Adaptive color space model based on dominant colors for image and video compression performance improvement

S. Madenda [1], *A. Darmayantie* [1]
[1] *Computer Engineering Department, Gunadarma University,*
Jl. Margonda Raya. No. 100, Depok – Jawa Barat, Indonesia

Abstract

This paper describes the use of some color spaces in JPEG image compression algorithm and their impact in terms of image quality and compression ratio, and then proposes adaptive color space models (ACSM) to improve the performance of lossy image compression algorithm. The proposed ACSM consists of, dominant color analysis algorithm and YC_oC_g color space family. The YC_oC_g color space family is composed of three color spaces, which are YC_cC_r, YC_pC_g and YC_yC_b. The dominant colors analysis algorithm is developed which enables to automatically select one of the three color space models based on the suitability of the dominant colors contained in an image. The experimental results using sixty test images, which have varying colors, shapes and textures, show that the proposed adaptive color space model provides improved performance of 3 % to 10 % better than YC_bC_r, YD_bD_r, YC_oC_g and YC_gC_o-R color spaces family. In addition, the YC_oC_g color space family is a discrete transformation so its digital electronic implementation requires only two adders and two subtractors, both for forward and inverse conversions.

Keywords: colors dominant analysis, adaptive color space, image compression, image quality, compression ratio.

Citation: Madenda S, Darmayantie A. Adaptive color space model based on dominant colors for image and video compression performance improvement. Computer Optics 2021; 45(3): 405-417. DOI: 10.18287/2412-6179-CO-780.

Acknowledgments: Thank you to Gunadarma University for providing funding support during the research and publication process.

1. Introduction

Raw images and video frames recorded by high-resolution cameras in information and communication technology (ICT) devices are in the form of three basic color components R (red), G (green) and B (blue). These color components are called RGB color spaces. Saving high-resolution raw images and video frames in RGB color space requires enormous memory. Image and video compression algorithms are the main solutions to reduce data size and minimize storage requirements. JPEG and JPEG2000 are two compression algorithms that are widely used in ICT devices. In general, these two algorithms have the same process flow, as shown in fig. 1 [1, 2, 3]. The Compression algorithm in fig. 1*a* consists of the following processes: color space conversion, spatial to frequency transform, quantization and coding. Contrary, the decompression algorithm (fig. 1*b*) consists of: decoding, inverse quantization, frequency to spatial transform and inverse color space conversion.

In image and video compression algorithms, the choice of color space is an important part, because it directly affects the compression ratio and image quality. This means that the use of the right color space can increase the compression ratio and image quality. Generally, the color space used is in accordance with the sensitivity of the human visual system to color changes, especially the color spaces that have luminance and chrominance components. The JPEG and JPEG2000 compression algorithms use color spaces YC_bC_r and YD_bD_r, which Y correspond to luminance, C_b and D_b are blue chrominance, and C_r and D_r are red chrominance components [3, 4, 5, 6]. The Color conversion process from RGB to YC_bC_r can be calculated through a mathematical transformation as given in Eq. (1), while the inverse conversion process from YC_bC_r to RGB is shown by Eq. (2). Matrix A is called a color transformation matrix to which $[A^{-1}]$ is the inverse of matrix $[A]$ and is called an inverse color transformation matrix. Numerical implementation of RGB to YC_bC_r conversion requires two shift-right operations, seven multiplications and six adders/subtractors. Numerical implementation of YC_bC_r to RGB conversion requires four multiplications and four adders / subtractors. This color space is an irreversible color transform because it is rounded during the process.

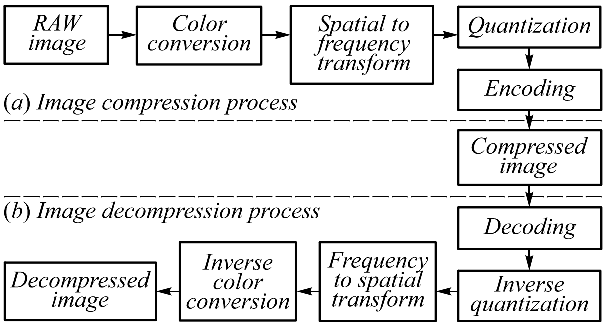

Fig. 1. General block diagram of JPEG and JPEG2000 algorithms: (a) Compression and (b) Decompression processes

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix},\ [A] = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1688 & -0.3312 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{bmatrix}, \quad (1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [A^{-1}] \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix},\ [A^{-1}] = \begin{bmatrix} 1.0 & 0.0 & 1.4020 \\ 1.0 & -0.3441 & -0.7141 \\ 1.0 & 1.7720 & 0.0 \end{bmatrix}. \quad (2)$$

The YD_bD_r color space used in the JPEG2000 compression algorithm is a variant of YC_bC_r [2, 7–9]. The matrices [A] and [A^{-1}] in Eqs. (3, 4) respectively show the transformation matrix for RGB to YD_bD_r conversion and inverse transformation matrix for YD_bD_r to RGB conversion. This color space is reversible color transform (RCT) and its numerical implementation is simpler because it only requires shift-right (symbolized by "»"), addition and subtraction operations, and without multiplication operation. For example, 0.5×R is equal to one-bit shift-right of R value (R » 1) and 0.25×B is identical to two-bits shift-right of B value (B » 2). Thus, for RGB to YD_bD_r conversion requires two shift-right operations and four adders / subtractors. While YD_bD_r to RGB inverse conversion requires three shift-right and seven addition and subtraction operations.

$$\begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix},\ [A] = \begin{bmatrix} 0.25 & 0.50 & 0.25 \\ 0.0 & -1.0 & 1.0 \\ 1.0 & -1.0 & 0.0 \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [A^{-1}] \begin{bmatrix} Y \\ D_b \\ D_r \end{bmatrix},\ [A^{-1}] = \begin{bmatrix} 1.0 & -0.25 & 0.75 \\ 1.0 & -0.25 & -0.25 \\ 1.0 & 0.75 & -0.25 \end{bmatrix}. \quad (4)$$

Another variant of the YC_bC_r color space is YC_oC_g, which has a conversion process easier and faster than YC_bC_r, and it is a reversible color transform [9, 10]. YC_oC_g is also known as the YC_gC_o color model. The matrix A is used for the conversion process from RGB to YC_oC_g and the inverse transformation matrix [A^{-1}] is applied for the conversion process from YC_oC_g to RGB which are given in Eqs. (5, 6). The numerical implementation of RGB *to* YC_oC_g conversion requires four shift-right, two addition and tow subtraction operations. Likewise for YC_oC_g to RGB conversion process requires only two addition and two subtraction operations.

$$\begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix},\ [A] = \begin{bmatrix} 0.25 & 0.50 & 0.25 \\ 0.50 & 0.0 & -0.50 \\ -0.25 & 0.50 & -0.25 \end{bmatrix}, \quad (5)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [A^{-1}] \begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix},\ [A^{-1}] = \begin{bmatrix} 1.0 & 1.0 & -1.0 \\ 1.0 & 0.0 & 1.0 \\ 1.0 & -1.0 & -1.0 \end{bmatrix}. \quad (6)$$

YC_oC_g color space is then developed into YC_gC-R with the aim to speed-up the conversion process by multiplying by 2 each value of the chrominance components [11, 12] (see Eqs. (7, 8)). Numerical implementation of RGB to YC_gC_o-R conversion and its inverse conversion form YC_gC_o-R to RGB, both only require two shift-right, two addition and two subtraction operations.

$$[A] = \begin{bmatrix} 0.25 & 0.50 & 0.25 \\ 1.0 & 0.0 & -1.0 \\ -0.50 & 1.0 & -0.50 \end{bmatrix}, \quad (7)$$

$$[A^{-1}] = \begin{bmatrix} 1.0 & 0.5 & -0.5 \\ 1.0 & 0.0 & 0.5 \\ 1.0 & -0.5 & -0.5 \end{bmatrix}. \quad (8)$$

In [13], the authors developed a new family of reversible low-complexity color transformation: Y_iUV_j. The YD_bD_r and YC_gC_o-R color spaces are included in this family. The $C_o = $ (R-G) / 2 and $C_g = $ (2G-(R+B))/4 as chrominance components of YC_oC_g color space (Eq. 5) are not yet included. From this family, the color space components Y_i, U_j and V_j, which have computational with lower complexity, are given in table 1. The performance of each color space has been compared with the focus on lossy compression and by applying an automatic color space selection algorithm based on entropy H.

Table 1. Y_iUV_j reversible color transformation.

i	Y_i	j	U_j	V_j
1	(R + 2G + B) / 4	1	G – (R + B) / 2	R – B
2	(R + G + 2B) / 4	2	B – (R + G) / 2	R – G
3	(2R + G + B) / 4	3	R – (G + B) / 2	B – G
4	(R + G) / 2	4	B – (R + G) / 2	R – G
5	(R + B) / 2	5	G – (R + B) / 2	R – B
6	(R + B) / 2	6	R – (G + B) / 2	B – G

The image in R, G and B color components format must first be converted into each color component of Y_i, U_j, and V_j, and then the entropy value is calculated for each of $H(Y_i)$, $H(U_j)$, and $H(V_j)$. It is assumed that the one having the smallest entropy value ($H = H(Y_i) + H(U_j) + H(V_j)$), as the components of the color space, will be selected to be applied. To avoid high computation time, an image is divided into $p \times q$ macro blocks, where the number of blocks depends on the image size. For example, an image is divided into 9 blocks ($p = q = 3$), in each block the same components can be applied, or may also be different, according to their entropy calculation result. Both adaptive color space and macro block models have been developed in [14, 15] and implemented in 4: 4: 4 video coding.

The first three color spaces in table 1, Y_1UV_1, Y_2UV_2 and Y_3UV_3 (each for $r = i = j$ is called YUV_r: YUV_1, YUV_2, and YUV_3), can be represented as matrix multiplication and each accompanied by its inverse transformation, respectively shown by Eqs. (9, 10, 11, 12, 13, 14). The YUV_1 color space is the same as the YC_gC_o-R color space, while YUV_2 and YUV_3 are the permutations of the YUV_1 color space. These three color spaces can be applied for lossless and lossy image compression.

RGB to YUV_1 and YUV_1 to RGB :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \left[A_1\right] \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \text{ where } \left[A_1\right] = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ -0.5 & 1.0 & -0.5 \\ 1.0 & 0.0 & -1.0 \end{bmatrix}, \quad (9)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \left[A_1^{-1}\right] \begin{bmatrix} Y \\ U \\ V \end{bmatrix}, \text{ where } \left[A_1^{-1}\right] = \begin{bmatrix} 1.0 & -0.5 & 0.5 \\ 1.0 & 0.5 & 0.0 \\ 1.0 & -0.5 & -0.5 \end{bmatrix}. \quad (10)$$

RGB to YUV_2 and YUV_2 to *RGB*:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \left[A_2\right] \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \text{ where } \left[A_2\right] = \begin{bmatrix} 0.25 & 0.25 & 0.5 \\ -0.5 & -0.5 & 1.0 \\ 1.0 & -1.0 & 0.0 \end{bmatrix}, \quad (11)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \left[A_2^{-1}\right] \begin{bmatrix} Y \\ U \\ V \end{bmatrix}, \text{ where } \left[A_2^{-1}\right] = \begin{bmatrix} 1.0 & -0.5 & 0.5 \\ 1.0 & -0.5 & -0.5 \\ 1.0 & 0.5 & 0.0 \end{bmatrix}. \quad (12)$$

RGB to YUV_3 and YUV_3 to *RGB* :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \left[A_3\right] \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \text{ where } \left[A_3\right] = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 1.0 & -0.5 & -0.5 \\ 0.0 & -1.0 & 1.0 \end{bmatrix}, \quad (13)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \left[A_3^{-1}\right] \begin{bmatrix} Y \\ U \\ V \end{bmatrix}, \text{ where } \left[A_3^{-1}\right] = \begin{bmatrix} 1.0 & 0.5 & 0.0 \\ 1.0 & -0.5 & -0.5 \\ 1.0 & -0.5 & 0.5 \end{bmatrix}. \quad (14)$$

2. Performance evaluation of existing color spaces

In terms of numerical implementation, it can be concluded that YUV_1 (= YC_oC_g-R), YUV_2 and YUV_3 are simpler and faster than YC_oC_g, YD_bD_r and YC_bC_r color spaces. So, what about the performance in terms of compression ratio and image quality, if each color space is implemented in the lossy image compression algorithm?

In this part, we describe performance evaluation of color spaces YC_bC_r, YD_bD_r, YC_oC_g, YUV_1, YUV_2 and YUV_3, applied in JPEG compression algorithm. As known that image quality and compression ratio of color image compression algorithms are not only determined by the diversity of color contained in an image, but also influenced by texture and shape variations. For these reasons, in the experiments, 60 images were selected containing various colors, shapes and textures from six sources of image dataset [16, 17, 18, 19, 20, 21]. Twelve of them are shown in fig. 2. For example, the Lena image has the dominant colors red, orange and red-purple, many homogeneous areas, slight in texture and shape variation. The Baboon image has random textures and shapes, and its dominant colors are green-yellow, cyan and red. The Frymire image is very rich in textures and shapes, as well as containing variant colors with almost the same composition between yellow, red, orange, green, cyan, blue and purple. The Roses image has dominant colors red, red-purple (pink), blue-cyan, blue-purple, yellow, white and black. The Woman image has a little color variation, some areas are relatively homogeneous and varying texture patterns in the hair. Yellow_orchid has a dominant homogeneous area, yellow and cyan colors, as well as various shapes and slight variations in texture.

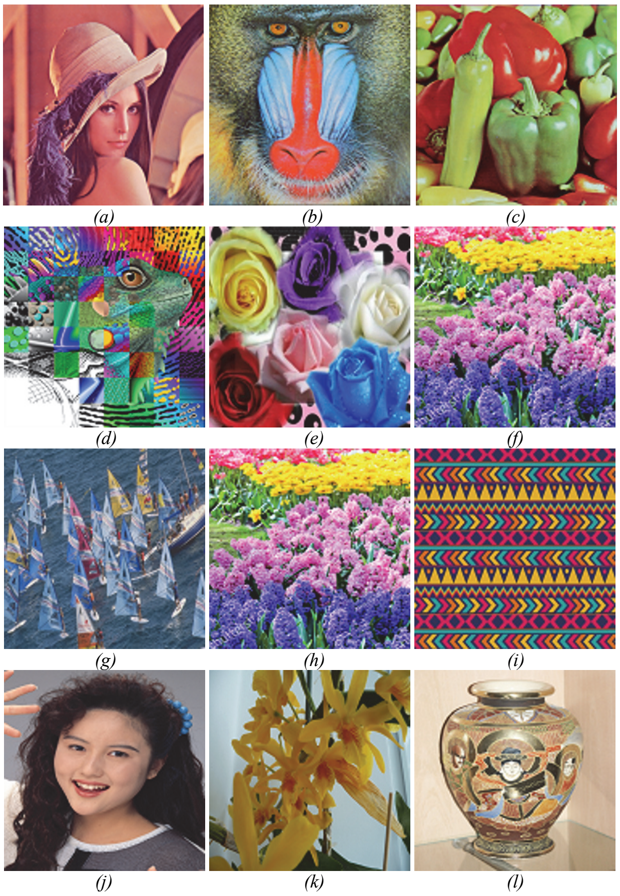

Fig. 2. Twelve test images: (a) Lena, (b) Baboon, (c) Peppers, (d) Frymire, (e) Roses, (f) Tulips, (g) Sails, (h) Monarch, (i) Textile, (j) Woman, (k) yellow_orchid, and (l) ucid00295

Next, we outline the performance comparison of six color space YC_bC_r, YD_bD_r, YC_oC_g, YUV_1, YUV_2 and YUV_3, regarding their compression ratio and image quality, when each of them is used in the lossy image compression algorithm. Block size 8 × 8 pixels [1], color components without sub-sampling for high quality image compression [23, 24, 25], discrete cosine transform (8 × 8 DCT and iDCT) [1, 26, 27, 28, 29], and Huffman statistic encoding / decoding [1, 2, 8, 22, 30] are chosen to carry-out the experiments of JPEG lossy image compression. For each color space used, it is done by determining the same quality for the same image and then compares their compression ratio. In this case, the experiment was carried out by using one Photoshop's quantization matrix for medium quality (Q_9) as shown in Eq. (15) [3, 22]. Q_{Lum} and Q_{Cr} are respectively quantization matrices for luminance and chrominance components, and *q* is a variable that determines the quantization value, so that the seven color spaces can produce the same quality for the same image, thus the compression ratio can be compared.

$$Q_{Cr} = q.\begin{bmatrix} 4 & 6 & 12 & 22 & 20 & 20 & 17 & 17 \\ 6 & 8 & 12 & 14 & 14 & 12 & 12 & 12 \\ 12 & 12 & 14 & 14 & 12 & 12 & 12 & 12 \\ 22 & 14 & 14 & 12 & 12 & 12 & 12 & 12 \\ 20 & 14 & 12 & 12 & 12 & 12 & 12 & 12 \\ 20 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

and

$$Q_{Lum} = q.\begin{bmatrix} 4 & 3 & 4 & 7 & 9 & 11 & 14 & 17 \\ 3 & 3 & 4 & 7 & 9 & 12 & 12 & 12 \\ 4 & 4 & 5 & 9 & 12 & 12 & 12 & 12 \\ 7 & 7 & 9 & 12 & 12 & 12 & 12 & 12 \\ 9 & 9 & 12 & 12 & 12 & 12 & 12 & 12 \\ 11 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 14 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 17 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \end{bmatrix} \quad (15)$$

Table 2. Compression ratio of nine images using JPEG algorithm for six color spaces

File Image (png)	**Quali-ty PSNR (dB)**	**Compression Ratio (CR) Using Color Space:**					
		YC_bC_r	***YD_bD_r***	***YC_oC_g***	***YUV_1 YC_oC_g-R***	***YUV_2***	***YUV_3***
Lena	36.117	9.490	8.430	**10.024**	8.910	7.987	9.076
Baboon	32.805	3.554	3.486	**3.798**	3.543	3.446	3.506
Peppers	34.057	6.161	5.952	**6.580**	6.230	6.419	5.995
Frymire	34.227	3.696	3.500	**3.765**	3.577	3.585	3.597
Roses	46.776	**19.96**	19.118	19.709	18.960	19.213	19.340
Tulips	33.764	**3.784**	3.405	3.679	3.500	3.669	3.601
Sails	37.456	7.451	7.228	**7.750**	7.220	7.081	7.013
Monarch	39.788	12.288	11.560	**12.553**	11.834	11.788	11.702
Textile	36.838	5.442	5.422	**5.464**	5.280	5.439	5.166
Woman	38.011	10.156	10.400	**10.442**	9.021	8.862	9.062
Yellow_Or	41.050	21.398	22.915	23.518	23.355	**24.131**	22.996
Ucid00295	36.614	7.642	7.794	**7.908**	7.228	7.086	7.174
CR Average		9.252	9.101	**9.599**	9.055	9.059	9.019
Note: ↑Increase ↓Decrease			↓1.6%	↑3.8%	↓2.1%	↓2.1%	↓2.5%

Table 2 shows the compression ratio (CR) of JPEG algorithm using color spaces: YC_bC_r, YD_bD_r, YC_oC_g, YUV_1, YUV_2 and YUV_3. In the first line, for Lena image with compression quality of PSNR = 36.117 dB, the CR produced by YC_bC_r is 9.490, YD_bD_r is 8.430, YC_oC_g is 10.024 and respectively for YUV_1, YUV_2 and YUV_3 are 8.910, 7.987, and 9.076. Similarly, for Baboon image in the second row with a compression quality of PSNR = 32.805 dB, six consecutive color spaces produced CR of: 3.554, 3.486, 3.798, 3.543, 3.446 and 3.506. Furthermore, the CR for Peppers, Frymire, Roses, Tulips, Sails, Monarch, Textile, Woman, yellow_orchid and ucid00295 images can be seen in the next lines.

From the compression ratio results in table 2, it can be seen that, YC_oC_g color space yields highest CR for Lena, Baboon, Peppers, Frymire, Sails, Monarch, Woman and Ucid00295 images compared to the five others. YC_bC_r results in better CR for Roses and Tulips images, while Yellow_Orchid image has highest CR given by YUV_2. In the two last rows of table 2, the CR average and the performance produced by each color space is given. When referring to these CR average values and comparing them to the CR average value of JPEG-standard YC_bC_r color space, YC_oC_g provides better performance with increased CR of 3.8 % on average, while YD_bD_r, YUV_1, YUV_2 and YUV_3 resulted CR averages of 1.6 %, 2.1 %, 2.1 %, and 2.5 % lower than YC_bC_r respectively.

The experimental results of the other forty-eight images are given in tables 4 – 5. Table 4 represents the compression ratios for the same image quality and table 5 reflects the image quality for the same compression ratio. For now, we focus on the first seven columns to evaluate the compression results of the forty-eight images using YC_bC_r, YC_oC_g, YUV_1, YUV_2 and YUV_3 color spaces. There are thirty-seven images (75.51 %) that have the highest compression ratio and image quality generated by the YC_oC_g color space (in the fourth column), nine images (18.37 %) given by the YC_bC_r color space (in the third column), and the remaining two, one and zero images respectively resulted by YUV_1, YUV_2 and YUV_3 color spaces (in the fifth, sixth and seventh column). In the last two rows of each table, the CR average and the performance yielded by each color space is given. YC_oC_g color space has the highest average value of 9.599, followed by YC_bC_r = 9.252, YD_bD_r = 9.101, YUV_2 = 9.059, YUV_1 = 9.055 and YUV_2 = 9.019. These CR average values show that for those forty-eight images, the use of YC_oC_g in the JPEG lossy compression algorithm can increase the CR average of 3.623 %, while for YUV_1, YUV_2 and YUV_3 there is a decrease of 3.158 %, 5.056 % and 4.693 % respectively.

These results indicate that in general YC_oC_g color space provides better performance than the five other color spaces and YC_bC_r has a higher compression ratio than YUV_1, YUV_2, YUV_3 and YD_bD_r color spaces. It may be assumed that the compatibility between the percentage composition of each color in a color space formula and the dominant colors contained in an image can affect the quality and compression ratio. The first analysis is about the percentage of color representation in a color space formula. For example, a pixel has pure red or pure blue, in YC_oC_g color space, its color will be accommodated 100 % (25 % in Y, 50 % in C_o and 25 % in C_g components), while in YC_bC_r, YD_bD_r and YC_gC_o-R family (YUV_1, YUV_2, YUV_3) color spaces respectively accommodate 97.775 %, 125 % and 175 %. Furthermore, for pure green, it will be recorded 100 %, 250 %, 150 % and 133.695 % respectively by YC_oC_g, YC_bC_r, YD_bD_r and YC_oC_g-R. Ideally to get good quality and compression ratio, the color conversion formula must still record 100 % of the dynamic value for each color component R, G and B. This ideal condition is only owned by YC_oC_g color space and it is proven that its performance is better than the three other color spaces. The second analysis concerns chrominance covered by the color space formula. For example, YC_bC_r color space tends to be dominant in color combinations with hues (for all chrominance and luminance): red, orange, yellow, green, green-yellow, cyan, and blue, whereas the YC_oC_g color space covers more color variations with hues: red, orange, green-yellow, green, green-cyan, blue, purple. This means, there are several hues that have not been well covered by each color space formula. The third analysis is the compatibility between the color space used and the dominant colors contained in an image. Based on the three thoughts above, this paper proposes a color dominant analysis algorithm and an adaptive color space model.

3. Proposed methods

The proposed methods consist of three parts: developing adaptive color space, analyzing dominant colors in an image to determine which color space will be used, and then the experimentations. The development of adaptive color space model refers to YC_oC_g color space which provides better quality and compression ratio in lossy image compression (see results in table 1). It has been described above that this color space has not evenly covered all existing chrominance. This weakness has an impact on quality and compression ratio: it is best suited for images having colors according to their characteristics and decreased for images having less suitable color.

3.1. Dominant colors analysis

As described in the example of twelve images (fig. 2), where each image has a different dominant color. Thus, there is a need to adapt the suitability of the color space used and the image to be compressed. For this reason, the dominant color analysis contained in the image must be carried out first. The colors information can be presented in the form of cylindrical coordinates hue-saturation as shown in fig. 3*a*. We propose to divide hue coordinates into 12 parts, so that 12 different hues will be formed with angle range: red = 0º – 15º and 345º – 360º, orange = 15º – 45º, yellow = 45º – 75º, green-yellow = 75º – 105º, green = 105º – 135º, green-cyan = 135º – 165º, cyan = 165º – 195º, blue-cyan = 195º – 225º, blue = 225º – 255º, blue-purple = 255º – 285º, purple = 285º – 315º, red-purple = 315º – 345º.

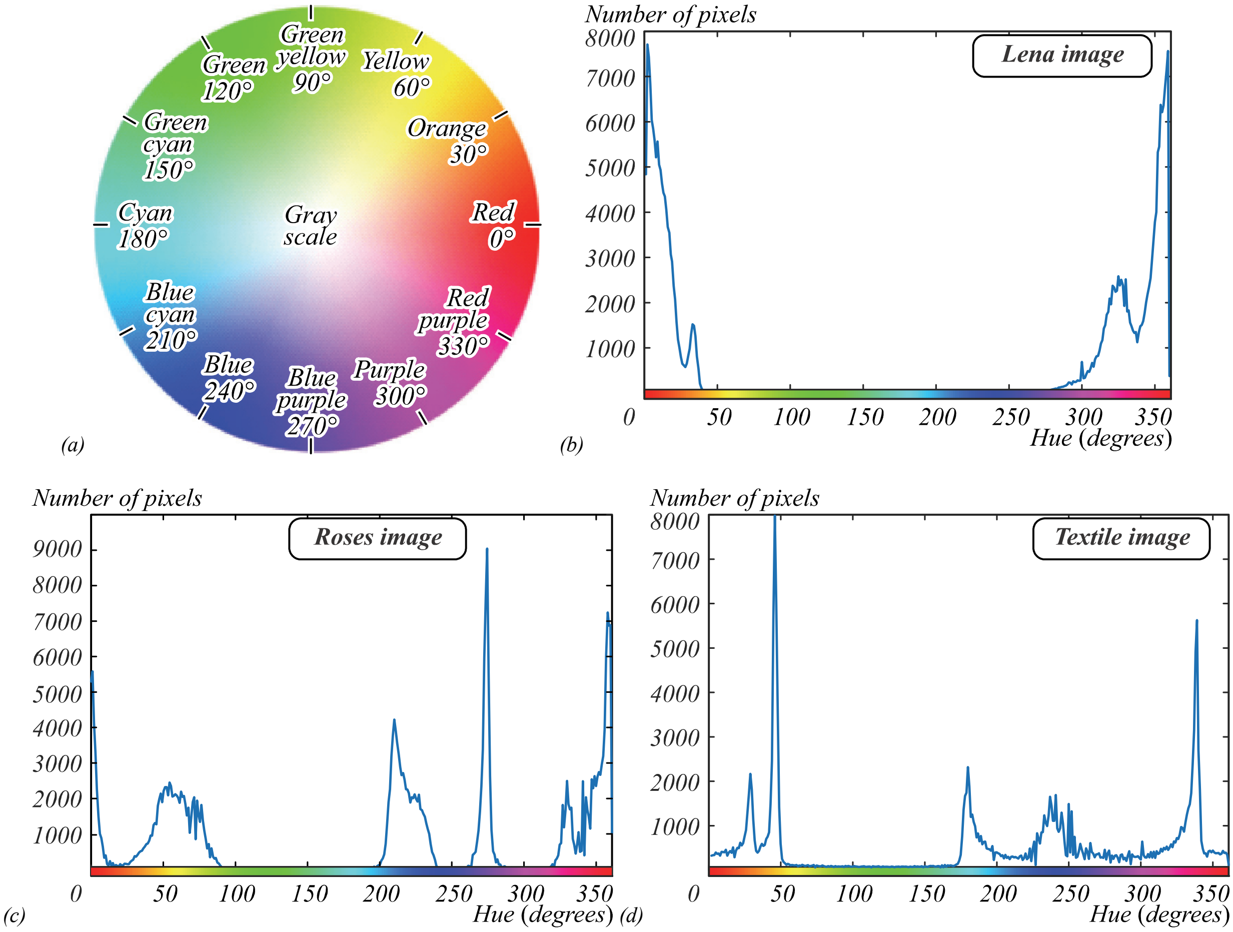

Fig. 3. (a) Cylindrical coordinates of hue; and hue histograms of (b) Lena, (c) Roses and (d) Textile images

The Hue of each pixel of an *RGB* image can be calculated using H (hue) component formula of HSL, HSV or HCL color spaces [31, 32]. From the hue value of each pixel, a histogram hue can be calculated. A Histogram hue is a distribution of the number of pixels having the same hue value in an image. Furthermore, this histogram hue can be used to determine what chrominance are dominants in an image. figs. 3*b* – *d* respectively show histogram hues for Lena, Roses, and Textile images. The dominant hue in Lena image is red, orange, and red-purple. These three hues are more in line with YC_oC_g that is why its compression ratio performance is 5.63 % better than YC_bC_r (see table 2). Roses image has dominant hues: red, red-purple, blue-cyan, blue-purple, yellow, white, and black. The majority of these hues are less compatible with YC_oC_g, this has an impact on decreasing its compression ratio performance to 1.27% lower than YC_bC_r. For Textile images, its dominant hues are yellow, orange, cyan, blue, red-purple. For this Textile image, YC_oC_g and YC_bC_r provide a relatively similar compression ratio.

The proposed algorithm for the dominant colors analysis process in an image is given in Algo-1. The syntax of Hue_histo(red), Hue_histo(cyan), and so on represent the number of pixels in an image that have the red hues (0° – 15° and 345° – 360°), cyan hues (165° – 195°), and so forth. The dominant colors are grouped into three parts according to the three color space models described in section 3.2. The first part (Model-1) is a combination of red, cyan, green-yellow, blue-purple, and gray-level colors. The second part (Model-2) is a combination of green, purple, orange, blue-cyan, and gray-level colors. The last part (Model-3) is a combination of blue, yellow, green-cyan, red-purple, and gray-level colors.

Algo-1. Dominant color analysis algorithm

```
Input  : RGB_Image
Output : Color_ Space
Hue_histo = Calculate_hue_histogram(RGB_Image)
DC_1 = Hue_histo(red) + Hue_histo( cyan) +
           Hue_histo(green-yellow) + Hue_histo(blue-purple)
DC_2 = Hue_histo(green) + Hue_histo(purple) +
           Hue_histo(orange) + Hue_histo( blue-cyan)
DC_3 = Hue_histo(blue) + Hue_histo(yellow) +
           Hue_histo(green-cyan) + Hue_histo(red-purple)
if (DC_1 ≥ DC_2) and (DC_1 ≥ DC_3)
    Color_Space = "Model_1"
else if (DC_2 ≥ DC_1) and (DC_2 ≥ DC_3)
    Color_Space = "Model_2"
else
    Color_Space = "Model_3"
end if
```

3.2. Proposed adaptive color space model

In this part, we adopted the permutation model proposed in [13] to produce two additional color spaces which are variants of YC_oC_g and to be a solution to its weaknesses. The two color spaces are shown in Eqs. (16, 17), and Eqs. (20, 21). Thus, the proposed adaptive color space model (ACSM) consists of three color spaces and can be said as the YC_oC_g color space family (*ACSM-*YC_oC_g). As a side note, if each of their all chrominance components is multiplied by 2, they will be identical to the YC_gC_o-R color space family (YUV_1, YUV_2 and YUV_3).

Model-1: YC_cC_r (luminance Y, chrominance cyan C_c and chrominance red C_r) with dominant hues (colors) coverage: red, cyan, green-yellow, blue-purple, and grey-level.

$$\begin{bmatrix} Y \\ C_c \\ C_r \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \ [A] = \begin{bmatrix} 0.50 & 0.25 & 0.25 \\ 0.0 & 0.50 & -0.50 \\ 0.50 & -0.25 & -0.25 \end{bmatrix}, \quad (16)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [A^{-1}] \begin{bmatrix} Y \\ C_c \\ C_r \end{bmatrix}, \ [A^{-1}] = \begin{bmatrix} 1.0 & 1.0 & -1.0 \\ 1.0 & -1.0 & -1.0 \\ 1.0 & 0.0 & 1.0 \end{bmatrix}. \quad (17)$$

Model-2: YC_pC_g (luminance Y, chrominance purple C_p and chrominance green C_g) is equal to YC_oC_g (Eq. 5) with dominant hues coverage: green, purple, orange, blue-cyan, and grey-level.

$$\begin{bmatrix} Y \\ C_p \\ C_g \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \ [A] = \begin{bmatrix} 0.25 & 0.50 & 0.25 \\ 0.50 & 0.0 & -0.50 \\ -0.25 & 0.50 & -0.25 \end{bmatrix}, \quad (18)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [A^{-1}] \begin{bmatrix} Y \\ C_p \\ C_g \end{bmatrix}, \ [A^{-1}] = \begin{bmatrix} 1.0 & 1.0 & -1.0 \\ 1.0 & 0.0 & 1.0 \\ 1.0 & -1.0 & -1.0 \end{bmatrix}. \quad (19)$$

Model-3: YC_yC_b (luminance Y, chrominance yellow C_y and chrominance blue C_b) with dominant hues coverage: blue, yellow, green-cyan, red-purple, and grey-level.

$$\begin{bmatrix} Y \\ C_y \\ C_b \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \ [A] = \begin{bmatrix} 0.25 & 0.25 & 0.50 \\ 0.50 & -0.50 & 0.0 \\ -0.25 & -0.25 & 0.5 \end{bmatrix}, \quad (20)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [A^{-1}] \begin{bmatrix} Y \\ C_y \\ C_b \end{bmatrix}, \ [A^{-1}] = \begin{bmatrix} 1.0 & 1.0 & -1.0 \\ 1.0 & -1.0 & -1.0 \\ 1.0 & 0.0 & 1.0 \end{bmatrix}. \quad (21)$$

The color image conversion algorithm from *RGB* space to ACSM and its inverse conversion from ACSM to *RGB* are presented by Algo-2 and Algo-3. The choice of color s pace (*Color_Space*), for the color image conversion process, depends on the dominant colors calculated using Algo-1. If the *Color_Space* = "*Model_1*", then the color space YC_cC_r will be used, or the color space YC_pC_g that will be selected if *Color_Space* = "*Model_2*", whereas if *Color_Space* = "*Model_3*", then the color space YC_yC_b will be used. The color image conversion process from *RGB_Image* to *ACSM_ Image* is done in the last line of the algorithm. This conversion process is carried out pixel by pixel. The Algo-3 is used for the inverse color conversion process from *ACSM_ Image* to *RGB_Image*.

Algo-2. ***RGB_Image* to *ACSM_ Image* conversion algorithm**

```
Input   : RGB_Image, Color_ Space
Output :  ACSM_ Image
if Color_ Space = "Model_1"          %% Color  model 1
    ACSM=[0.50   0.25    0.25
          0.00   0.50   -0.50
          0.50  -0.25   -0.25]
else if Color_ Space = "Model_2"     %% Color  model 2
    ACSM=[0.25   0.50    0.25
          0.50   0.00   -0.50
         -0.25   0.50   -0.25]
else
    ACSM=[0.25   0.25    0.50        %% Color model 3
          0.50  -0.50    0.00
         -0.25  -0.25    0.50]
end if
ACSM_ Image =  ACSM * RGB_Image
```

Algo-3. ***ACSM_ Image* to *RGB_Image* conversion algorithm**

```
Input   : ACSM_ Image, Color_ Space
Output :  RGB_Image
if Color_ Space = "Model_1"          %% Color  model 1
    iACSM = [1.0   0.0    1.0
             1.0   1.0   -1.0
             1.0  -1.0   -1.0]
else if Color_ Space = "Model_2"     %% Color  model 2
    iACSM = [1.0   1.0   -1.0
             1.0   0.0    1.0
             1.0  -1.0   -1.0]
else
    iACSM = [1.0   1.0   -1.0        %% Color model 3
             1.0  -1.0   -1.0
             1.0   0.0    1.0]
end if
RGB_Image =  iACSM * ACSM_ Image
```

4. Results and discussions

The proposed dominant color analysis and ACSM algorithms have been integrated into the JPEG lossy image compression algorithm. Besides that, we also implemented the ACSM algorithm using YC_gC_o-R color space family (YUV_1, YUV_2 and YUV_3). Both were tested using sixty images from [17, 18, 19, 20, 21]. The experimental results of the first twelve images are given in table 3 and the next forty-eight images are shown in tables 4 – 5. In table 3, there are eleven of the twelve images with the highest CR value offered by the ACSM-YC_oC_g color space family (last column), while the ACSM-YC_gC_o-R family only produces one image with the highest CR value (fourth column).

For Lena image, the adaptive color algorithm automatically selects and uses model-1 YC_cC_r color space, and obtained CR of 10.34 % better than YC_bC_r. Color space YUV_3 is also automatically selected when ACSM-YC_gC_o-R family is implemented. This color space is 4.362 % lower than YC_bC_r. Color space Model-1 is also selected for Roses and Ucid00295 images and their CR are better than YC_bC_r and ACSM-YC_gC_o-R family. An increase in CR is also seen in Peppers, Frymire, Tulips, and Textile images, where the color space chosen is model-3 YC_yC_b. Furthermore, for Baboon, Sails, Monarch, and Woman images, model-2 YC_pC_g color space is used, where the compression ratios results are also better than YC_bC_r and ACSM-YC_gC_o-R family. The higher CR for Yellow_Orchid image is given by YUV_3 of ACSM-YC_gC_o-R family. Its compression ratio is 12.77 % better than YC_bC_r and 2.607 % better than the ACSM-YC_oC_g family.

Furthermore, in tables 4 – 5, it can be observed that there are forty-one out of forty-eight images having the highest quality and compression ratio produced by the proposed ACSM-YC_oC_g family. YC_bC_r and ACSM-YC_gC_o-R family provide the highest quality and compression ratio of the other four and three images, respectively. On average, ACSM-YC_oC_g family can enhance the CR by 5,046 % for the first 12 images, while for the next 48 images the CR and compression quality increase 4,555 % and 0.303 dB comparatively to YC_bC_r. Conversely, there was a decrease in CR of 0.420 % for the first 12 images and a decrease in CR of 2.087 % and compression quantity of 0.439 dB for the next 48 images, when ACSM-YC_gC_o-R family was used.

Table 3. Compression ratio of YC_bC_r, YC_oC_g and adaptive color space: YC_cC_r, YC_pC_g, YC_yC_b

File Image (png)	**Quality PSNR (dB)**	**Compression Ratio (CR) Using Color Space:**		
		YC_bC_r	**ACSM** appled to:	
			YUV_1, YUV_2, YUV_3	YC_cC_r, YC_pC_g, YC_yC_b
Lena	36.117	9.490	9.076 (YUV_3)	**10.471** (YC_cC_r)
Baboon	32.805	3.554	3.543 (YUV_1)	**3.798** (YC_pC_g)
Peppers	34.057	6.161	6.419 (YUV_2)	**6.695** (YC_yC_b)
Frymire	34.227	3.696	3.597 (YUV_3)	**3.771** (YC_yC_b)
Roses	46.776	19.96	19.340 (YUV_3)	**20.145** (YC_cC_r)
Tulips	33.764	3.784	3.669 (YUV_2)	**3.926** (YC_yC_b)
Sails	37.456	7.451	7.220 (YUV_1)	**7.750** (YC_pC_g)
Monarch	39.788	12.288	11.834 (YUV_1)	**12.553** (YC_pC_g)
Textile	36.838	5.442	5.439 (YUV_2)	**5.650** (YC_yC_b)
Woman	38.011	10.156	9.062 (YUV_3)	**10.442** (YC_pC_g)
Yellow_Orc	41.050	21.398	**24.131** (YUV_2)	23.518 (YC_yC_b)
Ucid00295	36.614	7.642	7.228 (YUV_1)	**7.908** (YC_cC_r)
CR Average		9.252	9.213	9.719
Note: ↑ Increase ↓ Decrease			↓ 0.420%	↑ 5.046%

Furthermore, the adaptive color space performance test is performed using seven Photoshop's quantization matrices from the high Q_{12} to the low Q_6 compression quality (i.e. Q_{12}, Q_{11}, Q_{10}, Q_9, Q_8, Q_7, Q_6) [22, 30]. Tables 6 – 8 show the quality (PSNR) and compression ratio for Lena, Baboon and Textile images and graphically show by the curves in Figs. 4 – 6. The adaptive color space curves for the three images are above the YC_bC_r color space curve. This shows that the proposed adaptive model has a better performance.

Table 4. Compression ratio of YC_bC_r, YC_oC_g, ACSM-YC_gC_o-R (YUV_1, YUV_2, YUV_3) and ACSM-YC_oC_g (YC_cC_r, YC_pC_g, YC_yC_b)

Image file name	**YC_bC_r**		**YC_0C_g**	**YUV_1 YC_gC_0-R**	**YUV_2**	**YUV_2**	**ACSM: *YUV_1*, *YUV_2*, *YUV_3***	**ACSM: *YC_cC_r*, *YC_pC_g*, *YC_yC_b***
	PSNR (dB)	CR	CR	CR	CR	CR	CR	CR
5colors_544×544	38.611	7.468	**7.569**	7.309	7.322	7.385	7.385	**7.583**
408px-Killersudoku_color	41.071	**11.429**	11.205	10.756	10.756	10.868	10.868	**11.730**
73755	39.277	15.136	**15.140**	14.639	14.436	14.855	14.855	**15.297**
3975590069_7d5e05207e_o	45.767	37.588	**39.581**	38.840	39.484	40.538	40.538	**39.638**
article-0-0B9771B5000005...	44.027	17.167	**17.509**	16.092	16.354	16.093	16.354	**17.509**
Best-Science-Images-2007-I...	47.342	**20.269**	18.591	18.353	17.759	18.346	18.353	19.147
Best-Science-Images-2007-II...	40.302	10.604	10.245	**11.039**	10.268	10.883	**11.039**	10.775
bike_orig_1280x1600	36.740	8.383	**8.785**	7.435	7.754	7.683	7.754	**8.785**
butterfly_3	45.780	**19.965**	19.933	19.184	19.442	19.399	19.442	**20.062**
cafe_orig_1280x1600	34.084	4.365	**4.598**	4.200	4.239	4.154	4.239	**4.598**
computer-science-ultimate	38.722	**8.3158**	7.658	7.289	7.634	7.187	7.634	8.095
dc7b126a-20f9-4569-bfd4-ce...	39.482	9.139	**9.489**	8.671	8.541	8.564	8.671	**9.489**
F1_large	40.337	**10.828**	10.718	10.446	10.647	10.740	10.740	**11.056**
glas_coloured_6	41.043	17.102	**17.389**	17.084	17.344	17.020	17.344	**17.617**
p-radiologist-art1_1467422c	37.968	6.709	7.076	**7.125**	6.810	6.797	**7.125**	7.076
p01_orig_1280x1600	39.844	**11.145**	10.604	10.098	10.793	10.495	10.793	11.020
p06_orig_1280x1600	39.026	12.194	**12.303**	11.505	11.373	11.510	11.510	**12.303**
p30_orig_1280x1600	36.627	**9.296**	8.581	7.863	8.490	8.031	8.490	9.236
Screen-Searchmetri_968×576	40.886	14.033	**15.431**	14.640	13.719	13.979	14.640	**15.431**
Spectrscopic_mapping_speeds.	44.479	**11.834**	11.717	11.251	11.195	11.177	11.251	**11.878**
stadtplan-museum-o_880×600	35.788	5.641	**5.750**	5.228	5.319	5.108	5.319	**5.776**
stewart_tartan_2	39.212	7.33	**7.373**	6.689	6.842	6.881	6.881	**7.373**
surface_2	42.713	20.925	**22.584**	22.143	21.481	21.555	21.481	**22.584**
ucid00032	36.554	7.995	**8.564**	7.537	7.049	7.213	7.537	**8.564**
ucid00041	37.931	11.579	**12.815**	11.217	10.565	10.399	11.217	**12.815**
ucid00045	37.645	11.031	**12.140**	10.323	9.691	9.432	10.323	**12.140**
ucid00059	37.414	10.691	**11.495**	9.417	9.276	8.954	9.417	**11.495**
ucid00066	38.694	14.181	**15.383**	13.215	12.588	12.126	13.215	**15.383**
ucid00110	33.792	5.597	**5.962**	5.265	5.013	4.889	5.265	**5.962**
ucid00262	37.118	8.702	**8.710**	8.009	7.872	8.522	8.522	**9.235**
ucid00275	35.955	6.611	**6.876**	6.317	6.039	6.438	6.438	**6.905**
ucid00295	36.614	7.642	**7.908**	7.228	7.086	7.174	7.228	**7.908**
ucid00302	37.754	9.697	**10.531**	9.709	9.148	9.446	9.446	**10.531**
ucid00317	38.474	15.023	**16.341**	14.326	13.878	13.723	14.326	**16.341**
ucid00368	35.466	5.635	**6.255**	5.798	5.726	5.446	5.798	**6.255**
ucid00410	34.695	4.793	**5.008**	4.673	4.534	4.545	4.673	**5.066**
ucid00456	36.374	6.892	**7.193**	6.639	6.431	6.565	6.639	**7.193**
ucid00457	39.619	14.076	**14.367**	12.804	12.447	12.900	12.900	**14.367**
ucid00489	37.097	8.831	**8.946**	8.043	7.881	8.041	8.043	**8.946**
ucid00514	34.844	5.053	**5.055**	4.650	4.608	4.744	4.744	**5.172**
ucid00648	36.711	8.827	**9.440**	8.443	7.982	8.197	8.443	**9.440**
ucid00687	34.979	5.153	**5.444**	5.027	4.812	5.016	5.027	**5.444**
ucid00758	36.705	8.854	**9.563**	8.618	8.340	8.100	8.618	**9.563**
ucid00804	39.598	16.251	**19.805**	19.041	15.760	15.950	19.041	**19.805**
ucid00811	38.649	12.234	**12.820**	11.524	10.998	11.412	11.524	**12.820**
ucid00860	37.882	11.197	**11.251**	10.604	10.467	10.272	10.604	**11.251**
ucid00884	35.413	6.021	**6.439**	5.731	5.397	5.619	5.731	**6.439**
WFCAM_JHK_colour-comp...	42.822	14.177	14.819	14.718	**15.038**	14.194	**15.038**	14.819
CR average:		11.117	**11.520**	10.766	10.555	10.595	10.885	**11.623**
Note: ↑ Increase ↓ Decrease			↑3.626%	↓3.158%	↓5.056%	↓4.693%	↓2.087%	↑4.555%

Table 5. Compression quality of YC_bC_r, YC_oC_g, ACSM-YC_gC_o-R (YUV_1, YUV_2, YUV_3) and ACSM-YC_oC_g (YC_cC_r, YC_pC_g, YC_yC_b)

Image file name	**YC_bC_r**		**YC_gC_0**	**YUV_1**	**YC_gC_0-R YUV_2**	**YUV_3**	**ACSM: YUV_1, YUV_2, YUV_3**	**ACSM: YC_cC_r, YC_pC_g, YC_yC_b**
	CR	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
5colors_544×544	7.468	38.611	38.875	38.084	38.131	38.329	38.329	38.944
408px-Killersudoku_color	11.429	41.071	40.697	39.511	39.511	39.857	39.857	41.530
73755	15.136	39.277	39.278	39.165	39.112	39.225	39.225	39.308
3975590069_7d5e05207e_o	37.588	45.767	45.927	45.859	45.918	45.961	45.961	45.927
article-0-0B9771B5000005...	17.167	44.027	44.338	42.735	43.074	42.736	43.074	44.344
Best-Science-Images-2007-I...	20.269	47.342	41.826	43.761	42.977	43.719	43.761	45.231
Best-Science-Images-2007-II...	10.604	40.302	39.729	40.836	39.814	40.673	40.836	40.579
bike_orig_1280×1600	8.383	36.740	37.053	35.866	36.123	36.125	36.125	37.053
butterfly_3	19.965	45.780	45.762	45.330	45.465	45.452	45.465	45.833
cafe_orig_1280×1600	4.365	34.084	34.559	33.592	33.716	33.439	33.716	34.559
computer-science-ultimate	8.3158	38.722	36.989	36.445	37.058	35.884	37.058	38.254
dc7b126a-20f9-4569-bfd4-ce...	9.139	39.482	39.982	38.640	38.386	38.456	38.640	39.982
F1_large	10.828	40.337	40.210	39.788	40.086	40.216	40.216	40.581
glas_coloured_6	17.102	41.043	41.097	41.039	41.091	41.024	41.091	41.144
p-radiologist-art1_1467422c	6.709	37.968	38.448	38.472	38.141	38.128	38.128	38.448
p01_orig_1280×1600	11.145	39.844	39.486	39.041	39.581	39.361	39.581	39.760
p06_orig_1280×1600	12.194	39.026	39.090	38.530	38.443	38.526	38.530	39.090
p30_orig_1280×1600	9.296	36.627	36.132	35.202	35.761	35.423	35.761	36.583
Screen-Searchmetri_968×576	14.033	40.886	42.320	41.709	40.424	40.809	41.709	42.320
Spectrscopic_mapping_speed...	11.834	44.479	44.400	43.772	43.592	43.617	43.772	44.507
stadtplan-museum-o_880×600	5.641	35.788	36.008	34.766	34.961	34.437	34.961	36.062
stewart_tartan_2	7.33	39.212	39.350	36.897	37.722	37.746	37.746	39.350
surface_2	20.925	42.713	42.987	42.930	42.812	42.827	42.930	42.987
ucid00032	7.995	36.554	37.073	36.052	35.464	35.636	36.052	37.073
ucid00041	11.579	37.931	38.482	37.756	37.446	37.321	37.756	38.482
ucid00045	11.031	37.645	38.137	37.302	36.995	36.791	37.302	38.137
ucid00059	10.691	37.414	37.810	36.621	36.649	36.387	36.649	37.810
ucid00066	14.181	38.694	39.014	38.419	38.269	38.069	38.419	39.014
ucid00110	5.597	33.792	34.105	33.182	32.668	32.334	33.182	34.105
ucid00262	8.702	37.118	37.125	36.379	36.243	36.949	36.949	37.547
ucid00275	6.611	35.955	36.279	35.530	35.052	35.715	35.715	36.535
ucid00295	7.642	36.614	36.919	36.041	35.861	35.995	36.041	36.921
ucid00302	9.697	37.754	38.344	37.763	37.287	37.555	37.763	38.344
ucid00317	15.023	38.474	38.764	38.296	38.197	38.131	38.296	38.764
ucid00368	5.635	35.466	36.443	35.783	35.642	35.081	35.783	36.443
ucid00410	4.793	34.695	35.170	34.358	33.859	33.975	33.975	35.265
ucid00456	6.892	36.374	36.770	35.979	35.666	35.879	35.979	36.770
ucid00457	14.076	39.619	39.765	38.984	38.825	39.045	39.045	39.765
ucid00489	8.831	37.097	37.195	36.332	36.093	36.318	36.332	37.195
ucid00514	5.053	34.844	34.847	33.866	33.750	34.113	34.113	35.050
ucid00648	8.827	36.711	37.176	36.421	36.000	36.212	36.421	37.176
ucid00687	5.153	34.979	35.527	34.682	34.132	34.651	34.651	35.527
ucid00758	8.854	36.705	37.260	36.514	36.294	36.068	36.514	37.260
ucid00804	16.251	39.598	40.755	40.465	39.417	39.487	39.487	40.755
ucid00811	12.234	38.649	38.962	38.231	37.919	38.178	38.231	38.962
ucid00860	11.197	37.882	37.920	37.471	37.382	37.249	37.471	37.920
ucid00884	6.021	35.413	36.007	34.897	34.266	34.707	34.897	36.007
WFCAM_JHK_colour-comp...	14.177	42.822	43.251	43.290	43.371	42.836	43.371	43.251
PSNR average (dB):		**38.707**	**38.826**	**38.179**	**38.013**	**38.055**	**38.268**	**39.009**
Note: ↑ Increase ↓ Decrease (dB):			↑ 0.119	↓ 0.528	↓ 0.693	↓ 0.652	↓ 0.439	↑ 0.303

Table 6. PSNR and CR of YC_bC_r and ACSM for Lena image

Matrix Quality	YC_bC_r		ACSM: YC_cC_r, YC_pC_g, YC_yC_b	
	PSNR (dB)	CR	PSNR (dB)	CR
Q_6	34.545	13.027	34.691	13.719
Q_7	35.365	11.504	35.509	12.072
Q_8	35.396	11.170	35.550	11.757
Q_9	36.117	9.490	36.275	9.957
Q_{10}	36.934	7.457	37.105	7.789
Q_{11}	39.111	4.512	39.386	4.621
Q_{12}	46.143	2.401	46.564	2.429

Table 7. PSNR and CR of YC_bC_r and ACSM for Baboon image

Matrix Quality	YC_bC_r		ACSM: YC_cC_r, YC_pC_g, YC_yC_b	
	PSNR (dB)	CR	PSNR (dB)	CR
Q_6	30.914	4.152	31.452	4.180
Q_7	32.050	3.872	32.509	3.906
Q_8	31.912	3.857	32.492	3.883
Q_9	32.805	3.554	33.403	3.582
Q_{10}	33.814	3.144	34.417	3.17
Q_{11}	37.696	2.328	38.262	2.347
Q_{12}	46.102	1.534	46.567	1.545

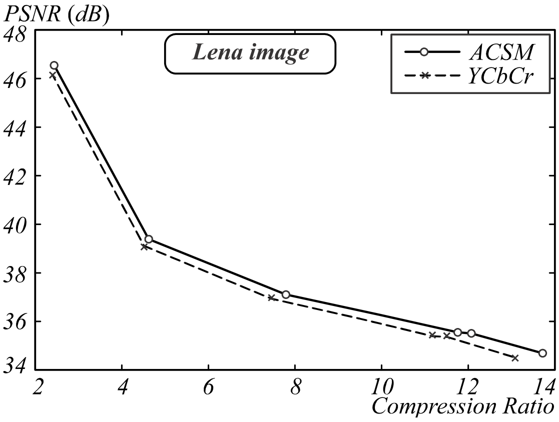

Fig. 4. PSNR vs CR curves of YC_bC_r and ACSM for Lena image

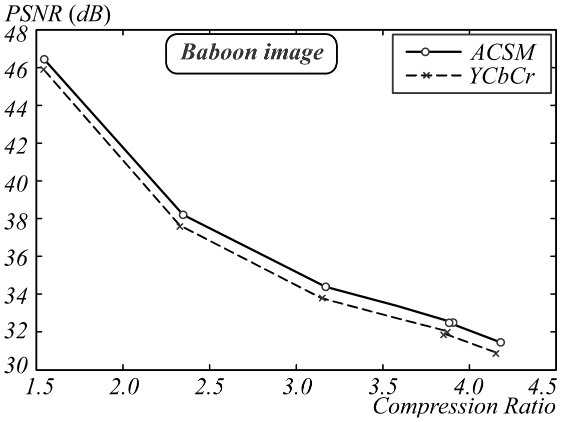

Fig. 5. PSNR vs CR curves of YC_bC_r and ACSM for Baboon image

Table 8. PSNR and CR of YC_bC_r and ACSM for Textile image

Matrix Quality	YC_bC_r		ACSM: YC_cC_r, YC_pC_g, YC_yC_b	
	PSNR (dB)	CR	PSNR (dB)	CR
Q_6	33.246	7.048	33.461	7.168
Q_7	35.442	6.208	35.622	6.303
Q_8	34.961	6.236	35.219	6.324
Q_9	36.838	5.442	37.140	5.537
Q_{10}	38.873	4.654	39.075	4.717
Q_{11}	42.191	3.572	42.411	3.60
Q_{12}	48.269	2.453	48.4971	2.426

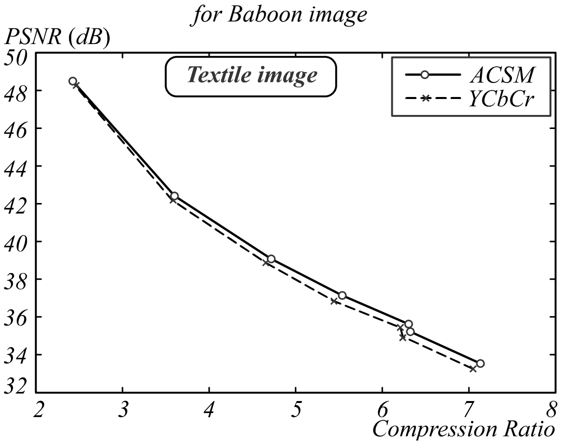

Fig. 6. PSNR vs CR curves of YC_bC_r and ACSM for Textile image

Besides its better performance, the implementation of the proposed adaptive color space in the form of an electronic circuit is very simple and only requires adders and subtractors. It is as simple as the electronic implementation of YC_gC_o-R color space family [13]. Suppose each color data R, G and B of an image is encoded with eight bits, that is: $R = r_7, ..., r_0$; $G = g_7, ..., g_0$; and $B = b_7, ..., b_0$. Arithmetic operations for these data are carried out in binary operations: addition, subtraction, and logic. The symbol "»1", is a one-bit shift-right operation, used instead of multiply by 0.5. For example, the Eq. (10) can be simplified as:

$$Y = 0.5R + 0.25G + 0.25B \rightarrow Y = (R + ((G + B)\gg 1))\gg 1,$$
$$\mathrm{C_c} = 0.5\mathrm{G} - 0.5\mathrm{B} \rightarrow \mathrm{C_c} = (\mathrm{G}\text{-}\mathrm{B})\gg 1,$$
$$\mathrm{C_r} = 0.5\mathrm{R} - 0.25\mathrm{G} - 0.25\mathrm{B} \rightarrow \mathrm{C_r} = (\mathrm{R} - ((\mathrm{G} + \mathrm{B})\gg 1))\gg 1.$$

Referring to this simplification model, the electronic schematic of RGB to YC_cC_r conversion can be created as shown in Fig. 7. This scheme only requires 2 adders and 2 subtractors. The luminance Y component requires 10 bits, consisting of 8 bits for integer ($s_7, ..., s_0$) value and 2 bits for fractional (s_{-1}, s_{-2}) value. While the cyan chrominance C_c component needs 9 bits, that are 1 bit of borrow-out (s_7 as sign bit: "0" positive or "1" negative), 7 bits integer value ($s_6, ..., s_0$), and 1 bit fractional (s_{-1}). Furthermore, the red chrominance C_r component takes 10 bits consisting of 1 bit borrow-out (s_7 as sign bit), 7 bits inte ger ($s_6, ..., s_0$), and 2 bits (s_{-1}, s_{-2}) fractional.

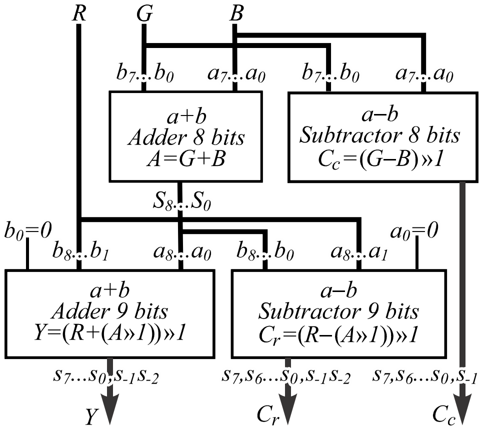

Fig. 7. Electronic schematic of RGB to YC_cC_r color conversion

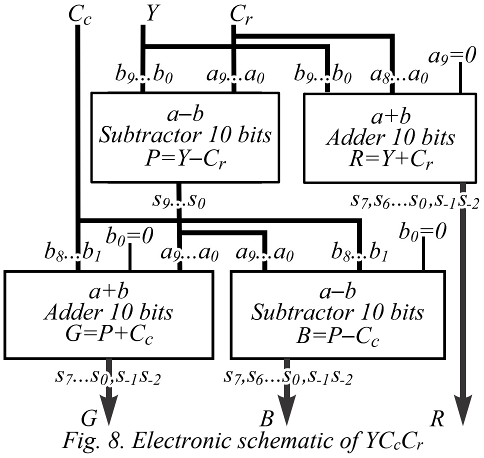

Fig. 8. Electronic schematic of YC_cC_r to RGB color conversion

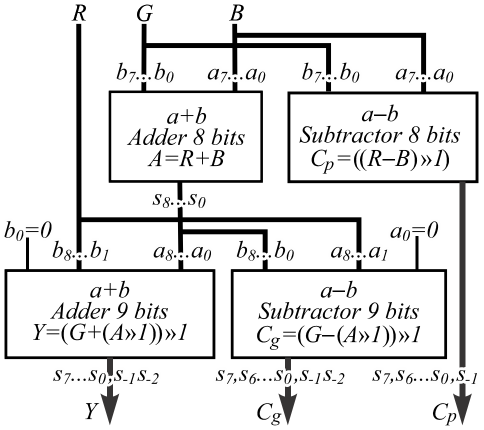

Fig. 9. Electronic schematic of RGB to YC_pC_g color conversion

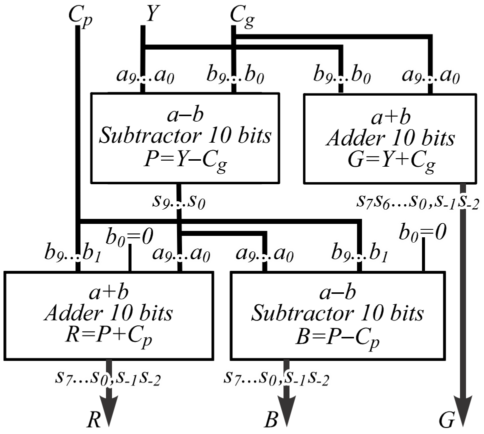

Fig. 10. Electronic schematic of YC_pC_g to RGB color conversion

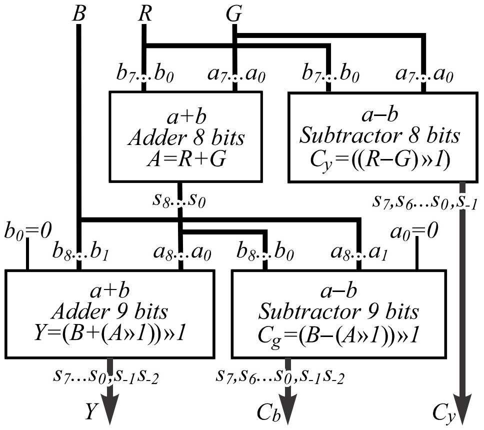

Fig. 11. Electronic schematic of RGB to YC_yC_b color conversion

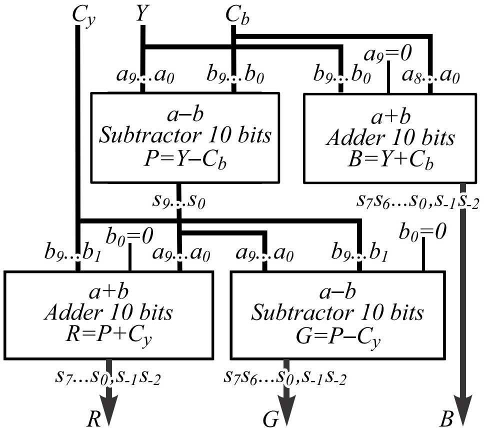

Fig. 12. Electronic schematic of YC_yC_b to RGB color conversion

The inverse conversion process from YC_cC_r to RGB in Eq. (11): $R=Y+C_c$,

$G=(Y-C_r)+C_c$,

$B=(Y-C_r)-C_c$.

The electronic schematic of this equation is given by fig. 8. It can be seen that this inverse conversion only needs 2 adders and 2 subtractors.

In the same ways, the color model-2 from RGB to YC_pC_g can be simplified as:

$$Y=0.25R+0.5G+0.25B \rightarrow Y=(G+((R+B)\gg 1))\gg 1,$$

$$\mathrm{C_p}=0.5\mathrm{R}-0.5\mathrm{B} \rightarrow \mathrm{C_p}=(\mathrm{R}-\mathrm{B})\gg 1,$$

$$\mathrm{C_g}=-0.25\mathrm{R}+0.5\mathrm{G}-0.25\mathrm{B} \rightarrow \mathrm{C_g}=(\mathrm{G}-((\mathrm{R}+\mathrm{B})\gg 1))\gg 1.$$

and its inverse conversion from YC_pC_g to RGB is:

$$\mathrm{R}=(\mathrm{Y}-\mathrm{C_g})+\mathrm{C_p},$$

$$\mathrm{G}=(\mathrm{Y}+\mathrm{C_g}),$$

$$\mathrm{B}=(\mathrm{Y}-\mathrm{C_g})-\mathrm{C_p}.$$

The electronic schematics diagrams of both *RGB* to *YC_pC_g*

Similarly for the color model-3 *RGB* to YC_yC_b is simplified as:

$$\begin{aligned}
&Y = 0.25R + 0.25G + 0.5B \rightarrow Y = (B + ((R + G) \gg 1)) \gg 1,\\
&C_y = 0.5R - 0.5G \rightarrow C_y = (R - G) \gg 1,\\
&C_b = -0.25R - 0.25G + 0.5B \rightarrow C_b = (B - ((R + G) \gg 1)) \gg 1.
\end{aligned}$$

and its inverse conversion from YC_yC_b to *RGB* is:

$$\begin{aligned}
&R = (Y - C_b) + C_y,\\
&G = (Y - C_b) - C_y,\\
&B = (Y + C_b).
\end{aligned}$$

The electronic schematics diagrams of both *RGB* to YC_yC_b and YC_yC_b to *RGB* colors conversion are sequentially presented in figs. 11 – 12.

5. Conclusion

In this paper, three color spaces YC_cC_r, YC_pC_g, and YC_yC_b (YC_oC_g color space family) have been developed that can be adapted according to the dominant colors contained in an image. The color analysis algorithm can calculate and determine the dominant colors in an image, and then automatically determines the appropriate color space to be applied in that image. The experimental results using sixty test images, which have varying colors, shapes, and textures, indicate that the proposed adaptive color spaces model provides improved performance of 3 % to 10 % better than other color spaces. Likewise, their electronic schematics require only two adders and two subtractors, both for forward and inverse conversions.

Our future research is to apply the dominant color analysis and adaptive color space model (ACSM) algorithm to the JPEG2000 image compression algorithm, and MPEG for video compression.

Reference

[1] Wallace GK. The JPEG still picture compression standard. IEEE Trans Consum Electron 1992; 38(1): 18-34.

[2] Skodras A, Christopoulos C, Ebrahimi T. The JPEG 2000 still image compression standard. IEEE Signal Process Mag 2001; 18(5): 36-58.

[3] Saptariani T, Madenda S, Ernastuti, Silfianti W. Accelerating compression time of the standard JPEG by employing the quantized YCbCr color space algorithm. Int J Electr Comput Eng 2018; 8(6): 4343-4351.

[4] Rao KR, Hwang JJ. Techniques and standards for image, video and audio coding. Englewood Cliffs, NJ: Prentice-Hall; 1996.

[5] Bhaskaran V, Konstantinides K. Image and video compression standards: Algorithms and applications. 2nd ed. Norwell, MA: Kluwer; 1997.

[6] Progressive lossy to lossless core experiment with a region of interest: Results with the S, S+P, two-ten integer wavelets and with the difference coding method. ISO/IEC JTC1/SC29/WG1 N741, March 1998.

[7] Nadenau MJ, Reichel J. Opponent color, human vision and wavelets for image compression. Proc 7th Color Imaging Conf 1999; 237-242.

[8] Taubman DS, Marcellin MW. JPEG2000 image compression fundamentals, standards and practice. Kluwer Academic Publishers; 2002.

[9] Pasteau F, Strauss C, Babel M, Déforges O, Bédat L. Improved colour decorrelation for lossless colour image compression using the LAR codec. European Signal Processing Conference (EUSIPCO'09) 2009; 1-4.

[10] Malvar HS, Sullivan GJ. Transform, scaling & color space impact of professional extensions. ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 Document JVT-H031 May 2003.

[11] Malvar HS, Sullivan GJ. YCoCg-R: A color space with *RGB* reversibility and low dynamic range, joint video team (JVT) of ISO/IEC MPEG & ITU-T VCEG, (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), JVT PExt Ad Hoc Group Meeting 22-24 July 2003.

[12] Malvar HS, Sullivan GJ, Srinivasan S. Lifting-based reversible color transformations for image compression. Proc SPIE 2008; 7073: 707307.

[13] Strutz, T. Multiplierless reversible colour transforms and their automatic selection for image data compression. IEEE Trans Circuits Syst Video Technol 2013; 23(7): 1249-1259. DOI: 10.1109/TCSVT.2013.2242612.

[14] Marpe D, Kirchhoffer H, George V, Kauff P, Wiegand T. An adaptive color transform approach and its application in 4:4:4 video coding. Proc EUSIPCO 2006; 1-5.

[15] Strutz T, Leipnitz A. Adaptive colour-space selection in high efficiency video coding. 2017 25th European Signal Processing Conference (EUSIPCO) 2017; 1534-1538. DOI: 10.23919/EUSIPCO.2017.8081466.

[16] Schaefer G, Stich M. UCID – An uncompressed colour image database. Proc SPIE 2004; 5307: 472-480.

[17] Index of /strutz/Papers/Testimages. Source: ⟨https://www1.hft-leipzig.de/strutz/Papers/Testimages/⟩.

[18] Resources of ACSS. Source: ⟨https://www1.hft-leipzig.de/strutz/Papers/ACSS-resources/⟩.

[19] Index of /strutz/Papers/Testimages. Source: ⟨http://jasoncantarella.com/downloads/ucid.v2.tar.gz⟩.

[20] Public-domain test images for homeworks and projects. Source: ⟨https://homepages.cae.wisc.edu/~ece533/images/⟩.

[21] SeedArea. Rose – seeds (Mix-color). Source: ⟨http://www.seedarea.com/rose-seeds/136-rose-seeds-mix-color.html⟩.

[22] ImpulseAdventure. JPEG compression quality from quantization tables. Source: ⟨http://www.impulseadventure.com/photo/jpeg-quantization.html⟩.

[23] Winkler S, van den Branden Lambrecht CJ, Kunt M. Vision and video: Models and applications. In Book: van den Branden Lambrecht ChJ, ed. Vision models and applications to image and video processing. Boston: Springer; 2001: 201-229.

[24] Poynton Ch. Digital video and HDTV: Algorithms and interfaces. US Morgan Kaufmann Publishers; 2003.

[25] Uhrina M, Bienik J, Mizdos T. Chroma subsampling influence on the perceived video quality for compressed sequences in high resolutions. Adv Electr Electron Eng 2017; 15(4): 692-700.

[26] Ahmed N, Natarajan T, Rao KR. Discrete cosine transform. IEEE Trans Comput 1974; C-23(1): 90-93.

[27] Narasinha NJ, Peterson AM. On the computation of the discrete cosine transform. IEEE Trans Commun 1978; COM-26(6): 966-968.

[28] Lee BG. A new algorithm to compute the discrete cosine transform. IEEE Trans Acoust Speech Signal Process 1984; ASSP-32(6): 1243-1245.

[29] Mandyam GD, Ahmed NU, Magotra N. DCT-based scheme for lossless image compression. Proc SPIE 1995; 2419: 474-478. DOI: 10.1117/12.206386.

[30] Madenda S, Pengolahan citra dan video digital: Teori, algoritma dan pemrograman Matlab. Jakarta: Erlangga, 2005.

[31] Gonzalez RC, Woods RE. Digital image processing. 2nd ed. Prentice Hall; 2002.

[32] Madenda S, Missaoui R. A new perceptually uniform color space with associated color similarity measure for content-based image and video retrieval. Proceedings of Multimedia Information Retrieval Workshop, 28th annual ACM Sigir Conference 2005; 1-8.

Authors' information

Sarifuddin Madenda is a researcher and professor at Gunadarma University, Indonesia. He was born in Raha, Indonesia 7th April 1963. He finished his: B.S degree in Physic Instrumentation at University of Indonesia in 1989; M. S. degree in Electronics from INSA-Lyon, France in 1992; and Ph.D degree in Electronics and Image Processing from Universite de Bourgogne, France in 1995. He is also lecturer at UQO, Quebec - Canada since 2003. Actually, he is Director of Doctoral Program in Information Technology. His research interests are focused on image and video processing, multimedia data compression, content based image and video retrieval, steganography (encryption, decryption, coding) and decoding of multimedia secret documents, real time system architecture (FPGA and ASIC design).
E-mail: *sarif@staff.gunadarma.ac.id* .

Astie Darmayantie was born in Jakarta, Indonesia 27th March 1990. She acquires her Bachelor Degree in Computer Science 2011. She finished her master in 2013, in which brings her to be entitled as Master in Information System Management from Universitas Gunadarma and Master in Computer Vision and Robotics from Universite De Bourgogne. She completed her doctoral degree in information technology in 2018. Her research interests vary from machine learning to health informatics and socio-technical aspects in technology. E-mail: *astie@staff.gunadarma.ac.id* .

Received July 2, 2020. The final version – February 16, 2021.