Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# A Random Forest-Based Low-Complexity Codec for Efficient Luma and Chroma Compression in Wireless Multimedia Communication

**R. Ezhilarasi**
University College of Engineering Arni
**B. Pradeep Khanth**
pradeepkhanth32212@gmail.com

University College of Engineering Arni

**Research Article**

**Keywords:** Video compression, Chroma down-sampling, Luma down-sampling, Random forest, Color space conversion, Wireless multimedia communication

**Posted Date:** November 25th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11768718/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# A Random Forest-Based Low-Complexity Codec for Efficient Luma and Chroma Compression in Wireless Multimedia Communication

R. Ezhilarasi[1] · B. Pradeep Khanth[1,*]

[1] Department of Electronics and Communication Engineering, University College of Engineering Arni, Arni, Tamil Nadu, India

* Corresponding author: B. Pradeep Khanth, pradeepkhanth32212@gmail.com; R. Ezhilarasi, ezhilr2k88@gmail.com

## Abstract

Wireless multimedia services operate under tight bandwidth and energy budgets, so every sample removed before encoding lowers the transmission cost. Conventional chroma down-sampling reduces the chroma resolution by spatial averaging or filtering within a single frame and does not exploit the temporal correlation between successive frames. This paper presents a low-complexity pre-processing codec framework with four modules: block-based lossless compression using spatial redundancy removal (BLC-SRR), dual chroma down-sampling (DCDS), random forest-based efficient luma down-sampling (RF-ELDS), and color transformation kernel matrix approximation (CTKMA). BLC-SRR encodes each 4 × 4 luma block with a representative value and a binary mask. DCDS averages the chroma planes of two successive frames before conventional sub-sampling. RF-ELDS uses a random forest regressor to predict a macroblock-adaptive threshold for temporal luma down-sampling. CTKMA replaces the multiplications of the RGB-to-YCbCr conversion with shift-and-add operations. In MATLAB simulations on 320 × 240 test sequences, BLC-SRR removed 18–20% of the luma samples without loss, DCDS raised the chroma sample reduction from 75% to 87.5% at a peak signal-to-noise ratio (PSNR) cost of about 0.4–2.0 dB, and RF-ELDS removed about 20% of the luma samples at a PSNR cost of about 0.6 dB. CTKMA maintained a PSNR above 49 dB and, in field-programmable gate array synthesis, reached a combinational path delay of 8.243 ns against 16.677 ns for the direct fixed-point conversion. These simulation results indicate that the framework is suited to bandwidth- and power-constrained wireless video terminals.

**Keywords** Video compression · Chroma down-sampling · Luma down-sampling · Random forest · Color space conversion · Wireless multimedia communication

## 1 Introduction

Video services place heavy demands on wireless access networks. Services such as mobile video conferencing, surveillance over wireless sensor nodes, and live streaming from battery-powered cameras must deliver acceptable picture quality over channels whose bandwidth and energy are limited. In such terminals, the number of samples handed to the encoder and the arithmetic cost of the pre-processing stages determine both the transmitted bit rate and the power drawn by the device [1].

Image and video coding standards compress color data in the luma–chroma (YCbCr) domain, in which each RGB pixel is transformed into one luma component (Y) and two chroma components (Cb and Cr) [2, 3]. Because the human visual system is less sensitive to chroma detail than to luma detail, the chroma planes are down-sampled before compression to reduce the bit rate, the bandwidth, and the computational load, and they are interpolated back to full resolution after decoding. The pre-processing

chain of a wireless video terminal therefore contains two operations that are executed for every pixel of every frame: the color space conversion and the chroma down-sampling.

Existing chroma down-sampling methods fall into two groups. Conventional schemes such as 4:2:0(A), 4:2:0(L), 4:2:0(R), and the MPEG-B filter derive each down-sampled chroma value from a small spatial neighborhood. More recent methods couple the down-sampling step to the interpolation used at the decoder or to the color filter array of the sensor, and they obtain the down-sampled values by optimization [4–9].

These methods share three limitations. First, they operate within a single frame; the strong correlation between the chroma planes of successive frames is not used during down-sampling. Second, the luma plane is transmitted at full resolution, although co-located luma blocks of successive frames are often nearly identical in static regions. Where luma down-sampling has been attempted, the decision rule relies on a fixed threshold that does not adapt to the content of each block [10]. Third, the standard RGB-to-YCbCr conversion requires nine multiplications per pixel, which is costly in the low-power hardware typical of wireless terminals.

The research gap addressed here is therefore the absence of a pre-processing framework that (i) exploits inter-frame correlation in both the chroma and the luma planes, (ii) adapts the luma down-sampling decision to local content, and (iii) removes the multipliers from the color conversion. Closing this gap is relevant to wireless multimedia communication because each of the three items lowers either the number of samples to be transmitted or the energy spent per pixel.

This paper proposes a codec framework with four modules. Block-based lossless compression using spatial redundancy removal (BLC-SRR) encodes repeated luma values within $4 \times 4$ blocks. Dual chroma down-sampling (DCDS) averages the chroma planes of two successive frames before conventional sub-sampling. Random forest-based efficient luma down-sampling (RF-ELDS) predicts a macroblock-adaptive threshold that decides whether co-located luma macroblocks of two frames are merged. Color transformation kernel matrix approximation (CTKMA) replaces the conversion matrix with power-of-two coefficients so that only shifts and additions remain. All results reported in this paper are obtained by software simulation and by hardware synthesis; no over-the-air transmission experiment is included.

## 1.1 Contributions

The contributions of this work are as follows:

1. A dual chroma down-sampling method that merges the chroma planes of two successive frames before 4:2:0 sub-sampling, which raises the chroma sample reduction from 75% to 87.5% relative to the 4:4:4 format and can be combined with any existing down-sampling and up-sampling pair.
2. A luma down-sampling method in which a random forest regressor predicts a threshold for each macroblock from eight spatial and temporal features, replacing the fixed threshold of the earlier efficient luma down-sampling scheme [10].
3. A lossless representation of $4 \times 4$ luma blocks based on a representative value and a 16-bit mask, together with its storage-cost expression.
4. A multiplier-free approximation of the RGB-to-YCbCr kernel that needs only two distinct shift amounts, evaluated by simulation on still images and by synthesis on a Spartan-3E device.

The remainder of the paper is organized as follows. Section 2 reviews related work, Section 3 describes the materials and methods, Section 4 reports the simulation results, Section 5 discusses them, Section 6 states the limitations, and Section 7 concludes the paper.

## 2 Related Work

### 2.1 Conventional chroma sub-sampling

The 4:2:0 family of schemes produces one (Cb, Cr) pair for each 2 × 2 block. The variant 4:2:0(A) averages the four chroma samples of the block, 4:2:0(L) and 4:2:0(R) average the left or the right column, 4:2:0(DIRECT) keeps a single sample, and 4:2:0(MPEG-B) applies a 13-tap filter. The variants 4:2:0(BRIGHT) and 4:2:0(BRIGHT_MEAN) use the luma intensity to guide the selection, but they are not considered further in this study. All of these schemes are computed inside one frame.

### 2.2 Sub-sampling for color filter array images

Most consumer cameras capture color through a Bayer color filter array (CFA) [11], and conventional 4:2:0 and 4:2:2 schemes ignore the CFA structure, which degrades the reconstructed RGB values. Two families of solutions have been studied: structure-conversion-first compression, which codes the CFA data directly [12–14], and the more common demosaicking-first compression [6, 9]. Chen et al. [6] observed that the reconstructed R and B values are dominated by the (Y, V) and (Y, U) pairs, respectively, and selected the chroma sampling positions according to the Bayer pattern position (BPP). Lin et al. [9] formulated the selection as a block-distortion minimization (BM) problem, although their reliance on COPY interpolation limited the quality gain. For the 4:2:2 format, Chung et al. [15] showed that U and V influence B and R more strongly than G, but their method does not analyze the distortion of the Bayer sub-block mathematically.

### 2.3 Interpolation-aware and optimization-based down-sampling

Interpolation-dependent image down-sampling (IDID) [4] determines the down-sampled values for a given interpolation filter. Joint chroma down-sampling and up-sampling (JCDU) [7] optimizes both steps together for screen content. Gradient-descent (GD) based sub-sampling [5] iteratively refines the chroma pair for Bayer CFA images. Luma modification-based chroma down-sampling (LMCD) [8] additionally adjusts the luma values of the block, and a cubic-convolution-based method [16] extends the approach to nonlinear interpolation for Bayer and RGBW images. These methods are normally paired with BILINEAR, BICUBIC, or edge-directed interpolation at the decoder to overcome the weakness of COPY-based recovery. In earlier work, the present authors proposed luma modification-based chroma down-sampling with a block-edge-based chroma up-sampling (BECU) step and an efficient luma down-sampling (ELDS) scheme governed by a fixed threshold [10].

### 2.4 Low-complexity color space conversion

The RGB-to-YCbCr conversion of ITU-R BT.601 [3] is commonly implemented with fixed-point multipliers [17]. Yang et al. [18] replaced the multiplications of the YCbCr-to-RGB conversion with shift-and-add operations by expanding every coefficient into a sum of powers of two, which removes the multipliers but increases the number of adders. The reversible color transform of JPEG 2000 [19] avoids multiplications by using power-of-two luma weights and simple color differences, but it does not produce BT.601-style Cb and Cr components.

### 2.5 Learning-based coding decisions and wireless video

Machine-learning classifiers have been adopted to accelerate decisions inside standard encoders. Random forest classifiers have been used to predict the coding-unit partition in H.266/VVC intra coding [20, 21], including encoders designed for power-constrained applications [22]. These approaches reduce

encoder complexity, but the number of samples presented to the encoder is unchanged. For wireless delivery, low-complexity encoding for wireless multimedia sensor networks has been studied through power–rate–distortion optimization [1], and deep joint source–channel coding maps pixel values directly to channel symbols [23]. Both lines of work act on the encoder or on the channel coder rather than on the sample reduction that precedes encoding, which is the stage addressed here.

### 2.6 Comparison and research gap

Representative studies are compared in Table 1. The table shows that the existing methods improve the quality obtained at a fixed chroma sampling ratio; none of them reduces the number of transmitted samples beyond that of the 4:2:0 format, and none uses the correlation between successive frames at the down-sampling stage.

**Table 1** Comparison of representative chroma down-sampling and color conversion studies

| Study | Target content | Principle | Limitation with respect to the present work |
|---|---|---|---|
| Zhang et al. [4] | Natural images | Down-sampling derived for a given interpolation filter (IDID) | Single-frame; sampling ratio unchanged |
| Chen et al. [6] | Bayer-pattern video | Chroma positions adjusted to the Bayer pattern (BPP) | Single-frame; CFA-specific |
| Lin et al. [9] | Mosaic video, arbitrary CFA | Block-distortion minimization (BM) | Assumes COPY interpolation |
| Wang et al. [7] | Screen content images | Joint down- and up-sampling (JCDU) | Single-frame; higher complexity |
| Chung et al. [5] | Bayer CFA images | Gradient-descent refinement (GD) | Iterative; single-frame |
| Chung et al. [8] | Bayer CFA images | Luma modification with chroma down-sampling (LMCD) | Single-frame; luma kept at full resolution |
| Chung et al. [16] | Bayer and RGBW images | Cubic-convolution-based sub-sampling | Single-frame; higher complexity |
| Ahilan et al. [10] | Video frames | LMCD with BECU up-sampling; fixed-threshold ELDS | Threshold not adapted to block content |
| Zhang et al. [20]; Si et al. [21]; Chen et al. [22] | VVC intra coding | Random forest coding-unit partition decision | Acts inside the encoder; sample count unchanged |
| Yang et al. [18] | Color conversion | Shift-and-add coefficient expansion | Many adders; long carry path |

In summary, existing studies have concentrated on spatial optimization within one frame and on the fidelity of the reconstructed image at the 4:2:0 or 4:2:2 ratio, while learning-based methods have been applied mainly to accelerate decisions inside the encoder. Limited attention has been given to inter-frame redundancy at the down-sampling stage, to content-adaptive luma down-sampling, and to the joint arithmetic cost of the pre-processing chain. The present work addresses these three points with DCDS, RF-ELDS, and CTKMA, respectively, and adds BLC-SRR as a lossless option for the luma plane.

## 3 Materials and Methods

### 3.1 Test data and simulation environment

The video experiments used three publicly available sample clips [24, 25], referred to as Videos 1–3, with frame rates of 30, 25, and 15 frames/s, respectively. Each clip was decoded into RGB frames of 320 × 240 pixels, so that every luma frame contains 76,800 samples. The rate–quality and threshold experiments in Section 4.3 used the Kodak [26] and IMAX [27] color image datasets in the same two-frame configuration. The color conversion module was additionally evaluated on four standard still test images.

All algorithms were implemented in MATLAB R2022a. The reconstructed frames were reassembled into video files with FFmpeg 6.1.1 [28], and the sequence-level quality metrics were computed with FFMetrics v1.4.8b using the vmaf_v0.6.1 model. The hardware cost of the color conversion was obtained by synthesizing Verilog descriptions with the Xilinx ISE Design Suite 14.7 for a Spartan-3E XC3S100E device. All results in Section 4 are therefore simulation or synthesis results.

### 3.2 Overview of the proposed framework

At the transmitter, each RGB frame is converted to YCbCr. The luma plane is processed either by BLC-SRR (lossless) or by RF-ELDS (lossy, temporal): BLC-SRR is selected when the luma plane must be preserved exactly, and RF-ELDS when a small luma distortion is acceptable in exchange for a larger sample reduction. The chroma planes are processed by DCDS followed by a conventional 4:2:0 scheme. At the receiver, the chroma planes are up-sampled with BECU [10], the luma plane is restored, and the frames are converted back to RGB.

The conventional conversion from RGB to YCbCr is given by Eq. (1) [2, 3]:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} O_R \\ O_G \\ O_B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{1}$$

where $O_R$, $O_G$, and $O_B$ are the original color components normalized to the range [0, 1], the 3 × 3 coefficient matrix is the color transformation kernel $\mathbf{T}$, and the constant vector is the offset $\mathbf{C}$ that sets the signal range. At the receiver, the RGB components are reconstructed by the inverse transformation in Eq. (2):

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} = \mathbf{T}^{-1} \left( \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} - \mathbf{C} \right) \tag{2}$$

where $R_R$, $R_G$, and $R_B$ denote the reconstructed components.

### 3.3 Block-based lossless compression using spatial redundancy removal

BLC-SRR operates on the luma plane, which is partitioned into non-overlapping 4 × 4 blocks. Let $B = \{b_{i,j}\}$, $i, j = 1, \dots, 4$, be one block. The block is scanned row by row into the vector of Eq. (3):

$$\mathbf{v} = (v_1, v_2, \dots, v_{16}) \tag{3}$$

The representative value $r$ of the block is its most frequent value, as defined in Eq. (4):

$$r = \underset{x}{\operatorname{argmax}} N(x), \qquad N(x) = |\{k : v_k = x\}| \tag{4}$$

where $N(x)$ is the number of occurrences of the value $x$ in $\mathbf{v}$. A 16-bit mask $\mathbf{M} = (m_1, \dots, m_{16})$ marks the positions that hold the representative value, according to Eq. (5):

$$m_k = \begin{cases} 1, & v_k = r \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The remaining values are collected in scan order in the vector of Eq. (6), and the encoded block is the triple in Eq. (7):

$$\mathbf{O} = (v_k \mid m_k = 0) \tag{6}$$

$$E(B) = (r, \mathbf{M}, \mathbf{O}) \tag{7}$$

The decoder traverses the mask and rebuilds each sample by Eq. (8):

$$v'_k = \begin{cases} r, & m_k = 1 \\ o_{t(k)}, & m_k = 0 \end{cases} \tag{8}$$

where $o_{t(k)}$ is the next unread element of $\mathbf{O}$ and $t(k)$ counts the zero-valued mask bits up to position $k$. The vector $\mathbf{v}'$ is reshaped into a 4 × 4 block, which equals the original block exactly; the method is therefore lossless.

For 8-bit samples, the storage cost of one encoded block follows directly from Eq. (7) and is given by Eq. (9):

$$S(B) = 8 + 16 + 8\big(16 - N(r)\big) \text{ bits} \tag{9}$$

Compared with the 128 bits of the raw block, a saving is obtained only when $N(r) \geq 4$. Smooth blocks therefore compress strongly, whereas highly textured blocks do not compress.

### 3.4 Dual chroma down-sampling

DCDS operates in two stages. Let $F_1$ and $F_2$ be two successive frames whose chroma planes after color conversion are $(C_{b1}, C_{r1})$ and $(C_{b2}, C_{r2})$. In the first stage, the two chroma planes are averaged sample by sample with a ceiling operation, as defined in Eqs. (10) and (11):

$$\hat{C}_b(i,j) = \left\lceil \frac{C_{b1}(i,j) + C_{b2}(i,j)}{2} \right\rceil \tag{10}$$

$$\hat{C}_r(i,j) = \left\lceil \frac{C_{r1}(i,j) + C_{r2}(i,j)}{2} \right\rceil \tag{11}$$

In the second stage, a conventional down-sampling operator DS(·), for example 4:2:0(A), is applied to the averaged planes, as expressed in Eq. (12):

$$(U_d, V_d) = \Big(\mathrm{DS}\big(\hat{C}_b\big),\ \mathrm{DS}\big(\hat{C}_r\big)\Big) \tag{12}$$

where $(U_d, V_d)$ is the single down-sampled chroma pair transmitted for both frames.

For two successive frames of $N \times N$ pixels, the 4:2:0 format transmits $2N^2$ luma samples and $4(N/2)^2 = N^2$ chroma samples, which is a reduction of 75% of the original $4N^2$ chroma samples. DCDS transmits $2(N/2)^2 = N^2/2$ chroma samples, which is a reduction of 87.5%. The transmitted chroma data are thus halved, and the total number of samples for the frame pair falls from $3N^2$ to $2.5N^2$, that is, from 50% to 41.7% of the $6N^2$ samples of the 4:4:4 format.

The workflow is illustrated in Fig. 1. At the receiver, the chroma planes are up-sampled with BECU [10] and the same reconstructed chroma planes are used for both frames.

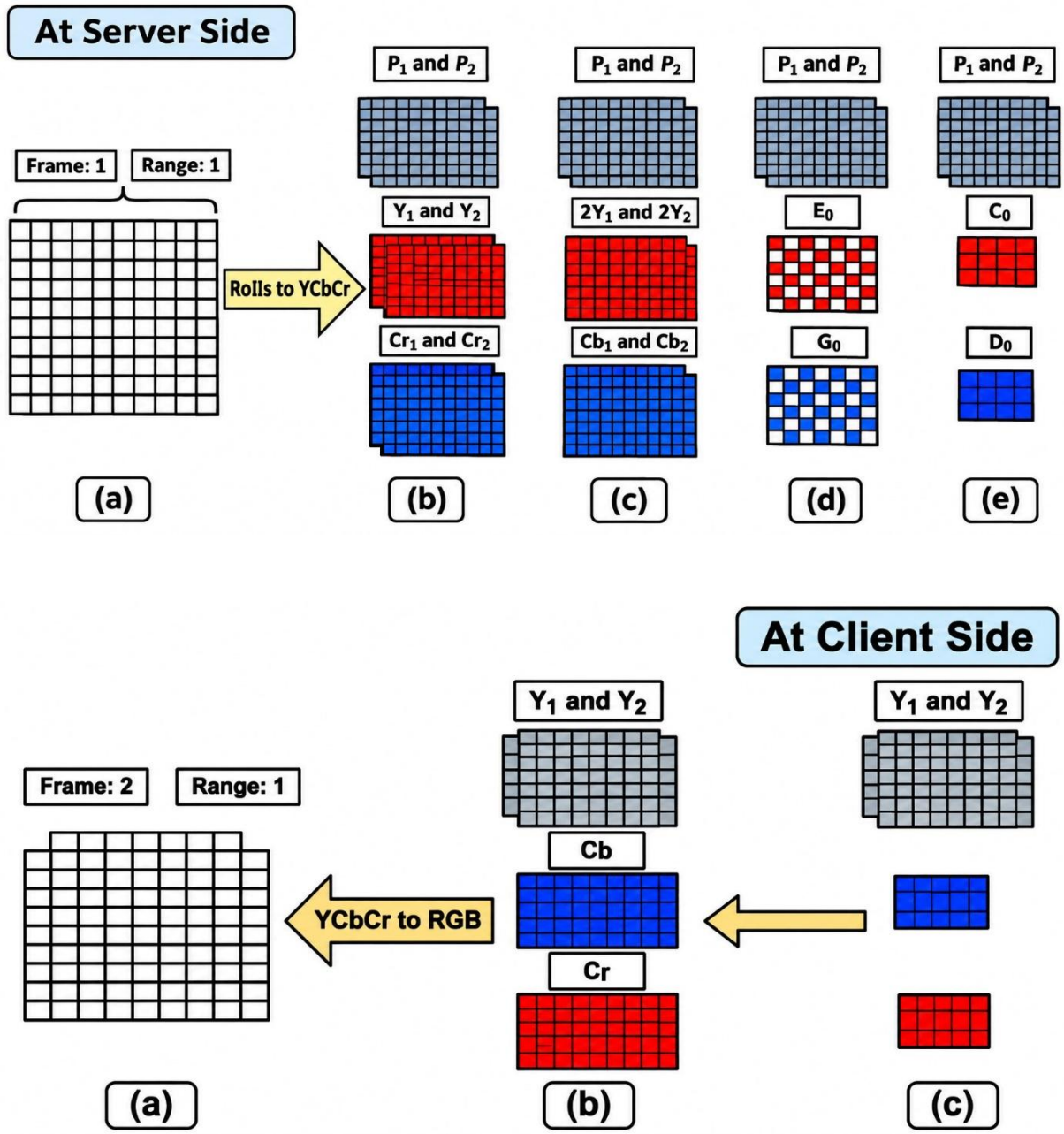



**Fig. 1** Workflow of the proposed DCDS. Transmitter (top): **a** two successive RGB frames, **b** YCbCr planes of both frames, **c** frame-averaged chroma planes, **d** chroma sub-sampling, **e** transmitted luma planes and down-sampled chroma planes. Receiver (bottom): **c** received planes, **b** chroma planes up-sampled by BECU [10], **a** reconstructed RGB frames

### 3.5 Random forest-based efficient luma down-sampling

RF-ELDS decides, for each macroblock, whether the co-located luma data of two successive frames are similar enough to be transmitted once. The decision threshold is predicted by a random forest regressor [29] from the spatial and temporal features of the macroblock.

**Macroblock formation and motion search.** The current luma frame $Y_t$ and the reference frame $Y_{t-1}$ are divided into non-overlapping macroblocks of size $B \times B$, with $B \in \{4,8\}$. The $i$-th macroblock of the current frame is denoted $MB_i$. It is compared with candidate macroblocks $MB_{\mathrm{ref},k}$ of the reference

frame at $K = 8$ neighboring search positions using the sum of absolute differences (SAD) in Eq. (13), and the minimum is taken as in Eq. (14):

$$SAD_k = \sum_{x=1}^{B} \sum_{y=1}^{B} \left| MB_i(x,y) - MB_{\mathrm{ref},k}(x,y) \right|, \qquad k = 1, \ldots, K \tag{13}$$

$$SAD_{\min} = \min_k SAD_k \tag{14}$$

A small $SAD_{\min}$ indicates high temporal similarity.

**Feature extraction.** For every macroblock, the feature vector of Eq. (15) is formed:

$$\mathbf{F}_i = [SAD_{\min},\ \mu,\ \sigma^2,\ H,\ M,\ E,\ T,\ B] \tag{15}$$

Its elements are the minimum SAD, the mean intensity $\mu$ and the variance $\sigma^2$ of Eqs. (16) and (17), the entropy $H$ of Eq. (18), the motion vector magnitude $M$ of Eq. (19), the edge density $E$ of Eq. (20), the temporal difference $T$ of Eq. (21), and the macroblock size $B$:

$$\mu = \frac{1}{B^2} \sum_{x=1}^{B} \sum_{y=1}^{B} M\, B_i(x,y) \tag{16}$$

$$\sigma^2 = \frac{1}{B^2} \sum_{x=1}^{B} \sum_{y=1}^{B} (MB_i(x,y) - \mu)^2 \tag{17}$$

$$H = -\sum_{l=0}^{255} p\,(l) \log_2 p(l) \tag{18}$$

$$M = \sqrt{MV_x^2 + MV_y^2} \tag{19}$$

$$E = \frac{N_{\mathrm{edge}}}{B^2} \tag{20}$$

$$T = \frac{1}{B^2} \sum_{x=1}^{B} \sum_{y=1}^{B} |Y_t(x,y) - Y_{t-1}(x,y)| \tag{21}$$

where $p(l)$ is the relative frequency of gray level $l$ in the macroblock, $(MV_x, MV_y)$ is the displacement of the best-matching candidate, and $N_{\mathrm{edge}}$ is the number of edge pixels returned by an edge detector.

**Training-set generation.** Let $TV_j$, $j = 1, \ldots, m$, be the candidate threshold values. For each candidate, luma down-sampling is executed and the result is scored with the weighted quality measure of Eq. (22):

$$Q_j = \alpha\, PSNR_j + \beta\, SSIM_j + \gamma\, CP_j, \qquad \alpha + \beta + \gamma = 1 \tag{22}$$

where $PSNR_j$, $SSIM_j$, and $CP_j$ are the peak signal-to-noise ratio, the structural similarity index, and the compression percentage obtained with $TV_j$, and $\alpha$, $\beta$, and $\gamma$ are weighting coefficients. The target threshold of the macroblock is given by Eq. (23), and each training sample is the pair $(\mathbf{F}_i, TV_i^*)$:

$$TV^* = \operatorname*{argmax}_{TV_j} Q_j \tag{23}$$

**Regression and decision.** The forest contains $N_T$ regression trees. The $k$-th tree outputs $f_k(\mathbf{F}_i)$, and the predicted threshold is the average in Eq. (24):

$$TV_{RF} = \frac{1}{N_T} \sum_{k=1}^{N_T} f_k\left(\mathbf{F}_i\right) \tag{24}$$

The down-sampling decision $D$ follows Eq. (25):

$$D = \begin{cases} 1, & SAD_{\min} < TV_{RF} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

When $D = 1$, the two co-located macroblocks are replaced by their average, as in Eq. (26), and the averaged macroblock is transmitted once; when $D = 0$, both original macroblocks are transmitted:

$$\hat{Y}_{MB}(x, y) = \left\lfloor \frac{Y_{1,MB}(x, y) + Y_{2,MB}(x, y)}{2} \right\rfloor \tag{26}$$

The decoder copies $\hat{Y}_{MB}$ into both frames. One flag bit per macroblock must accompany the data so that the decoder knows the decision.

The settings of the regressor are listed in Table 2, and the combined RF-ELDS and DCDS workflow is shown in Fig. 2.

**Table 2** Settings of the random forest regressor used in RF-ELDS

| **Parameter** | **Setting** |
|---|---|
| Number of trees, $N_T$ | 10 |
| Sampling | Bootstrap |
| Node-splitting criterion | Mean squared error |
| Maximum tree depth | 8 |
| Minimum leaf size | 5 samples |
| Input features | 8 (Eq. (15)) |
| Output | Threshold $TV_{RF}$ per macroblock |

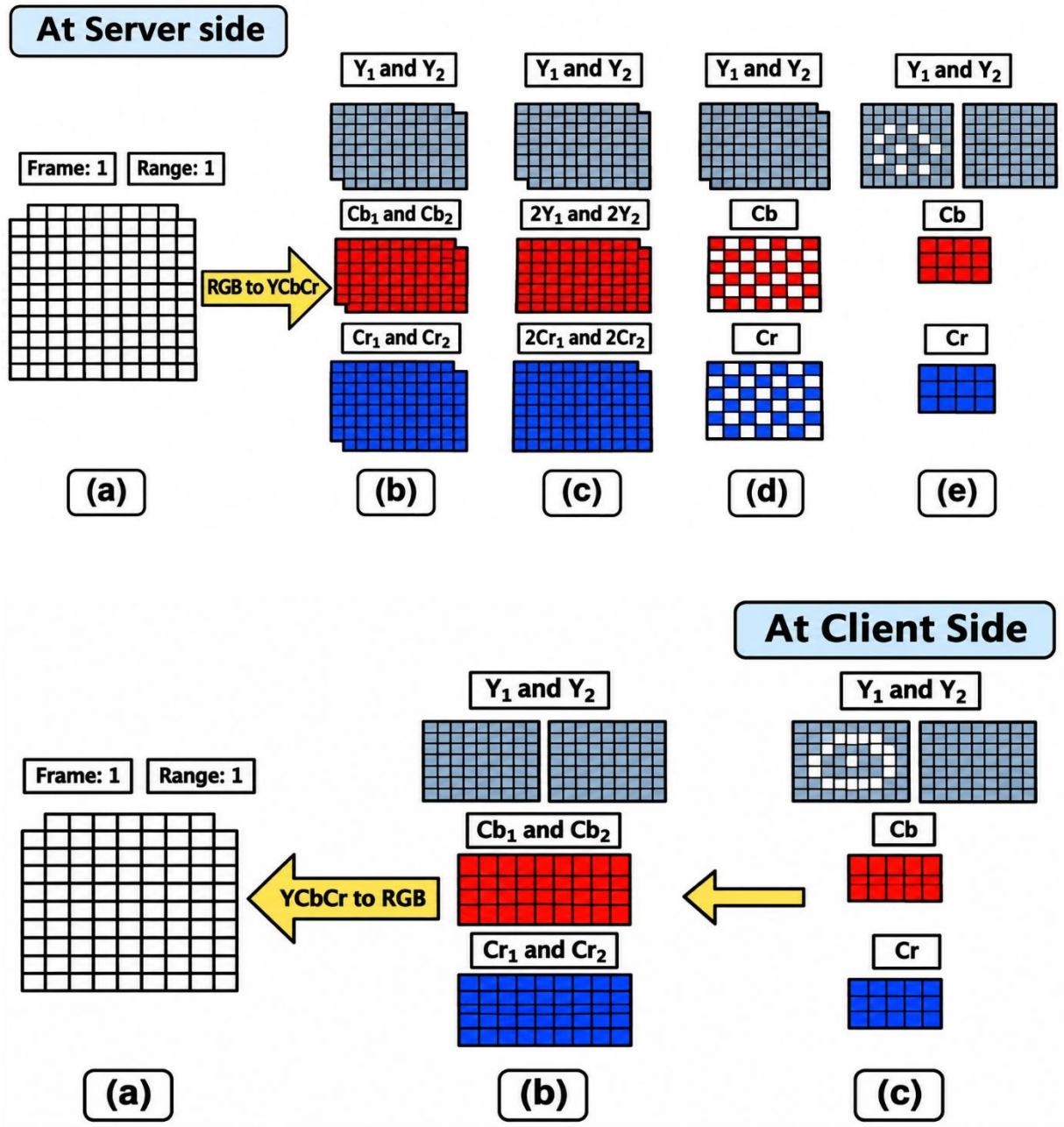



**Fig. 2** Workflow of the proposed RF-ELDS combined with DCDS. Transmitter (top): **a** two successive RGB frames, **b** YCbCr planes, **c** frame-averaged chroma planes, **d** chroma sub-sampling, **e** luma planes after RF-ELDS and down-sampled chroma planes. Receiver (bottom): **c** received planes, **b** luma restored by copying and chroma up-sampled by BECU [10], **a** reconstructed RGB frames

### 3.6 Color transformation kernel matrix approximation

CTKMA removes the multiplications of Eq. (1) in three steps. First, the kernel is written in 8-bit fixed-point form with integer coefficients, as in Eq. (27):

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \frac{1}{2^8}\begin{bmatrix} 65 & 128 & 25 \\ -38 & -74 & 112 \\ 112 & -94 & -18 \end{bmatrix}\begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{27}$$

where $R$, $G$, and $B$ are 8-bit values. Second, each coefficient is replaced by the nearest power of two, which gives the matrix $\mathbf{T}_1$ of Eq. (28):

$$\mathbf{T}_1 = \begin{bmatrix} 64 & 128 & 32 \\ -32 & -64 & 128 \\ 128 & -64 & -16 \end{bmatrix} \tag{28}$$

Third, the approximation error is reduced by restoring two properties of the original kernel: the second and third rows sum to zero, so that gray inputs produce no chroma, and the first row is given a sum of $2^8$. The resulting matrix $\mathbf{T}_2$ is given in Eq. (29), and it decomposes into the two terms of Eq. (30), whose entries are 0, 1, and −1:

$$\mathbf{T}_2 = \begin{bmatrix} 64 & 128 & 64 \\ -64 & -64 & 128 \\ 128 & -64 & -64 \end{bmatrix} \tag{29}$$

$$\mathbf{T}_{\text{mod}} = 2^7 \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} + 2^6 \begin{bmatrix} 1 & 0 & 1 \\ -1 & -1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \tag{30}$$

Substituting $2^{-8}\,\mathbf{T}_{\text{mod}}$ for the kernel yields the multiplier-free conversion of Eq. (31):

$$\begin{aligned} Y_{\text{mod}} &= (R \gg 2) + (G \gg 1) + (B \gg 2) + 16 \\ C_{b,\text{mod}} &= -(R \gg 2) - (G \gg 2) + (B \gg 1) + 128 \\ C_{r,\text{mod}} &= (R \gg 1) - (G \gg 2) - (B \gg 2) + 128 \end{aligned} \tag{31}$$

where $\gg$ denotes a right shift. Only two distinct shift amounts (one and two bits) are required.

The receiver recovers the RGB components with the inverse of the modified kernel, given in Eq. (32):

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 3 & 1 & 5 \\ 3 & -3 & -3 \\ 3 & 5 & 1 \end{bmatrix} \left( \begin{bmatrix} Y_{\text{mod}} \\ C_{b,\text{mod}} \\ C_{r,\text{mod}} \end{bmatrix} - \mathbf{C} \right) \tag{32}$$

### 3.7 Evaluation metrics

Reconstruction quality is measured by the mean squared error (MSE) of Eq. (33) and the PSNR of Eq. (34):

$$MSE = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \left( I(x,y) - \hat{I}(x,y) \right)^2 \tag{33}$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \tag{34}$$

where $I$ and $\hat{I}$ are the original and reconstructed images of size $W \times H$. The structural similarity index (SSIM) [30] compares luminance, contrast, and structure and is defined in Eq. (35):

$$SSIM(a,b) = \frac{(2\mu_a\mu_b + c_1)(2\sigma_{ab} + c_2)}{(\mu_a^2 + \mu_b^2 + c_1)(\sigma_a^2 + \sigma_b^2 + c_2)} \tag{35}$$

where $\mu_a$ and $\mu_b$ are local means, $\sigma_a^2$ and $\sigma_b^2$ are local variances, $\sigma_{ab}$ is the covariance of the image patches $a$ and $b$, and $c_1$ and $c_2$ are stabilizing constants. The multi-scale SSIM (MS-SSIM) [31] evaluates the same comparison over several scales and is more robust to changes in viewing conditions. Video multi-method assessment fusion (VMAF) [32, 33] is a full-reference video quality metric that fuses elementary measures, namely visual information fidelity, the detail loss metric, and a temporal difference feature, with a support vector regressor; it builds on earlier fusion-based quality indices [34–

36]. The sample reduction achieved by a method is expressed by the compression percentage (CP) of Eq. (36):

$$CP = \frac{N_{\text{orig}} - N_{\text{enc}}}{N_{\text{orig}}} \times 100\% \tag{36}$$

where $N_{\text{orig}}$ and $N_{\text{enc}}$ are the numbers of samples before and after processing.

### 3.8 Statistical analysis

The results are reported as per-frame and per-sequence values. Repeated runs, confidence intervals, and significance tests were not performed.

## 4 Results

### 4.1 Lossless luma compression with BLC-SRR

The compression percentage obtained with BLC-SRR on four luma frames is given in Table 3. Between 18% and 20% of the luma samples were removed without any loss in the reconstructed frame.

**Table 3** Compression percentage of BLC-SRR on 320 × 240 luma frames of the test video [25]

| Frame | Original size (samples) | Encoded size (samples) | CP (%) |
|---|---|---|---|
| 1 | 76,800 | 61,440 | 20.0 |
| 2 | 76,800 | 62,208 | 19.0 |
| 3 | 76,800 | 62,976 | 18.0 |
| 4 | 76,800 | 61,824 | 19.5 |

### 4.2 Chroma compression with DCDS

DCDS was applied before the conventional 4:2:0(A), 4:2:0(R), and 4:2:0(L) schemes and before the BPP [6] and GD [5] methods. The results for two successive frames are listed in Table 4, and the comparison with interpolation-aware methods is given in Table 5. In every case the chroma CP increases from 75% to 87.5%. Except for GD in frame 1, where the PSNR is slightly higher with DCDS, the accompanying PSNR reduction lies between about 0.4 dB and 2.0 dB, that is, between about 0.9% and 4.4% of the baseline value.

**Table 4** PSNR, MSE, and MS-SSIM of conventional down-sampling methods without and with the proposed DCDS for two successive frames. The chroma CP is 75% without DCDS and 87.5% with DCDS in all rows

| Base method | Frame | PSNR (dB), base | PSNR (dB), with DCDS | MSE, base | MSE, with DCDS | MS-SSIM, base (R / G / B) |
|---|---|---|---|---|---|---|
| 4:2:0(A) | 1 | 42.9112 | 41.8542 | 2.6421 | 2.9521 | 0.9993 / 0.9998 / – |
| 4:2:0(A) | 2 | 42.8966 | 41.1597 | 2.6510 | 2.9950 | 0.9994 / 0.9998 / 0.9992 |
| 4:2:0(R) | 1 | 42.6669 | 41.6581 | 2.7951 | 2.9152 | 0.9993 / 0.9994 / 0.9996 |
| 4:2:0(R) | 2 | 44.1286 | 43.1228 | 2.8069 | 3.0245 | 0.9994 / 0.9994 / 0.9991 |
| 4:2:0(L) | 1 | 40.1449 | 38.5984 | 4.9956 | 3.8354 | 0.9992 / 0.9997 / 0.9988 |
| 4:2:0(L) | 2 | 40.0783 | 38.3258 | 5.0729 | 5.9824 | 0.9993 / 0.9998 / 0.9990 |
| BPP [6] | 1 | 45.5774 | 44.3258 | 1.8562 | 2.0254 | 0.9995 / 0.9998 / 0.9993 |
| BPP [6] | 2 | 44.2057 | 43.1245 | 1.9652 | 2.2158 | 0.9994 / 0.9997 / 0.9991 |
| GD [5] | 1 | 44.2774 | 45.1285 | 1.6582 | 1.8025 | 0.9996 / 0.9997 / 0.9994 |
| GD [5] | 2 | 45.1757 | 44.7587 | 1.7254 | 2.0154 | 0.9995 / 0.9997 / 0.9994 |

**Table 5** PSNR, MSE, and MS-SSIM of interpolation-aware down-sampling methods without and with the proposed DCDS. The chroma CP for every two frames is 75% without DCDS and 87.5% with DCDS in all rows

| Base method | PSNR (dB), base | PSNR (dB), with DCDS | MSE, base | MSE, with DCDS | MS-SSIM, base (R / G / B) | MS-SSIM, with DCDS (R / G / B) |
|---|---|---|---|---|---|---|
| IDID–NEDI [4] | 43.5896 | 41.8925 | 2.7584 | 3.2548 | 0.9993 / 0.9997 / 0.9992 | 0.9922 / 0.9997 / 0.9991 |
| JCDU–BILI [7] | 44.1256 | 42.9852 | 1.9652 | 2.2158 | 0.9994 / 0.9997 / 0.9993 | 0.9994 / 0.9994 / 0.9991 |
| LMCD–BILI [8] | 46.5982 | 44.5812 | 1.6125 | 1.9531 | 0.9996 / 0.9998 / 0.9994 | 0.9996 / 0.9997 / 0.9993 |
| LMCD–BECU [10] | 47.3546 | 46.1532 | 1.4259 | 1.6125 | 0.9998 / 0.9998 / 0.9996 | 0.9996 / 0.9997 / 0.9995 |

The per-frame PSNR and MSE over a test sequence are plotted in Fig. 3. The curves obtained with DCDS remain below the corresponding baseline curves throughout the sequence without a downward trend, which indicates that the loss introduced by the temporal averaging does not accumulate from frame to frame.

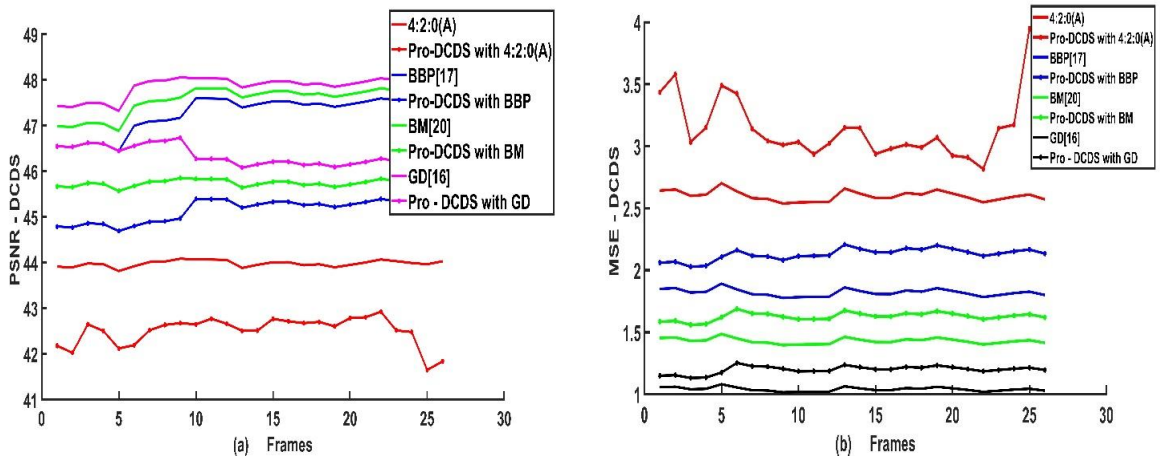



**Fig. 3** Per-frame comparison of existing down-sampling methods without and with the proposed DCDS: **a** PSNR, **b** MSE (BPP [6]; BM [9]; GD [5])

Sequence-level PSNR, SSIM, and VMAF values for Videos 1 and 2 are listed in Table 6. With 4:2:0(A), the PSNR falls from 59.68 dB to 57.97 dB for Video 1 and from 44.03 dB to 43.46 dB for Video 2; the VMAF score changes by less than 0.2 in every case.

**Table 6** Sequence-level PSNR, SSIM, and VMAF of reconstructions obtained with conventional 4:2:0 down-sampling and with the proposed DCDS (–: not evaluated)

| Method | Video 1 PSNR (dB) | Video 1 SSIM | Video 1 VMAF | Video 2 PSNR (dB) | Video 2 SSIM | Video 2 VMAF |
|---|---|---|---|---|---|---|
| 4:2:0(A) | 59.6771 | 0.9977 | 97.2764 | 44.0316 | 0.9922 | 97.1323 |
| 4:2:0(L) | 58.3583 | 0.9968 | 97.2494 | 43.1864 | 0.9900 | 97.0325 |
| 4:2:0(R) | 59.4641 | 0.9977 | 97.3334 | 43.9509 | 0.9921 | 97.0291 |
| DCDS + 4:2:0(A) | 57.9747 | 0.9972 | 97.3201 | 43.4591 | 0.9921 | 97.0211 |
| DCDS + 4:2:0(L) | 57.9747 | 0.9986 | 97.3201 | – | – | – |
| DCDS + 4:2:0(R) | 57.7542 | 0.9986 | 97.3443 | 42.7268 | 0.9881 | 96.9975 |

### 4.3 Luma compression with RF-ELDS

The per-frame PSNR and MSE obtained when RF-ELDS is applied to the luma plane in combination with existing chroma down-sampling methods are plotted in Fig. 4, and the sequence-level metrics are listed in Table 7.

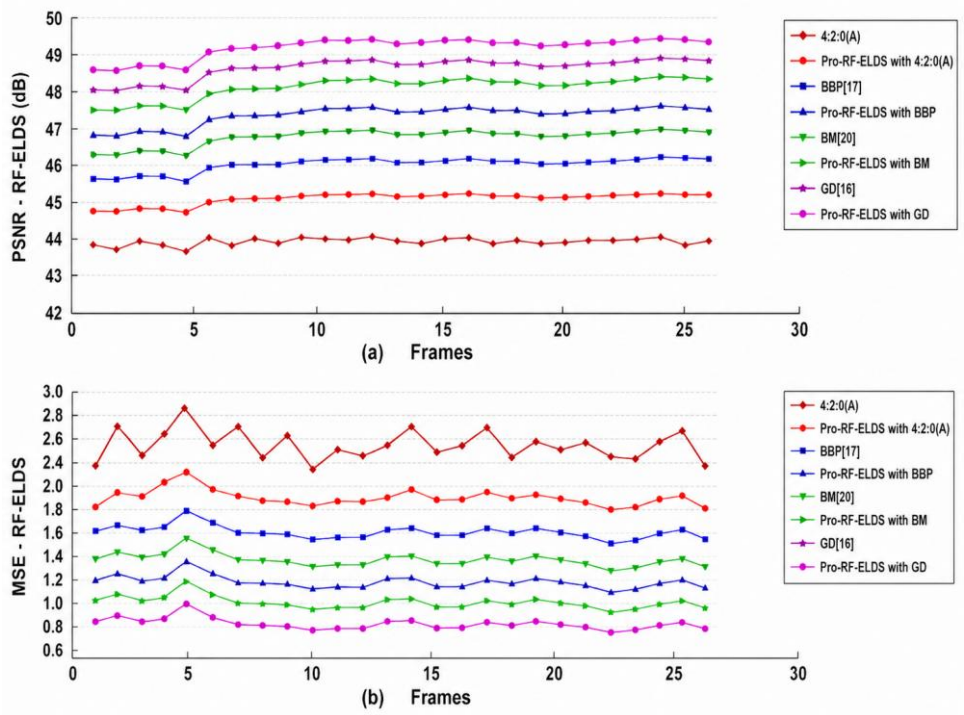



**Fig. 4** Per-frame comparison of existing chroma down-sampling methods without and with the proposed RF-ELDS on the luma plane: **a** PSNR, **b** MSE (BPP [6]; BM [9]; GD [5])

**Table 7** Sequence-level PSNR, SSIM, and VMAF of reconstructions obtained with conventional 4:2:0 down-sampling and with the proposed RF-ELDS (–: not evaluated)

| Method | Video 1 PSNR (dB) | Video 1 SSIM | Video 1 VMAF | Video 3 PSNR (dB) | Video 3 SSIM | Video 3 VMAF |
|---|---|---|---|---|---|---|
| 4:2:0(A) | 59.6771 | 0.9977 | 97.2764 | 53.7387 | 0.9971 | 99.4293 |
| 4:2:0(L) | 58.3583 | 0.9968 | 97.2494 | 51.9116 | 0.9963 | 99.4154 |
| 4:2:0(R) | 59.4641 | 0.9977 | 97.3334 | 53.3642 | 0.9970 | 99.4154 |
| RF-ELDS + 4:2:0(A) | 53.4761 | 0.9953 | 96.9178 | 50.9915 | 0.9952 | 93.1483 |
| RF-ELDS + 4:2:0(L) | 53.1095 | 0.9940 | 96.9178 | 50.9214 | 0.9960 | 99.1454 |
| RF-ELDS + 4:2:0(R) | 53.4291 | 0.9953 | 96.9784 | – | – | – |

At the sequence level (Table 7), RF-ELDS lowers the PSNR with 4:2:0(A) from 59.68 dB to 53.48 dB for Video 1 and from 53.74 dB to 50.99 dB for Video 3, while the SSIM remains at or above 0.994.

The rate–quality behavior is shown in Fig. 5. At equal bit rate, each method combined with RF-ELDS attains a PSNR equal to or slightly higher than that of the same method without RF-ELDS.

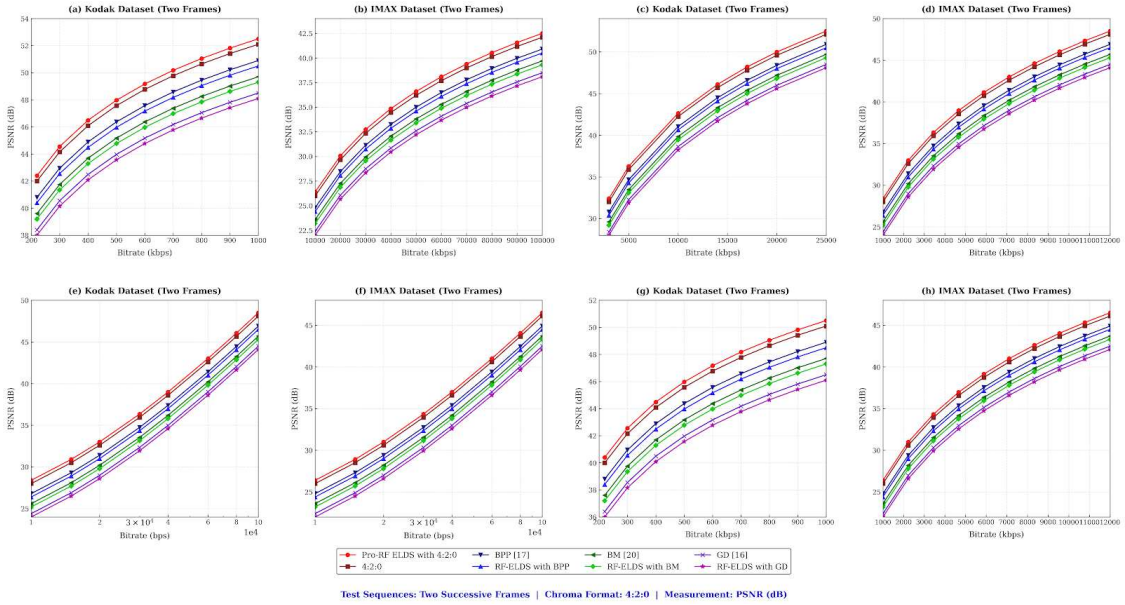



**Fig. 5** PSNR versus bit rate of existing chroma down-sampling methods without and with the proposed RF-ELDS for two successive frames in the 4:2:0 format (BPP [6]; BM [9]; GD [5])

The benefit of the predicted threshold over a fixed threshold is shown in Fig. 6. With the adaptive threshold, a luma CP of 20% is reached at a PSNR reduction of about 0.6 dB (from about 42.2 dB to 41.6 dB on the first data set and from about 40.5 dB to 39.9 dB on the second), whereas the fixed-threshold scheme loses about 3.2 dB at the same CP. Beyond 20%, the PSNR of RF-ELDS also declines more steeply, so 20% is taken as the operating point.

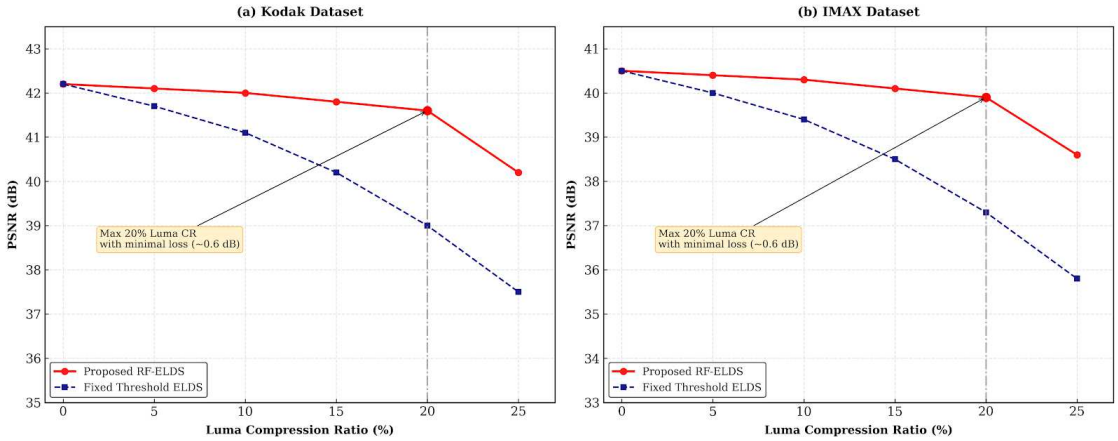



**Fig. 6** PSNR versus luma compression percentage for the proposed RF-ELDS and the fixed-threshold ELDS on two data sets

## 4.4 Combined RF-ELDS and DCDS

When RF-ELDS is applied to the luma plane and DCDS to the chroma planes, the per-frame results in Fig. 7 are obtained. The total sample reduction for two successive frames is 50% for the existing 4:2:0 methods and 58.3% with DCDS, in agreement with the sample counts of Section 3.4, and it rises to 68–70% when RF-ELDS is added (Fig. 7c), which corresponds to a luma CP of about 29–35% for these frames.

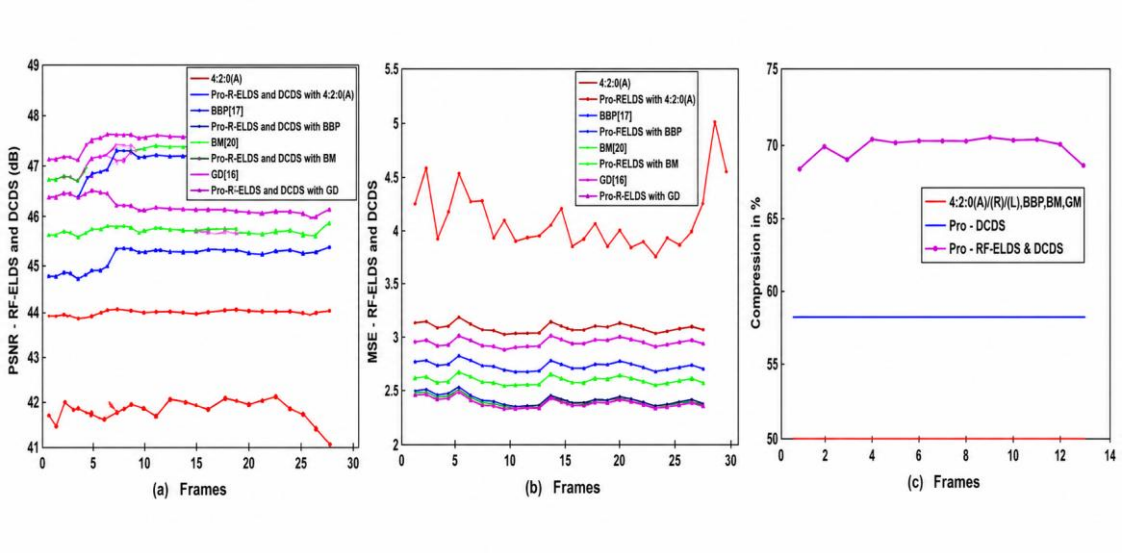



**Fig. 7** Per-frame comparison of existing methods with the proposed RF-ELDS on luma and DCDS on chroma for two successive frames: **a** PSNR, **b** MSE, **c** total compression percentage (BPP [6]; BM [9]; GD [5])

The sequence-level metrics of the combined scheme are listed in Table 8. The PSNR reduction relative to conventional 4:2:0 down-sampling ranges from 0.7 dB to 6.7 dB (1.5% to 11.2%) over the five evaluated combinations of video and 4:2:0 variant, the SSIM remains above 0.986, and the VMAF score remains above 96.5.

**Table 8** Sequence-level PSNR, SSIM, and VMAF of reconstructions obtained with conventional 4:2:0 down-sampling and with the proposed RF-ELDS and DCDS combined (–: not evaluated)

| Method | Video 1 PSNR (dB) | Video 1 SSIM | Video 1 VMAF | Video 2 PSNR (dB) | Video 2 SSIM | Video 2 VMAF |
|---|---|---|---|---|---|---|
| 4:2:0(A) | 59.6771 | 0.9977 | 97.2764 | 44.0316 | 0.9922 | 97.1323 |
| 4:2:0(L) | 58.3583 | 0.9968 | 97.2494 | 43.1864 | 0.9900 | 97.0325 |
| 4:2:0(R) | 59.4641 | 0.9977 | 97.3334 | 43.9509 | 0.9921 | 97.0291 |
| RF-ELDS + DCDS + 4:2:0(A) | 52.9795 | 0.9972 | 97.9201 | 42.1688 | 0.9862 | 96.5636 |
| RF-ELDS + DCDS + 4:2:0(L) | 52.9340 | 0.9986 | 96.9801 | – | – | – |
| RF-ELDS + DCDS + 4:2:0(R) | 57.9340 | 0.9948 | 96.9801 | 43.2863 | 0.9901 | 96.6263 |

## 4.5 Color conversion with CTKMA

The quality of images converted with the modified kernel of Eq. (31) and reconstructed with Eq. (32) is reported in Table 9. The PSNR exceeds 49 dB and the MS-SSIM exceeds 0.976 for all four images, and Fig. 8 shows no visible difference from the images obtained with the original kernel.

**Table 9** Reconstruction quality of the proposed CTKMA on standard test images

| Image | PSNR (dB) | MSE | SSIM | MS-SSIM |
|---|---|---|---|---|
| Pepper | 49.295 | 0.7648 | 0.9851 | 0.9888 |
| Lena | 49.130 | 0.7944 | 0.9834 | 0.9935 |
| Baboon | 49.123 | 0.7957 | 0.9741 | 0.9767 |
| Airplane | 49.287 | 0.7662 | 0.9830 | 0.9957 |




**Fig. 8** Test images reconstructed with the original conversion matrix of Eq. (1) (top row) and with the proposed CTKMA (bottom row)

The arithmetic operations required per pixel are compared in Table 10. The direct fixed-point implementation of Eq. (1) needs nine multiplications. The shift-based method of Yang et al. [18] removes them by expanding each coefficient into powers of two (for example, $0.2969 \approx 2^{-2} + 2^{-5} + 2^{-6}$), which costs 35 additions. CTKMA needs no multiplication, two shift amounts, and six additions.

**Table 10** Arithmetic operations of RGB-to-YCbCr conversion methods (constant-offset additions are not counted)

| Method | Additions | Multiplications | Right shifts | Left shifts |
|---|---|---|---|---|
| Direct fixed-point, Eq. (1) | 6 | 9 | 0 | 0 |
| Fixed-point with left shifts [18] | 35 | 0 | 0 | 9 |
| Proposed CTKMA | 6 | 0 | 2 | 0 |

The synthesis results are given in Table 11. CTKMA has the shortest combinational path delay, 8.243 ns, which is 50.6% below that of the direct fixed-point design and 35.5% below that of the shift-based design. Its slice and look-up table counts are far below those of the direct design and slightly above those of the shift-based design.

**Table 11** Synthesis results of RGB-to-YCbCr conversion methods on a Spartan-3E XC3S100E device (Xilinx ISE 14.7)

| Method | Maximum combinational path delay (ns) | Slices | 4-input LUTs | Bonded IOBs |
|---|---|---|---|---|
| Direct fixed-point, Eq. (1) | 16.677 | 150 | 205 | 72 |
| Fixed-point with left shifts [18] | 12.780 | 9 | 16 | 10 |
| Proposed CTKMA | 8.243 | 11 | 18 | 54 |

## 5 Discussion

The principal finding is that the correlation between successive frames can be used at the down-sampling stage, before any standard encoder, to remove samples that conventional 4:2:0 processing retains. DCDS halves the transmitted chroma data at a PSNR cost of at most about 2 dB in the frame-level results, and RF-ELDS removes about one fifth of the luma samples at a cost of about 0.6 dB at its operating point.

Compared with the methods reviewed in Section 2, the proposed modules act on a different axis. IDID, JCDU, BPP, BM, GD, and LMCD improve the fidelity obtained at a fixed sampling ratio, whereas DCDS changes the sampling ratio itself and can be placed in front of any of them, as Tables 4 and 5 show. In Table 5 the ranking of the base methods is preserved after DCDS is applied, which suggests that optimization-based down-sampling and temporal merging are complementary.

The advantage of RF-ELDS over the fixed-threshold scheme can be explained by the content dependence of the SAD. A single threshold that is safe for textured or moving macroblocks is too conservative for static smooth regions, and a threshold tuned to static regions merges macroblocks whose differences are visible. The regressor resolves this conflict by raising the threshold where the features indicate low activity and lowering it elsewhere. The shallow forest of ten trees with a depth of eight keeps the inference cost at no more than 80 comparisons per macroblock.

CTKMA shortens the critical path because each output is formed from three shifted operands, and shifts by a constant cost only wiring in hardware. The price is that the modified kernel no longer conforms to ITU-R BT.601: the luma weights become (1/4, 1/2, 1/4), so the transmitter and the receiver must both use the modified kernel, and a standard decoder would display shifted colors. In addition, the inverse in Eq. (32) contains divisions by three for the R and B outputs, so the receiver is not multiplier-free. The luma row of the modified kernel coincides with that of the reversible color transform of JPEG 2000 [19], which forms the luma as $(R + 2G + B)/4$ and the chroma as the differences $B - G$ and $R - G$. CTKMA differs in its chroma rows, $(2B - R - G)/4$ and $(2R - G - B)/4$, which keep the sign structure and the offsets of the BT.601 components.

For wireless multimedia communication, the practical implication is a smaller payload before entropy coding and a cheaper front end. For each frame pair, the number of samples delivered to the encoder falls from $3N^2$ with 4:2:0 to $2.5N^2$ with DCDS and to about $2.1N^2$ when 20% of the luma samples are also removed, a reduction of 30% relative to 4:2:0. Fewer samples reduce the encoder workload and, at a given quantization setting, the bit rate that must be carried by the radio link, which in turn reduces airtime and transmit energy. The multiplier-free conversion lowers the per-pixel switching activity of the front end, which matters in battery-powered nodes. These implications are inferred from sample counts and synthesis results; they have not been measured on a wireless link.

Two trade-offs should be noted. Both temporal modules process frames in pairs, which introduces a delay of one frame at the transmitter, and the merged chroma or luma data are shared by both frames, so fast motion or scene changes increase the distortion. Two observations also indicate that the effect of the modules depends on content: one configuration (GD, frame 1) shows a slightly higher PSNR with DCDS than without, and the sequence-level PSNR loss of RF-ELDS is larger than the frame-level loss at the 20% operating point. Both should be examined on a larger and more varied test set.

# 6 Limitations

The study has the following limitations. First, all quality and compression figures are simulation results obtained in MATLAB, and the hardware figures are synthesis estimates; neither a complete encoder–decoder implementation nor on-board measurements are included. Second, the video evaluation uses a small number of short, low-resolution clips; results at higher resolutions, on high-motion content, and on the common-test-condition sequences of the video coding standards are not yet available. Third, the generalization of the random forest to content outside its training sequences was not evaluated separately. Fourth, no repeated runs or statistical tests support the reported differences. Fifth, the side

information required by the decoder (block masks in BLC-SRR and decision flags in RF-ELDS) and the interaction of the modules with the entropy coder of a standard codec are not quantified. Sixth, the effect of channel errors and packet loss on frame pairs that share chroma or luma data has not been studied, although it is central to wireless transmission. Seventh, the color space produced by CTKMA is not standard-compliant, and its luma range exceeds eight bits unless the output is clipped.

## 7 Conclusion

This paper presented a low-complexity pre-processing codec framework that combines BLC-SRR, DCDS, RF-ELDS, and CTKMA. In simulation, BLC-SRR removed 18–20% of the luma samples without loss by representing 4 × 4 blocks with a representative value and a binary mask. DCDS raised the chroma sample reduction from 75% to 87.5% by merging the chroma planes of successive frames, at a frame-level PSNR cost of about 0.4–2.0 dB. RF-ELDS, which predicts a macroblock-adaptive threshold with a random forest regressor, removed about 20% of the luma samples at a PSNR cost of about 0.6 dB at its operating point and outperformed a fixed threshold. CTKMA replaced the nine multiplications of the RGB-to-YCbCr conversion by two shift amounts and six additions, preserved a PSNR above 49 dB on the test images, and reached the shortest combinational path delay, 8.243 ns, in synthesis. Within the tested conditions, the framework reduces the number of samples and the arithmetic cost ahead of the encoder, which makes it a candidate for bandwidth- and power-limited wireless multimedia terminals. Future work will validate the framework on standard test sequences inside a complete HEVC or VVC coding chain, quantify the side information, and evaluate its behavior over error-prone wireless channels.

## Declarations

**Funding** No funding was received for conducting this study.

**Competing interests** The authors have no competing interests to declare that are relevant to the content of this article.

**Data availability** The test videos and image datasets are publicly available from the sources cited in the reference list. The simulation results are available from the corresponding author on reasonable request.

**Code availability** The MATLAB and Verilog code is available from the corresponding author on reasonable request.

**Author contributions** Both authors contributed to the conception and design of the study, the development of the methods, the simulations, the hardware synthesis, and the analysis of the results. Both authors wrote, read, and approved the final manuscript.

## References

[1] Kang, L.-W., Lu, C.-S., & Lin, C.-Y. (2012). Low-complexity video coding via power–rate–distortion optimization. *Journal of Visual Communication and Image Representation, 23*(3), 569–585.

[2] Poynton, C. A. (1996). *A technical introduction to digital video*. John Wiley & Sons.

[3] International Telecommunication Union. (1995). *Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios* (Recommendation ITU-R BT.601-5). ITU.

[4] Zhang, Y., Zhao, D., Zhang, J., Xiong, R., & Gao, W. (2011). Interpolation-dependent image downsampling. *IEEE Transactions on Image Processing, 20*(11), 3291–3296. https://doi.org/10.1109/TIP.2011.2158226

[5] Chung, K.-L., Lee, Y.-L., & Chien, W.-C. (2019). Effective gradient descent-based chroma subsampling method for Bayer CFA images in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology, 29*(11), 3281–3290. https://doi.org/10.1109/TCSVT.2018.2879095

[6] Chen, H., Sun, M., & Steinbach, E. (2009). Compression of Bayer-pattern video sequences using adjusted chroma subsampling. *IEEE Transactions on Circuits and Systems for Video Technology, 19*(12), 1891–1896. https://doi.org/10.1109/TCSVT.2009.2031370

[7] Wang, S., Gu, K., Ma, S., & Gao, W. (2016). Joint chroma downsampling and upsampling for screen content image. *IEEE Transactions on Circuits and Systems for Video Technology, 26*(9), 1595–1609. https://doi.org/10.1109/TCSVT.2015.2461891

[8] Chung, K.-L., Liu, T.-Y., & Cheng, J.-S. (2020). Novel and optimal luma modification-based chroma downsampling for Bayer color filter array images. *IEEE Open Journal of Circuits and Systems, 1*, 48–59. https://doi.org/10.1109/OJCAS.2020.2996624

[9] Lin, C.-H., Chung, K.-L., & Yu, C.-W. (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 Transactions on Circuits and Systems for Video Technology, 26*(9), 1722–1733.

[10] Ahilan, A., Pradeep Khanth, B., Ezhilarasi, R., & Muthukumaran, N. (2024). Efficient luma modification-based chroma down-sampling and novel luma down-sampling with adaptive interpolation. *Signal, Image and Video Processing, 18*(2), 1415–1428. https://doi.org/10.1007/s11760-023-02814-6

[11] Bayer, B. E. (1976). *Color imaging array* (U.S. Patent No. 3,971,065). U.S. Patent and Trademark Office.

[12] Doutre, C., Nasiopoulos, P., & Plataniotis, K. N. (2008). H.264-based compression of Bayer pattern video sequences. *IEEE Transactions on Circuits and Systems for Video Technology, 18*(6), 725–734. https://doi.org/10.1109/TCSVT.2008.919111

[13] Gastaldi, F., Koh, C. C., Carli, M., Neri, A., & Mitra, S. K. (2005). Compression of videos captured via Bayer patterned color filter arrays. In *Proceedings of the 13th European Signal Processing Conference (EUSIPCO)* (pp. 1–4).

[14] Doutre, C., & Nasiopoulos, P. (2009). Modified H.264 intra prediction for compression of video and images captured with a color filter array. In *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP)* (pp. 3401–3404). https://doi.org/10.1109/ICIP.2009.5413873

[15] Chung, K.-L., Huang, Y.-H., & Lin, C.-H. (2017). Improved universal chroma 4:2:2 subsampling for color filter array video coding in HEVC. *Signal, Image and Video Processing, 11*(6), 1041–1048. https://doi.org/10.1007/s11760-017-1055-1

[16] Chung, K.-L., Leung, T.-C., Liu, T.-Y., & Tseng, Y.-C. (2022). A cubic convolution interpolation-based chroma subsampling method for Bayer and RGBW CFA raw images. *IEEE Access, 10*, 22687–22699. https://doi.org/10.1109/ACCESS.2022.3154487

[17] Hamsavahini, R., & Rashmi, N. (2016). Implementation of RGB to YCbCr color space transformation for ARINC transceiver. *International Journal of Advanced Research in Computer and Communication Engineering, 5*(1), 394–398. https://doi.org/10.17148/IJARCCE.2016.5198

[18] Yang, Y., Peng, Y., & Liu, Z. (2007). A fast algorithm for YCbCr to RGB conversion. *IEEE Transactions on Consumer Electronics, 53*(4), 1490–1493. https://doi.org/10.1109/TCE.2007.4429242

[19] Taubman, D. S., & Marcellin, M. W. (2002). *JPEG2000: Image compression fundamentals, standards and practice*. Springer. https://doi.org/10.1007/978-1-4615-0799-4

[20] Zhang, Q., Wang, Y., Huang, L., & Jiang, B. (2020). Fast CU partition and intra mode decision method for H.266/VVC. *IEEE Access, 8*. https://doi.org/10.1109/ACCESS.2020.3004580

[21] Si, L., Zhu, W., & Zhang, Q. (2023). Fast adaptive CU partition decision algorithm for VVC intra coding. *IEEE Access, 11*. https://doi.org/10.1109/ACCESS.2023.3327519

[22] Chen, L., Cheng, B., Zhu, H., Qin, H., Deng, L., & Luo, L. (2024). Fast versatile video coding (VVC) intra coding for power-constrained applications. *Electronics, 13*(11), 2150. https://doi.org/10.3390/electronics13112150

[23] Bourtsoulatze, E., Kurka, D. B., & Gündüz, D. (2019). Deep joint source-channel coding for wireless image transmission. *IEEE Transactions on Cognitive Communications and Networking, 5*(3), 567–579. https://doi.org/10.1109/TCCN.2019.2919300

[24] Samplelib. (n.d.). *Sample MP4 video files*. https://samplelib.com/sample-mp4.html

[25] Learning Container. (n.d.). *Sample MP4 video files download*. https://www.learningcontainer.com/mp4-sample-video-files-download/

[26] Franzen, R. (n.d.). *Kodak lossless true color image suite*. http://r0k.us/graphics/kodak/

[27] Zhang, L., Wu, X., Buades, A., & Li, X. (2011). Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging, 20*(2), 023016. https://doi.org/10.1117/1.3600632

[28] Gyan Doshi. (n.d.). *FFmpeg builds for Windows*. https://www.gyan.dev/ffmpeg/builds/

[29] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

[30] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*(4), 600–612. https://doi.org/10.1109/TIP.2003.819861

[31] Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers* (Vol. 2, pp. 1398–1402). https://doi.org/10.1109/ACSSC.2003.1292216

[32] Rassool, R. (2017). VMAF reproducibility: Validating a perceptual practical video quality metric. In *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1–2). https://doi.org/10.1109/BMSB.2017.7986143

[33] Netflix Technology Blog. (2016). *Toward a practical perceptual video quality metric*. https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652. Accessed 15 July 2017.

[34] Liu, T.-J., Lin, Y.-C., Lin, W., & Kuo, C.-C. J. (2013). Visual quality assessment: Recent developments, coding applications and future trends. *APSIPA Transactions on Signal and Information Processing, 2*, e4. https://doi.org/10.1017/ATSIP.2013.5

[35] Lin, J. Y., Liu, T.-J., Wu, E. C.-H., & Kuo, C.-C. J. (2014). A fusion-based video quality assessment (FVQA) index. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 1–5). https://doi.org/10.1109/APSIPA.2014.7041705

[36] Lin, J. Y., Wu, C.-H., Katsavounidis, I., Li, Z., Aaron, A., & Kuo, C.-C. J. (2015). EVQA: An ensemble-learning-based video quality assessment index. In *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)* (pp. 1–5). https://doi.org/10.1109/ICMEW.2015.7169760

## Author biographies



**R. Ezhilarasi** received the B.E. degree in Electronics and Communication Engineering from the Government College of Technology, Coimbatore, India, the M.E. degree in Applied Electronics from Thanthai Periyar Government Institute of Technology, Vellore, India, and the Ph.D. degree from Anna University, Chennai, India. Dr. Ezhilarasi is currently an Assistant Professor in the Department of Electronics and Communication Engineering, University College of Engineering Arni, India.



**B. Pradeep Khanth** received the B.E. degree from Priyadarshini Engineering College, Tamil Nadu, India, the M.E. degree in Embedded Systems from Bannari Amman Institute of Technology, Erode, India, and the Ph.D. degree from Anna University, Chennai, India. Dr. Pradeep Khanth is currently a Teaching Fellow in the Department of Electronics and Communication Engineering, University College of Engineering Arni, India.

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ClientSideYCbCrReconstructionDiagram.webp
- ClientSideYCbCrtoRGBDiagram.png
- ServerSideMultiScalePatchPipeline.webp
- ServerSideYCbCrProcessingDiagram.webp