J Inf Process Syst, Vol.13, No.1, pp.114~127, February 2017
https://doi.org/10.3745/JIPS.02.0053

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Color Image Coding Based on Shape-Adaptive All Phase Biorthogonal Transform

Xiaoyan Wang*, Chengyou Wang*, Xiao Zhou*, and Zhiqiang Yang*,**

Abstract

This paper proposes a color image coding algorithm based on shape-adaptive all phase biorthogonal transform (SA-APBT). This algorithm is implemented through four procedures: color space conversion, image segmentation, shape coding, and texture coding. Region-of-interest (ROI) and background area are obtained by image segmentation. Shape coding uses chain code. The texture coding of the ROI is prior to the background area. SA-APBT and uniform quantization are adopted in texture coding. Compared with the color image coding algorithm based on shape-adaptive discrete cosine transform (SA-DCT) at the same bit rates, experimental results on test color images reveal that the objective quality and subjective effects of the reconstructed images using the proposed algorithm are better, especially at low bit rates. Moreover, the complexity of the proposed algorithm is reduced because of uniform quantization.

Keywords

Color Image Coding, Shape-Adaptive All Phase Biorthogonal Transform (SA-APBT), Color Space Conversion, Chain Code

1. Introduction

In conventional block-based image coding, an image is uniformly partitioned into many $N \times N$ rectangular blocks. These rectangular blocks are then encoded in turn. Because discrete cosine transform (DCT) has many advantages in terms of energy compaction and coefficient decorrelation [1], it has been widely used in image and video compression standards, such as H.264/AVC [2,3] and HEVC/H.265 [4,5]. Many block-based image coding methods use DCT in the transform step. However, image coding based on DCT also has some disadvantages such as the complex quantization table and serious blocking effects at low bit rates. To solve the above problems, in 2009, Hou et al. [6] proposed the all phase biorthogonal transform (APBT) based on Walsh-Hadamard transform (WHT), DCT, and inverse DCT (IDCT) and used it in JPEG-like image coding. In 2014, Wang et al. [7] applied APBT to simple profile MPEG-4 video coding in which I frame was transformed by APBT and uniform quantization step was used to the coefficients subsequently. In the above work, the blocking effects of the reconstructed images and videos using the algorithm based on APBT are greatly reduced compared

Manuscript received October 16, 2015; first revision February 1, 2016; second revision September 26, 2016; third revision November 11, 2016; accepted December 3, 2016.
Corresponding Author: Chengyou Wang (wangchengyou@sdu.edu.cn)
* School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China (swwxy00800313@163.com, wangchengyou@sdu.edu.cn, zhouxiao@sdu.edu.cn)
** Integrated Electronic Systems Lab Co. Ltd., Jinan, China (yangzhiqiang@ieslab.cn)

with the one based on DCT. To improve the speed of image compression, Wang et al. [8] proposed the parallel APBT-JPEG algorithm based on GPU in 2015.

Although block-based image coding method is simple, efficient, and robust, its disadvantage is the bad subjective effects of the reconstructed images, especially in very low bit rate applications [9]. To solve this problem, region-based image coding has been widely developed in the past few years.

Region-based image coding is a flexible coding approach. Firstly, an image is segmented into region-of-interest (ROI) and background area. Then, to make full use of the visual characteristics of human eyes, the ROI is encoded with a low compression ratio and the background area is encoded with a high compression ratio. In this way, the final reconstructed images can obtain good subjective effects. Shape-adaptive DCT (SA-DCT) [10] is a transform method for arbitrarily shaped image segmentation coding and has been widely used in region-based image coding [11]. However, the disadvantage of the region-based image coding using SA-DCT is the serious blocking effects at low bit rates. To solve this problem, this paper proposes a color image coding algorithm based on shape-adaptive APBT (SA-APBT) [12]. This algorithm is implemented with four procedures: color space conversion, image segmentation, shape coding, and texture coding. In texture coding, SA-APBT is used. Experimental results show that compared with the color image coding algorithm based on SA-DCT, the objective and subjective effects of the reconstructed color images using the proposed algorithm are better, especially at low bit rates.

The rest of this paper is organized as follows. Section 2 briefly introduces the color space of the color image, including the RGB color space and the YCbCr color space. The transform process of SA-APBT is explained in Section 3. Then, in Section 4, a color image coding algorithm based on SA-APBT is proposed. The overall algorithm description of region-based color image coding is firstly given. Thereafter, the shape coding and texture coding based on SA-APBT are described in detail. Experimental results and comparisons between the color image coding algorithm based on SA-DCT and the proposed algorithm based on SA-APBT are given in Section 5. In addition, some reconstructed color images are presented. Conclusions and further work are given in Section 6.

2. Color Space

The color space determines the representation method of brightness and chromaticity. There are two typical color spaces: RGB and YCbCr. In the RGB color space, the three components R, G, and B are evenly distributed, which means each pixel has the same number of bits of the three components. The YCbCr color space is composed of three components Y, Cb, and Cr, where Y is the luminance component while Cb and Cr are color difference components. The human visual system (HVS) is less sensitive to chromaticity than to brightness. Based on this visual characteristic, the YCbCr color space completely retains the luminance component and samples the color difference components to represent the color image more efficiently. There are three sampling methods, 4:4:4, 4:2:2, and 4:2:0, as shown in Fig. 1. We suppose each component is represented with 8 bits. If 4:4:4 sampling is used, on average, 24 bits for each pixel are required. If 4:2:2 sampling is used, on average, 16 bits for each pixel are required. If 4:2:0 sampling is used, on average, 12 bits for each pixel are required. Therefore, three sampling methods are applicable to situations with different quality grades. For example, 4:2:0 sampling is widely used for video conferencing, in which real-time demand is high but the resolution requirement is not high.

Fig. 1. Three sampling methods in the YCbCr color space.

Color space conversion is performed according to specific requirements. The mathematical expression of color space conversion from RGB to YCbCr is

$$\begin{aligned} &\mathrm{Y} = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B}, \\ &\mathrm{Cb} = -0.168\mathrm{R} - 0.3313\mathrm{G} + 0.5\mathrm{B} + 128, \\ &\mathrm{Cr} = 0.5\mathrm{R} - 0.4187\mathrm{G} - 0.0813\mathrm{B} + 128. \end{aligned} \tag{1}$$

On the contrary, the mathematical expression of color space conversion from YCbCr to RGB is

$$\begin{aligned} &\mathrm{R} = \mathrm{Y} + 1.402(\mathrm{Cr} - 128), \\ &\mathrm{G} = \mathrm{Y} - 0.34414(\mathrm{Cb} - 128) - 0.71414(\mathrm{Cr} - 128), \\ &\mathrm{B} = \mathrm{Y} + 1.772(\mathrm{Cb} - 128). \end{aligned} \tag{2}$$

3. SA-APBT

Based on the theory of all phase digital filtering, three kinds of all phase biorthogonal transforms based on WHT, DCT, and IDCT were proposed and the matrices of APBT were deduced in [6]. APBT can be used in block-based image coding to transform the image from the spatial domain to the frequency domain, but it is not applicable to region-based image coding.

Taking the all phase inverse discrete cosine biorthogonal transform (APIDCBT) as an example, the process of two-dimensional APBT is introduced as follows. Let $\boldsymbol{X}$ and $\boldsymbol{E}$ represent an image block and an APIDCBT matrix with size of $N \times N$, respectively. After two-dimensional APIDCBT transform, transform coefficient block $\boldsymbol{Y}$ can be denoted by

$$\boldsymbol{Y} = \boldsymbol{E}\boldsymbol{X}\boldsymbol{E}^{\mathrm{T}}, \tag{3}$$

$$\boldsymbol{E}(i,j) = \begin{cases} \dfrac{1}{N}, & i = 0,\ j = 0,1,\cdots,N-1, \\ \dfrac{N - i + \sqrt{2} - 1}{N^2} \cos\dfrac{i(2j+1)\pi}{2N}, & i = 1,2,\cdots,N-1,\ j = 0,1,\cdots,N-1. \end{cases} \tag{4}$$

where $\boldsymbol{E}^{\mathrm{T}}$ is the transpose matrix of $\boldsymbol{E}$. We use

$$\boldsymbol{X} = \boldsymbol{E}^{-1}\boldsymbol{Y}(\boldsymbol{E}^{-1})^{\mathrm{T}}, \tag{5}$$

to reconstruct the image, where $\boldsymbol{E}^{-1}$ is the inverse matrix of $\boldsymbol{E}$.

Similar to SA-DCT [11], SA-APBT can be used in region-based gray image coding [12] and MPEG-4 video compression [13]. There are four procedures for SA-APBT: vertically shift the pixels to be transformed to the uppermost position, one-dimensional variable-length APBT in the vertical direction, horizontally shift the pixels to the left border, and one-dimensional variable-length APBT in the horizontal direction.

Taking shape-adaptive APIDCBT (SA-APIDCBT) for example, the mathematical expression of one-dimensional variable-length APIDCBT in the vertical direction is

$$\boldsymbol{a}_j = \boldsymbol{E}_{n_j} \boldsymbol{x}_j, \tag{6}$$

where j is the column number; $\boldsymbol{x}_j$ and $\boldsymbol{a}_j$ are the column vectors before and after the one-dimensional APIDCBT in the vertical direction, respectively; n_j is the length of the column vector; and $\boldsymbol{E}_{n_j}$ is the APIDCBT matrix with size of $n_j \times n_j$. The mathematical expression of one-dimensional variable-length APIDCBT in the horizontal direction is

$$\boldsymbol{c}_i = \boldsymbol{E}_{m_i} \boldsymbol{b}_i, \tag{7}$$

where i is the row number; $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ are the row vectors before and after the one-dimensional APIDCBT in the horizontal direction, respectively; m_i is the length of the row vector; and $\boldsymbol{E}_{m_i}$ is the APIDCBT matrix with size of $m_i \times m_i$.

On the contrary, during inverse SA-APIDCBT process, the mathematical expressions of one-dimensional variable-length inverse APIDCBT in the horizontal and vertical directions are

$$\boldsymbol{b}_i^* = \boldsymbol{E}_{m_i}^{-1} \boldsymbol{c}_i^*, \tag{8}$$

$$\boldsymbol{x}_j^* = \boldsymbol{E}_{n_j}^{-1} \boldsymbol{a}_j^*, \tag{9}$$

where the stars indicate the resulting data after the SA-APIDCBT coefficients are quantized and dequantized; $\boldsymbol{E}_{m_i}^{-1}$ is the inverse matrix of $\boldsymbol{E}_{m_i}$; and $\boldsymbol{E}_{n_j}^{-1}$ is the inverse matrix of $\boldsymbol{E}_{n_j}$.

4. Color Image Coding Based on SA-APBT

4.1 Overall Algorithm Description

Fig. 2 shows the main procedures for the encoding and decoding framework of the region-based color image compression. In the encoding process, after the color space conversion from RGB to YCbCr [14] is performed on the original color image according to Eq. (1), the YCbCr color image is obtained. With the 4:4:4 sampling method, the luminance component and the color difference components are retained completely in the YCbCr color space. Then, the luminance component (Y) and the color difference components (Cb and Cr) are separated. After the separation process, the image

segmentation procedure is performed for each component (Y, Cb, and Cr) to obtain the ROI and background area. After that, on the one hand, the contour of the ROI goes through shape coding to obtain the shape bit stream. On the other hand, texture coding is performed to obtain the texture bit stream, where the texture coding areas include the ROI and background area of the three components. Finally the shape bit stream and texture bit stream are multiplexed to obtain the color image bit stream, where the shape bit stream is prior to the texture bit stream.

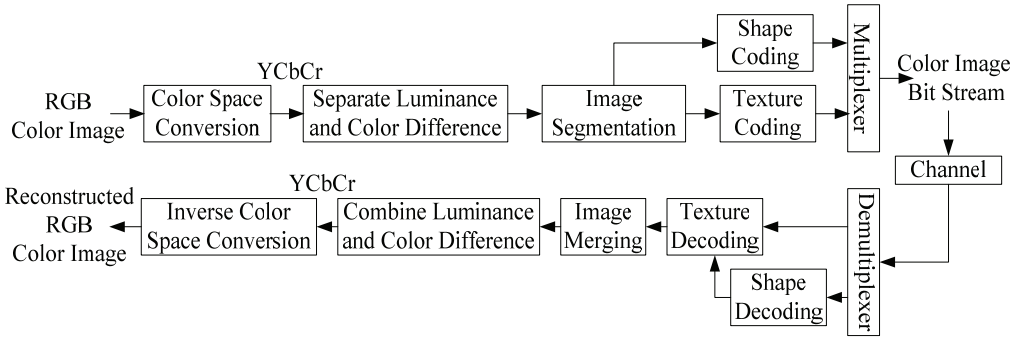


Fig. 2. Diagram of region-based color image compression.

In the decoding process, firstly, the received color image bit stream is demultiplexed to obtain the shape bit stream and texture bit stream. Then on the one hand, the shape bit stream goes through shape decoding to obtain the contour of the ROI. On the other hand, the texture bit stream performs texture decoding to obtain the reconstructed ROI and the reconstructed background area of the three components, where the decoded contour information is needed in texture decoding. After that, the reconstructed ROI and the reconstructed background area are merged to obtain the reconstructed components (Y, Cb, and Cr). The reconstructed components are combined to obtain the reconstructed YCbCr color image. Finally, after the color space conversion from YCbCr to RGB is performed on the reconstructed YCbCr color image according to Eq. (2), the reconstructed RGB color image is obtained.

4.2 Shape Coding

To encode the positions of all pixels on the boundary of the ROI, shape coding is performed. Shape coding is implemented by the chain code, which has eight directions with values of 0 to 7. Fig. 3 shows the settings of the chain code directions. The contour of the ROI can be completely reconstructed by the chain code which is a lossless shape coding method.

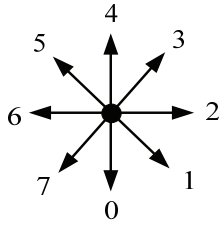


Fig. 3. Chain code direction settings.

The encoding process of the chain code is introduced as follows. Firstly, the boundary pixel located in the top left corner is found as the starting point. The row and column numbers of the starting points are then recorded. Then the next adjacent pixel on the boundary is found in clockwise direction. The corresponding chain code direction value is obtained according to the chain code direction settings, the position of the current pixel and the position of the next adjacent pixel. After that, the same search process is continued from this new pixel until all pixels on the boundary are traversed. The row number and column number of the starting point, and all chain code direction values obtained during the traversal process are the final chain code results.

The decoding process of the chain code is opposite the encoding process of the chain code. Firstly, the position of the pixel located in the top left corner is determined by the row number and column number of the starting point. The pixel located in the top left corner is set as the current pixel. Then, the position of the next adjacent pixel is determined clockwise according to the settings of the chain code directions and the chain code direction value of next adjacent pixel on the boundary. After that, under the condition that the newly found pixel is set as the current pixel, the position of the next adjacent pixel is found in the same way. When the position of the next adjacent pixel is the top left corner, the chain code decoding process is finished.

4.3 Texture Coding

The texture coding of the ROI is prior to the background area. Fig. 4 shows the main procedures of the image texture coding algorithm based on SA-APBT. When the texture coding is performed on the ROI, firstly the ROI is divided into many blocks with size of 8×8. In the transform step, for the block in the ROI, APBT is used; for the block outside the ROI, the transform procedure is omitted; for the block including the boundary of the ROI, SA-APBT is used.

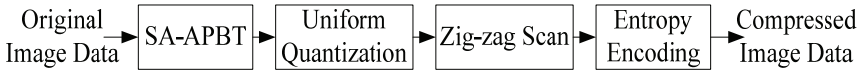


Fig. 4. Image texture coding algorithm based on SA-APBT.

Then, in the quantization step, uniform quantization is adopted. However, in the coding algorithm based on SA-DCT, two complex quantization tables proposed in JPEG standard are used. The luminance quantization table for the luminance component and the chrominance quantization table for the color difference components are shown in Fig. 5(a) and (b), respectively. Because the APBT coefficients have different frequency weightings during the transform process, the quantization effect of the proposed algorithm after using uniform quantization is similar to that of the coding algorithm based on SA-DCT.

After quantization, the two-dimensional transform coefficient block is converted into a one-dimensional data sequence by zig-zag scanning. Finally, to compress the data further, entropy encoding is performed, where the Huffman table used for the luminance component is different from the Huffman table used for the color difference components.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b)

Fig. 5. Two quantization tables used in SA-DCT: (a) luminance quantization table and (b) chrominance quantization table.

Fig. 6 shows the main procedures for the image texture decoding algorithm based on SA-APBT. The decoding process of the ROI is introduced as follows. The compressed image data go through an entropy decoding process to obtain the scan sequence of the quantized transform coefficients. Thereafter, inverse zig-zag scanning converts the one-dimensional data sequence into the two-dimensional transform coefficient block. After inverse uniform quantization, different inverse transforms are performed according to the different types of blocks. For the block in the ROI, inverse APBT is adopted; for the block outside the ROI, the inverse transform procedure is omitted; for the block including the boundary of the ROI, inverse SA-APBT (SA-IAPBT) is adopted.

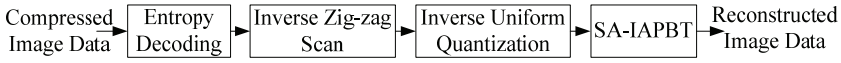


Fig. 6. Image texture decoding algorithm based on SA-APBT.

The texture coding and decoding processes of the background area are the same as the ROI.

5. Experimental Results

To test the performance of the proposed color image coding algorithm based on SA-APBT, simulation experiments are conducted with MATLAB 7.8. The performance comparisons between the color image coding algorithm based on SA-DCT and the proposed algorithm based on SA-APBT are performed in terms of both the objective quality and subjective quality of the reconstructed color images. The objective quality of the color image is measured by the composite peak signal-to-noise ratio (CPSNR) [15]:

$$\mathrm{CPSNR} = 10\log_{10}\left[\frac{255^2}{\frac{1}{3MN}\sum_{k=1}^{3}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I_{\mathrm{in}}(i,j,k)-I_{\mathrm{out}}(i,j,k)\right]^2}\right](\mathrm{dB}), \tag{10}$$

where I_{in} and I_{out} are the original color image and reconstructed color image, respectively; M and N denote the number of rows and columns of the color image, respectively; and i, j and k are the row, column and color plane number, respectively.

In the simulation experiments, typical test color images named Lena and Airplane (24 bits/pixel, 512×512) are used. Firstly, the original color image Lena (Fig. 7(a)) is segmented manually into ROI (Fig. 7(b)) and background area (Fig. 7(c)). Similarly, the original color image Airplane (Fig. 8(a)) is segmented manually into ROI (Fig. 8(b)) and background area (Fig. 8(c)).

(a) (b) (c)

Fig. 7. Original color image Lena and its segmentation results: (a) Lena, (b) ROI, and (c) background area.

(a) (b) (c)

Fig. 8. Original color image Airplane and its segmentation results: (a) Airplane, (b) ROI, and (c) background area.

The ROI and background area of the original color image are encoded using the proposed color image coding algorithm based on SA-APBT. Then the corresponding decoding procedure is performed on the coded bit stream to obtain the reconstructed ROI and background area. Finally, the reconstructed color image is obtained by merging the reconstructed ROI and background area. Based on the original color image and the reconstructed color image using the proposed algorithm, CPSNR is computed according to Eq. (10). For comparison with the color image coding algorithm based on SA-DCT, the ROI and background area are encoded using the algorithm based on SA-DCT. Then, the corresponding decoding procedure is performed on the coded bit stream to obtain the reconstructed ROI and background area. Finally, the reconstructed color image is obtained by merging the reconstructed ROI and background area. Based on the original color image and the reconstructed color image using the algorithm based on SA-DCT, CPSNR is computed according to Eq. (10).

It's worth pointing out that because the size of the APWBT [6] matrix is $N \times N$, where $N = 2^k$, $k \in \boldsymbol{Z}$, SA-APWBT cannot be used for arbitrarily shaped image segmentation coding.

Therefore, the color image coding algorithms based on SA-DCT [11], SA-APDCBT, and SA-APIDCBT are tested in the following simulation experiments.

Tables 1 and 2 show the experimental results of the color image coding using SA-DCT and SA-APBT in terms of the CPSNR of the reconstructed color images at different bit rates applied to Lena and Airplane color images, respectively. The bit rates in Tables 1 and 2 are the total bit rates of the full color image including shape coding and texture coding. In the color image coding algorithms using SA-DCT and SA-APBT, the quantization matrix used by the ROI is the same as the background area. To show the experimental results clearly, the distortion curves of Lena and Airplane are plotted based on the data in Tables 1 and 2, as shown in Fig. 9(a) and (b), respectively.

Table 1. Experimental results of color image coding using SA-DCT and SA-APBT applied to Lena image

Bit rate (bpp)	SA-DCT [11]		SA-APDCBT		SA-APIDCBT	
	Matrix Q multiply	CPSNR (dB)	Step Q	CPSNR (dB)	Step Q	CPSNR (dB)
0.40	5.10	23.71	6.08	28.13	6.20	28.19
0.45	3.60	27.22	4.15	29.37	4.24	29.50
0.50	2.75	28.63	3.21	30.18	3.29	30.33
0.55	2.25	29.77	2.57	30.85	2.63	31.04
0.60	1.88	30.49	2.15	31.38	2.20	31.60
0.70	1.39	31.88	1.62	32.21	1.65	32.47
0.80	1.10	32.72	1.28	32.88	1.31	33.16

Table 2. Experimental results of color image coding using SA-DCT and SA-APBT applied to Airplane image

Bit rate (bpp)	SA-DCT [11]		SA-APDCBT		SA-APIDCBT	
	Matrix Q multiply	CPSNR (dB)	Step Q	CPSNR (dB)	Step Q	CPSNR (dB)
0.40	6.50	24.61	7.21	26.25	7.25	26.50
0.45	4.59	25.69	5.08	27.42	5.07	27.84
0.50	3.50	27.09	3.93	28.40	3.89	28.86
0.55	2.80	28.12	3.20	29.13	3.15	29.67
0.60	2.32	28.78	2.66	29.65	2.65	30.21
0.70	1.72	29.92	1.98	30.89	1.97	31.40
0.80	1.33	31.21	1.55	31.79	1.55	32.27

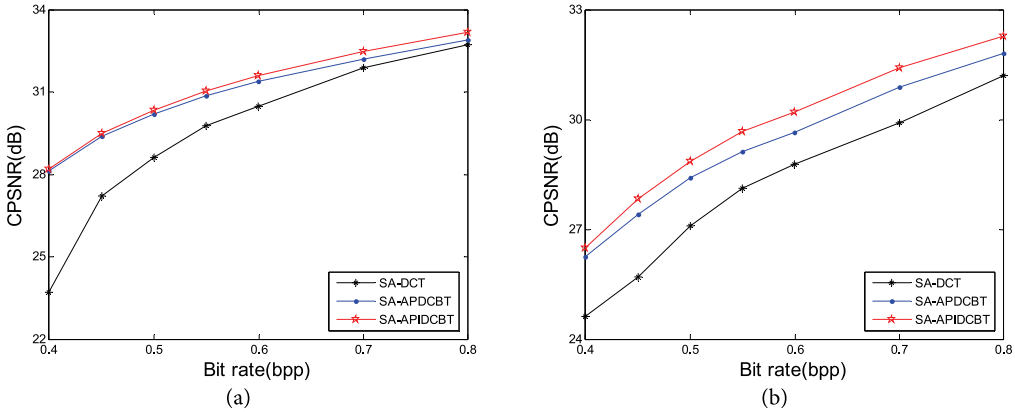
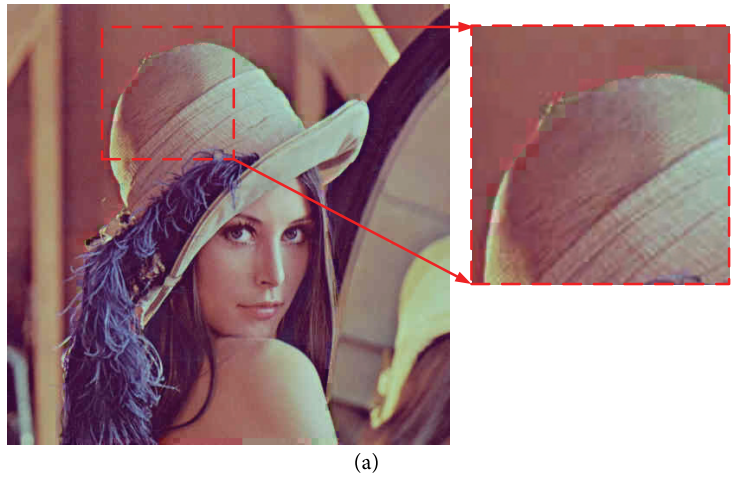
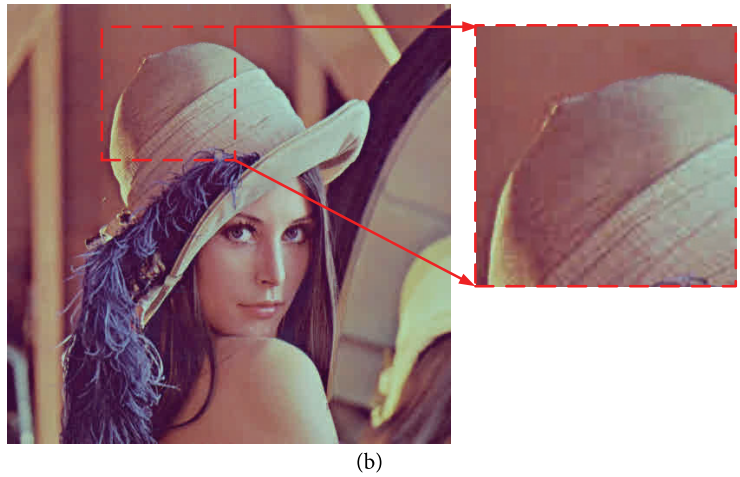


Fig. 9. Rate distortion curves: (a) Lena and (b) Airplane.



(a)



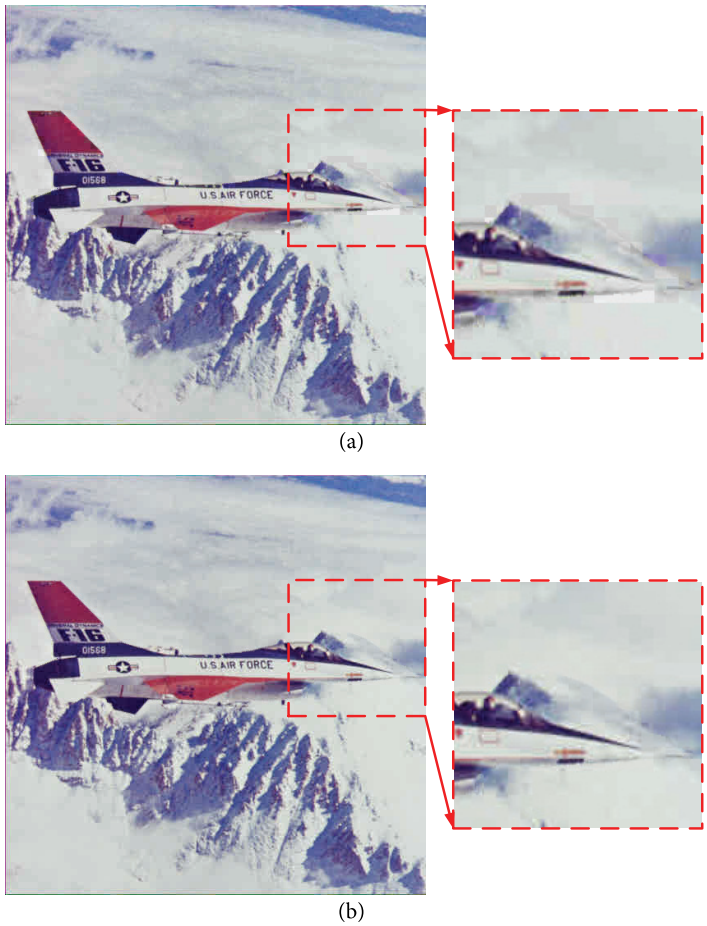
(b)

Fig. 10. Reconstructed color images of Lena using two different algorithms: (a) SA-DCT and (b) SA-APIDCBT.

From the experimental results in Table 1 and Fig. 9(a), we can conclude that higher CPSNR values are obtained at different bit rates using the proposed color image coding algorithm based on SA-APBT compared with that based on SA-DCT. Especially at low bit rates, the gap in CPSNR values between the two different algorithms is larger. With the increase in bit rate, the CPSNR gap between the two algorithms becomes smaller. From the experimental results of Airplane shown in Table 2 and Fig. 9(b), the same conclusions can be obtained.

To compare the coding performance in terms of subjective effects, Fig. 10 shows the reconstructed color images Lena using the color image coding algorithm based on SA-DCT and the proposed algorithm based on SA-APBT when the bit rate is 0.70 bpp. Fig. 11 shows the reconstructed color images of Airplane using the color image coding algorithm based on SA-DCT and the proposed algorithm based on SA-APBT when the bit rate is 0.75 bpp. From the experimental results in Figs. 10(a) and 11(a), we can see that the blocking effects of the reconstructed color images of Lena and Airplane using the color image coding algorithm based on SA-DCT are obvious, especially at some intense areas of change, such as the top border of the hat in Lena and the peripheral location of the fuselage in

Airplane. In contrast, the subjective effects of the reconstructed color images of Lena and Airplane using the proposed color image coding algorithm based on SA-APBT are better, as shown in Figs. 10(b) and 11(b). Their differences can be seen clearly from the inside of the dotted boxes. To show this more clearly, we enlarge the selected part and put it on the right side of each image.



(a)

(b)

Fig. 11. Reconstructed color images of Airplane using two different algorithms: (a) SA-DCT and (b) SA-APIDCBT.

In some cases, to reduce bit rate and make use of the visual characteristics of human eyes, the ROI was encoded with a low compression ratio, while the background area was encoded with a high compression ratio. Fig. 12(a) shows reconstructed color images of Lena using the proposed color image coding algorithm based on SA-APIDCBT under a bit rate of 0.50 bpp, when the quantization step Q of the ROI and background area is 3.29. Fig. 12(b) shows a reconstructed color image of Lena using the proposed color image coding algorithm based on SA-APIDCBT under a bit rate of 0.50 bpp, when the quantization step Q of the ROI is 2.65 and the quantization step Q of the background area is 5.80. From Fig. 12(a) and (b), it can be seen that the subjective quality of the reconstructed ROI in Fig. 12(b) is better and the subjective quality of the reconstructed background area is worse. If observers focus on the ROI, Fig. 12(b) shows better subjective effects at the same bit rate.

Finally, we would like to point out that the proposed scheme has been applied to other test images, and similar conclusions can be obtained. In conclusion, the proposed color image coding algorithm based on SA-APBT shows better performance in terms of both objective and subjective quality compared with the algorithm based on SA-DCT. Moreover, because the proposed algorithm uses uniform quantization, the quantization process is simplified and the memory space of the quantization table is conserved, thus easing the hardware implementation of the proposed algorithm.

(a) (b)

Fig. 12. Reconstructed color images of Lena: (a) same compression ratio for the ROI and background and (b) low compression ratio for ROI and high compression ratio for background area.

6. Conclusions

In this paper, a color image coding algorithm based on SA-APBT is proposed. The implementation of this algorithm consists of four steps: color space conversion, image segmentation, shape coding, and texture coding. SA-APBT and uniform quantization are adopted in texture coding. Two areas including the ROI and background area are encoded and decoded independently. Simulation experiments on typical color images are performed using MATLAB 7.8. Compared with the color image coding algorithm based on SA-DCT at the same bit rates, the CPSNR and the subjective effects of the reconstructed images using the proposed algorithm are better, especially at low bit rates (an increase of approximately 8% in terms of CPSNR). Moreover, by using uniform quantization, the proposed algorithm is simplified and the hardware implementation becomes easier.

Although high performance is obtained, the proposed method takes more time than conventional image coding algorithms since it includes the process of extracting the ROI. Therefore, in real applications, the tradeoff between image quality and computation complexity should be considered.

Finally, it should be noted that the performance of the ROI segmentation affects the performance of the proposed method. In our work, we mainly focused on comparing the performance of SA-APBT with that of SA-DCT. To achieve accurate segmentation, the ROI is extracted manually in our simulation. However, to achieve practical applications, automatic segmentation methods should be studied in depth. In addition, the impact of different segmentation methods on the proposed method should be discussed in detail. Moreover, the comparison between artificial segmentation and automatic segmentation should be considered. These issues are left for future researches.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61201371), the Natural Science Foundation of Shandong Province, China (No. ZR2015PF004), and the Research Award Fund for Outstanding Young and Middle-Aged Scientists of Shandong Province, China (No. BS2013DX022).

References

[1] E. C. Acocella and A. Alcaim, "Alignment by phase of vertical coefficients in SA-DCT," *IEEE Signal Processing Letters*, vol. 8, no. 2, pp. 42-44, 2001.

[2] *Information technology - Coding of audio-visual objects - Part 10: Advanced video coding*, Recommendation ITU-T H.264 and ISO/IEC 14496-10: 2012, 2014.

[3] B. A. Sarif, M. T. Pourazad, P. Nasiopoulos, V. C. Leung, and A. Mohamed, "Fairness scheme for energy efficient H.264/AVC-based video sensor network," *Human-centric Computing and Information Sciences*, vol. 5, article no. 7, pp. 1-29, 2015.

[4] M. B. Dissanayake and D. L. B. Abeyrathna, "Performance comparison of HEVC and H.264/AVC standards in broadcasting environments," *Journal of Information Processing Systems*, vol. 11, no. 3, pp. 483-494, 2015.

[5] *Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding*, ISO/IEC 23008-2: 2015, 2015.

[6] Z. X. Hou, C. Y. Wang, and A. P. Yang, "All phase biorthogonal transform and its application in JPEG-like image compression," *Signal Processing: Image Communication*, vol. 24, no. 10, pp. 791-802, 2009.

[7] X. Y. Wang, B. C. Jiang, C. Y. Wang, Z. Q. Yang, and C. X. Zhang, "All phase biorthogonal transform and its application in MPEG-4 video compression," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 4, pp. 13-22, 2014.

[8] C. Y. Wang, R. Y. Shan, and X. Zhou, "APBT-JPEG image coding based on GPU," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 4, pp. 1457-1470, 2015.

[9] K. Belloulata, A. Belalia, and S. P. Zhu, "Object-based stereo video compression using fractals and shape-adaptive DCT," *AEU-International Journal of Electronics and Communications*, vol. 68, no. 7, pp. 687-697, 2014.

[10] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 59-62, 1995.

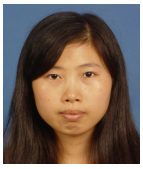
[11] Y. M. Zheng, X. Y. Wang, and C. Y. Wang, "Shape-adaptive DCT and its application in region-based image coding," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 1, pp. 99-108, 2014.

[12] B. C. Jiang, A. P. Yang, C. Y. Wang, and Z. X. Hou, "Shape adaptive all phase biorthogonal transform and its application in image coding," *Journal of Communications*, vol. 8, no. 5, pp. 330-336, 2013.

[13] X. Y. Wang, C. Y. Wang, X. Zhou, and Z. Q. Yang, "Video coding based on shape-adaptive all phase biorthogonal transform and MPEG-4," *Journal of Communications*, vol. 10, no. 12, pp. 1004-1011, 2015.

[14] C. Y. Wang, S. Z. Xie, and X. Zhou, "Bayer patterned image compression based on structure conversion and APBT," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 2, pp. 333-340, 2015.

[15] F. F. Yang, C. Y. Wang, and X. Zhou, "JPEG-like color image compression based on all phase biorthogonal transform and improved quantization table," *Journal of Communications*, vol. 10, no. 11, pp. 896-902, 2015.



Xiaoyan Wang http://orcid.org/0000-0002-7745-4794

She was born in Shandong province, China, in 1990. She received her B.S. degree in electronic information science and technology from Shandong University, Weihai, China, in 2012 and her M.E. degree in circuits and systems from Shandong University, China, in 2015. She is currently with the Huawei Technologies Co. Ltd., Nanjing, China. Her research interests include digital image/video processing, and communication technology.

Chengyou Wang http://orcid.org/0000-0002-0901-2492

He was born in Shandong province, China, in 1979. He received his B.E. degree in electronic information science and technology from Yantai University, China, in 2004, and his M.E. and Ph.D. degrees in signal and information processing from Tianjin University, China, in 2007 and 2010, respectively. He is currently an associate professor and supervisor of postgraduate students with Shandong University, Weihai, China. His current research interests include digital image/video processing and analysis (transform coding, digital watermarking, deblocking, demosaicking, dehazing, image quality assessment, etc.), computer vision (tamper detection, content-based image retrieval, etc.), pattern recognition, and machine learning.

Xiao Zhou http://orcid.org/0000-0002-1331-7379

She was born in Shandong province, China, in 1982. She received her B.E. degree in automation from Nanjing University of Posts and Telecommunications, China, in 2003; her M.E. degree in information and communication engineering from Inha University, Korea, in 2005; and her Ph.D. degree in information and communication engineering from Tsinghua University, China, in 2013. She is currently a lecturer and supervisor of postgraduate students with Shandong University, Weihai, China. Her current research interests include wireless communication technology, digital image processing, and computer vision.

Zhiqiang Yang http://orcid.org/0000-0002-7599-5722

He was born in Shandong province, China, in 1954. He received his B.S. degree and M.E. degree in electronics from Shandong University, China. He is currently an associate professor and supervisor of postgraduate students with Shandong University, Weihai, China and he is also with the Integrated Electronic Systems Lab Co. Ltd., Jinan, China. His current research interests include electronic information system, software and power system automation.