Video Coding Based on Shape-Adaptive All Phase Biorthogonal Transform and MPEG-4

Xiaoyan Wang[1], Chengyou Wang[1], Xiao Zhou[1], and Zhiqiang Yang[1,2]
[1]School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China
[2]Integrated Electronic Systems Lab Co. Ltd., Jinan 250100, China
Email: swwxy00800313@163.com; {wangchengyou, zhouxiao}@sdu.edu.cn; yangzhiqiang@ieslab.cn

Abstract —This paper proposes an efficient video coding algorithm based on shape-adaptive all phase biorthogonal transform (SA-APBT) and MPEG-4. Firstly, the input video sequence is segmented into many arbitrarily shaped video objects. Then the motion, shape and texture of all video objects are encoded in accordance with priority. Shape coding uses context-based arithmetic encoding (CAE). SA-APBT and uniform quantization method are adopted in texture coding. Experimental results show that reconstructed video sequence using video coding algorithm based on SA-APBT and MPEG-4 obtains better effects including objective quality and subjective quality compared with the one based on SA-DCT. And the proposed algorithm uses uniform quantization instead of complex quantization table in MPEG-4, which makes hardware implementation easier.

Index Terms—Video coding, MPEG-4, shape-adaptive all phase biorthogonal transform (SA-APBT), video object

I. Introduction

In video coding, the block-based coding has been widely used in many video compression standards, including MPEG-2 [1], H.263 [2] and H.264 [3]. The disadvantage of the block-based coding is bad subjective effects of reconstructed video sequence at low bit rates [4]. Now in many applications such as video conference, telemedicine and remote monitoring, observers tend to focus on some objects in the video sequence, whereas they are not interested in other objects, which promote the development of the object-based video coding. Because observed object can also get good reconstructed quality using object-based coding even at low bit rates [5]. In MPEG-4 [6], video sequence is comprised of many video objects with different physical significance. Each video object can be accessed and manipulated separately [7].

Shape-Adaptive Discrete Cosine Transform (SA-DCT) [8] for arbitrarily shaped image segment coding is widely adopted and included in MPEG-4 standard because of low complexity and effective coefficients decorrelation. But the shortcoming of it is the serious blocking effects at low bit rates [9]. In order to solve this problem, this paper proposes video coding algorithm based on Shape-Adaptive All Phase Biorthogonal Transform (SA-APBT) and MPEG-4. Firstly, the video sequence is segmented into many arbitrarily shaped video objects. Then the motion, shape and texture of all video objects are encoded in accordance with priority. In texture coding, SA-APBT and uniform quantization method are used. Experimental results show that the objective quality and subjective effects of the reconstructed video sequence using the proposed algorithm based on SA-APBT are better than that based on SA-DCT at same bit rates.

The rest of this paper is organized as follows. Section II introduces two shape-adaptive transforms: SA-DCT and SA-APBT. MPEG quantization and H.263 quantization methods used in MPEG-4 are explained in Section III. And then in Section IV, MPEG-4 video coding algorithm using SA-APBT is proposed. The overall algorithm description of MPEG-4 video coding using SA-APBT is firstly given. Thereafter the shape coding and texture coding for all video objects are described in detail. Experimental results and comparisons between the proposed algorithm based on SA-APBT and the one based on SA-DCT are presented in Section V. Conclusions of the paper and discussion for future work are given finally in Section VI.

II. Shape-Adaptive Transform

A. SA-DCT

Conventional two-dimensional Discrete Cosine Transform (DCT) [10] can only be used in rectangular image coding and is not applicable to arbitrarily shaped image segment coding. Let $\boldsymbol{X}$ and $\boldsymbol{C}$ represent an image block and DCT matrix with size of $N \times N$ respectively. After two-dimensional DCT transform, transform coefficients block $\boldsymbol{Y}$ can be denoted by

$$\boldsymbol{Y} = \boldsymbol{C}\boldsymbol{X}\boldsymbol{C}^{\mathrm{T}} \tag{1}$$

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

Manuscript received May 30, 2015; revised December 7, 2015.
This work was supported by the National Natural Science Foundation of China under Grant No. 61201371, the promotive research fund for excellent young and middle-aged scientists of Shandong Province, China under Grant No. BS2013DX022, and the Natural Science Foundation of Shandong Province, China under Grant No. ZR2015PF004.
Corresponding author email: wangchengyou@sdu.edu.cn.
doi:10.12720/jcm.10.12.1004-1011

where $\boldsymbol{C}^{\mathrm{T}}$ is the transpose matrix of C.

The processes of SA-DCT are comprised of two one-dimensional variable-length DCT subtransforms in the vertical and horizontal directions [8], as shown in Fig. 1. Fig. 1(a) shows the original boundary image block with size of 8×8 segmented into foreground (brunet) and background (white). In order to perform DCT transform in the vertical direction, the position l_j of the first pixel of each column in the foreground is found and the length n_j of each column in the foreground is counted. Then the all pixels of each column in the foreground are shifted to the uppermost position and grouped into column vector $\boldsymbol{x}_j$, as shown in Fig. 1(b). Each column vector $\boldsymbol{x}_j$ goes through one-dimensional DCT transform length of n_j, the column vector $\boldsymbol{a}_j$ is obtained (Fig. 1(c)). After one-dimensional variable-length DCT subtransform in the vertical direction, the transform result $\boldsymbol{a}_j$ can be denoted by

$$\boldsymbol{a}_j = s_{n_j} \boldsymbol{C}_{n_j} \boldsymbol{x}_j \tag{3}$$

where $\boldsymbol{C}_{n_j}$ is the DCT transform matrix with size of $n_j \times n_j$ and s_{n_j} is the SA-DCT transform prefactor value of $\sqrt{2/n_j}$.

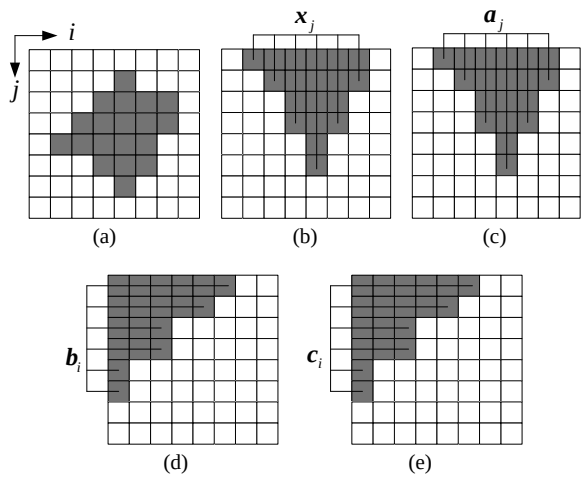


Fig. 1. The processes of SA-DCT on the foreground within an 8×8 image block: (a) Original boundary block, (b) Boundary block after the all pixels in the foreground are shifted to the uppermost position, (c) Boundary block after one-dimensional DCT transform in the vertical direction, (d) Boundary block after the all pixels in the foreground are shifted to the left border, and (e) Final SA-DCT transform coefficients boundary block after one-dimensional DCT transform in the horizontal direction.

In order to perform DCT transform in the horizontal direction, the position u_i of the first coefficient of each row in the foreground is found and the length m_i of each row in the foreground is counted. Then the all coefficients of each row in the foreground are shifted to the left border and grouped into row vector $\boldsymbol{b}_i$, as shown in Fig. 1(d). Each row vector $\boldsymbol{b}_i$ goes through one-dimensional DCT transform length of m_i, the row vector $\boldsymbol{c}_i$ is obtained (Fig. 1(e)). After one-dimensional variable-length DCT subtransform in the horizontal direction, the final transform result $\boldsymbol{c}_i$ can be denoted by

$$\boldsymbol{c}_i = s_{m_i} \boldsymbol{C}_{m_i} \boldsymbol{b}_i \tag{4}$$

where $\boldsymbol{C}_{m_i}$ is the DCT transform matrix with size of $m_i \times m_i$ and s_{m_i} is the SA-DCT transform prefactor value of $\sqrt{2/m_i}$.

B. SA-APBT

On the basis of all phase digital filtering [11], three kinds of all phase biorthogonal transforms based on the WHT, DCT and IDCT were proposed and the matrices of APBT were deduced in [12], which were defined as APWBT, APDCBT, and APIDCBT. Similar to DCT, it can be used in rectangular image coding and is not applicable to arbitrarily shaped image segment coding.

Taking APIDCBT as an example, the image block $\boldsymbol{X}$ with size of $N \times N$ goes through two-dimensional APIDCBT transform, the transform coefficients block $\boldsymbol{Y}$ can be denoted by

$$\boldsymbol{Y} = \boldsymbol{E}\boldsymbol{X}\boldsymbol{E}^{\mathrm{T}} \tag{5}$$

where $\boldsymbol{E}$ is the APIDCBT matrix with size of $N \times N$ and $\boldsymbol{E}^{\mathrm{T}}$ is the transpose matrix of $\boldsymbol{E}$. The APIDCBT matrix is

$$\boldsymbol{E}(i,j) = \begin{cases} \dfrac{1}{N}, & i = 0,\ j = 0,1,\cdots,N-1, \\ \dfrac{N-i+\sqrt{2}-1}{N^2} \cos\dfrac{i(2j+1)\pi}{2N}, & i = 1,2,\cdots,N-1,\ j = 0,1,\cdots,N-1. \end{cases} \tag{6}$$

SA-APBT can be used in arbitrarily shaped image segment coding. It contains SA-APWBT, SA-APDCBT, and SA-APIDCBT based on APWBT, APDCBT, and APIDCBT, respectively. The processes of SA-APBT are comprised of two one-dimensional variable-length APBT subtransforms in the vertical and horizontal directions. Taking SA-APIDCBT as an example, the mathematical expression of one-dimensional APIDCBT subtransform in the vertical direction is

$$\boldsymbol{a}_j = \boldsymbol{E}_{n_j} \boldsymbol{x}_j \tag{7}$$

The mathematical expression of one-dimensional APIDCBT subtransform in the horizontal direction is

$$\boldsymbol{c}_i = \boldsymbol{E}_{m_i} \boldsymbol{b}_i \tag{8}$$

III. Quantization Method in MPEG-4

A. MPEG Quantization

In MPEG quantization method [6], the quantization matrix which is adopted in intra-coded block is different from the quantization matrix which is adopted in inter-coded block, as shown in Fig. 2(a) and Fig. 2(b).

8	17	18	19	21	23	25	27
17	18	19	21	23	25	27	28
20	21	22	23	24	26	28	30
21	22	23	24	26	28	30	32
22	23	24	26	28	30	32	35
23	24	26	28	30	32	35	38
25	26	28	30	32	35	38	41
27	28	30	32	35	38	41	45

(a)

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

(b)

Fig. 2. MPEG quantization matrix: (a) Quantization matrix for intra-coded block, and (b) Quantization matrix for inter-coded block.

The mathematical expression of inverse quantization used by the direct current (DC) coefficient of intra-coded block is

$$F^{"}[0][0] = dc\_scaler \times QF[0][0] \tag{9}$$

where in short header mode dc_scaler is 8, otherwise dc_scaler is calculated according to Table I; $QF[0][0]$ and $F^{"}[0][0]$ mean the DC coefficients before and after inverse quantization respectively. The mathematical expression of inverse quantization used by the alternating current (AC) coefficients of intra-coded block and the all coefficients of inter-coded block are

$$\begin{aligned} F^{"}[v][u] &= [(2 \times QF[v][u] + k) \times W[w][v][u] \times QP] / 16, \\ k &= \begin{cases} 0, & \text{intra-coded block,} \\ sign(QF[v][u]), & \text{inter-coded block,} \end{cases} \end{aligned} \tag{10}$$

where $W[w][v][u]$ is quantization matrix, w takes the value 0 or 1, $W[0][v][u]$ is used for intra-coded block, $W[1][v][u]$ is used for inter-coded block; $QF[v][u]$ and $F^{"}[v][u]$ mean respectively block transform coefficients before and after inverse quantization; QP is quantization parameter.

TABLE I: VALUES OF dc_scaler PARAMETER DEPENDING ON QP RANGE.

Type	$QP \le 4$	$5 \le QP \le 8$	$9 \le QP \le 24$	$25 \le QP$
Luminance block	8	$2 \times QP$	$QP + 8$	$(2 \times QP) - 16$
Chrominance block	8	$(QP + 13) / 2$	$(QP + 13) / 2$	$QP - 6$

After the quantization coefficients go through inverse quantization, computed results are limited to the range $[-2048, 2047]$. Thus MPEG inverse quantization process is completed. MPEG quantization process is opposite to MPEG inverse quantization process.

B. H.263 Quantization

In H.263 quantization method, the mathematical expression of inverse quantization used by the DC coefficient of intra-coded block is also (9). The mathematical expression of inverse quantization used by the AC coefficients of intra-coded block and the all coefficients of inter-coded block are

$$\left|F^{"}[v][u]\right| = \begin{cases} 0, & QF[v][u] = 0, \\ 2 \times \left|QF[v][u]\right| \times QP + QP, & QF[v][u] \neq 0 \text{ and } QP \text{ is odd,} \\ 2 \times \left|QF[v][u]\right| \times QP + QP - 1, & QF[v][u] \neq 0 \text{ and } QP \text{ is even,} \end{cases} \tag{11}$$

where QP is quantization parameter, $QF[v][u]$ and $F^{"}[v][u]$ mean respectively block transform coefficient before and after inverse quantization. The computed results from (11) are absolute values of inverse quantization results, combined with positive and negative information, we get

$$F^{"}[v][u] = sign(QF[v][u]) \times \left|F^{"}[v][u]\right| \tag{12}$$

Finally the inverse quantization results from (12) are limited to the range $[-2048, 2047]$. Thus H.263 inverse quantization process is completed. H.263 quantization process is opposite to H.263 inverse quantization process.

IV. VIDEO CODING BASED ON SA-APBT AND MPEG-4

A. Overall Algorithm Description

In order to achieve flexible and high-quality low bit rate compression, MPEG-4 defines video object (VO) as compression unit. Firstly, the input video sequence is segmented into many arbitrarily shaped video objects. Each video object is an area of video scene and has specific physical meaning. Then the motion, shape and texture of all video objects are encoded in accordance with priority.

A video object consists of many Video Object Planes (VOP). Because the shape of VOP is generally irregular, shape information also needs to be encoded before texture coding. If the input VOP is intra-coded, Motion Estimation (ME) and Motion Compensation (MC) processes are omitted. Shape coding is performed to get shape bit stream. Texture coding is done to obtain texture bit stream and reconstructed current VOP. Then on the one hand shape bit stream and texture bit stream are multiplexed to store in cache, on the other hand the reconstructed current VOP is stored in the VOP cache. If the input VOP is inter-coded, the previous reconstructed VOP in VOP cache is set as the reference VOP of current VOP to perform ME and MC. Consequently, Motion Vector (MV) and the predicted VOP of current VOP are obtained. After that, motion coding and shape coding are performed to get motion bit stream and shape bit stream. The predicted VOP is subtracted by the current VOP to get the residual VOP. Then texture coding is performed on the residual VOP to obtain texture bit stream and reconstructed residual VOP. Finally, on the one hand the motion bit stream, shape bit stream and texture bit stream are multiplexed to store in cache, on the other hand the sum of reconstructed residual VOP and the predicted VOP is stored in VOP cache. The sum of reconstructed residual VOP and the predicted VOP is reconstructed

current VOP. The proposed MPEG-4 video coding framework is shown in Fig. 3.

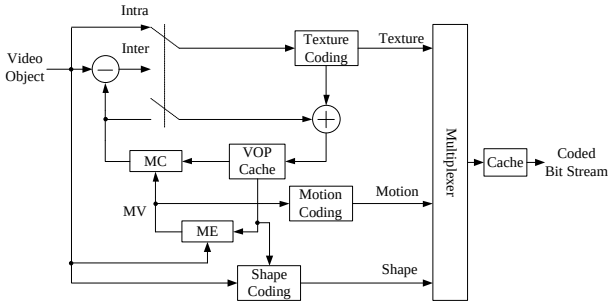


Fig. 3. Proposed MPEG-4 video coding framework.

The decoding process is composed of three parts: shape decoding, motion decoding and texture decoding. Firstly the received coded bit stream passes the demultiplexer to get shape bit stream, motion bit stream and texture bit stream. Shape bit stream goes through shape decoding procedure to obtain the shape information of current VOP. When texture bit stream goes through texture decoding, the decoded shape information is used. If the current VOP is intra-coded, the reconstructed current VOP is obtained after texture decoding. If the current VOP is inter-coded, the reconstructed residual VOP is got after texture decoding. After that, the motion bit stream is performed motion decoding to get MV. The MC is performed to obtain the predicted VOP according to the decoded MV, previous reconstructed VOP and decoded shape information. Finally the predicted VOP and reconstructed residual VOP is summed to get reconstructed current VOP. Reconstructed current VOP is stored in VOP cache. Proposed MPEG-4 video decoding framework is shown in Fig. 4.

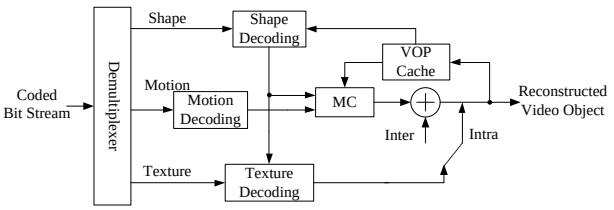


Fig. 4. Proposed MPEG-4 video decoding framework.

B. Shape Coding

There are two methods for the shape coding of VO: binary shape coding and grey shape coding. If binary shape coding is adopted, the binary alpha mask for each boundary macroblock is coded. The value of the binary alpha mask pixel for each boundary macroblock is either 0 or 1. The pixel value of 0 is outside the VOP. The pixel value of 1 is in the VOP. If grey shape coding is used, the binary alpha mask and grey-scale alpha plane for each boundary macroblock are coded. The value of grey-scale alpha plane pixel for each boundary macroblock is between 0 and 255, where 0 indicates that the pixel is completely transparent, 255 indicates that the pixel is completely opaque. The coding algorithm used by grey-scale alpha plane for boundary macroblock is the same as texture coding algorithm.

Before binary shape coding, the optimum rectangular area including the current VOP is calculated [13]. The optimum rectangular area has the following requirements: the horizontal and vertical coordinates of the left and upper vertex are even, the length and width are multiples of 16, the number of the coded macroblocks with size of 16×16 is least. Then the all macroblocks in the optimum rectangular area are divided into three groups:

a) Binary shape coding is not performed on the macroblock in the VOP.
b) Binary shape coding is not performed on the macroblock outside the VOP.
c) Binary shape coding is performed on the macroblock including the boundary of VOP.

Binary shape coding is implemented by context-based arithmetic encoding (CAE). There are two CAE types: intra-frame CAE and inter-frame CAE.

C. Texture Coding

For inter-frame coded VOP, each macroblock in the optimum rectangular area goes through ME and MC process to get MV. The MV may point to the region outside the opaque area of the reference VOP. Therefore the macroblock-based texture padding is performed on the boundary macroblock and external macroblock of the reference VOP [6]. The macroblock-based texture padding defines the luminance and chrominance components data of transparent pixels in these macroblocks. The padding of luminance component is based on macroblock with size of 16×16. The padding of chrominance component is based on block with size of 8×8. The texture padding process for boundary macroblock is divided into horizontal repetitive padding and vertical repetitive padding, as shown in Fig. 5. The texture padding process for external macroblock uses extended padding. For intra-frame coded VOP, the macroblock-based texture padding process is not performed.

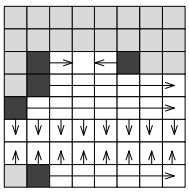


Fig. 5. Texture padding process for boundary macroblock.

Fig. 6 shows the main procedures for VOP texture coding algorithm based on SA-APBT. Texture coding process is performed on the macroblocks in optimum rectangular area. Since the input to MPEG-4 encoder is a YUV video sequence in 4:2:0 progressive format, there are six blocks with size of 8×8 to be texture coded for each macroblock, where the luminance component Y has four, the chrominance component U has one, the chrominance component V has one.

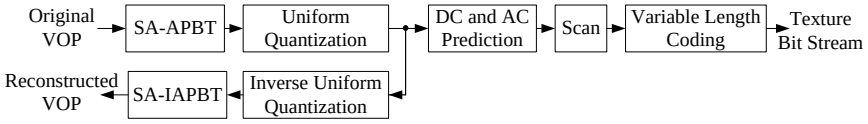


Fig. 6. Diagram of VOP texture coding algorithm based on SA-APBT.

Firstly the all macroblocks are transformed by different transform methods. In this step, the all macroblocks in the optimum rectangular area are divided into three groups:

a) For the macroblock in the VOP, its six blocks are transformed by APBT.
b) For the macroblock outside the VOP, its six blocks are not transformed.
c) For the macroblock including the boundary of the VOP, its six blocks are transformed by SA-APBT according to the binary shape information.

Then the transform coefficients are quantified by quantization table. Different from the complex quantization method in MPEG-4, uniform quantization is adopted in the proposed algorithm because SA-APBT has better energy compaction characteristic in image compression than SA-DCT. After uniform quantization, in order to prevent error propagation, the coding algorithm on the one hand goes on texture coding, on the other hand goes through inverse uniform quantization and shape-adaptive inverse all phase biorthogonal transform (SA-IAPBT) procedures to get reconstructed current VOP. SA-IAPBT means the inverse transform process of SA-APBT, namely, SA-IAPBT is inverse SA-APBT. Inter-frame coded VOP uses the reconstructed VOP instead of the original VOP as the reference VOP to perform ME and MC.

Thereafter the six blocks in the macroblock go through DC and AC prediction procedure, where AC prediction is optional. The DC coefficient of current block is predicted from the DC coefficient of upper or left block. The direction of the smallest DC gradient is chosen as the prediction direction of the DC coefficient of the current block. The prediction direction of AC coefficients is the same as the prediction direction of DC coefficient. If the prediction direction of DC coefficient is horizontal, the first column of AC coefficients in current block is predicted by the first column of AC coefficients in left block. If the prediction direction of DC coefficient is vertical, the first row of AC coefficients in current block is predicted by the first row of AC coefficients in upper block. When the sum of absolute values of the difference between the AC coefficients and its prediction values is larger than the sum of the absolute values of AC coefficients, AC coefficients are not predicted.

Then the transform coefficients are prepared by scan for variable length coding (VLC). There are three scan types: Zig-zag scan, alternate-horizontal scan and alternate-vertical scan.

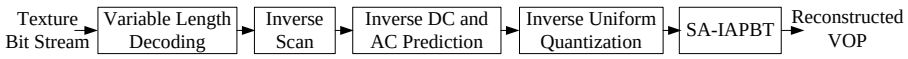


Fig. 7. Diagram of VOP texture decoding algorithm based on SA-APBT.

Finally VLC is performed to get the texture bit stream. MPEG-4 defines different VLC tables for intra-coded transform coefficients and inter-coded transform coefficients. Motion vector differences including x and y component are also coded by using VLC. To improve the error resilience, MPEG-4 defines reversible variable length coding (RVLC) and the corresponding RVLC tables.

The texture decoding process is essentially opposite to the texture encoding process. Fig. 7 shows the main procedures for VOP texture decoding algorithm based on SA-APBT. Firstly the texture bit stream goes through variable length decoding to get the scan sequence of the quantized transform coefficients. Thereafter the inverse scan procedure converts the one-dimensional quantized transform coefficients into two-dimensional quantized transform coefficients. Then inverse DC and AC prediction are performed. After inverse uniform quantization, the different inverse transforms are done according to different macroblock types. For the macroblock in the VOP, its six blocks are transformed by inverse APBT. For the macroblock outside the VOP, its six blocks are not transformed. For the macroblock including the boundary of the VOP, its six blocks are transformed by SA-IAPBT.

V. Experimental Results

In order to test the performance of proposed video coding algorithm based on SA-APBT and MPEG-4, simulation experiments are conducted using VC++ 6.0. The test video sequence Foreman is in YUV format, with the size of 352×288. Foreman is segmented into two video objects: foreground and background, where the first frame and its segmentation results are shown in Fig. 8.

(a) (b) (c)

Fig. 8. The first frame of Foreman and its segmentation results: (a) The first frame of Foreman, (b) Foreground, and (c) Background.

The foreground is respectively encoded according to MPEG-4 video coding algorithm based on SA-APBT and the one based on SA-DCT using the advanced coding

efficiency profile. The simulation experiments are carried out with a group of pictures (GOP) size of 12 pictures. The GOP includes one I frame and eleven P frames. The frame rate is 30 frames/s and the bit rate is 300kbps. MPEG quantization method is adopted in MPEG-4 video coding algorithm based on SA-DCT. To get the reconstructed foreground by using two different algorithms, corresponding decoding procedures are performed on the coded bit stream. In terms of objective quality, Y component's peak signal to noise ratio (PSNR) is computed according to each VOP of the foreground and each VOP of the reconstructed foreground. Fig. 9 shows the Y component's PSNR using MPEG-4 video coding algorithm based on SA-APBT and the one based on SA-DCT for the different VOPs of the foreground.

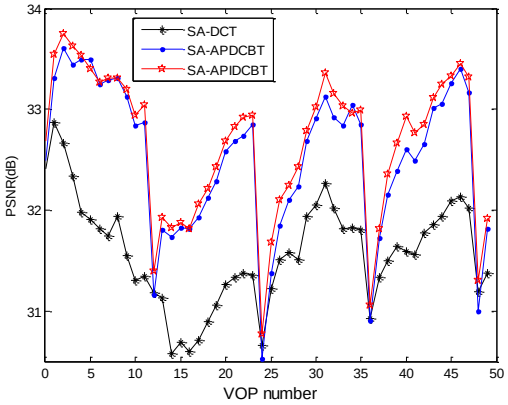


Fig. 9. Experimental results of different algorithms applied to the foreground of video sequence Foreman.

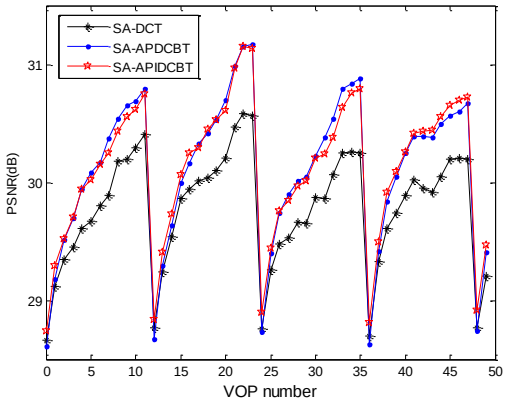
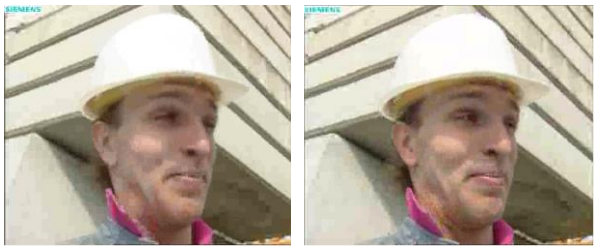


Fig. 10. Experimental results of different algorithms applied to the background of the video sequence Foreman.

Similarly, the background is respectively encoded according to MPEG-4 video coding algorithm based on SA-APBT and the one based on SA-DCT using the advanced coding efficiency profile. The coding parameters setting of the background is the same as the coding parameters setting of the foreground. To get the reconstructed background using two different algorithms, corresponding decoding procedures are performed on the coded bit stream. In terms of objective quality, Y component's PSNR is computed according to each VOP of the background and each VOP of the reconstructed background. Fig. 10 shows the Y component's PSNR using MPEG-4 video coding algorithm based on SA-APBT and the one based on SA-DCT for the different VOPs of the background.

From the experimental results in Fig. 9, we conclude that Y component's PSNR curve obtained by the video coding algorithm based on SA-APBT and MPEG-4 has basically the same change rule as the one obtained by the video coding algorithm based on SA-DCT and MPEG-4 for the foreground of Foreman. And compared with the video coding algorithm based on SA-DCT and MPEG-4, PSNR values using the proposed algorithm based on SA-APBT are higher for the majority of VOPs at different VOP numbers. For the background of Foreman, the same conclusion is got from the experimental results in Fig. 10. In brief, the proposed algorithm is 0.10~1.70dB better than the one adopting SA-DCT.

In order to compare the compression performance subjectively, the reconstructed foreground and the reconstructed background using two different algorithms are merged into the reconstructed Foreman. Fig. 11(a) and Fig. 11(b) show that the 6th frames of the reconstructed Foreman video sequences obtained by MPEG-4 video coding algorithm using SA-DCT, SA-APIDCBT. Fig. 12(a) and Fig. 12(b) show that the 9th frames of the reconstructed Foreman video sequences obtained by MPEG-4 video coding algorithm using SA-DCT, SA-APIDCBT. Fig. 13(a) and Fig. 13(b) show that the 45th frames of the reconstructed Foreman video sequences obtained by MPEG-4 video coding algorithm using SA-DCT, SA-APIDCBT.



(a) (b)

Fig. 11. The 6th frames: (a) SA-DCT, and (b) SA-APIDCBT.

(a) (b)

Fig. 12. The 9th frames: (a) SA-DCT, and (b) SA-APIDCBT.

From Fig. 11(a), Fig. 12(a), and Fig. 13(a), it can be seen that the block artifacts of the three frames of the reconstructed Foreman using the MPEG-4 video coding algorithm based on SA-DCT are obvious especially at areas such as eyes and mouth. In Fig. 11(b), Fig. 12(b), and Fig. 13(b), the block artifacts of the three frames of the reconstructed Foreman using the proposed algorithm

based on SA-APBT are reduced greatly and subjective effects are better. Note that the proposed algorithm based on SA-APBT outperforms the baseline MPEG-4 based on SA-DCT. And blocking artifacts have been removed significantly.

(a) (b)

Fig. 13. The 45th frames: (a) SA-DCT, and (b) SA-APIDCBT.

In conclusion, better performance is obtained using video coding algorithm based on SA-APBT and MPEG-4 in terms of both objective quality and subjective quality. And because of uniform quantization, the computational complexity of the proposed algorithm is reduced and hardware implementation is easier.

VI. Conclusions

In this paper, video coding algorithm based on SA-APBT and MPEG-4 is proposed. Firstly, the input video sequence is segmented into many arbitrarily shaped video objects. Then the motion, shape and texture of all video objects are encoded in accordance with priority. In texture coding, SA-APBT and uniform quantization method are adopted. The experiments to typical test video sequence are done using VC++ 6.0. Compared with the video coding algorithm based on SA-DCT and MPEG-4, the proposed algorithm based on SA-APBT obtains better performance including objective quality and subjective effects. And the proposed algorithm is easier for hardware implementation because of using uniform quantization. In the future, we will continue to explore the quick and efficient optimization algorithm for video coding based on SA-APBT and MPEG-4 using graphic processing unit (GPU) parallel technology.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61201371), the promotive research fund for excellent young and middle-aged scientists of Shandong Province, China (Grant No. BS2013DX022), and the Natural Science Foundation of Shandong Province, China (Grant No. ZR2015PF004). The authors would like to thank Qiming Fu, Fanfan Yang, and Liping Wang for their kind help and valuable suggestions.

References

[1] ISO/IEC, Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video, ISO/IEC 13818-2: 2013, Sep. 2013.

[2] J. Tavares, A. Silva, and A. Navarro, "H.263 video codec performance with a fast 8×8 integer IDCT," in *Proc. IEEE International Conference on Multimedia and Expo*, Toronto, Canada, Jul. 9-12, 2006, pp. 2009-2012.

[3] Joint Video Team of ITU-T and ISO/IEC, Information Technology – Coding of Audio-Visual Objects – Part 10: Advanced Video Coding, ITU-T Rec. H.264 | ISO/IEC 14496-10: 2012, Jan. 2014.

[4] 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, Jul. 2014.

[5] S. P. Zhu, Y. S. Hou, Z. K. Wang, and K. Belloulata, "Fractal video sequences coding with region-based functionality," *Applied Mathematical Modelling*, vol. 36, no. 11, pp. 5633-5641, Nov. 2012.

[6] ISO/IEC, Information Technology – Coding of Audio-Visual Objects – Part 2: Visual, ISO/IEC 14496-2: 2004, Jun. 2004.

[7] Y. Liu, K. N. Ngan, and F. Wu, "3-D shape-adaptive directional wavelet transform for object-based scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 888-899, Jul. 2008.

[8] 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, Feb. 1995.

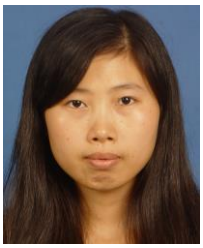
[9] B. C. Jiang, A. P. Yang, C. Y. Wang, and Z. X. Hou, "Shape adaptive all phase biorthogonal transform and its application in image coding," *Journal of Communications*, vol. 8, no. 5, pp. 330-336, May 2013.

[10] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18-34, Feb. 1992.

[11] Z. X. Hou and X. Yang, "The all phase DFT filter," in *Proc. 10th IEEE Digital Signal Processing Workshop and the 2nd IEEE Signal Processing Education Workshop*, Pine Mountain, Georgia, USA, Oct. 13-16, 2002, pp. 221-226.

[12] 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, Nov. 2009.

[13] K. B. Lee, J. Y. Lin, and C. W. Jen, "A multisymbol context-based arithmetic coding architecture for MPEG-4 shape coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 283-295, Feb. 2005.



Xiaoyan Wang 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. Now she is with the Huawei Technologies Co. Ltd., Nanjing, China. Her research interests include digital image & video processing, and communication technology.

Chengyou Wang 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. degree in signal and information processing from Tianjin University, China in 2007 and 2010 respectively. Now he is an associate professor and supervisor of postgraduate in the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China. His current research interests include digital

image/video processing and analysis, multidimensional signal and information processing.

Xiao Zhou 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. Now she is a lecturer in the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China. Her current research interests include wireless communication technology, digital image processing and analysis.

Zhiqiang Yang was born in Shandong province, China in 1954. He received his B.S. degree and M.E. degree in electronics from Shandong University, China. Now he is an associate professor and supervisor of postgraduate in the School of Mechanical, Electrical and Information Engineering, 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.