IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 22, 2020, accepted May 25, 2020, date of publication June 4, 2020, date of current version July 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999910

Effective Chroma Subsampling and Luma Modification for RGB Full-Color Images Using the Multiple Linear Regression Technique

KUO-LIANG CHUNG, (Senior Member, IEEE), JEN-SHUN CHENG, AND HONG-BIN YANG
Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10672, Taiwan

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

This work was supported by Grant MOST-107-2221-E-011-108-MY3 and Grant MOST-108-2221-E-011-077-MY3.

ABSTRACT Differing from the traditional chroma subsampling on the YUV image converted from a RGB full-color image, in this paper, we propose a novel and effective chroma subsampling and luma modification (CSLM) method. For each 2 × 2 YUV block, first, a newly reconstructed 2 × 2 RGB full-color block-distortion model is proposed, and then we propose a multiple linear regression approach to tackle our CSLM method such that the reconstructed 2 × 2 RGB full-color block-distortion can be minimized, achieving significant quality improvement of the reconstructed RGB full-color image. Based on the Kodak and IMAX datasets, the comprehensive experimental results demonstrated that on the versatile video coding (VVC) platform VTM-8.0, our method achieves substantial quality and quality-bitrate tradeoff improvement of the reconstructed RGB full-color images relative to six traditional methods and the three state-of-the-art methods.

INDEX TERMS Chroma subsampling, distortion model, Luma modification, multiple linear regression, quality-bitrate tradeoff, RGB full-color image, versatile video coding (VVC).

I. INTRODUCTION

As a target image for human visual perception, the RGB full-color image I^{RGB} is the most important medium. In the traditional coding system, as shown in Fig. 1, prior to compression, I^{RGB} is first converted to the YUV image I^{YUV} by the following RGB-to-YUV conversion formula:

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

where in each 2 × 2 YUV block, (Y_i, U_i, V_i), $1 \le i \le 4$, denotes the converted YUV triple-value in zigzag order; (R_i, G_i, B_i) denotes the collocated original RGB triple-value. Traditionally, the study on the chroma subsampling for I^{YUV} at the server side, as shown in the upper part of Fig. 1, can be classified into two categories, namely, 4:2:0 and 4:2:2. 4:2:0 subsamples the (U, V)-pair for each 2 × 2 UV block B^{UV}; 4:2:2 subsamples the (U, V)-pair for each row of B^{UV}. In this paper, our discussion focuses on 4:2:0, although it is applicable to 4:2:2. 4:2:0 has been used in Blu-ray discs (BDs) and digital versatile discs (DVDs) for recording movies, sports, and so on.

After decompressing the encoded subsampled YUV image by the decoder, each decoded subsampled YUV image is upsampled at the client side, as shown in the lower part of Fig. 1. Furthermore, each upsampled YUV pixel is converted into a RGB full-color pixel by the following YUV-to-RGB conversion formula:

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

As a result, the reconstructed RGB full-color image is produced.

Note that all discussion in this paper can be applied to other color spaces, such as the YC_bC_r color space, because the RGB-to-YC_bC_r and YC_bC_r-to-RGB color conversions are also linear as the color conversion between the RGB space and the YUV space appeared in Eqs. (1)-(2). The digital YUV data in Eqs. (1)-(2) are originally converted from analog signals, while the YC_bC_r data are originally digital. The YUV

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

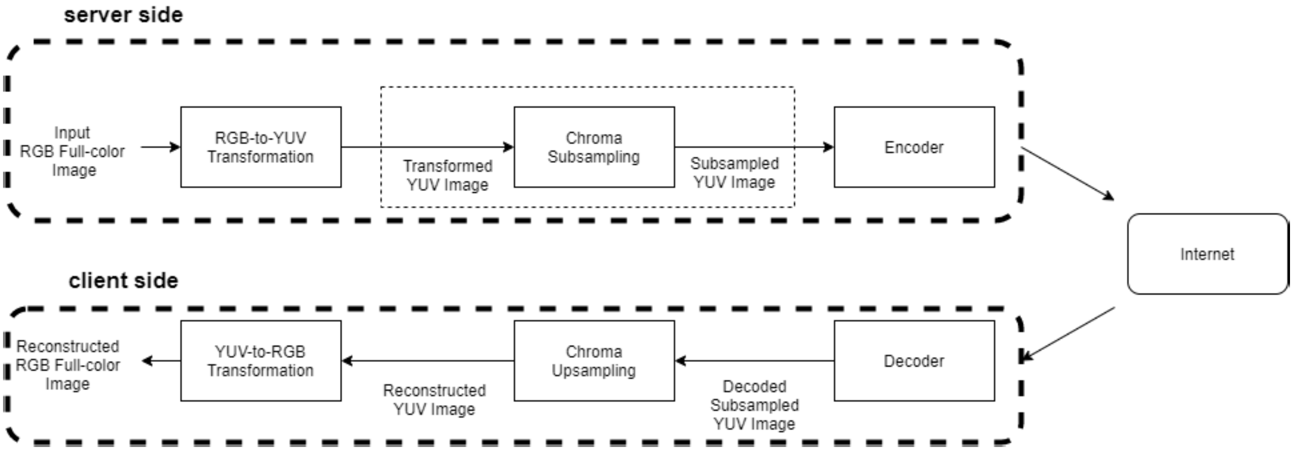


FIGURE 1. The chroma subsampling model in the coding system for the input RGB full-color image.

color space is often used in analogue color TV broadcasting; the YC$_b$C$_r$ color space is often used in digital videos and BT.601.

A. RELATED WORK

In this subsection, we introduce six traditional chroma subsampling methods and three state-of-the-art chroma subsampling methods [1], [6], [12]. All of them will be included in the comparative methods.

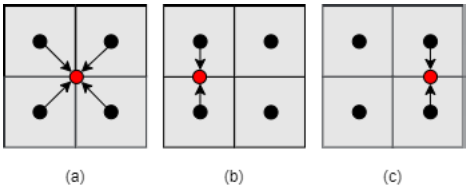


FIGURE 2. The depiction of three traditional chroma subsampling methods. (a) 4:2:0(A). (b) 4:2:0(L). (c) 4:2:0(R).

1) Six traditional chroma subsampling methods: We first introduce the six traditional chroma subsampling methods which are 4:2:0(A), 4:2:0(L), 4:2:0(R), 4:2:0(MPEG-B) [7], 4:2:0(BRIGHT), and 4:2:0(BRIGHT_MEAN). For each 2×2 UV block B^{UV}, 4:2:0(A) subsamples the (U, V)-pair by averaging the four U and V components of B^{UV}. 4:2:0(L) and 4:2:0(R) subsample the (U, V)-pairs by averaging the chroma components in the left and right columns of B^{UV}, respectively. For convenience, we have drawn 4:2:0(A), 4:2:0(L), and 4:2:0(R), as depicted in Fig. 2. 4:2:0(MPEG-B) determines the subsampled (U, V)-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^{UV}. 4:2:0(BRIGHT) determines the subsampled (U, V)-pair from the location with the largest luma value in the collocated 2×2 luma block B^Y. 4:2:0(BRIGHT_MEAN) is equal to 4:2:0(BRIGHT) when the ratio of the largest luma value in B^Y over the smallest is larger than 2; otherwise, it is equal to 4:2:0(A).

The common weakness of the above traditional chroma subsampling methods is the failure to take the reconstructed RGB full-color block distortion into account, limiting the quality improvement of the reconstructed RGB full-color image.

2) Three state-of-the-art chroma subsampling methods: To overcome the weakness existing in the traditional methods, three state-of-the-art methods were developed.

Based on the new edge-directed interpolation (NEDI) [5], Zhang *et al.* [12] proposed an interpolation-dependent image downsampling (IDID) method for chroma subsampling, and their combination IDID-NEDI, in which IDID is used at the server side and NEDI is used at the client side, can tackle the chroma downsampling well.

Lin *et al.* [6] proposed a modified chroma 4:2:0(A) subsampling method, namely modified 4:2:0(A), by considering the truncation and carry operations influence; among the four considered variants, they select the best subsampled (U, V)-pair. At the client side, they improved the previous chroma upsampling process [10] by considering the distance between each estimated upsampled chroma value and its three neighboring (TN) pixels; their combination, modified 4:2:0(A)-TN, achieves good quality of the reconstructed RGB full-color image.

After performing a chroma subsampling, e.g. 4:2:0(A), under the COPY-based chroma upsampling process in which each estimated (U, V)-pair of each 2×2 UV block B^{UV} just copies the subsampled (U, V)-pair of B^{UV}, Chung *et al.* [1] proposed a pixel-based approach to adjust each luma pixel in the 2×2 luma block to improve the quality of the reconstructed RGB full-color image. The first weakness in [1] is that the COPY-based chroma upsampling process used at the server side is too simple to meet the chroma upsampling capability at the client side, such as the BILI upsampling method, thus limiting the quality improvement. The second weakness in [1] is the failure to consider chroma subsampling and luma adjustment simultaneously to achieve better quality improvement.

B. CONTRIBUTIONS

In this paper, we propose a novel chroma subsampling and luma modification (CSLM) method for I^{RGB}. The three contributions of this paper are clarified in the following aspects.

In the first contribution of our CSLM method, for each 2×2 YUV block, a novel reconstructed 2×2 RGB full-color block-distortion model is proposed. Considering the neighboring subsampled (U, V)-pairs of the current 2×2 UV block, we deploy the BILI interpolation into the block-distortion model to better meet the chroma upsampling capability at the client side.

In the second contribution of CSLM, the reconstructed 2×2 RGB full-color block-distortion model can be transformed to an overdetermined system by a multiple linear regression approach. Furthermore, the matrix pseudoinverse technique is applied to determine the subsampled (U, V)-pair and the four luma values such that the reconstructed block-distortion can be minimized.

In the third contribution, based on the Kodak and IMAX datasets, the comprehensive experimental results demonstrated that on the versatile video coding (VVC) platform VTM-8.0 [8], our CSLM method achieves significant quality and quality-bitrate tradeoff improvement of the reconstructed RGB full-color images relative to the six traditional methods and three state-of-the-art methods in [1], [6], [12]. Here, the quality metrics used are CPSNR (color peak-signal-noise-ratio), SSIM (structure similarity index) [9], and FSIMc (feature similarity index) [11]; the quality-bitrate tradeoff metric is illustrated by the RD-curves (rate-distortion curves) for different quantization parameter (QP) values. In addition, based on the video sequence "Boat" which can be accessed from [6], the quality-bitrate tradeoff merit of our method is reported.

The rest of this paper is organized as follows. In Section II, the reconstructed 2×2 RGB full-color block-distortion model is first presented, and then the corresponding overdetermined system is derived. In Section III, the matrix pseudoinverse technique is applied to determine the subsampled (U, V)-pair and the four modified luma values. In Section IV, the comprehensive experimental results are demonstrated to justify the significant quality and quality-bitrate tradeoff merits of our CSLM method. In Section V, some concluding remarks are addressed.

II. THE RECONSTRUCTED BLOCK-DISTORTION MODEL AND THE DERIVED OVERDETERMINED SYSTEM

In this section, we first present a mathematical model to estimate the reconstructed 2×2 RGB full-color block-distortion, and then we derive the overdetermined system by deploying the two chroma subsampled parameters and four luma modification parameters into the block-distortion model.

A. ESTIMATING THE RECONSTRUCTED 2 × 2 RGB FULL-COLOR BLOCK-DISTORTION MODEL

Before presenting the proposed reconstructed 2×2 RGB full-color block-distortion model at the server side, we first describe how to estimate the four (U, V)-pairs of the current 2×2 UV block B^{UV} by referring to the eight neighboring subsampled (U, V)-pairs of B^{UV}.

Because our CSLM method combines chroma subsampling and luma modification together, and is performed on I^{YUV} in a row-major order, for the current 2×2 UV block B^{UV}, the eight neighboring subsampled (U, V)-pairs of B^{UV} consist of four already known subsampled (U, V)-pairs obtained by our CSLM method and four future subsampled (U, V)-pairs which can be obtained by performing any traditional chroma subsampling scheme, e.g. 4:2:0(A), on the four future reference 2×2 UV blocks. As depicted in Fig. 3, the eight reference subsampled (U, V)-pairs are denoted by (U_{tl}, V_{tl}), (U_t, V_t), (U_{tr}, V_{tr}), (U_l, V_l), (U_r, V_r), (U_{bl}, V_{bl}), (U_b, V_b), and (U_{br}, V_{br}).

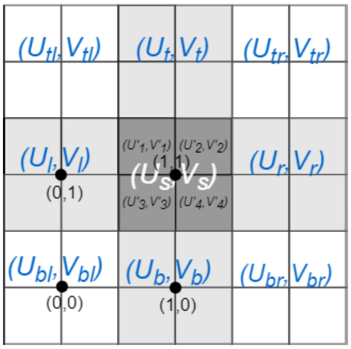


FIGURE 3. Notations used in the estimation of the 2 × 2 chroma block $B^{rec,UV}$ at the server side.

It is known that the subsampled (U, V)-pair of the current 2×2 block B^{UV} is denoted by the parameter-pair (U_s, V_s). Let the four estimated (U, V)-pairs of B^{UV} be denoted by (U_1', V_1'), (U_2', V_2'), (U_3', V_3'), and (U_4', V_4'), as shown in Fig. 3. Without loss of generality, we derive the estimation only for (U_3', V_3') in detail, and then provide the general estimation formula for the four estimated (U, V)-pairs of B^{UV}.

For estimating (U_3', V_3'), the four reference subsampled (U, V)-pairs are (U_l, V_l), (U_{bl}, V_{bl}), (U_b, V_b), and (U_s, V_s). As shown in Fig. 3, the subsampled (U, V)-pair (U_{bl}, V_{bl}) is defined to be located at the original coordinate (0, 0); (U_l, V_l), (U_b, V_b), and (U_s, V_s) are thus located at (0, 1), (1, 0), and (1, 1), respectively. According to the bilinear interpolation, (U_3', V_3') is estimated by

$$\begin{aligned}
&(U_3', V_3')\\
&= (\frac{3}{4})(1-\frac{1}{4})(U_s, V_s) + (1-\frac{3}{4})(1-\frac{1}{4})(U_l, V_l)\\
&\quad +(1-\frac{3}{4})(\frac{1}{4})(U_{bl}, V_{bl}) + (\frac{3}{4})(\frac{1}{4})(U_b, V_b)\\
&= \frac{9}{16}(U_s, V_s) + \frac{3}{16}(U_l, V_l) + \frac{1}{16}(U_{bl}, V_{bl}) + \frac{3}{16}(U_b, V_b)\\
&= (\frac{9}{16}U_s + \frac{3}{16}U_l + \frac{1}{16}U_{bl} + \frac{3}{16}U_b,\ \frac{9}{16}Vs + \frac{3}{16}V_l + \frac{1}{16}V_{bl}\\
&\quad +\frac{3}{16}V_b)\\
&= (\frac{9}{16}U_s + \delta(U_3),\ \frac{9}{16}V_s + \delta(V_3)) \qquad (3)
\end{aligned}$$

In the same estimation way as for (U'_3, V'_3), the estimation of (U'_1, V'_1), (U'_2, V'_2), and (U'_4, V'_4) can be followed. In general, we have

$$(U'_i, V'_i) = (\frac{9}{16}U_s + \delta(U'_i), \frac{9}{16}V_s + \delta(V'_i)) \tag{4}$$

with

$$\begin{aligned}
\delta(U'_1) &= \frac{3}{16}U_l + \frac{1}{16}U_{tl} + \frac{3}{16}U_t \\
\delta(V'_1) &= \frac{3}{16}V_l + \frac{1}{16}V_{tl} + \frac{3}{16}V_t \\
\delta(U'_2) &= \frac{3}{16}U_r + \frac{1}{16}U_{tr} + \frac{3}{16}U_t \\
\delta(V'_2) &= \frac{3}{16}V_r + \frac{1}{16}V_{tr} + \frac{3}{16}V_t \\
\delta(U'_3) &= \frac{3}{16}U_l + \frac{1}{16}U_{bl} + \frac{3}{16}U_b \\
\delta(V'_3) &= \frac{3}{16}V_l + \frac{1}{16}V_{bl} + \frac{3}{16}V_b \\
\delta(U'_4) &= \frac{3}{16}U_r + \frac{1}{16}U_{br} + \frac{3}{16}U_b \\
\delta(V'_4) &= \frac{3}{16}V_r + \frac{1}{16}V_{br} + \frac{3}{16}V_b
\end{aligned} \tag{5}$$

B. DERIVING THE OVERDETERMINED SYSTEM

Considering the ith entry of the current 2 × 2 YUV block B^{YUV}, $1 \leq i \leq 4$, its YUV triple-value is denoted by (Y_i, U_i, V_i). By Eqs. (3)-(5), the ith entry of the estimated 2×2 UV block of B^{UV} is denoted by (U'_i, V'_i); the ith modified luma value is denoted by Y'_i. Replacing the three parameters Y_i, U_i, and V_i at the right ride of Eq. (2) with Y'_i, U'_i, and V'_i, $1 \leq i \leq 4$, respectively, it yields the following three equalities:

$$\begin{aligned}
R'_i &= 1.164 \times (Y'_i - 16) + 1.596 \times (V'_i - 128) \\
G'_i &= 1.164 \times (Y'_i - 16) - 0.391 \times (U'_i - 128) \\
&\quad -0.813 \times (V'_i - 128) \\
B'_i &= 1.164 \times (Y'_i - 16) + 2.018 \times (U'_i - 128)
\end{aligned} \tag{6}$$

For $1 \leq i \leq 4$, it yields the following twelve equalities:

$$\begin{aligned}
R'_1 &= 1.164 \times (Y'_1 - 16) + 1.596 \times (V'_1 - 128) \\
G'_1 &= 1.164 \times (Y'_1 - 16) - 0.391 \times (U'_1 - 128) \\
&\quad -0.813 \times (V'_1 - 128) \\
B'_1 &= 1.164 \times (Y'_1 - 16) + 2.018 \times (U'_1 - 128) \\
R'_2 &= 1.164 \times (Y'_2 - 16) + 1.596 \times (V'_2 - 128) \\
G'_2 &= 1.164 \times (Y'_2 - 16) - 0.391 \times (U'_2 - 128) \\
&\quad -0.813 \times (V'_2 - 128) \\
B'_2 &= 1.164 \times (Y'_2 - 16) + 2.018 \times (U'_2 - 128) \\
R'_3 &= 1.164 \times (Y'_3 - 16) + 1.596 \times (V'_3 - 128) \\
G'_3 &= 1.164 \times (Y'_3 - 16) - 0.391 \times (U'_3 - 128) \\
&\quad -0.813 \times (V'_3 - 128) \\
B'_3 &= 1.164 \times (Y'_3 - 16) + 2.018 \times (U'_3 - 128) \\
R'_4 &= 1.164 \times (Y'_4 - 16) + 1.596 \times (V'_4 - 128) \\
G'_4 &= 1.164 \times (Y'_4 - 16) - 0.391 \times (U'_4 - 128) \\
&\quad -0.813 \times (V'_4 - 128) \\
B'_4 &= 1.164 \times (Y'_4 - 16) + 2.018 \times (U'_4 - 128)
\end{aligned} \tag{7}$$

From Eq. (7), the reconstructed 2 × 2 RGB full-color block-distortion model is naturally denoted by $BD(Y'_1, Y'_2, Y'_3, Y'_4, U'_1, U'_2, U'_3, U'_4, V'_1, V'_2, V'_3, V'_4)$. Because by Eq. (3)-(5), we know that U'_1, U'_2, U'_3, and U'_4 are the functions with the parameter U_s; V'_1, V'_2, V'_3, and V'_4 are the functions with the parameter V_s, so the reconstructed 2 × 2 RGB full-color block-distortion is defined by

$$\begin{aligned}
&BD(Y'_1, Y'_2, Y'_3, Y'_4, U_s, V_s) \\
&\quad = \sum_{i=1}^{4}[(R_i - R'_i)^2 + (G_i - G'_i)^2 + (B_i - B'_i)^2]
\end{aligned} \tag{8}$$

Ideally, the solution of $\boldsymbol{X}$ $(= (Y'_1, Y'_2, Y'_3, Y'_4, U_s, V_s))$ aims to zeroize the reconstructed 2 × 2 RGB full-color block-distortion in Eq. (8). At the right side of Eq. (7), for $1 \leq i \leq 4$, we replace U'_i and V'_i with $(\frac{9}{16}U_s + \delta(U'_i))$ and $(\frac{9}{16}V_s + \delta(V'_i))$ (see Eq. (4)), respectively. Therefore, ideally, the solution of $\boldsymbol{X}$ aims to satisfy the following overdetermined system:

$$\begin{aligned}
R_1 &= 1.164 \times (Y'_1 - 16) + 1.596 \times (\frac{9}{16}V_s + \delta(V'_1) - 128) \\
G_1 &= 1.164 \times (Y'_1 - 16) - 0.391 \times (\frac{9}{16}U_s + \delta(U'_1) - 128) \\
&\quad -0.813 \times (\frac{9}{16}V_s + \delta(V'_1) - 128) \\
B_1 &= 1.164 \times (Y'_1 - 16) + 2.018 \times (\frac{9}{16}U_s + \delta(U'_1) - 128) \\
R_2 &= 1.164 \times (Y'_2 - 16) + 1.596 \times (\frac{9}{16}V_s + \delta(V'_2) - 128) \\
G_2 &= 1.164 \times (Y'_2 - 16) - 0.391 \times (\frac{9}{16}U_s + \delta(U'_2) - 128) \\
&\quad -0.813 \times (\frac{9}{16}V_s + \delta(V'_2) - 128) \\
B_2 &= 1.164 \times (Y'_2 - 16) + 2.018 \times (\frac{9}{16}U_s + \delta(U'_2) - 128) \\
R_3 &= 1.164 \times (Y'_3 - 16) + 1.596 \times (\frac{9}{16}V_s + \delta(V'_3) - 128) \\
G_3 &= 1.164 \times (Y'_3 - 16) - 0.391 \times (\frac{9}{16}U_s + \delta(U'_3) - 128) \\
&\quad -0.813 \times (\frac{9}{16}V_s + \delta(V'_3) - 128) \\
B_3 &= 1.164 \times (Y'_3 - 16) + 2.018 \times (\frac{9}{16}U_s + \delta(U'_3) - 128) \\
R_4 &= 1.164 \times (Y'_4 - 16) + 1.596 \times (\frac{9}{16}V_s + \delta(V'_4) - 128) \\
G_4 &= 1.164 \times (Y'_4 - 16) - 0.391 \times (\frac{9}{16}U_s + \delta(U'_4) - 128) \\
&\quad -0.813 \times (\frac{9}{16}V_s + \delta(V'_4) - 128) \\
B_4 &= 1.164 \times (Y'_4 - 16) + 2.018 \times (\frac{9}{16}U_s + \delta(U'_4) - 128)
\end{aligned} \tag{9}$$

In the above overdetermined system, there are six parameters, namely Y'_1, Y'_2, Y'_3, Y'_4, U_s, and V_s, to be solved. Because it is intractable to solve $\boldsymbol{X}$ such that all equalities in Eq. (9)

are totally satisfied, in the next subsection, a matrix pseudoinverse technique is proposed to solve $\boldsymbol{X}$ such that the sum of the square errors between the left side and right side of Eq. (9) could be minimized, obtaining the best solution of $\boldsymbol{X}$.

III. DETERMINING THE SUBSAMPLED CHROMA PAIR AND MODIFIED LUMA VALUES

In this section, we first transform the overdetermined system in Eq. (9) to a matrix form, and then we show that it can be solved by the matrix pseudoinverse technique, determining the solution of $\boldsymbol{X}$ for each 2×2 YUV block B^{YUV}. Finally, the whole procedure to realize our CSLM method is provided.

A. SOLVING THE OVERDETERMINED SYSTEM WITH THE MATRIX PSEUDOINVERSE TECHNIQUE

Moving the constant terms at the right side of Eq. (9) to the left side, in matrix form, Eq. (9) is expressed as Eq. (10), as shown at the bottom of the page. Let

$$b = \begin{bmatrix} R_1 - 1.159\delta(V_1') + 222.912 \\ G_1 + 0.391\delta(U_1') + 0.813\delta(V_1') - 135.488 \\ B_1 - 2.018\delta(U_1') + 276.928 \\ R_2 - 1.159\delta(V_2') + 222.912 \\ G_2 + 0.391\delta(U_2') + 0.813\delta(V_2') - 135.488 \\ B_2 - 2.018\delta(U_2') + 276.928 \\ R_3 - 1.159\delta(V_3') + 222.912 \\ G_3 + 0.391\delta(U_3') + 0.813\delta(V_3') - 135.488 \\ B_3 - 2.018\delta(U_3') + 276.928 \\ R_4 - 1.159\delta(V_4') + 222.912 \\ G_4 + 0.391\delta(U_4') + 0.813\delta(V_4') - 135.488 \\ B_4 - 2.018\delta(U_4') + 276.928 \end{bmatrix}$$

which is often called the response vector [2]. Therefore, Eq. (10) is simplified to Eq. (11), where the matrix T is often called the design matrix [2].

$$b = TX \tag{11}$$

Based on the geometry relation, $(b - TX)$ is perpendicular to the range of T, namely $R(T)$, which is spanned by the column vectors of T. Therefore, it yields $T^t(b - TX) = 0$, and then the normal equation $T^t b = T^t TX$ holds. Because the design matrix T is full rank and the rank is 6, the pseudoinverse $(T^tT)^{-1}T^t$ exists [2]. Therefore, with our CSLM method, the solution of $\boldsymbol{X}$ for Eq. (11) can be obtained by

$$X = (T^tT)^{-1}(T^t)b \tag{12}$$

where the design matrix T and the response vector b have been defined in Eqs. (10)-(11).

In fact, the pseudoinverse $(T^tT)^{-1}T^t$ can be computed in advance as a fixed 6×12 matrix which is shown in Eq. (13), as shown at the bottom of the page, achieving the execution-time reduction effect.

B. THE WHOLE PROCEDURE TO REALIZE OUR CSLM METHOD

Consequently, using our CSLM method, for the current 2×2 YUV block B^{YUV}, by Eqs. (11)-(12), the four modified luma values, Y_1', Y_2', Y_3', and Y_4', and the two subsampled chroma values, V_s and U_s, can be determined quickly such that the reconstructed 2×2 RGB full-color block-distortion could be minimized in the least square errors sense. The whole procedure to realize our CSLM method is listed below.

IV. EXPERIMENTAL RESULTS

Based on the Kodak dataset with 24 images [4] and the IMAX dataset with 18 images [3], all the considered experiments are carried out on the VTM-8.0 platform. To compare the quality performance among the considered chroma subsampling methods, the three quality metrics used are CPSNR, SSIM, and FSIMc. Besides the three quality merits of our CSLM method, the quality-bitrate tradeoff merit of our method is also demonstrated for different QP values. In addition, the luma mean-preserving effect of CSLM is reported. The execution time comparison of the considered methods

$$\begin{bmatrix} R_1 - 1.159\delta(V_1') + 222.912 \\ G_1 + 0.391\delta(U_1') + 0.813\delta(V_1') - 135.488 \\ B_1 - 2.018\delta(U_1') + 276.928 \\ R_2 - 1.159\delta(V_2') + 222.912 \\ G_2 + 0.391\delta(U_2') + 0.813\delta(V_2') - 135.488 \\ B_2 - 2.018\delta(U_2') + 276.928 \\ R_3 - 1.159\delta(V_3') + 222.912 \\ G_3 + 0.391\delta(U_3') + 0.813\delta(V_3') - 135.488 \\ B_3 - 2.018\delta(U_3') + 276.928 \\ R_4 - 1.159\delta(V_4') + 222.912 \\ G_4 + 0.391\delta(U_4') + 0.813\delta(V_4') - 135.488 \\ B_4 - 2.018\delta(U_4') + 276.928 \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 0 & 0 & 0 & 0.8977 \\ 1.164 & 0 & 0 & 0 & -0.220 & -0.4573 \\ 1.164 & 0 & 0 & 0 & 1.135 & 0 \\ 0 & 1.164 & 0 & 0 & 0 & 0.8977 \\ 0 & 1.164 & 0 & 0 & -0.220 & -0.4573 \\ 0 & 1.164 & 0 & 0 & 1.135 & 0 \\ 0 & 0 & 1.164 & 0 & 0 & 0.8977 \\ 0 & 0 & 1.164 & 0 & -0.220 & -0.4573 \\ 0 & 0 & 1.164 & 0 & 1.135 & 0 \\ 0 & 0 & 0 & 1.164 & 0 & 0.8977 \\ 0 & 0 & 0 & 1.164 & -0.220 & -0.4573 \\ 0 & 0 & 0 & 1.164 & 1.135 & 0 \end{bmatrix} \begin{bmatrix} Y_1' \\ Y_2' \\ Y_3' \\ Y_4' \\ U_s \\ V_s \end{bmatrix} \tag{10}$$

$$(T^tT)^{-1}T^t = \begin{bmatrix} 0.2790 & 0.3409 & 0.2392 & -0.0074 & 0.0545 & -0.0472 & -0.0074 & 0.0545 & -0.0472 & -0.0074 & 0.0545 & -0.0472 \\ -0.0074 & 0.0545 & -0.0472 & 0.2790 & 0.3409 & 0.2392 & -0.0074 & 0.0545 & -0.0472 & -0.0074 & 0.0545 & -0.0472 \\ -0.0074 & 0.0545 & -0.0472 & -0.0074 & 0.0545 & -0.0472 & 0.2790 & 0.3409 & 0.2392 & -0.0074 & 0.0545 & -0.0472 \\ -0.0074 & 0.0545 & -0.0472 & -0.0074 & 0.0545 & -0.0472 & -0.0074 & 0.0545 & -0.0472 & 0.2790 & 0.3409 & 0.2392 \\ -0.0371 & -0.0727 & 0.1098 & -0.0371 & -0.0727 & 0.1098 & -0.0371 & -0.0727 & 0.1098 & -0.0371 & -0.0727 & 0.1098 \\ 0.1098 & -0.0920 & -0.0178 & 0.1098 & -0.0920 & -0.0178 & 0.1098 & -0.0920 & -0.0178 & 0.1098 & -0.0920 & -0.0178 \end{bmatrix} \tag{13}$$

TABLE 1. CPSNR, SSIM, and FSIMc MERITS (QP = 0) of our CSLM method relative to the considered eighteen combinations.

		CPSNR	SSIM	FSIMc	Time (s)
COPY	4:2:0(A)	40.6484	0.9761	0.99984	0.00671
	4:2:0(L)	39.2445	0.9703	0.99957	0.00637
	4:2:0(R)	39.1950	0.9701	0.99957	0.00532
	4:2:0(MPEG-B)	38.4808	0.9664	0.99938	0.00731
	4:2:0(BRIGHT)	37.6030	0.9639	0.99910	0.00325
	4:2:0(BRIGHT_MEAN)	40.3534	0.9762	0.99973	0.00335
BILI	4:2:0(A)	41.8774	0.9793	0.99973	╲
	4:2:0(L)	40.9531	0.9767	0.99962	╲
	4:2:0(R)	40.6006	0.9762	0.99960	╲
	4:2:0(MPEG-B)	42.6818	0.9803	0.99972	╲
	4:2:0(BRIGHT)	39.3171	0.9718	0.99931	╲
	4:2:0(BRIGHT_MEAN)	41.6642	0.9793	0.99968	╲
BICU	4:2:0(A)	43.1302	0.9834	0.99984	╲
	4:2:0(L)	42.1107	0.9784	0.99965	╲
	4:2:0(R)	41.0471	0.9776	0.99964	╲
	4:2:0(MPEG-B)	42.9225	0.9814	0.99968	╲
	4:2:0(BRIGHT)	38.9529	0.9700	0.99928	╲
	4:2:0(BRIGHT_MEAN)	42.5837	0.9826	0.99976	╲
BILI	**Proposed CSLM**	**44.1290**	**0.9863**	**0.99985**	0.06991

```
Procedure: CSLM
Input: 2×2 RGB block $B^{RGB}$ and the converted 2×2 YUV
       block $B^{YUV}$.
Output: the solution of $X = (Y'_1, Y'_2, Y'_3, Y'_4, U_s, V_s)$.
Step 1: By Eq. (2.1), estimate the four reconstructed $(U, V)$-pairs of $B^{UV}$.
Step 2: By Eqs. (3.2)-(3.3), obtain the response vector $b$ and the design matrix $T$.
Step 3: By Eq. (3.5), obtain the pseudoinverse matrix $S = (T^tT)^{-1}T^t$.
Step 4: By Eq. (3.4), calculate $X = Sb$ to determine the four modified luma values and the subsampled $(U, V)$-pair, $(U_s, V_s)$.
Return $X$.
```

is also made. In addition, based on the video sequence "Boat", the quality-bitrate tradeoff merit of our method is also reported.

All the considered methods are implemented on a computer with an Intel Core i7-4790 CPU 3.6 GHz and 24 GB RAM. The operating system is the Microsoft Windows 10 64-bit operating system. The program development environment is Visual C++ 2017.

A. CPSNR, SSIM, AND FSIMc MERITS

The quality metric CPSNR is defined by

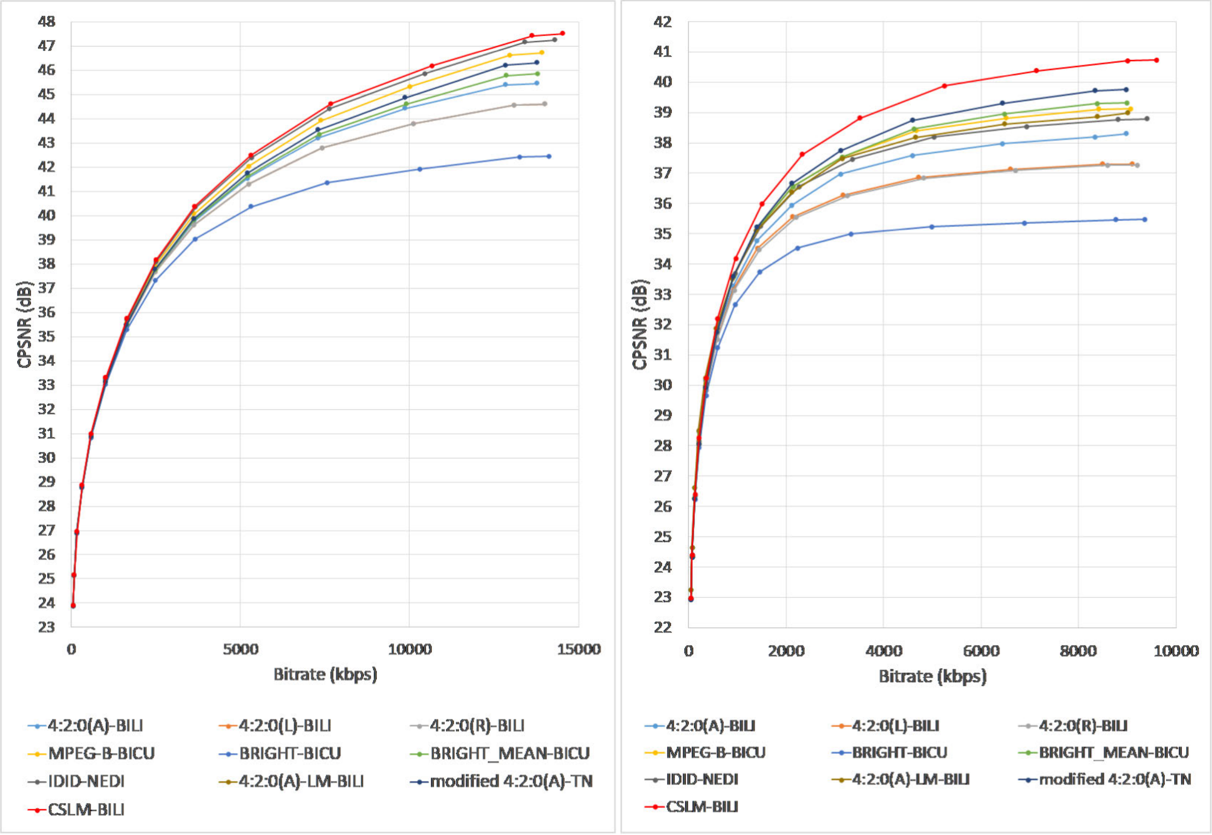
$$\mathrm{CPSNR} = \frac{1}{N}\sum_{n=1}^{N} 10\log_{10}\frac{255^2}{MSE} \tag{14}$$

in which N denotes the number of test images used in the dataset; $MSE = \frac{1}{3WH}\sum_{p\in P}\sum_{c\in\{R,G,B\}}[I_{n,c}^{RGB}(p) - I_{n,c}^{rec,RGB}(p)]^2$ where "W×H" denotes the size of the test image. $I_{n,c}^{RGB}(p)$ and $I_{n,c}^{rec,RGB}(p)$ denote the c-color values of the pixels at position p in the nth input RGB full-color image and the reconstructed one, respectively.

SSIM [9] 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. To measure the FSIMc metric value [11], we first utilize the contrast invariant feature "phase congruency (PC)" and the minor feature "gradient magnitude" to obtain the local quality map. Further, we utilize PC as a weighting function to calculate the quality score as the FSIMc metric value. Note that the available code for FSIMc can be accessed from [11]. To justify the CPSNR, SSIM, and FSIMc merits of our CSLM method, we set QP to zero and the related results are computed by passing the compression and decompression process.

TABLE 2. CPSNR, SSIM, and FSIMc MERITS (QP = 0) of our CSLM-BILI combination relative to the three state-of-the-art combinations.

	CPSNR	SSIM	FSIMc	Time (s)
IDID-NEDI [12]	43.0151	0.9819	0.99974	9.06112
4:2:0(A)-LM-BILI [1]	42.6105	0.9816	0.99897	0.13663
modified 4:2:0(A)-TN [6]	43.0305	0.9828	0.99985	0.01119
Proposed CSLM-BILI	**44.1290**	**0.9863**	**0.99985**	0.06991



(a)

(b)

FIGURE 4. The quality-bitrate tradeoff merit of our CSLM method. (a) For the Kodak dataset on VTM-8.0. (b) For the IMAX dataset on VTM-8.0.

For the reconstructed RGB full-color images, Table 1 tabulates the quality comparison in terms of CPSNR, SSIM, and FSIMc. Here, the three chroma upsampling processes at the client side, namely COPY, BILI, and BICU, are included. From Table 1, we observe that our CSLM method under the BILI chroma upsampling process has the highest CPSNR, SSIM, and FSIMc in boldface when compared with the eighteen combinations for the six considered traditional chroma subsampling methods and the three considered chroma upsampling processes.

In terms of CPSNR, SSIM, and FSIMc, Table 2 tabulates the quality comparison among the proposed CSLM-BILI combination and the other three state-of-the-art combinations, IDID-NEDI [12], 4:2:0(A)-LM-BILI [1], and modified 4:2:0(A)-TN [6]. From Table 2, we observe that our combination has the highest CPSNR, SSIM, and FSIMc in boldface when compared with the three state-of-the-art combinations.

B. QUALITY-BITRATE TRADEOFF MERIT, LUMA MEAN PRESERVATION EFFECT, AND EXECUTION TIME COMPARISON

In this subsection, we first present the quality-bitrate tradeoff merit of our CSLM method, and then present the luma mean preservation effect. Finally, the execution time comparison is reported.

1) QUALITY-BITRATE TRADEOFF MERIT

When setting QP = 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, and 51, the quality-bitrate tradeoff of each considered method is depicted by the RD curve for the reconstructed RGB full-color images. The bitrate of one compressed dataset is defined by

$$\text{bitrate} = \frac{B}{N} \tag{15}$$

where B denotes the total number of bits required in compressing N test images in that dataset. On VVC platform, the RD curves corresponding to the Kodak dataset and the IMAX dataset are depicted in Fig. 4(a) and Fig. 4(b), respectively, in which the X-axis denotes the average bitrate required and the Y-axis denotes the average CPSNR value of the reconstructed RGB full-color images. Fig. 4 indicates that under the same bitrate, our CSLM method has the highest CPSNR among the nine considered methods. Based on the testing video sequence ''Boat'', Fig. 5 indicates that under the same bitrate, our CSLM method still has the highest CPSNR.

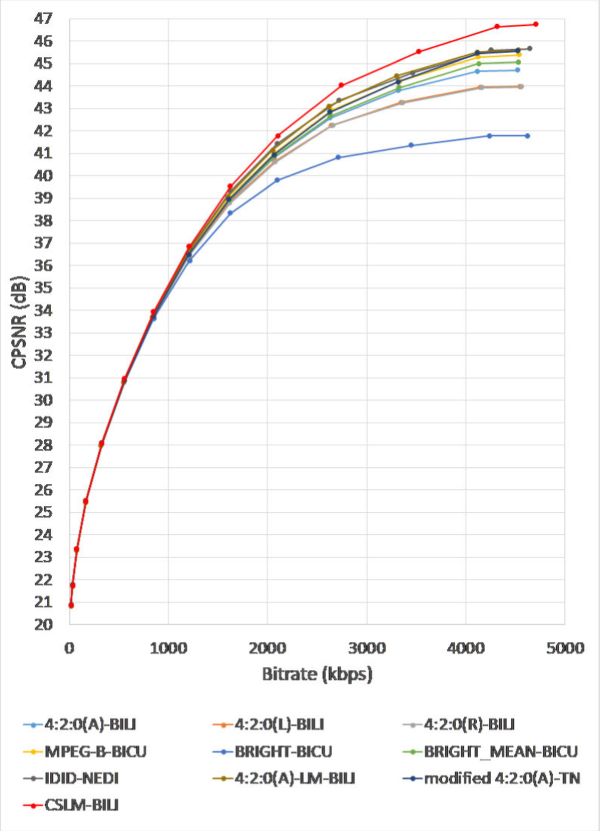


FIGURE 5. **The quality-bitrate tradeoff merit of our CSLM method for the video sequence ''Boat'' on VTM-8.0.**

2) LUMA MEAN PRESERVATION EFFECT

The luma mean-loss of our CSLM method is measured by

$$\frac{1}{N}\sum_{k=1}^{N} |\bar{I}_k^Y - \bar{I}_k^{rec,Y}| \tag{16}$$

where N denotes the number of test images in the dataset; $\bar{I}_k^Y$ and $\bar{I}_k^{rec,Y}$ denote the luma mean values of original kth luma image and reconstructed kth analogue, respectively.

Although most of the comparative methods do not consider modifying the luma values in chroma subsampling, their luma mean-loss values are the same, empirically 0.0028 dB, due to the floating point-to-integer conversion error before compression; it indicates a nearly perfect luma mean-preservation effect. For the Kodak dataset, the average luma mean-loss value by our CSLM is 0.0157 dB and for the IMAX dataset, the average luma mean-loss by our CSLM is 0.0151 dB. On average, the luma mean-loss value is 0.0154 dB, indicating a good luma mean-preservation effect of our CSLM method.

3) EXECUTION TIME COMPARISON

Table 1 tabulates the execution time (in seconds) comparison among the six traditional chroma subsampling methods and our CSLM method. For simplicity, the execution time of each traditional chroma subsampling method is listed once in Table 1, and from Table 1, although our method takes more time than the traditional methods, our method has clearly better quality. In Table 2, we observe that besides the quality merit, our CSLM method is also much faster than the two state-of-the-art methods, IDID [12], and 4:2:0(A)-LM [1]; our CSLM method has worse execution time performance but has better quality performance relative to modified 4:2:0(A)-TN [6].

V. CONCLUSION

We have presented the proposed CSLM chroma subsampling method for RGB full-color images. First, a newly reconstructed 2 × 2 RGB full-color block-distortion model is presented. Then, an overdetermined system is derived to deploy the two chroma subsampled parameters and four luma modification parameters into the distortion model. Furthermore, we show that the derived overdetermined system can be solved by the matrix pseudoinverse technique, determining the solution of the required chroma subsampled pair and four modified luma values for each 2 × 2 YUV block. Finally, a whole procedure is provided to realize our CSLM method. Based on the Kodak and IMAX datasets, the comprehensive experimental results have justified the quality and quality-bitrate tradeoff merits of our CSLM method relative to six traditional chroma subsampling methods and three state-of-the-art methods. In addition, based on the video sequence ''Boat'', the quality-bitrate tradeoff merit of our method has been justified.

How to extend the delivered process to estimate the four chroma pairs of each 2 × 2 chroma block, as described in Subsection II.A, using the other nonlinear upsampling processes, such as the bicubic interpolation-based estimation process, is our first future work. In our second future work, we hope to combine CSLM and the discrete cosine transform based (DCT-based) downsampling approach, and then compare it with the current work [13] in which the downsampling process is only done in the DCT domain, while it does nothing on the chroma subsampling and luma modification prior to the compression.

ACKNOWLEDGMENT

The authors appreciate the help of Associate Editor Prof. Z. Pan and the two anonymous reviewers for their valuable comments to improve the manuscript. They also appreciate the proofreading help of Ms. C. Harrington to improve the manuscript.

REFERENCES

[1] K.-L. Chung, T.-C. Hsu, and C.-C. Huang, "Joint chroma subsampling and distortion-minimization-based luma modification for RGB color images with application," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4626–4638, Oct. 2017.

[2] B. N. Datta, *Numerical Linear Algebra and Applications*, 1st ed. Pacific Grove, CA, USA: Brooks/Cole, 1995, pp. 315–324.

[3] *IMAX True Color Image Collection* Accessed: Aug. 2014. [Online]. Available: http://www.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm

[4] *Kodak True Color Image Collection*. Accessed: Aug. 2014. [Online]. Available: http://r0k.us/graphics/kodak

[5] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[6] 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.*, early access, Sep. 2019, doi: 10.1109/TCSVT.2019.2939280.

[7] *Spatial Scalability Filters*, document ISO/IEC JTC1/SC29/WG11 ITU-T SG 16 Q.6, Jul. 2005.

[8] *Versatile Video Coding (VVC)*. Accessed: Feb. 2020. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

[9] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[10] Y. C. Yu, J. W. Jhang, X. Wei, H. W. Tseng, Y. Wen, and Z. Liu, "Chroma upsampling for YCbCr 420 videos," in *Proc. IEEE Int. Conf. Consum. Electron.*, Jun. 2017, pp. 163–164.

[11] 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.

[12] 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.

[13] S. Zhu, C. Cui, R. Xiong, Y. Guo, and B. Zeng, "Efficient chroma subsampling 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 National Taiwan University, Taipei, in 1982, 1984, and 1990, respectively. He has been the Chair Professor of the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, since 2009. His research interests include deep learning, image processing, and video compression. He was a recipient of the Distinguished Research Award, from 2004 to 2007 and from 2019 to 2022, and a Distinguished Research Project Award, from 2009 to 2012, from the Ministry of Science and Technology of Taiwan. In 2020, he received the K. T. Li Fellow Award from the Institute of Information and Computing Machinery, Taiwan. He has been an Associate Editor of *Journal of Visual Communication and Image Representation*, since 2011.

JEN-SHUN CHENG received the B.S. degree in computer science and information engineering from Fu Jen Catholic University, New Taipei City, in 2016. He is currently pursuing the M.S. degree in computer science and information engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include image processing and video compression.

HONG-BIN YANG is currently pursuing the bachelor's degree in computer science and information engineering with the National Taiwan University of Science and Technology, Taiwan. He has participated in some programming contests such as ACM ICPC. Under the supervision of Prof. K. L. Chung, he is also working on a project in image processing.