All phase biorthogonal transform and its application in JPEG-like image compression


School of Electronic Information Engineering, Tianjin University, Tianjin 300072, PR China

ABSTRACT

This paper proposes new concepts of the all phase biorthogonal transform (APBT) and the dual biorthogonal basis vectors. In the light of all phase digital filtering theory, three kinds of all phase biorthogonal transforms based on the Walsh transform (WT), the discrete cosine transform (DCT) and the inverse discrete cosine transform (IDCT) are proposed. The matrices of APBT based on WT, DCT and IDCT are deduced, which can be used in image compression instead of the conventional DCT. Compared with DCT-based JPEG (DCT-JPEG) image compression algorithm at the same bit rates, the PSNR and visual quality of the reconstructed images using these transforms are approximate to DCT, outgoing DCT-JPEG at low bit rates especially. But the advantage is that the quantization table is simplified and the transform coefficients can be quantized uniformly. Therefore, the computing time becomes shorter and the hardware implementation easier.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

To achieve a greater convenience for image storage and transmission, the development of multimedia and Internet requires a more suitable method for image compression. Discrete cosine transform (DCT) [1] has been adopted widely in international standards for image compression, for example, JPEG [2,3], MPEG-2 [4], MPEG-4 [5] and H.264/AVC [6,7]. In spite of the emerging wavelet-based standard JPEG2000 [8], the DCT-based JPEG standard remains to be the most commonly employed lossy compression algorithm for still images due to its high effectiveness and low computational complexity. In DCT-based lossy image compression algorithm, the quantization table taken into consideration the vision characteristics of human eyes is used. However, adopting different quantization steps for different DCT coefficients makes the quantization table more complex. In particular, more complex multiplications are required when adjusting the bit rates.

In this paper, we bring forward a new transform, namely the all phase biorthogonal transform (APBT), which is based on Walsh transform (WT), DCT and inverse discrete cosine transform (IDCT). It can be applied to image compression successfully instead of the conventional DCT. The advantage of the method is that the quantization table is simpler; and the
transform coefficients can be quantized uniformly, the computation simplified, and the computation time shortened. It can be easily implemented by software and hardware. Besides, it can achieve the same performance as DCT in image compression, outgoing DCT-JPEG at low bit rates especially.

The rest of this paper is organized as follows. Section 2 briefly introduces the conventional DCT and JPEG image compression algorithm. Based on sequency filtering, three kinds of all phase biorthogonal transforms, including the all phase Walsh biorthogonal transform (APWBT), the all phase discrete cosine biorthogonal transform (APDCBT) and the all phase inverse discrete cosine biorthogonal transform (APIDCBT) are proposed in Section 3. The applications of APWBT, APDCBT and APIDCBT in image compression are described in Section 4. In Section 5, experimental results and comparisons with conventional JPEG algorithm are presented; rate distortion curves of two typical images are plotted; and some reconstructed images are also displayed. Finally, conclusions and discussions for further research are given in Section 6.

2. The conventional DCT and JPEG image compression algorithm

2.1. The conventional DCT

The conventional 2-D DCT is always implemented separately by two 1-D DCTs. Let us use \( X \) and \( C \) to denote an image block and the DCT matrix with size of \( N \times N \), respectively. After the conventional 2-D DCT, the transform coefficient block \( Y \) can be expressed as

\[
Y = CX^T, 
\]

where \( C^T \) is the transpose matrix of \( C \),

\[
C(i,j) = \begin{cases} \frac{1}{\sqrt{N}}, & i = 0, \ 0 \leq j \leq N - 1, \\ \frac{2}{\sqrt{2N}} \cos \left( \frac{(2j + 1)\pi}{2N} \right), & 1 \leq i \leq N - 1, \ 0 \leq j \leq N - 1. \end{cases} 
\]  

(1)

Since DCT is an orthogonal transform, i.e. \( C^T = C^{-1} \), we use

\[
X = C^{-1}Y(C^T)^{-1} = C^{-1}Y(C^{-1})^T = C^{-1}YC
\]

to reconstruct the image. The column vectors of \( C \) and \( C^{-1} \) are the basis vectors of the forward transform (DCT) and inverse transform (IDCT), respectively. The outer products of column vectors of \( C^{-1} \) compose 2-D basis images for DCT. Fig. 1 shows the basis vectors of \( C \) and \( C^{-1} \).

2.2. DCT-based JPEG image compression algorithm (DCT-JPEG)

The encoder of DCT-JPEG is mainly composed of four parts: forward discrete cosine transform (FDCT), quantization, zig-zag scan, and entropy encoder. In the encoding process, the input image is grouped into blocks. Prior to computing the FDCT, the input image data are level shifted to a signed two’s complement representation. For 8-bit input precision, the level shift is achieved by subtracting 128. And then, each block is transformed by the FDCT into 64 DCT coefficients, the DC coefficient and the 63 AC coefficients. After quantization, the DC coefficient of each block is coded in a differential pulse code modulation (DPCM). The 63 quantized AC coefficients are converted into a 1-D zig-zag sequence, preparing for

Fig. 1. The basis vectors \((N = 8): (a) the basis vectors of \( C \) and (b) the basis vectors of \( C^{-1} \).
entropy encoding. The decoder also contains four major parts: entropy decoder, inverse zig-zag scan, dequantization, and IDCT, which performs essentially the inverse of its corresponding main procedure within the encoder.

3. Construction of APBT matrices

3.1. Sequency filtering in time domain

‘Frequency’ is defined as: the whole period number (or a half of the number of crossing-zero point) of a sine function in a time unit. However, frequency is just applicable to sine (periodic) function. In this case, crossing-zero points of this function are equidistant in an interval. ‘Extensive frequency’ [9] which is also named sequency [10] is defined as: a half of the number of crossing-zero point of a function in a time unit. It is used to differentiate these functions whose crossing-zero points are non-equidistant in an interval. Meanwhile, they do not have to be periodic. As to sine function, sequency and frequency are identical.

Fig. 2 shows the scheme of conventional sequency filtering in transform domain, where \( X = [x(0), x(1), \ldots, x(N-1)] \) is the N-D vector obtained from the digital sequence \( \{x(n)\} \) in time domain; \( T \) and \( T^{-1} \) are the forward and inverse transform matrices of discrete orthogonal transform with size of \( N \times N \), respectively; \( F \) is the N-D sequency response vector; \( Y = [y(0), y(1), \ldots, y(N-1)] \) is the signal after filtering in time domain. The scheme of sequency filtering in time domain is shown in Fig. 3, where \( H \) is the matrix of sequency filtering. According to Figs. 2 and 3 yields

\[
Y = T^{-1} (F \otimes TX),
\]

where the symbol \( \otimes \) means the multiplication operator of the corresponding elements. The elements of \( Y \) are

\[
y(i) = \sum_{l=0}^{N-1} T^{-1}(i,l) \left\{ F(l) \sum_{j=0}^{N-1} T(l,j)x(j) \right\} = \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} T^{-1}(i,l)T(l,j)F(l)x(j) = \sum_{j=0}^{N-1} H(i,j)x(j), \quad i = 0, 1, \ldots, N-1,
\]

i.e.

\[
Y =HX,
\]

where

\[
H(i,j) = \sum_{l=0}^{N-1} T^{-1}(i,l)T(l,j)F(l), \quad i, j = 0, 1, \ldots, N-1.
\]

3.2. All phase sequency filtering

The basic idea of all phase filtering [11] is derived from the superimposing digital filter. Based on mathematical analysis, a general approach to designing the superimposing digital filters based on WT, Fourier transform and IDCT was proposed by Hou [12]. A convolution algorithm in temporal domain for the discrete cosine sequency filtering was deducted and a new linear phase digital filter was introduced in [13].

For a digital signal sequence \( \{x(n)\} \), there are \( N \) N-D vectors that contain \( x(n) \) and have different intercepted phases:

\[
X_0 = [x(n), x(n + 1), \ldots, x(n + N - 1)]^T,
\]

\[
X_i = z^{-i}X_0 = [x(n - i), x(n - i + 1), \ldots, x(n - i + N - 1)]^T,
\]

\[
X \xrightarrow{T} \bigcirc \xrightarrow{T^{-1}} Y
\]

\[F\]

**Fig. 2.** Conventional sequency filtering in transform domain.

\[
X \xrightarrow{\otimes} Y
\]

\[H\]

**Fig. 3.** Sequency filtering in time domain.
where \( i = 1, 2, \ldots, N - 1 \), \( z^{-1} \) is the delay operator. Obviously, \( x(n) \) is the intersection of these vectors, \( x(n) = X_0 \cap X_1 \cap \cdots \cap X_{N-1} \). Applying sequency filtering to each vector in time domain, we are interested in the response of the same point \( x(n) \). It would yield \( N \) different responses, respectively,

\[
Y_i(i) = HX_i, \quad i = 0, 1, \ldots, N - 1.
\]

In order to eliminate the different meanings of the filtering values and reduce the blocking artifacts, we take the mean of these \( N \) values as the filtering output:

\[
y(n) = \frac{1}{N} \sum_{i=0}^{N-1} Y_i(i).
\]

According to Eqs. (4)–(6), we get

\[
y(n) = \frac{1}{N} \sum_{i=0}^{N-1} H(i,j)x(n - i + j) = \sum_{j=0}^{N-1} h(j)x(n - j),
\]

where \( h(j) \) is the unit impulse response of all phase sequency filter:

\[
h(j) = \begin{cases} 
\frac{1}{N} \sum_{i=j}^{N-1} H(i,j), & j = 0, 1, \ldots, N - 1, \\
\frac{1}{N} \sum_{i=0}^{N-1-j} H(i,j), & j = -1, -2, \ldots, -N + 1. 
\end{cases}
\]

If \( H \) is a symmetric matrix, i.e. \( H(i,j) = H(j,i) \), yields

\[
h(j) = h(-j), \quad j = 0, 1, \ldots, N - 1.
\]

Eq. (9) indicates \( h \) has zero-phase characteristic. Let \( \vec{h} = [h(0), h(1), \ldots, h(N-1)]^T \), obviously, if we know \( \vec{h} \), we can get \( h(j), \quad j = -N + 1, -N + 2, \ldots, N - 2, N - 1 \) by using the symmetric property.

### 3.3. All phase sequency filtering based on WT

When applying WT to sequency filtering, suppose \( N = 2^k, \quad k \in \mathbb{N} \), we have

\[
T(i,j) = W_k(i,j), \quad T^{-1}(i,j) = W_k^{-1}(i,j) = \frac{1}{N} W_k(i,j),
\]

where \( W_k \) is the WT matrix with size of \( N \times N \), \( W_k(i,j) = W_k(j,i) \). Suppose \( H_k \) be the matrix of sequency filtering, according to Eq. (3), the elements of \( H_k \) are

\[
H_k(i,j) = \frac{1}{N} \sum_{l=0}^{N-1} W_k(i,l)W_k(l,j)F_k(l) = \frac{1}{N} \sum_{l=0}^{N-1} W_k(i \oplus j,l)F_k(l) = \hat{F}_k(i \oplus j),
\]

where \( \hat{F}_k \) is the Walsh transform of the response vector \( F_k \), and the symbol \( \oplus \) means the dyadic sum. Eq. (10) shows that \( H_k \) is a symmetric matrix. According to Eq. (9), we know that the all phase sequency filter \( h \) is a zero-phase digital filter. Suppose \( \vec{h}_k = [h_k(0), h_k(1), \ldots, h_k(N-1)]^T \), obviously, if we know \( \vec{h}_k \), we can get \( h_k(j), \quad j = -N + 1, -N + 2, \ldots, N - 2, N - 1 \) by using the symmetric property. According to Eqs. (8) and (10), we have

\[
h_k(j) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{l=0}^{N-1} W_k(i,l)W_k(l,i-j)F_k(l) = \sum_{l=0}^{N-1} \left[ \frac{1}{N^2} \sum_{i=j}^{N-1} W_k(i,l)W_k(l,i-j) \right] F_k(l) = \sum_{l=0}^{N-1} S_k(j,l)F_k(l), \quad j = 0, 1, \ldots, N - 1,
\]

i.e.

\[
h_k = S_k F_k,
\]

where \( F_k \) is \( N-D \) sequency response vector; \( S_k \) is the all phase WT matrix with size of \( N \times N \),

\[
S_k(j,l) = \frac{1}{N^2} \sum_{i=j}^{N-1} W_k(i,l)W_k(l,i-j) = \frac{1}{N^2} \sum_{i=0}^{N-1-j} W_k(i+j,l)W_k(l,i).
\]

\[
S_k(j,l) = \frac{1}{N^2} \sum_{i=j}^{N-1} W_k(i,l)W_k(l,i-j) = \frac{1}{N^2} \sum_{i=0}^{N-1-j} W_k(i+j,l)W_k(l,i).
\]
For example, when \( k = 3, N = 2^k = 8 \),

\[
W_3 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
\end{bmatrix}
\]

\[
S_j = \frac{1}{64} \begin{bmatrix}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
7 & 5 & 3 & 1 & -1 & -3 & -5 & -7 \\
6 & -2 & -6 & -6 & 2 & 2 & 6 & 6 \\
5 & -1 & -3 & 1 & 3 & 1 & -5 & -5 \\
4 & -4 & 4 & 4 & -4 & -4 & 4 & 4 \\
3 & -3 & -1 & 1 & -1 & 1 & 3 & -3 \\
2 & -2 & 2 & -2 & 2 & -2 & 2 & 2 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
\end{bmatrix}
\]

In order to optimize the all phase WT matrix, we let \( k = 3, W_k = V_0, S_k = V_1, \) Eq. (13) can be rewritten as

\[
V_{ij} = \frac{1}{64} \sum_{i=0}^{7-j} V_0(i + j, l) V_0(l, i),
\]

(14)

and then we can summarize it to the following expression:

\[
V_m(i, j) = \frac{1}{64} \sum_{i=0}^{7-j} V_{m-1}(i + j, l) V_{m-1}(l, i), \quad m \in \mathbb{N}.
\]

(15)

For example, after four iterations, we have

\[
V_5 = \frac{1}{64} \begin{bmatrix}
8.0000 & 8.0000 & 8.0000 & 8.0000 & 8.0000 & 8.0000 & 8.0000 & 8.0000 \\
7.0000 & 5.1833 & 3.6220 & 2.0186 & 0.7104 & -4.8184 & -4.5786 & -9.1372 \\
5.0000 & -1.0781 & -4.1499 & -3.4166 & -0.9594 & 4.8113 & 6.1078 & -6.3152 \\
4.0000 & -1.8218 & -3.9411 & -0.3862 & 4.2072 & 0.2307 & -5.3196 & 3.0309 \\
3.0000 & -3.3642 & -1.1966 & 2.2302 & 0.4331 & -3.5140 & 3.5731 & -1.1616 \\
2.0000 & -3.1933 & 0.9833 & 1.7602 & -2.6011 & 2.1770 & -1.3342 & 0.2080 \\
1.0000 & -2.2280 & 1.8363 & -1.3542 & 1.0357 & -0.3971 & -0.0758 & 0.1831 \\
\end{bmatrix}
\]

(16)

### 3.4. All phase sequency filtering based on DCT

When applying DCT to sequency filtering, we have

\[
T(i, j) = C(i, j), \quad T^{-1}(i, j) = C^{-1}(i, j),
\]

where \( C \) is the DCT matrix with size of \( N \times N, C^{-1}(i, j) = C^T(i, j). \) According to Eq. (3), the elements of \( H \) are

\[
H(i, j) = \sum_{l=0}^{N-1} C^{-1}(i, l) C(j, l) F(l) = \sum_{l=0}^{N-1} C^T(i, l) C^T(j, l) F(l), \quad i, j = 0, 1, \ldots, N - 1.
\]

(17)

According to Eqs. (1) and (17), we have

\[
H(i, j) = \frac{1}{N} \left[ F(0) + \sum_{l=1}^{N-1} 2 \cos \left( \frac{2l + 1}{2N} \pi \right) \cos \left( \frac{2j + 1}{2N} \pi \right) F(l) \right].
\]

(18)

Eq. (18) indicates that \( H(i, j) = H(j, i) \). According to Eq. (9), the discrete cosine sequency filter \( h \) also has zero-phase property. Suppose \( h_{1/2} = [h(0), h(1), \ldots, h(N - 1)]^T \), if we know \( h_{1/2} \), we can get \( h \) by using the symmetric property.

According to Refs. [11,13], the design of 1-D all phase discrete cosine sequency filter \( h \) with the length of \( 2N - 1 \) is composed of Eqs. (9), (19) and (20):

\[
h_{1/2} = [h(0), h(1), \ldots, h(N - 1)]^T = V F,
\]

(19)
where the elements of $V$ are

$$V(i, j) = \begin{cases} 
\frac{N-i}{N^2}, & i = 0, 1, \ldots, N-1, \ j = 0, \\
\frac{1}{N^2} \left[ (N-i) \cos \frac{j \pi i}{N} \csc \frac{j \pi}{N} \sin \frac{j \pi i}{N} \right], & i = 0, 1, \ldots, N-1, \ j = 1, 2, \ldots, N-1.
\end{cases}$$

(20)

For example, when $N = 8$,

$$V = \begin{bmatrix}
0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 & 0.1250 \\
0.1094 & 0.0854 & 0.0617 & 0.0262 & -0.0156 & -0.0575 & -0.0930 & -0.1167 \\
0.0938 & 0.0374 & -0.0221 & -0.0783 & -0.0938 & -0.0543 & 0.0221 & 0.0952 \\
0.0781 & -0.0078 & -0.0709 & -0.0657 & 0.0156 & 0.0787 & 0.0396 & -0.0676 \\
0.0625 & -0.0408 & -0.0625 & 0.0169 & 0.0625 & -0.0169 & -0.0625 & 0.0408 \\
0.0469 & -0.0557 & -0.0175 & 0.0498 & -0.0156 & -0.0368 & 0.0488 & -0.0198 \\
0.0313 & -0.0510 & 0.0221 & 0.0101 & -0.0313 & 0.0341 & -0.0221 & 0.0068 \\
0.0156 & -0.0301 & 0.0267 & -0.0216 & 0.0156 & -0.0096 & 0.0046 & -0.0012
\end{bmatrix}$$

3.5. All phase sequency filtering based on IDCT

When applying IDCT to sequency filtering, we have

$$T(i, j) = C^{-1}(i, j), \quad T^{-1}(i, j) = C(i, j),$$

where $C$ is the DCT matrix with size of $N \times N$, $C^{-1}(i, j) = C^T(i, j)$. According to Eq. (3), the elements of $H$ are

$$H(i, j) = \sum_{l=0}^{N-1} C(i, l) C^{-1}(l, j) F(l) = \sum_{l=0}^{N-1} C(i, l) C^T(l, j) F(l) = \sum_{l=0}^{N-1} C(i, l) C(j, l) F(l), \quad i, j = 0, 1, \ldots, N-1.$$ (21)

Eq. (21) indicates that $H(i, j) = H(j, i)$. According to Eq. (9), the discrete inverse cosine sequency filter $h$ also has zero-phase property. Suppose $h_{1/2} = [h(0), h(1), \ldots, h(N-1)]^T$, if we know $h_{1/2}$, we can get $h$ by using the symmetric property.

According to Refs. [11,14], the design of 1-D all phase inverse discrete cosine sequency filter $h$ with the length of $2N-1$ is composed of Eqs. (9), (19) and (22). Here, the elements of $V$ in Eq. (19) are

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

(22)

For example, when $N = 8$,

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

3.6. All phase biorthogonal transform (APBT)

Just like transform matrix $C$ in DCT, according to Eqs. (15), (20) and (22), we know that the column vectors of $V(V_m)$ have the sequency properties too, i.e. the sequency increases with the increasing column number. In addition, since they are full rank matrices, they have the inverse matrices $V^{-1}(V_m^{-1})$. Though $V(V_m)$ are not orthogonal matrices, i.e. $V^T \neq V^{-1}$, they have the following biorthogonal relationships:

$$V V^{-1} = I, \quad V^{-1} V = I.$$ (23)

where $I$ is the unit matrix with size of $N \times N$. Therefore, $V$ can be considered as new transform matrices. Similar to DCT matrix $C$, they can be used in image compression transforming the image from spatial domain to frequency domain too.
Definition 1. For 1-D vector \( x \), we define the APBT as \( y = Vx \) and the inverse APBT (IAPBT) \( x = V^{-1}y \). We define the 2-D APBT as
\[
Y = VXV^T,
\]
and the IAPBT is
\[
X = V^{-1}Y(V^{-1})^T,
\]
where \( X \) denotes an image matrix, \( Y \) denotes the matrix of transform coefficients. The corresponding transforms are defined as the all phase Walsh biorthogonal transform (APWBT), the all phase discrete cosine biorthogonal transform (APDCBT), and the all phase inverse discrete cosine biorthogonal transform (APIDCBT), respectively.

Definition 2. We define \( V(V_m) \) and \( V^{-1}(V_m^{-1}) \) as the APBT matrices which are dual biorthogonal, and the column vectors of \( V \) and \( V^{-1} \) are the basis vectors of APBT and IAPBT. They make up a pair of dual biorthogonal basis vectors. The outer product of column vectors of \( V^{-1} \) compose 2-D basis images for APBT.
When \( N = 8 \), suppose \( V = [x_0, x_1, \ldots, x_8] \), \( V^{-1} = [\beta_0, \beta_1, \ldots, \beta_7] \), \( Y = [y(i,j)(i,j = 0, 1, \ldots, 7)] \), where \( x_j(i = 0, 1, \ldots, 7) \) and \( \beta_j(j = 0, 1, \ldots, 7) \) are the basis vectors of \( V \) and \( V^{-1} \), respectively. Then Eq. (25) can be rewritten as

\[
X = V^{-1}Y(V^{-1})^T = [\beta_0, \beta_1, \ldots, \beta_7]Y[\beta_0, \beta_1, \ldots, \beta_7]^T = \sum_{i=0}^{7} \sum_{j=0}^{7} y(i,j)\beta_i\beta_j^T = \sum_{i=0}^{7} \sum_{j=0}^{7} y(i,j)B_{ij},
\]

where \( B_{ij} = \beta_i\beta_j^T \) are the basis images of APBT. According to Eq. (26), we have

\[
y(i,j) = \langle X, B_{ij} \rangle.
\]

Eq. (26) shows that the image block \( X \) can be expressed as the weighted sum of transform coefficients \( y(i,j) \) and the corresponding basis images \( B_{ij} \), while Eq. (27) shows that each coefficient \( y(i,j) \) is the inner product of image block \( X \) and \( B_{ij} \).

For example, Fig. 4 shows the basis vectors of \( V \) and \( V^{-1} \) in APIIDCBT; Fig. 5 shows the basis images of APDCBT and APIDCBT when \( N = 8 \).

4. Application of APBT in image compression

4.1. Quantized uniformly

From Figs. 4(b) and 1(b), we can see that the similarity between the matrices \( V^{-1} \) of APBT and the matrix \( C^{-1} \) of DCT is that the sequency increases with the increasing column number. The difference between them is that all basis vectors of \( C^{-1} \) have the same norm, but the norm of basis vectors in \( V^{-1} \) increases with the increasing sequency.

We define the energy of image as the Frobenius norm of the image matrix

\[
E_{B_{ij}} = \|B_{m,n}\|^2_F = \sum_{i=0}^{7} \sum_{j=0}^{7} |b_{ij}|^2,
\]

where \( B_{ij} = [b_{ij}], i,j = 0, 1, \ldots, 7 \). For example, the normalized energy of APIDCBT basis images is as follows:

\[
E_{\text{APIDCBT}} = \begin{bmatrix} 1.00 & 2.33 & 3.11 & 4.37 & 6.57 & 10.98 & 21.96 & 64.00 \\ 2.33 & 5.42 & 7.24 & 10.17 & 15.30 & 25.57 & 51.14 & 149.03 \\ 3.11 & 7.24 & 9.68 & 13.59 & 20.44 & 34.16 & 68.33 & 199.11 \\ 4.37 & 10.17 & 13.59 & 19.07 & 28.68 & 47.95 & 95.90 & 279.46 \\ 6.57 & 15.30 & 20.44 & 28.68 & 43.15 & 72.13 & 144.27 & 420.42 \\ 10.98 & 25.57 & 34.16 & 47.95 & 72.13 & 120.57 & 241.15 & 702.76 \\ 21.96 & 51.14 & 68.33 & 95.90 & 144.27 & 241.15 & 482.30 & 1405.50 \\ 64.00 & 149.03 & 199.11 & 279.46 & 420.42 & 702.76 & 1405.50 & 4096.00 \end{bmatrix}
\]

According to Fig. 5 and Eq. (29), we can see that the energy of basis images for APBT increases with the increasing frequency along the down/right diagonal, which results in the high-frequency attenuation characteristics of the coefficients of APBT. That is to say, the value of transform coefficients in each block has a decreasing tendency along the down/right diagonal. Therefore, when the transform coefficients are quantized by the same step (i.e. uniformly), it is equivalent to the low-frequency coefficients are quantized by small step and high-frequency coefficients quantized by big one. Similar quantization effects are achieved similar to the case in DCT-JPEG that uses the complex quantization table.

From another point of view, we plot the normalized frequency responses of the filters in DCT and APBT, as shown in Fig. 6. We can see that APBT has better compression performance by giving more emphasis to the low-frequency band than that of DCT. In the APBT-JPEG algorithm, frequency weighting is already accomplished by the transform step, and therefore it is more appropriate to utilize a uniform quantizer instead of standard JPEG quantization matrix. And APBT is more suitable for image compression than DCT.

4.2. Modified JPEG algorithm based on APBT (APBT-JPEG)

The proposed scheme of APBT-JPEG image codec is shown in Fig. 7. The basic processes are similar to the DCT-JPEG algorithm. The differences between them are the transform (DCT or APBT) and quantizer (quantization table or uniform quantization step). Other steps of image compression algorithm are identical to DCT-based JPEG.

5. Experimental results and comparisons with DCT-JPEG

In order to test the proposed scheme and compare with the DCT-JPEG algorithm, in this section, we will present simulation results obtained by applying the proposed transforms in Section 3 to test typical images Lena and Barbara, both of size \( 512 \times 512 \), 8 bpp, monochrome images. Throughout this paper, all experiments are conducted with Matlab 6.5.
In APWBT-JPEG algorithm, $S_3$ and $V_m$ can be used. Experimental results show that $V_5$ works better than $S_3$ and other $V_m$. Therefore, we choose $V_5$ as the APWBT matrix in the following experiment. In APDCBT-JPEG and APIDCBT-JPEG, $V$ (when $N = 8$) is used as the transform matrix. In the quantization part, the uniform quantization step $q$ is adopted. While in simple baseline DCT-JPEG algorithm, $C$ is used in transform step with the following quantization table $Q$ (see [2]), and the

![Diagram](image.png)

**Fig. 6.** Frequency responses of the filters in DCT and APBT: (a) DCT, (b) APWBT, (c) APDCBT and (d) APIDCBT.

![Diagram](image.png)

**Fig. 7.** The proposed scheme of APBT-JPEG image codec.
quality of compressed image is controlled by the parameter:

\[
Q = \begin{bmatrix}
16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\
12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\
14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\
14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\
18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\
24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\
49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\
72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 
\end{bmatrix}
\] (30)

Both of the methods use the typical Huffman tables [2] which have been developed from the average statistics of a large set of images with 8-bit precision. The distortion is measured by the PSNR

\[
\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \text{ (dB)},
\]

where MSE denotes the mean squared error between the original and reconstructed images. Tables 1 and 2 show the experimental results with DCT-JPEG, APWBT-JPEG, APDCBT-JPEG and APIDCBT-JPEG in terms of PSNR at different bit rates, applied to image Lena and Barbara, respectively.

From Tables 1 and 2, we conclude that compared with DCT-JPEG algorithm at the same bit rates in terms of PSNR, the performance of APBT-JPEG algorithm is close to conventional DCT-JPEG in image compression. The proposed algorithm is superior about 0.3 dB at low bit rates. In particular, among the three transforms, APIDCBT-JPEG has the best performance, outgoing DCT-JPEG at various bit rates. For the sake of clarity, according to the experimental results in Tables 1 and 2, we plot the ratio distortion curves of Lena and Barbara, as shown in Figs. 8(a) and (b), respectively. In order to compare the compression performance subjectively, Fig. 8 shows the reconstructed image Lena of size 512 \times 512 obtained by using DCT-JPEG, APWBT-JPEG, APDCBT-JPEG and APIDCBT-JPEG at 0.20 bpp. It is clear that, as compared with the DCT-JPEG version shown in Fig. 9(a), better visual quality of APBT-JPEG version (shown in Figs. 9(b)–(d)) is achieved and the blocking artifacts has been reduced significantly. No obvious blocking artifacts can be perceived easily.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR comparison of DCT-JPEG and APBT-JPEG applied to image Lena.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit rate (bpp)</th>
<th>DCT-JPEG PSNR (dB)</th>
<th>APWBT-JPEG PSNR (dB)</th>
<th>APDCBT-JPEG PSNR (dB)</th>
<th>APIDCBT-JPEG PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>25.82</td>
<td>26.74</td>
<td>26.76</td>
<td>26.86</td>
</tr>
<tr>
<td>0.20</td>
<td>28.91</td>
<td>29.12</td>
<td>29.16</td>
<td>29.27</td>
</tr>
<tr>
<td>0.25</td>
<td>30.69</td>
<td>30.55</td>
<td>30.67</td>
<td>30.89</td>
</tr>
<tr>
<td>0.30</td>
<td>31.92</td>
<td>31.83</td>
<td>31.72</td>
<td>32.00</td>
</tr>
<tr>
<td>0.40</td>
<td>34.74</td>
<td>33.14</td>
<td>33.30</td>
<td>33.69</td>
</tr>
<tr>
<td>0.50</td>
<td>35.61</td>
<td>34.22</td>
<td>34.40</td>
<td>34.86</td>
</tr>
<tr>
<td>0.60</td>
<td>35.41</td>
<td>35.08</td>
<td>35.27</td>
<td>35.79</td>
</tr>
<tr>
<td>0.75</td>
<td>36.62</td>
<td>36.08</td>
<td>36.33</td>
<td>36.87</td>
</tr>
<tr>
<td>1.00</td>
<td>37.93</td>
<td>37.37</td>
<td>37.63</td>
<td>38.22</td>
</tr>
<tr>
<td>1.25</td>
<td>39.00</td>
<td>38.40</td>
<td>38.65</td>
<td>39.27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR comparison of DCT-JPEG and APBT-JPEG applied to image Barbara.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit rate (bpp)</th>
<th>DCT-JPEG PSNR (dB)</th>
<th>APWBT-JPEG PSNR (dB)</th>
<th>APDCBT-JPEG PSNR (dB)</th>
<th>APIDCBT-JPEG PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>22.04</td>
<td>22.52</td>
<td>22.56</td>
<td>22.60</td>
</tr>
<tr>
<td>0.20</td>
<td>23.47</td>
<td>23.54</td>
<td>23.66</td>
<td>23.79</td>
</tr>
<tr>
<td>0.25</td>
<td>24.36</td>
<td>24.26</td>
<td>24.39</td>
<td>24.62</td>
</tr>
<tr>
<td>0.30</td>
<td>25.11</td>
<td>24.96</td>
<td>25.14</td>
<td>25.46</td>
</tr>
<tr>
<td>0.40</td>
<td>26.55</td>
<td>26.35</td>
<td>26.59</td>
<td>27.04</td>
</tr>
<tr>
<td>0.50</td>
<td>27.90</td>
<td>27.60</td>
<td>27.85</td>
<td>28.39</td>
</tr>
<tr>
<td>0.60</td>
<td>29.20</td>
<td>28.70</td>
<td>28.99</td>
<td>29.67</td>
</tr>
<tr>
<td>0.75</td>
<td>30.88</td>
<td>30.10</td>
<td>30.49</td>
<td>31.39</td>
</tr>
<tr>
<td>1.00</td>
<td>33.22</td>
<td>32.11</td>
<td>32.53</td>
<td>33.59</td>
</tr>
<tr>
<td>1.25</td>
<td>35.10</td>
<td>33.74</td>
<td>34.19</td>
<td>35.38</td>
</tr>
</tbody>
</table>
From the experimental results, we can see that compared with conventional DCT-JPEG, the proposed algorithm improve the performance at low bit rates, both in terms of PSNR and visual quality. In APBT-JPEG scheme, due to the adoption of one quantization step and omission of the memory space of the quantization table, when adjusting the bit rates, each image block can save 63 multiplication operations between quantization factor and quantization table.

We execute time consumption experiments on an IBM ThinkPad R61i 7732-2DC Laptop (1.60 GHz Intel Pentium Dual-Core T2330 CPU, 1 GB RAM, SATA 5400 rpm HD) installed with Windows XP operating system. Here, the run-time is the time spent from the beginning to the end of encoding process. Experimental results on Lena image show that APBT-JPEG based on WT/DCT/IDCT is 0.2 s shorter than conventional DCT-JPEG at various bit rates on average.

Finally, we would like to point out that the proposed scheme is also applied to other test images, and similar results can be obtained. Since the APBT-JPEG algorithm does not modify the basic compression algorithm, they can be used in other DCT-based codec without introducing substantial modifications.
6. Conclusions

In this paper, the all phase biorthogonal transform (APBT) and the matrices of APWBT, APDCBT and APIDCBT were deduced based on the theory of all phase digital filtering. Considering the all phase thoughts, we successfully achieved the compression and reconstruction of image with these transforms by replacing DCT which was commonly used in image compression. Compared with DCT-based JPEG, similar performance was obtained. The advantage of the proposed algorithm was the simple quantization, taking uniform quantization for transform coefficients, especially saving much multiplication operations when adjusting the bit rates. A simpler and more effective algorithm is therefore developed, and can be easily implemented in both software and hardware.

Actually, the proposed algorithm has good performance at low bit rates. But when image is compressed at very low bit rates, the visually disturbing blocking artifacts generates. Better performance may be realized by taking post-processing techniques. In addition, we have not applied the proposed algorithm to other DCT-based schemes, such as MPEG-2, H.264/AVC, etc. Besides, we only discussed the application of APBT in image compression. We can continue to explore other applications of APBT, such as image representation, edge detection, etc. These issues are left for future research. We can foresee that the proposed transforms will be widely used in the fields of image processing.

Acknowledgments

This work was supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China (no. 20060056051). The authors would like to thank Master Sai-Yuan Zhao and Li-Xiao Zhang for their help and valuable suggestions. Also, they thank the reviewers of this paper and the area editor for their valuable and detailed comments that helped to improve the presentation of the paper.

References