

Video Coding Algorithm Based on High Efficiency Video Coding (HEVC) and Hybrid Transforms

Chengyou Wang, Rongyang Shan, and Xiao Zhou*

School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China
[e-mail: wangchengyou@sdu.edu.cn, sry@mail.sdu.edu.cn, zhouxiao@sdu.edu.cn]

*Corresponding author: Xiao Zhou

*Received November 3, 2017; revised April 6, 2018; accepted April 25, 2018;
published September 30, 2018*

Abstract

In recent years, due to its high efficiency and better performance, the high efficiency video coding (HEVC) has become the most common compression standard in the field of video coding. In this paper, the framework of HEVC is deeply analyzed, and an improved HEVC video coding algorithm based on all phase biorthogonal transform (APBT) is proposed, where APBT is utilized to replace the discrete cosine transform (DCT) and discrete sine transform (DST) in original HEVC standard. Based on the relationship between APBT and DCT, the integer APBT is deduced. To further improve the coding performance, an optimal HEVC video coding algorithm based on hybrid APBT is proposed. The coding performance of the proposed HEVC coding algorithm is improved without increasing the complexity. Experimental results show that compared with HEVC standard algorithm, the improved HEVC video coding algorithm based on hybrid APBT can improve the coding performance of chrominance components by about 0.3%.

Keywords: Video coding, high efficiency video coding (HEVC), discrete cosine transform (DCT), discrete sine transform (DST), all phase biorthogonal transform (APBT)

This work was supported by the National Natural Science Foundation of China (No. 61702303, No. 61201371); the Natural Science Foundation of Shandong Province, China (No. ZR2017MF020, No. ZR2015PF004); and the Research Award Fund for Outstanding Young and Middle-Aged Scientists of Shandong Province, China (No. BS2013DX022).

1. Introduction

With the continuous development of multimedia and information technology, people are paying more attention to the quality of images/videos. However, the large amount of data carried by them bring great challenges to the data storage, network transmission, and real-time data processing. At the same time, with the constant increasing of image/video resolution, it has become urgent to improve the speed of image/video coding to satisfy people's requirements for real-time image processing.

After entering the 21st century, although new international image compression standards have been developed successively, such as JPEG2000 [1] based on discrete wavelet transform (DWT) and JPEG-XR [2] based on lapped biorthogonal transform (LBT) [3], the JPEG standard [4] based on block discrete cosine transform (DCT) [5] is still the most widely used still image lossy compression standard due to its high performance and low complexity. Meanwhile, DCT is also the core transform of the video compression standards, such as MPEG-2 [6], MPEG-4 [7], H.263, H.264/AVC [8], and H.265/HEVC (high efficiency video coding) [9]. Compared with image coding, the framework of video coding is more complex because there are spatial and temporal redundancies in video sequences, which need to be eliminated.

Similar to the previous standards, the framework of HEVC coding still adopts hybrid model based on blocks. First of all, HEVC encoder divides the input video sequence into groups of pictures (GOP), and each frame in GOP is further divided into coding units (CUs). Next, according to the spatial and temporal correlation of the video sequence, the corresponding prediction coding is implemented for each CU. The spatial and temporal redundancies of the video sequence are eliminated by intra-prediction and inter-prediction, and then the corresponding residual coefficients are calculated by the difference between predicted image blocks and original image blocks. After transformation and quantization, the quantized transform coefficients can be obtained. Finally, the quantized coefficients are encoded and the HEVC bit stream is obtained. Recently, many improved and optimized HEVC coding methods are proposed. To improve the speed of the intra-coding stage, Sun et al. [10] proposed two methods, an early coding tree unit (CTU) depth interval prediction and an efficient CU partition method. Tai et al. [11] introduced a fast HEVC encoding method by researching the relationship between the current and collocated CU. Additionally, the relationship of the rate-distortion costs is investigated to build the mathematical model for representation of the rate-distortion costs. In this method, a search range algorithm is applied to improve the speed of the algorithm. Chen et al. [12] presented a novel method based on blocks for intra-coding in HEVC. This scheme is derived from the exploration that the prediction errors on some pixels in each prediction block (PB) that are neighboring to the reference pixels would be no bigger than the corresponding coding errors, which divides the pixels into two half parts. One is for coding via a novel padding technique together with a constrained quantization algorithm, and the other is for reconstruction by linear interpolations along a prediction direction. To decrease the complexity of the HEVC coding standard, Tang et al. [13] applied the Hadamard energy for residual information measurement to determine the quantization parameter (QP) value for better coding efficiency. To overcome the problem of the computational complexity in HEVC, Goswami and Kim [14] proposed a fast HEVC coding scheme on the basis of Markov chain Monte Carlo model and Bayesian classifier. Yang et al. [15] analyzed three aspects of video contents, including the pixel gradient, the block mean value, and the block variance of CU, to

realize fast CU depth level decision and fast prediction mode decision, decreasing the computational complexity.

It is worth noting that, besides DCT, the discrete sine transform (DST) [16] is firstly adopted in the transform stage. The DCT, DST, and quantization process are involved in the prediction stage to obtain the optimal transform unit (TU). Finally, the residual coefficients under the optimal mode are transformed, and the transformed and quantized coefficients are encoded. However, in block-based coding standards, serious block artifacts exist at low bit rates, and the redundancies between sub-blocks still remain.

To solve the above problems, according to the design theory of all phase digital filter (APDF) [17], in [18], three kinds of all phase biorthogonal transforms (APBTs) were deduced based on the orthogonal transforms: Walsh-Hadamard transform (WHT), DCT, and inverse DCT (IDCT). These three transforms are called all phase Walsh biorthogonal transform (APWBT), all phase discrete cosine biorthogonal transform (APDCBT), and all phase inverse discrete cosine biorthogonal transform (APIDCBT), respectively. Simultaneously, the JPEG-like image coding algorithm based on APBT was proposed. Since APBT has better characteristics in energy concentration and high frequency attenuation, it is used to replace the DCT for image transform. In order to achieve the effect of fine quantization for low frequency and coarse quantization for high frequency, the simple uniform quantization is used to replace the complex quantization table in JPEG compression which is designed based on the human visual system (HVS). Besides, by using this uniform quantization, the algorithm is also simplified. Experimental results show that the quality of the reconstructed image is better than the traditional DCT when APBT is used in image coding. The block artifacts in reconstructed image can be removed effectively, especially at low bit rates.

In recent years, shape-adaptive APBT (SA-APBT) [19], directional APBT (D-APBT) [20], and windowed APBT (WAPBT) [21] have been proposed successively by using the excellent characteristics of APBT. Further studies have been developed for color image coding [22], MPEG-2 [23], MPEG-4 [24], H.263 [25], and other video coding algorithms. Moreover, the APBT-JPEG [26] coding efficiency has been improved by adopting parallel GPU, and the JPEG-like image compression algorithm based on the all phase DST has been proposed [27]. The use of GPU and all phase DST enhances the coding performance and achieves nearly 100 times acceleration.

To further optimize the algorithm and enhance the performance of HEVC video coding, this paper applies APBT to HEVC video coding framework. Compared with conventional HEVC video coding algorithms, the subjective and objective quality of the reconstructed video frames are further improved.

The rest of this paper is organized as follows. Section 2 presents the derivation process of integer DCT and integer DST in detail. Section 3 introduces the principle and characteristics of APIDCBT, and then the integer APIDCBT is deduced. The application of APBT in HEVC video coding is described in Section 4. Experimental results and analysis about the proposed algorithm are presented in Section 5. Conclusions and remarks on possible further work are given finally in Section 6.

2. Related Work

The core parts of image and video compression are transform process and quantization process. Like H.264, the integer DCT is also used in HEVC standard. The main differences between H.264 and HEVC are that more transform sizes are introduced in HEVC and the DST is firstly introduced in transform process. In this section, the derivation processes of integer DCT and

integer DST will be introduced [9], which can lay the foundation for the derivation of integer APBT in Section 3.

2.1 Integer DCT in HEVC

In HEVC standard [9], the integer DCT is used for intra-frame coding blocks with sizes of 8×8 , 16×16 , 32×32 , and 64×64 , and it is also used for inter-frame coding blocks with sizes of 4×4 , 8×8 , 16×16 , 32×32 , and 64×64 . Taking 4×4 coding block as an example, we give the derivation process of the integer DCT in HEVC standard, and then it is compared with the integer DCT in H.264. The two-dimensional DCT in HEVC [9] and in H.264 [8] is defined as:

$$Y(k,l) = C(k)C(l) \sum_{m=0}^3 \sum_{n=0}^3 \left[x(m,n) \cos \frac{(2m+1)k\pi}{8} \right] \cos \frac{(2n+1)l\pi}{8}, \quad k,l = 0,1,2,3, \quad (1)$$

where

$$C(k) = C(l) = \begin{cases} 1/2, & k,l = 0, \\ 1/\sqrt{2}, & k,l = 1,2,3. \end{cases} \quad (2)$$

From Eq. (1), we have:

$$Y(k,l) = C(k) \sum_{m=0}^3 \left[C(l) \sum_{n=0}^3 x(m,n) \cos \frac{(2n+1)l\pi}{8} \right] \cos \frac{(2m+1)k\pi}{8}, \quad k,l = 0,1,2,3. \quad (3)$$

It can be seen from Eq. (3) that the two-dimensional DCT can be decomposed into two one-dimensional DCT in row and column directions, which can be represented as:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}, \quad (4)$$

where \mathbf{X} represents the image block; \mathbf{Y} is the transformed DCT coefficient matrix; and the transform matrix \mathbf{A} can be expressed as:

$$\mathbf{A} = \begin{bmatrix} \frac{1}{2} \cos 0 & \frac{1}{2} \cos 0 & \frac{1}{2} \cos 0 & \frac{1}{2} \cos 0 \\ \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{5\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{7\pi}{8} \\ \frac{1}{\sqrt{2}} \cos \frac{2\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{6\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{10\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{14\pi}{8} \\ \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{9\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{15\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{21\pi}{8} \end{bmatrix} \quad (5)$$

$$= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & -\frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & -\frac{1}{\sqrt{2}} \cos \frac{\pi}{8} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & -\frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & -\frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} \end{bmatrix}.$$

Let $a = \frac{1}{2}$, $b = \frac{1}{\sqrt{2}} \cos \frac{\pi}{8}$, and $c = \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8}$, the transform matrix \mathbf{A} can be rewritten as:

$$\mathbf{A} = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}. \quad (6)$$

In H.264 standard [8], the values b and c in integer DCT need to be adjusted appropriately, and the ratio of b and c is adjusted to 2. Therefore, the integer DCT for 4×4 coding block in H.264 can be represented as:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \otimes \begin{bmatrix} a & a & a & a \\ \frac{b}{2} & \frac{b}{2} & \frac{b}{2} & \frac{b}{2} \\ a & a & a & a \\ \frac{b}{2} & \frac{b}{2} & \frac{b}{2} & \frac{b}{2} \end{bmatrix} = \mathbf{C}_f \otimes \mathbf{E}_f, \quad (7)$$

$$\mathbf{Y} = \mathbf{A}\mathbf{X} = (\mathbf{C}_f \otimes \mathbf{E}_f)\mathbf{X} = (\mathbf{C}_f\mathbf{X}) \otimes \mathbf{E}_f, \quad (8)$$

where “ \otimes ” denotes the multiplication operation of the corresponding elements in two matrices, \mathbf{E}_f is the correction matrix, and f is the size of the matrix. The correction process in H.264 is implemented in quantization process after transformation.

Unlike H.264, the elements in DCT transform matrix are scaled up directly in HEVC standard, which is used to improve the accuracy. It makes the values of integer DCT coefficients closer to the floating point DCT coefficients, which improves the coding performance. The integer DCT transform matrix for 4×4 coding block can be obtained by multiplying the values of a , b , and c in Eq. (6) with 128. Besides, to satisfy the orthogonality, that is $2a^2 = b^2 + c^2$, the values of b and c should be fine tuned. Finally, the integer DCT transform matrix with size of 4×4 in HEVC can be represented as:

$$\mathbf{H}_4 = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}. \quad (9)$$

Similar to the correction matrix in H.264, the correction matrix in HEVC is defined as \mathbf{E}_4 . Then, the two-dimensional integer DCT for 4×4 coding block in HEVC can be expressed as:

$$\begin{aligned} \mathbf{Y} &= (\mathbf{H}_4\mathbf{X}\mathbf{H}_4^T) \otimes (\mathbf{E}_4 \otimes \mathbf{E}_4^T) \\ &= \left\{ \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix} \mathbf{X} \begin{bmatrix} 64 & 83 & 64 & 36 \\ 64 & 36 & -64 & -83 \\ 64 & -36 & -64 & 83 \\ 64 & -83 & 64 & -36 \end{bmatrix} \right\} \cdot \frac{1}{128} \cdot \frac{1}{128}. \end{aligned} \quad (10)$$

In HEVC, the integer DCT for other sizes can be obtained by the same method. The only difference is that the transform matrices for different sizes are amplified by different ratios. That means for a $N \times N$ transform matrix, the magnification times is $64\sqrt{N}$.

2.2 Integer DST in HEVC

DST is firstly introduced in HEVC standard. HEVC standard stipulates that in the residual coding of luminance components, the integer DST is used for 4×4 intra-prediction mode while the integer DCT is used for other prediction modes. The main reason is that the residual coefficients obtained by HEVC intra-prediction have the characteristic that the further the distance from the predicted pixels is, the larger the amplitude of prediction residual will be. The DST can be well adapted to this characteristic, and experimental results show that by using integer DST in 4×4 intra-prediction mode, the coding performance can be improved by 0.8%.

For 4×4 coding blocks of intra-frame luminance components in HEVC [9], the two-dimensional DST can be expressed as:

$$Y(k,l) = C(k) \sum_{m=1}^4 \left[C(l) \sum_{n=1}^4 x(m,n) \sin \frac{(2k-1)k\pi}{9} \right] \sin \frac{(2l-1)k\pi}{9}, \quad k,l=1,2,3,4, \quad (11)$$

where

$$C(k) = C(l) = \frac{2}{3}. \quad (12)$$

Similar to DCT, the two-dimensional DST can be decomposed into two one-dimensional DST which can be expressed as:

$$\mathbf{Y} = \mathbf{B}\mathbf{X}, \quad (13)$$

where \mathbf{X} represents the image block, and \mathbf{Y} is the transformed DST coefficient matrix. The transform matrix \mathbf{B} can be represented as:

$$\mathbf{B}(m,k) = C(k) \sin \frac{(2k-1)m\pi}{9}, \quad m,k=1,2,3,4. \quad (14)$$

Taking 4×4 image block for example, the transformation matrix \mathbf{B} can be expressed as:

$$\mathbf{B} = \frac{2}{3} \cdot \begin{bmatrix} \sin \frac{\pi}{9} & \sin \frac{2\pi}{9} & \sin \frac{3\pi}{9} & \sin \frac{4\pi}{9} \\ \sin \frac{3\pi}{9} & \sin \frac{3\pi}{9} & 0 & -\sin \frac{3\pi}{9} \\ \sin \frac{5\pi}{9} & -\sin \frac{\pi}{9} & -\sin \frac{3\pi}{9} & \sin \frac{2\pi}{9} \\ \sin \frac{7\pi}{9} & -\sin \frac{4\pi}{9} & \sin \frac{3\pi}{9} & -\sin \frac{\pi}{9} \end{bmatrix}. \quad (15)$$

Let $a = \frac{2}{3} \sin \frac{\pi}{9}$, $b = \frac{2}{3} \sin \frac{2\pi}{9}$, $c = \frac{2}{3} \sin \frac{3\pi}{9}$, and $d = \frac{2}{3} \sin \frac{4\pi}{9}$, then matrix \mathbf{B} can be rewritten as:

$$\mathbf{B} = \begin{bmatrix} a & b & c & d \\ c & c & 0 & -c \\ d & -a & -c & b \\ b & -d & c & -a \end{bmatrix}. \quad (16)$$

The elements in matrix \mathbf{B} are magnified 128 times, and the integer DST transform matrix can be obtained:

$$\mathbf{H} = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}. \quad (17)$$

Like the 4×4 integer DCT, the elements in integer DST transform matrix are also multiplied by 128. Therefore, the formula of the two-dimensional integer DST for 4×4 coding block in HEVC can be represented as:

$$\mathbf{Y} = (\mathbf{H}\mathbf{X}\mathbf{H}^T) \otimes (\mathbf{E} \otimes \mathbf{E}^T) \\ = \left\{ \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} \mathbf{X} \begin{bmatrix} 29 & 74 & 84 & 55 \\ 55 & 74 & -29 & -84 \\ 74 & 0 & -74 & 74 \\ 84 & -74 & 55 & -29 \end{bmatrix} \right\} \cdot \frac{1}{128} \cdot \frac{1}{128}. \quad (18)$$

Similar to the integer DCT, the correction process “ $\otimes(\mathbf{E} \otimes \mathbf{E}^T)$ ” of the integer DST will also be completed in quantization process after transform.

3. Derivation of Integer APBT

In [27], the APBT based on DST was deduced which is called the all phase discrete sine biorthogonal transform (APDSBT). It achieves good performance in JPEG-like coding than DCT. In this section, the derivation process of the integer APBT will be introduced. According to the HEVC standard, the integer APBT can be deduced directly by scaling up. Due to the special relationship between APIDCBT and DCT, the APBT can be applied to HEVC at the minimal cost. Moreover, the APIDCBT has good performance in JPEG-like coding. Therefore, APIDCBT will represent for APBT in the paper.

3.1 Principle and Characteristics of APIDCBT

APIDCBT can be obtained by APDF based on IDCT, and the derivation process is the same as APDSBT in [27]. The transform matrix $\mathbf{V}(i, j)$ can be represented as:

$$\mathbf{V}(i, j) = \begin{cases} \frac{1}{N}, & i = 0, j = 0, 1, \dots, N - 1, \\ \frac{N - i + \sqrt{2} - 1}{N^2} \cos \frac{i(2j + 1)\pi}{2N}, & i = 1, 2, \dots, N - 1, j = 0, 1, \dots, N - 1. \end{cases} \quad (19)$$

where N is the size of the original matrix, and i as well as j are the coordinates in the matrix. If $N = 8$,

$$\mathbf{V} = \begin{bmatrix} 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 \\ 0.1136 & 0.0963 & 0.0644 & 0.0226 & -0.0226 & -0.0644 & -0.0963 & -0.1136 \\ 0.0926 & 0.0384 & -0.0384 & -0.0926 & -0.0926 & -0.0384 & 0.0384 & 0.0926 \\ 0.0703 & -0.0165 & -0.0830 & -0.0470 & 0.0470 & 0.0830 & 0.0165 & -0.0703 \\ 0.0488 & -0.0488 & -0.0488 & 0.0488 & 0.0488 & -0.0488 & -0.0488 & 0.0488 \\ 0.0296 & -0.0523 & 0.0104 & 0.0444 & -0.0444 & -0.0104 & 0.0523 & -0.0296 \\ 0.0144 & -0.0349 & 0.0349 & -0.0144 & -0.0144 & 0.0349 & -0.0349 & 0.0144 \\ 0.0043 & -0.0123 & 0.0184 & -0.0217 & 0.0217 & -0.0184 & 0.0123 & -0.0043 \end{bmatrix}.$$

Among the three kinds of APBTs, APIDCBT has the best energy concentration characteristic. Similar to APDSBT, APIDCBT can also achieve good performance in JPEG-like image coding. APIDCBT is closely related to DCT, and in theory, it also has fast algorithm. What's more, it can further improve the coding performance of HEVC without changing the coding framework.

3.2 Integer APIDCBT

The integer APBT can be derived by two methods. The derivation of the first method is similar to the integer DCT in HEVC standard. The integer APBT transform matrix is magnified for $64\sqrt{N}$ times, and then the integer APBT transform matrix is used for HEVC video coding. However, this method has low computational efficiency because APBT has no fast algorithm currently. The second method obtains the integer APBT transform matrix by analyzing the relationship between APIDCBT and DCT. The integer APBT obtained by DCT can use the "butterfly operation" in DCT, which can improve the HEVC coding performance without increasing the complexity of the algorithm. Therefore, we choose the second method to derive the integer APBT.

According to [28], the relationship between the APIDCBT matrix \mathbf{V} and DCT transform matrix \mathbf{C} is that $\mathbf{V} = \mathbf{DC}$, where \mathbf{D} and its inverse matrix \mathbf{D}^{-1} are both diagonal matrices. The expressions of \mathbf{D} and \mathbf{D}^{-1} are represented as Eq. (20) and Eq. (21), respectively.

$$\mathbf{D} = \frac{1}{N\sqrt{N}} \begin{bmatrix} N & 0 & 0 & \cdots & 0 \\ 0 & 1 + \frac{N-2}{\sqrt{2}} & 0 & \cdots & 0 \\ 0 & 0 & 1 + \frac{N-3}{\sqrt{2}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad (20)$$

$$\mathbf{D}^{-1}(i,i) = \begin{cases} \sqrt{N}, & i=0, \\ \frac{N\sqrt{2N}}{N-i+\sqrt{2}-1}, & i=1,2,\dots,N-1. \end{cases} \quad (21)$$

Therefore, the inverse transform matrix \mathbf{V}^{-1} of APIDCBT can be expressed as $\mathbf{V}^{-1} = \mathbf{C}^{-1}\mathbf{D}^{-1} = \mathbf{C}^T\mathbf{D}^{-1}$. Because of $\mathbf{D}(i,i) = 1/\mathbf{D}^{-1}(i,i)$, the two-dimensional APIDCBT can be expressed as:

$$\begin{aligned} Y_{\text{APBT}} &= \mathbf{V}\mathbf{X}\mathbf{V}^T = (\mathbf{D}\mathbf{C})\mathbf{X}(\mathbf{D}\mathbf{C})^T = \mathbf{D}\mathbf{C}\mathbf{X}\mathbf{C}^T\mathbf{D}^T = \mathbf{D}(\mathbf{C}\mathbf{X}\mathbf{C}^T)\mathbf{D} \\ &= \mathbf{D}\mathbf{Y}_{\text{DCT}}\mathbf{D} = \mathbf{Y}_{\text{DCT}} \otimes \mathbf{Q}^*, \end{aligned} \quad (22)$$

where $\mathbf{Q}^*(u, v) = 1/[\mathbf{D}^{-1}(u, u)\mathbf{D}^{-1}(v, v)]$, $u = 0, 1, \dots, N-1$, $v = 0, 1, \dots, N-1$.

The two-dimensional integer DCT is expressed as $\mathbf{Y}_{\text{DCT}} = (\mathbf{C}\mathbf{X}\mathbf{C}^T) \otimes \mathbf{E}'$, where \mathbf{E}' is the correction matrix. Therefore, the two-dimensional integer APIDCBT can be expressed as $\mathbf{Y}_{\text{APIDCBT}} = (\mathbf{C}\mathbf{X}\mathbf{C}^T) \otimes \mathbf{E}' \otimes \mathbf{Q}^*$. In H.264 standard, the transform process only needs matrix multiplication operation “ $\mathbf{C}\mathbf{X}\mathbf{C}^T$ ” while the correction process “ $\otimes \mathbf{E}' \otimes \mathbf{Q}^*$ ” is conducted in quantification process. For example, in H.264, $\mathbf{C}_{8 \times 8}$ is the integer DCT transform matrix with size of 8×8 , and its corresponding correction matrix is $\mathbf{E}' = \mathbf{E}_{8 \times 8} \otimes \mathbf{E}_{8 \times 8}^T$. $\mathbf{C}_{8 \times 8}$ and $\mathbf{E}_{8 \times 8}$ are represented as:

$$\mathbf{C}_{8 \times 8} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}, \quad \mathbf{E}_{8 \times 8} = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & b & b & b & b & b & b & b \\ c & c & c & c & c & c & c & c \\ b & b & b & b & b & b & b & b \\ a & a & a & a & a & a & a & a \\ b & b & b & b & b & b & b & b \\ c & c & c & c & c & c & c & c \\ b & b & b & b & b & b & b & b \end{bmatrix}, \quad (23)$$

where $a = \frac{1}{16\sqrt{2}}$, $b = \frac{1}{17\sqrt{2}}$, and $c = \frac{1}{8\sqrt{5}}$. The correction matrix \mathbf{Q}^* between APIDCBT and DCT is:

$$\mathbf{Q}^* = \begin{bmatrix} 8.0000 & 12.2076 & 14.1108 & 16.7170 & 20.5041 & 26.5097 & 37.4903 & 64.0000 \\ 12.2076 & 18.6282 & 21.5324 & 25.5094 & 31.2883 & 40.4524 & 57.2083 & 97.6607 \\ 14.1108 & 21.5324 & 24.8893 & 29.4846 & 36.1662 & 46.7591 & 66.1273 & 112.8864 \\ 16.7170 & 25.5096 & 29.4864 & 34.9325 & 42.8461 & 55.3954 & 78.3410 & 133.7364 \\ 20.5041 & 31.2883 & 36.1662 & 42.8461 & 52.5525 & 67.9448 & 96.0884 & 164.0331 \\ 26.5097 & 40.4524 & 46.7591 & 55.3951 & 67.9448 & 87.8453 & 124.2320 & 212.0773 \\ 37.4903 & 57.2083 & 66.1273 & 78.3410 & 96.0884 & 124.2320 & 175.6906 & 299.9227 \\ 64.0000 & 97.6607 & 112.8864 & 133.7364 & 164.0331 & 212.0773 & 299.9227 & 512.0000 \end{bmatrix}.$$

In HEVC, the integer APIDCBT is obtained by multiplying it with a large number, and it can be decomposed into transforms that have close correlation with DCT. The correction matrix \mathbf{Q}^* is used in quantization process of HEVC standard. From the above analysis, the two-dimensional integer APIDCBT with size of 8×8 can be adopted in HEVC coding. The specific expression of integer APIDCBT for 8×8 image block is finally expressed as:

$$\mathbf{Y}_{\text{APIDCBT}} = (\mathbf{C}_{8 \times 8} \mathbf{X} \mathbf{C}_{8 \times 8}^T) \otimes \mathbf{E}_8 \otimes \mathbf{E}_8^T \otimes \mathbf{Q}^*, \quad (24)$$

where $\mathbf{E}_8 = \frac{1}{64\sqrt{8}} = \frac{1}{181}$. Similarly, the integer APIDCBT for other blocks with different sizes can also be obtained. Then, it can be applied in the HEVC framework for better performance.

4. Application of APBT in HEVC Video Coding Scheme

HEVC is a system with high coding efficiency, but it has high computational complexity in encoding process. DCT and DST are two main transforms in HEVC scheme. In this section, we will analyze their roles in HEVC coding. Based on this analysis, we use APBT to replace DCT to further improve the HEVC coding performance. The APBT-HEVC video coding algorithm is designed on the basis of HEVC standard model (HM).

Fig. 1 shows the calling relationship of some functions in HM. The test programs of HM involve eight parts: TAppCommon, TAppDecoderAnalyze, TAppDecoder, TAppEncoder, TLibCommon, TLibDecoder, TLibEncoder, and TLibVideoIO. The encoder program takes the encoding function as an entrance and calls CompressGOP to conduct the encoding process. Firstly, the CompressGOP traverses the GOP, and each frame in GOP is encoded individually. Next, each frame is divided into many slices, and each slice is further divided into CUs. Each CU is compressed by calling compressCU recursively. In xCompressCU function, the intra-prediction and inter-prediction are executed by calling functions xCheckRDCostIntra and xCheckRDCostInter. Taking the function xCheckRDCostIntra as an example, it can process the luminance component and chrominance component by calling functions estIntraPredQT and estIntraPredChromaQT, respectively. The selection of rate-distortion cost (RDCost) is controlled by function estIntraPredQT, while the direction prediction is achieved by the function predIntraLumaAng.

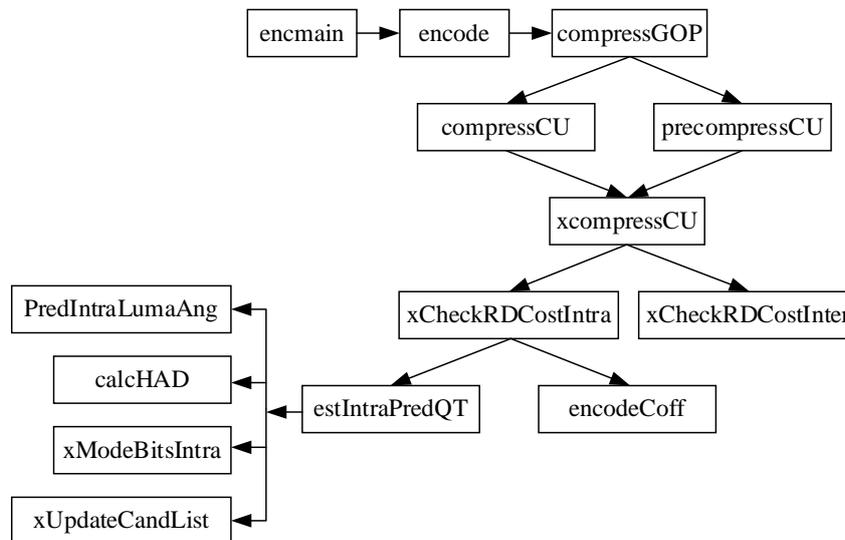


Fig. 1. The calling relationship of some functions in HM.

To improve the coding efficiency, each CTU in HEVC standard is split into smaller CUs according to quadtree-based segmentation. At each depth, the rough mode decision (RMD) and rate-distortion optimization (RDO) operations will be applied to the corresponding prediction unit (PU), and the RDCost can be obtained. Next, further division for current CU is implemented, and the sum of RDCost in four Sub-CUs is calculated and compared with the RDCost of the CU in previous layers. If the RDCost in Sub-CUs is smaller than that of CU, the current CU will be divided continually until the block size of CU is 8×8 . Otherwise, the current CU will not be divided. When the block size of CU is 8×8 , its corresponding PU has two sizes: 4×4 and 8×8 . The size of PU is the same as CU when the size of CU is larger than 8×8 . In prediction process of HEVC, the cost functions of all the prediction modes will be

calculated to obtain the optimal prediction mode. As shown in Fig. 2, the CUs will be encoded according to the order of Zigzag scanning.

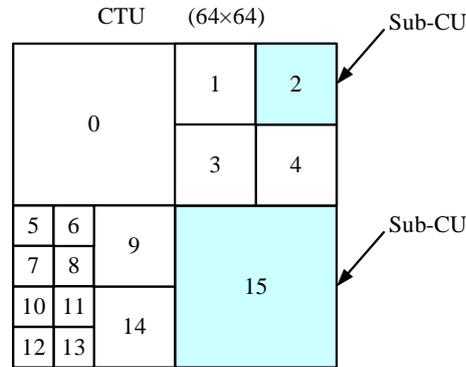


Fig. 2. Coding sequence of CU in HEVC.

4.1 APBT-Based HEVC Video Coding Algorithm

The framework of HEVC coding adopts a hybrid model based on blocks. Firstly, the HEVC encoder divides the input video sequence into GOPs, and each frame in GOP is further divided into CUs. According to the spatial and temporal correlation of video sequence, each CU is encoded in predictive encoding process. This process is adopted to eliminate the spatial and temporal redundancies in the video sequence by intra-prediction encoding and inter-prediction encoding. Next, the corresponding residual coefficients are obtained by the difference between blocks in predicted and original images, and then the residual coefficients are transformed and quantified. Finally, the quantized coefficients are encoded into HEVC bit stream by entropy coding.

In HEVC standard, DCT, DST, and the quantization process are involved in prediction stage to obtain the optimal TU. The residual coefficients in the optimal mode are transformed, and then the entropy coding is implemented to the quantized coefficients. The APBT-based HEVC video encoding framework in this paper is shown in Fig. 3, which has two differences compared with the standard HEVC coding algorithm using DCT and DST. One is that the DCT and DST are replaced by APBT, and the other is the quantization table in HEVC is replaced by the uniform quantization. There are two quantization schemes in HEVC: rate-distortion quantization and the quantization using standard quantization table. The ScalingList in configuration file is used to control the selection of quantization scheme. In this experiment, the quantization table is utilized to complete the quantization process. Correspondingly, the APBT-based HEVC video coding algorithm in this paper can be also implemented in two schemes. The first one uses the integer APBT with different sizes to replace DCT and DST in HEVC directly, and the uniform quantization is adopted in the quantization stage. The second one makes full use of the relationship between APBT and DCT, and conducts the correction process in quantization stage. Since there is no fast algorithm in APBT currently, the computational efficiency of the second method is slightly higher than that of the first one.

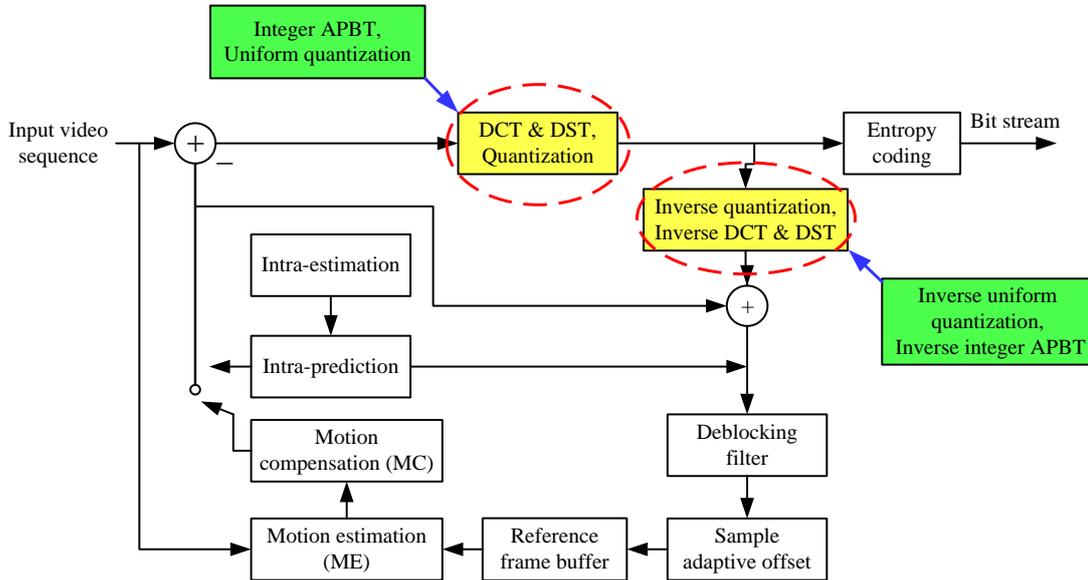


Fig. 3. APBT-based HEVC video encoding framework.

The HEVC decoding process is relatively simple, and the APBT-based HEVC video decoding framework is shown in Fig. 4. The decoding process is the inverse procedure of the encoding process. Firstly, entropy decoding is implemented to the bit stream obtained from HEVC coding. The decoded data is reordered according to the order in encoding process, and then the inverse quantization and inverse transform are applied to the reordered data correspondingly. Next, the reconstructed frame can be obtained by superimposing the residual coefficients after inverse transform and the image blocks after prediction in coding side. After repeating the above-mentioned processes, the reconstructed video sequence can be obtained finally.

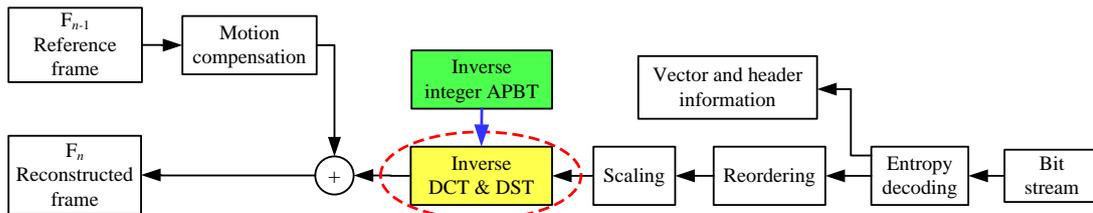


Fig. 4. APBT-based HEVC video decoding framework.

4.2 Optimization of APBT-HEVC Video Coding Algorithm

In fact, a lot of experiments show that the HEVC compression method based on single APBT cannot get a satisfactory result by directly replacing the DCT and DST in HEVC standard using APBT. To improve the performance of the APBT-based HEVC framework, we will make further optimization for transforms in this section. The hybrid transform scheme, which is mixed by APBT, DST, and DCT, is designed. For the transform blocks with different sizes in different modes, different transforms are adopted to obtain the optimal coding performance. To get a better coding framework, a large number of experiments are executed, and the following scheme is determined ultimately. For luminance components, the DCT and DST in

HEVC standard are still used; for chrominance components, APBT is adopted for transform blocks with sizes of 8×8 , 16×16 , and 32×32 while DCT is used for transform blocks with size of 4×4 . Finally, the framework of HEVC encoding based on hybrid APBT is formed, which is shown in Fig. 5. In the decoding side, the inverse DST (IDST), IDCT, and inverse APBT (IAPBT) are adopted to complete the inverse transform. As it can be seen in Fig. 5, the current frames are transformed by DST, DCT, and APBT, respectively, and the quantization step is performed in the following. Then, reordering and entropy decoding are executed for generating the bit stream. Besides, motion estimation (ME) and motion compensation (MC) are utilized for the reference frame to get the reconstructed frame.

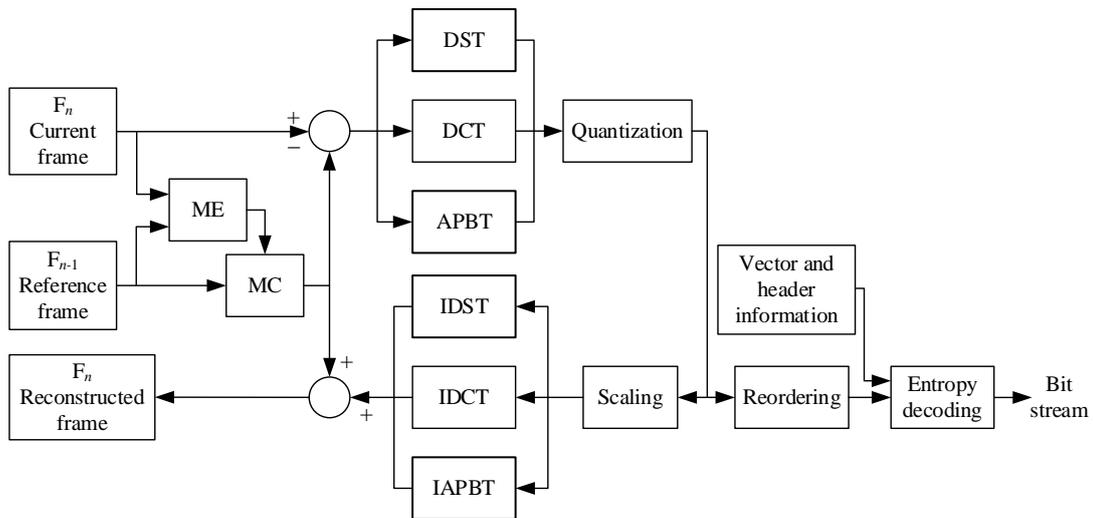


Fig. 5. HEVC encoding framework based on hybrid APBT.

5. Experimental Results and Analysis

To verify the performance of the APBT-based HEVC coding algorithm, some simulation experiments are carried out. The hardware environment in this paper is: Intel Core i3-2100 3.1 GHz CPU and 6 GB RAM. The software environment is: Visual Studio 2013, Windows 10 64-bit professional edition, and HEVC test model 16.0. The video sequences with different resolutions (2560×1600 , 1920×1080 , 1024×768 , and 832×480) are selected to test this algorithm. The configuration file `encoder_lowdelay_main.cfg` is used to set parameters of the algorithm. The `IntraPeriod` is set to 16 and the `GOP size` is set to 8. The `DecodingRefreshType` is set to 1. There is no P frame. Therefore, the frame structure is I, B, B, ..., B. The frame rate is set to 30. The `ScalingList` is set to 1 and the `QP` is set to 42. Other parameters in HM are set to the default values.

As mentioned above, the transforms for luminance components in HEVC standard are not changed in the proposed scheme. Therefore, we only compare the peak signal-to-noise ratio (PSNR) values of the chrominance components (average PSNR of the U and V) in each reconstructed frame, which are obtained by using different video coding schemes. The video sequence used in this experiment is BasketballDrill sequence (resolution: 832×480 , format: YUV) and the comparison result is shown in Fig. 6. From Fig. 6, it can be observed that the coding performance of chrominance components is slightly improved by using the HEVC video coding algorithm based on hybrid APBT.

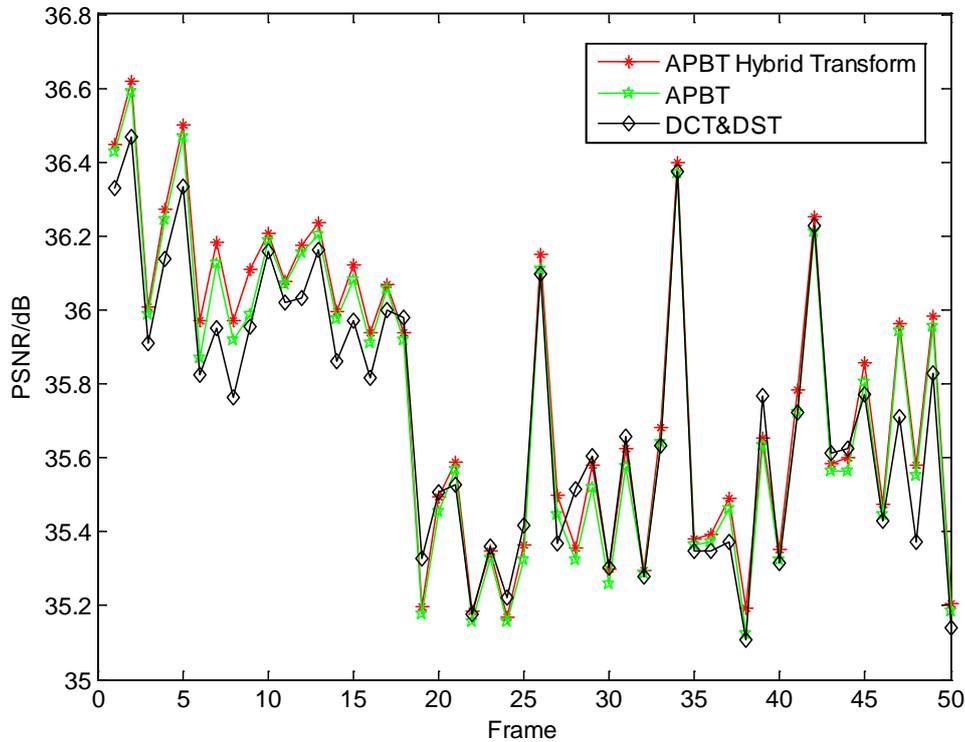


Fig. 6. PSNR comparisons of the chrominance components using different video coding algorithms for video sequence BasketballDrill.

To analyze the coding performance of the proposed scheme for video sequences with different resolutions, **Table 1** gives the coding performance comparisons of four different video sequences with different resolutions and the same QP. The four different sequences used in this paper are BasketballDrill (resolution: 832×480, format: YUV), ChinaSpeed (resolution: 1024×768, format: YUV), BQTerrace (resolution: 1920×1080, format: YUV), and Traffic (resolution: 2560×1600, format: YUV), respectively. To test the coding performance of the proposed scheme under different QPs, **Table 2** gives the coding performance comparisons of the chrominance components under different QPs. The video sequence used in **Table 2** is the BasketballDrill video sequence. From **Table 1** and **Table 2**, we can draw the conclusion that the coding performance of the proposed scheme is improved by about 0.3% for video sequences with different resolutions and QPs.

Table 1. Comparisons of experimental results for video sequences with different resolutions

Test sequence	PSNR/dB			Performance improvement /%
	DCT & DST	APBT	Hybrid APBT	
BasketballDrill	36.20	36.23	36.31	0.30
ChinaSpeed	37.58	37.57	37.73	0.39
BQTerrace	28.03	27.90	28.12	0.32
Traffic	31.67	31.59	31.75	0.25

Table 2. Experimental results of BasketballDrill under different QPs

QP	PSNR/dB			
	DCT & DST	APBT	Hybrid APBT	Performance improvement /%
22	44.77	44.79	44.91	0.31
28	41.29	41.30	41.40	0.26
32	39.65	39.70	39.78	0.33
36	37.92	37.91	38.07	0.39
40	36.90	36.89	37.01	0.29
44	35.89	35.89	35.98	0.25
47	34.17	34.20	34.30	0.38

To evaluate the subjective effect of the proposed HEVC coding algorithm based on hybrid APBT, the first frame in BasketballDrill video sequence is compressed and reconstructed by using the proposed method. The reconstructed frame is then compared with the reconstructed frame obtained by standard HM model. Fig. 7 shows the reconstructed frames obtained by the standard HM model and the proposed method when QP=22. It can be seen from Fig. 7 that there is almost no difference between the two reconstructed frames.



Fig. 7. Reconstructed frames of BasketballDrill (QP=22): (a) the standard HEVC and (b) the improved HEVC based on hybrid APBT.

Fig. 8 gives the reconstructed frames obtained by the standard HM model and the proposed method when $QP=47$. It indicates that the subjective performance of HEVC video coding based on hybrid APBT is superior to that of the HM model when QP has a relative large value. Besides, compared with the HEVC standard, the block artifacts of the proposed scheme are reduced to a certain extent.



Fig. 8. Reconstructed frames of BasketballDrill ($QP=47$): (a) the standard HEVC and (b) the improved HEVC based on hybrid APBT.

6. Conclusions

In this paper, the integer DCT and integer DST in HEVC standard are introduced firstly. By drawing from their derivation processes, the integer APBT is derived, and the relationship between integer APBT and integer DCT is analyzed. Based on this analysis, a new APBT-based HEVC video coding algorithm is proposed in this paper. To further improve its coding performance, the proposed scheme is then optimized by using the strategy of hybrid transforms. Experimental results show that the proposed algorithm can improve the coding performance of chrominance components by about 0.3%.

To further improve the coding efficiency, in the next research, we will apply the parallel computing technology based on GPU to the proposed algorithm. That means we will propose a

novel HEVC video coding based on parallel APBT. Besides, more comparisons with other recent work will be realized in the future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61702303, No. 61201371); the Natural Science Foundation of Shandong Province, China (No. ZR2017MF020, No. ZR2015PF004); and the Research Award Fund for Outstanding Young and Middle-Aged Scientists of Shandong Province, China (No. BS2013DX022). The authors thank Yunpeng Zhang, Heng Zhang, Zhi Zhang, and Xiaoyan Yu for their kind help and valuable suggestions. Moreover, the authors thank the anonymous reviewers for their valuable comments to improve the presentation of the paper.

References

- [1] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36-58, Sept. 2001. [Article \(CrossRef Link\)](#).
- [2] F. Dufaux, G. J. Sullivan, and T. Ebrahimi, "The JPEG XR image coding standard," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 195-199, Nov. 2009. [Article \(CrossRef Link\)](#).
- [3] H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1043-1053, Apr. 1998. [Article \(CrossRef Link\)](#).
- [4] ISO/IEC and ITU-T, "Information Technology -- Digital Compression and Coding of Continuous-tone Still Images - Part 1: Requirements and Guidelines," ISO/IEC 10918-1: 1994 | ITU-T Rec. T.81, Sept. 2011. [Article \(CrossRef Link\)](#).
- [5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 23, no. 1, pp. 90-93, Jan. 1974. [Article \(CrossRef Link\)](#).
- [6] ISO/IEC, "Information Technology -- Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video," ISO/IEC 13818-2: 2013, Sept. 2013. [Article \(CrossRef Link\)](#).
- [7] ISO/IEC, "Information Technology -- Coding of Audio-Visual Objects - Part 2: Visual," ISO/IEC 14496-2: 2004, Aug. 2015. [Article \(CrossRef Link\)](#).
- [8] 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: 2014, Aug. 2014. [Article \(CrossRef Link\)](#).
- [9] ISO/IEC, "Information Technology -- High Efficiency Coding and Media Delivery in Heterogeneous Environments - Part 2: High Efficiency Video Coding," ISO/IEC 23008-2: 2015, Aug. 2015. [Article \(CrossRef Link\)](#).
- [10] X. B. Sun, X. D. Chen, Y. Xu, Y. Wang, and D. Y. Yu, "Efficient coding unit partition strategy for HEVC intra coding," *Journal of Electronic Imaging*, vol. 26, no. 4, pp. 1-8, Jul. 2017. [Article \(CrossRef Link\)](#).
- [11] K. H. Tai, M. Y. Hsieh, M. J. Chen, C. Y. Chen, and C. H. Yeh, "A fast HEVC encoding method using depth information of collocated CUs and RD cost characteristics of PU modes," *IEEE Transactions on Broadcasting*, vol. 63, no. 4, pp. 680-692, Dec. 2017. [Article \(CrossRef Link\)](#).
- [12] C. Chen, S. Y. Zhu, B. Zeng, and M. Gabbouj, "A new block-based coding method for HEVC intra coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1727-1736, Aug. 2017. [Article \(CrossRef Link\)](#).
- [13] M. H. Tang, X. Y. Chen, J. T. Wen, and Y. X. Han, "Hadamard transform based optimized HEVC video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1-13, Feb. 2018. [Article \(CrossRef Link\)](#).

- [14] K. Goswami and B. G. Kim, "A design of fast high efficiency video coding (HEVC) scheme based on Markov chain Monte Carlo model and Bayesian classifier," *IEEE Transactions on Industrial Electronics*, pp. 1-11, Mar. 2018. [Article \(CrossRef Link\)](#).
- [15] Z. Y. Yang, Q. L. Shao, and S. X. Guo, "Fast coding algorithm for HEVC based on video contents," *IET Image Processing*, vol. 11, no. 6, pp. 343-351, Jun. 2017. [Article \(CrossRef Link\)](#).
- [16] M. Budagavi, A. Fuldseth, and G. Bjontegaard, "Core transform design in the high efficiency video coding (HEVC) standard," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1029-1041, Dec. 2013. [Article \(CrossRef Link\)](#).
- [17] Z. X. Hou and X. Yang, "The all phase DFT filter," in *Proc. of the 10th IEEE Digital Signal Processing (DSP) Workshop and the 2nd IEEE Signal Processing Education (SPE) Workshop*, Pine Mountain, Georgia, USA, Oct. 13-16, 2002, pp. 221-227. [Article \(CrossRef Link\)](#).
- [18] 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. [Article \(CrossRef Link\)](#).
- [19] 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. [Article \(CrossRef Link\)](#).
- [20] C. X. Zhang, C. Y. Wang, and B. C. Jiang, "Color image compression based on directional all phase biorthogonal transform," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 1, pp. 247-254, Jan. 2015. [Article \(CrossRef Link\)](#).
- [21] Q. M. Fu, X. Zhou, C. Y. Wang, and B. C. Jiang, "Windowed all phase biorthogonal transform and its application in JPEG-like image compression," *Journal of Communications*, vol. 10, no. 4, pp. 284-293, Apr. 2015. [Article \(CrossRef Link\)](#).
- [22] C. Y. Wang, B. C. Jiang, and S. Z. Xie, "Properties of all phase biorthogonal transform matrix and its application in color image compression," *Journal of Computational Information Systems*, vol. 9, no. 18, pp. 7227-7234, Sept. 2013. [Article \(CrossRef Link\)](#).
- [23] B. C. Jiang, C. X. Zhang, C. Y. Wang, and X. Y. Wang, "Video compression algorithm based on all phase biorthogonal transform and MPEG-2," *International Journal of Hybrid Information Technology*, vol. 8, no. 3, pp. 145-153, Mar. 2015. [Article \(CrossRef Link\)](#).
- [24] 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, Dec. 2015. [Article \(CrossRef Link\)](#).
- [25] C. X. Zhang, C. Y. Wang, and B. C. Jiang, "Video compression algorithm based on directional all phase biorthogonal transform and H.263," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 3, pp. 189-198, Mar. 2016. [Article \(CrossRef Link\)](#).
- [26] 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, Apr. 2015. [Article \(CrossRef Link\)](#).
- [27] R. Y. Shan, X. Zhou, C. Y. Wang, and B. C. Jiang, "All phase discrete sine biorthogonal transform and its application in JPEG-like image coding using GPU," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 9, pp. 4467-4486, Sept. 2016. [Article \(CrossRef Link\)](#).
- [28] Q. M. Fu, X. Zhou, C. Y. Wang, and B. C. Jiang, "Mathematical relation between APBT-based and DCT-based JPEG image compression schemes," *Journal of Communications*, vol. 11, no. 1, pp. 84-92, Jan. 2016. [Article \(CrossRef Link\)](#).



Chengyou Wang 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 at Shandong University, Weihai, China. His current research interests include digital image/video processing and analysis, computer vision, and machine learning.



Rongyang Shan received his B.E. degree in communication engineering from Shandong University, Weihai, China, in 2014, and his M.E. degree in signal and information processing from Shandong University, China, in 2017. He currently works in the Beijing Jingdong Century Trading Co., Ltd., Beijing, China. His research interests include image/video transform coding, parallel computing, and video watermarking.



Xiao Zhou 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 at Shandong University, Weihai, China. Her current research interests include wireless communication technology, image communication, and computer vision.