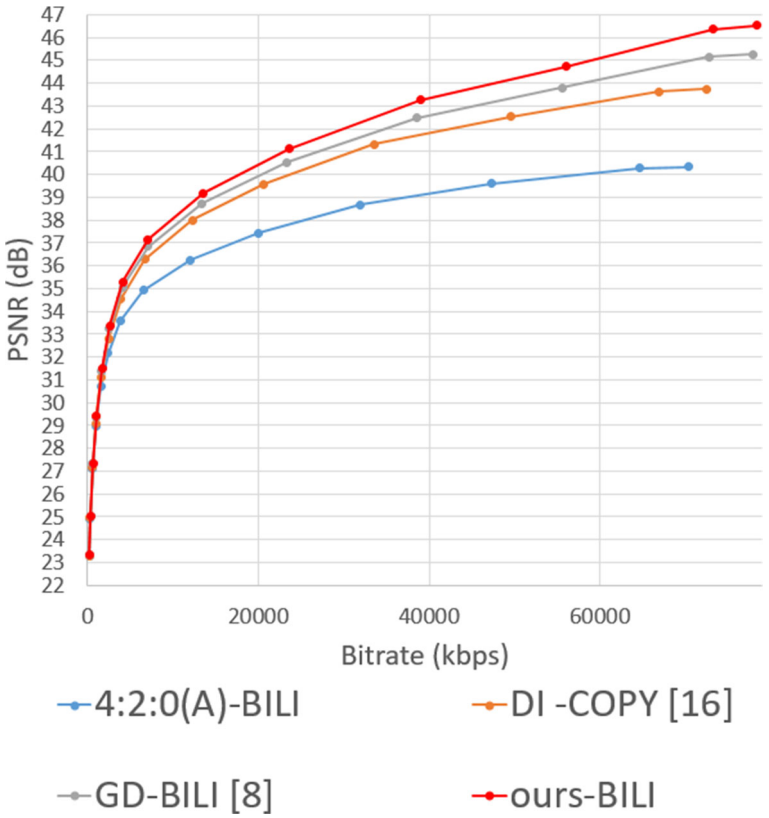

Fig. 8 The quality-bitrate tradeoff merit of our combination for I^{Bayer}

4.2.1 The quality-bitrate tradeoff merit in terms of RD curves

Let B_1, B_2, and B_3 denote the total bits required for the Kodak dataset with 24 images, the IMAX dataset with 18 images, and the SCI dataset with 26 images, respectively. Further, the average bitrate requirement of one image is defined as $\frac{1}{3}(\frac{B_1}{24} + \frac{B_2}{18} + \frac{B_3}{26})$.

For I^{RGB}, the RD curves of the considered combinations are illustrated in Fig. 7 and we observe that under the same bitrate condition, our combination 'ours-BILI' has the highest CPSNR among the considered combinations. For I^{Bayer} and I^{DTDI}, the RD curves of the considered combinations are depicted in Figs. 8 and 9, respectively, indicating that under the same bitrate condition, our combination still has the best quality among the considered combinations.

4.2.2 The BD-PSNR merit

Finally, we demonstrate the average BD-PSNR merit of our combination 'ours-BILI' over the other combinations relative to the baseline combination '4:2:0(A)-BILI'. Here, the average BD-PSNR reduction percentage [4] is used to evaluate the average PSNR (CPSNR) difference under the same bitrate. In Tables 5, 6 and 7, we list the BD-PSNR results for the

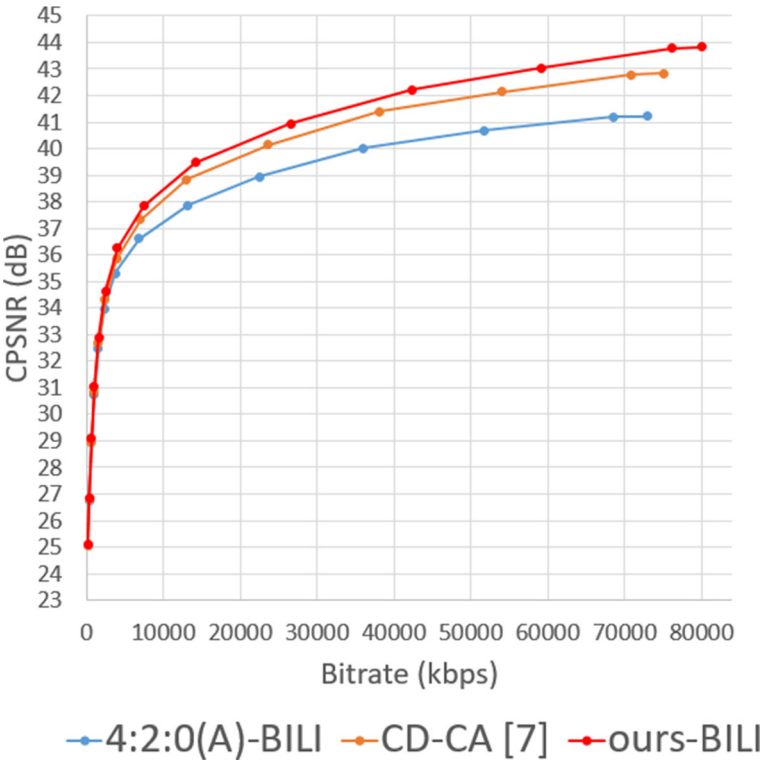

Fig. 9 The quality-bitrate tradeoff merit of our combination for I^{DTDI}

Table 5 Average BD-PSNR reduction(%) among the considered combinations for I^{RGB}

	QP 4-16	QP 12-24	QP 20-32	QP 28-48
4:2:0(L)-BILI	(0.4046)	(0.349)	(0.2273)	(0.1074)
	[0.1914]	[0.155]	[0.1006]	[0.0377]
	{0.0395}	{0.0146}	{-0.0393}	{-0.0668}
	0.345	0.1946	0.1033	0.0372
4:2:0(R)-BILI	(0.3615)	(0.2265)	(0.0576)	(-0.0192)
	[-0.4528]	[-0.2173]	[-0.1035]	[-0.0385]
	{0.0415}	{-0.0083}	{-0.1097}	{-0.2055}
	0.0694	0.0703	0.0162	-0.0505
4:2:0(D)-BILI	(0.5373)	(0.4722)	(0.2745)	(0.115)
	[-0.005]	[0.1144]	[0.1149]	[0.0447]
	{-0.091}	{-0.1247}	{-0.2391}	{-0.3264}
	0.4163	0.2225	0.0867	-0.0259
4:2:0(MPEG-B)-BILI	(0.2639)	(0.226)	(0.1277)	(0.0524)
	[0.1655]	[0.1355]	[0.0822]	[0.0276]
	{-0.1846}	{-0.1793}	{-0.1679}	{-0.1134}
	0.1799	0.0745	0.0126	-0.0098

Table 5 (continued)

	QP 4-16	QP 12-24	QP 20-32	QP 28-48
IDID-NEDI [30]	(0.4223)	(0.3808)	(0.2336)	(0.1185)
	[0.6467]	[0.2279]	[0.8435]	[2.8035]
	{0.5417}	{0.4655}	{0.2939}	{0.0751}
	0.6504	0.4504	0.2896	0.1262
MCIM-BICU [24]	(0.4596)	(0.4316)	(0.2821)	(0.157)
	[0.8188]	[0.1954]	[0.8117]	[2.778]
	{0.948}	{0.8234}	{0.5894}	{0.3058}
	0.8553	0.5673	0.3764	0.1887
ours-BILI	**(1.7688)**	**(1.2886)**	**(0.6687)**	**(0.2604)**
	[0.5891]	**[0.2774]**	**[0.8688]**	**[2.8124]**
	{**1.4411**}	{**1.2277**}	{**0.806**}	{0.2696}
	1.3976	**1.0794**	**0.7104**	**0.3075**

Table 6 Average BD-PSNR reduction(%) among the considered combinations for I^{Bayer}

	QP 4-16	QP12-24	QP 20-32	QP 28-40
DI -COPY [16]	(2.4526)	(1.2947)	(0.4836)	(0.121)
	[0.104]	[-0.0227]	[-0.0262]	[-0.0081]
	{4.6959}	{4.0379}	{2.5815}	{1.0707}
	2.6005	1.8401	1.1505	0.4415
GD-BILI [8]	(3.7312)	(2.1115)	(0.8921)	(0.3252)
	[0.5252]	[0.1554]	[0.057]	[0.0287]
	{4.7515}	{4.1333}	{2.7383}	{1.0753}
	3.3876	2.3528	1.5338	0.6371
ours-BILI	**(4.5996)**	**(2.4496)**	**(0.9574)**	(0.3242)
	[0.5348]	[0.1526]	[0.0537]	[0.0259]
	{**5.9544**}	{**5.0503**}	{**3.1936**}	{**1.1494**}
	4.1446	**2.803**	**1.7858**	**0.699**

Table 7 Average BD-PSNR reduction(%) among the considered combinations for I^{DTDI}

	QP 4-16	QP12-24	QP 20-32	QP 28-40
CD-CA [7]	(1.7643)	(1.0408)	(0.3377)	(0.036)
	[0.3244]	[0.2149]	[0.0388]	[-0.0107]
	{1.7304}	{1.5997}	{1.152}	{0.4076}
	1.2864	1.001	0.6072	0.1991
ours-BILI	**(2.678)**	**(1.5211)**	**(0.6813)**	**(0.1454)**
	[0.5771]	**[0.3611]**	**[0.1065]**	**[0.0382]**
	{**2.3748**}	{**2.1196**}	{**1.4493**}	{0.3953}
	1.894	**1.4666**	**0.9461**	**0.3525**

considered combinations relative to the baseline for I^{RGB}, I^{Bayer}, and I^{DTDI}, respectively, indicating the best BD-PSNR reduction merit of our combination.

5 Conclusion

In the past years, several chroma subsampling methods were developed for I^{RGB}, I^{Bayer}, and I^{DTDI} separately. In this paper, we have presented the proposed effective BILI-based iterative chroma subsampling method to tackle chroma subsampling for the above-mentioned three image types simultaneously, in particular, using the same demosaicking method for I^{Bayer} and I^{DTDI}. Instead of the COPY-based block-distortion function used in the previous works only for I^{Bayer}, we first propose a BILI-based block-distortion function for the considered three image types. Next, using a mathematical inequality, we prove that the proposed block-distortion function is a convex function. Then, we propose a more effective method to handle chroma subsampling for I^{RGB}, I^{Bayer}, and I^{DTDI} simultaneously. The comprehensive experimental results have justified that under the VVC compression platform VTM-8.0, our chroma subsampling method achieves substantial quality enhancement and BD-PSNR improvement effects when compared with the traditional methods and the existing methods [7, 8, 16, 17, 24, 30]. The future work is to apply our iterative chroma subsampling method to tackle the chroma subsampling operation for wireless capsule endoscopy compression [2].

Acknowledgments This work was supported by the contracts MOST-107-2221-E-011-108-MY3 and MOST-108-2221-E-011-077-MY3 of the Ministry of Science and Technology, Taiwan. The authors appreciate the valuable comments of the three anonymous referees and the proofreading help of Ms. C. Harrington to improve the manuscript.

References

1. Allebach J, Wong PW (1996) Edge-directed interpolation. In: Proceedings of 3rd IEEE international conference on image processing, vol 3, pp 707–710. https://doi.org/10.1109/ICIP.1996.560768
2. Babu C, Chandy DA, Karthigaikumar P (2020) Novel chroma subsampling patterns for wireless capsule endoscopy compression. Neural Comput & Applic 32(10):6353–6362
3. Bayer BE (1976) Color imaging array. US Patent 3,971,065
4. Bjøntegaard G (2001) Calculation of average psnr differences between rd-curves (vceg-m33). In: Document VCEG-m33, 13th ITU-t video coding experts group (VCEG) meeting, Austin, TX,USA, pp 2–4
5. Bodenstorfer E, Fürtler J, Brodersen J, Mayer KJ, Eckel C, Gravogl K, Nachtnebel H (2007) High-speed line-scan camera with digital time delay integration. In: Real-Time Image Processing 2007, International Society for Optics and Photonics, vol 6496, p 64960I. https://doi.org/10.1117/12.704516
6. Chen H, Sun M, Steinbach E (2009) Compression of Bayer-pattern video sequences using adjusted chroma subsampling. IEEE Trans Circuits Syst Video Technol 19(12):1891–1896. https://doi.org/10.1109/TCSVT.2009.2031370
7. Chung KL, Yang WJ, Chen CH, Liao HYM, Zeng SM (2011) Efficient chroma subsampling strategy for compressing digital time delay integration mosaic video sequences in h.264/AVC. Journal of Electronic Imaging 20(2):1–16. https://doi.org/10.1117/1.3586799
8. Chung KL, Lee YL, Chien WC (2019) Effective gradient descent-based chroma subsampling method for Bayer CFA images in HEVC. IEEE Trans Circuits Syst Video Technol 29(11):3281–3290. https://doi.org/10.1109/TCSVT.2018.2879095
9. Datta BN (1995) Numerical linear algebra and applications, 1st edn. Philadelphia PA, USA,Brooks/Cole
10. Eastman Kodak Company (2014) Kodak dataset. https://www.math.purdue.edu/lucier/PHOTO_CD/BMP_IMAGES/

11. ITU-R (2011) BT-601-5: Studio encoding parameters of digital television for standard 4:3 and widescreen 16:9 aspect ratios. International Telecommunications Union
12. ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group (2020) VTM-8.0. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM
13. Kiku D, Monno Y, Tanaka M, Okutomi M (2013) Residual interpolation for color image demosaicking. In: 2013 IEEE international conference on image processing, pp 2304–2308 https://doi.org/10.1109/ICIP.2013.6738475
14. Li X, Orchard MT (2001) New edge-directed interpolation. IEEE Trans Image Process 10(10):1521–1527. https://doi.org/10.1109/83.951537
15. Li X, Gunturk B, Zhang L (2008) Image demosaicing: A systematic survey. In: Visual communications and image processing 2008, International Society for Optics and Photonics, vol 6822, p 68221J. https://doi.org/10.1117/12.766768
16. Lin CH, Chung KL, Yu CW (2016) 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 26(9):1722–1733. https://doi.org/10.1109/TCSVT.2015.2472118
17. Lin TL, Yu YC, Jiang KH, Liang CF, Liaw PS (2020) Novel chroma sampling methods for CFA video compression in AVC, HEVC and VVC. IEEE Trans Circuits Syst Video Technol 30(9):3167–3180. https://doi.org/10.1109/TCSVT.2019.2939280
18. Lu Y, Li S, Shen H (2011) Virtualized screen: a third element for cloud–mobile convergence. IEEE Multimedia Magazine 18(2):4–11. https://doi.org/10.1109/MMUL.2011.33
19. Lukac R, Plataniotis KN (2005) Color filter arrays: design and performance analysis. IEEE Trans Consum Electron 51(4):1260–1267. https://doi.org/10.1109/TCE.2005.1561853
20. Ni Z, Ma KK, Zeng H, Zhong B (2020) Color image demosaicing using progressive collaborative representation. IEEE Trans Image Process 29:4952–4964. https://doi.org/10.1109/TIP.2020.2975978
21. Ridge J, Karczewicz M (2007) Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q. 6)
22. SCI Image Database (2021) ftp://140.118.175.164/SCI26
23. Tan DS, Chen WY, Hua KL (2018) DeepDemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks. IEEE Trans Image Process 27(5):2408–2419. https://doi.org/10.1109/TIP.2018.2803341
24. Wang S, Gu K, Ma S, Gao W (2016) Joint chroma downsampling and upsampling for screen content image. IEEE Trans Circuits Syst Video Technol 26(9):1595–1609. https://doi.org/10.1109/TCSVT.2015.2461891
25. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612. https://doi.org/10.1109/TIP.2003.819861
26. Yu YC, Jhang JW, Wei X, Tseng HW, Wen Y, Liu Z, Lin TL, Chen SL, Chiou YS, Lee HY (2017) Chroma upsampling for YCbCr 420 videos. In: 2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), pp 163–164. https://doi.org/10.1109/ICCE-China.2017.7991046
27. Zhang L, Wu X, Buades A, Li X (2011a) Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic Imaging 20(2):023016–1–023016–16. https://doi.org/10.1117/1.3600632
28. Zhang L, Zhang L, Mou X, Zhang D (2011b) FSIM: A feature similarity index for image quality assessment. IEEE Trans Image Process 20(8):2378–2386. https://doi.org/10.1109/TIP.2011.2109730
29. Zhang L, Wu X, Buades A, Li X (2014) IMAX dataset. http://www.comp.polyu.edu.hk/cslzhang/CDM_Dataset.htm
30. Zhang Y, Zhao D, Zhang J, Xiong R, Gao W (2011c) Interpolation-dependent image downsampling. IEEE Trans Image Process 20(11):3291–3296. https://doi.org/10.1109/TIP.2011.2158226

Kuo-Liang Chung received his B.S., M.S., and Ph.D. degrees from National Taiwan University, Taipei, Taiwan in 1982, 1984, and 1990, respectively. He has been one Chair Professor of the Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan since 2009. He was the recipient of the Distinguished Research Award (2004–2007; 2019–2022) from the Ministry of Science and Technology of Taiwan. He has been an Editor and Associate Editor of Signals and the Journal of Visual Communication and Image Representation since 2020 and 2011, respectively. His research interests include machine learning, image processing, and video compression.

Szu-Ni Chen received her B.S. degree in computer science and information engineering from the National Taiwan Ocean University National, Keelung, Taiwan, in 2018. She is currently pursuing her M.S. degree in computer science and information engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan. Her research interests include image processing and video compression.