Mathematical Relation between APBT-Based and DCT-Based JPEG Image Compression Schemes

Qiming Fu, Xiao Zhou, Chengyou Wang, and Baochen Jiang
School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China
Email: fqmsdu@163.com; {zhouxiao, wangchengyou, jbc}@sdu.edu.cn

Abstract —A novel linear transform, called All Phase Biorthogonal Transform (APBT), was generated from All Phase Digital Filter (APDF) theory. APBT can be used in image compression instead of the conventional Discrete Cosine Transform (DCT), and the corresponding image compression scheme is called APBT-based JPEG (APBT-JPEG) which can achieve better coding performance, especially at low bit rates. With in-depth mathematical analysis on the emerging APBT-JPEG and conventional DCT-based JPEG (DCT-JPEG), we bring forward a unique insight into the relation between them. The relation is that APBT-JPEG can be implemented by DCT-JPEG, using a new quantization table deduced from mathematical analysis. To the best of our knowledge, it is the first time to reveal it, which can be considered to the main contribution of this paper. Finally, experiment results obtained with the test images have verified our proposed conclusion both in terms of objective quality and subjective effect.

Index Terms—Image compression, All Phase Biorthogonal Transform (APBT), Discrete Cosine Transform (DCT), quantization table, JPEG, All Phase Digital Filter (APDF)

I. Introduction

In spite of the emerging Discrete Wavelet Transform (DWT)-based standard JPEG2000 [1] and Lapped Biorthogonal Transform (LBT)-based standard JPEG-XR [2], the Discrete Cosine Transform (DCT)-based JPEG standard [3] also known as the baseline JPEG remains to be the most commonly employed lossy compression algorithm for still images due to its high effectiveness and low computational complexity. A block diagram of DCT-based JPEG (DCT-JPEG) coding algorithm is shown in Fig. 1. In the encoding process, the input image is first divided into 8×8 blocks. Each block is transformed into 64 DCT coefficients via 2-dimensional DCT. The 64 DCT coefficients are then quantized by a 64-element quantization table. The DC coefficients of successive blocks are differentially coded and the 63 AC coefficients are zig-zag scanned, run-length coded, and followed by a Huffman coder. The decoding performs the reverse of these steps. In DCT-JPEG, the default quantization table provided by JPEG is widely used, which takes into consideration the vision characteristics of human eyes. Adopting different quantization step sizes for different DCT coefficients makes the quantization table more complex. In consequence, more complex multiplications are required when adjusting the bit rates. In addition, DCT-JPEG also suffers from some annoying blocking artifacts at low bit rates, which results in visible discontinuities across block boundaries [4].

At this point, three kinds of linear transforms: All Phase Biorthogonal Transforms (APBTs) based on Walsh-Hadamard Transform (WHT), DCT, and inverse DCT (IDCT) were proposed [5] to replace the conventional DCT in baseline JPEG. Corresponding image compression scheme is called APBT-based JPEG (APBT-JPEG), in which the transform coefficients can be quantized uniformly with simplified quantization table. In fact, among these APBTs, APBT based on IDCT always achieves best coding performance. When it applied to image compression, coding performance outperforms DCT-JPEG at various bit rates. Especially at low bit rates, the blocking artifacts of reconstructed image are reduced significantly. Thus, our research work focuses on APBT based on IDCT (called APBT in the following of paper).

Due to these superiorities of APBT, its theory and application have been researched further in recent years [6]-[10]. However, to the best of our knowledge, there is no fast algorithm for computing APBT. As a result, with APBT replacing DCT in baseline JPEG, extra burden on computational complexity is required in the transform step. Recently, in order to improve the real-time performance of APBT-JPEG, a parallel framework to speed up it on GPU was proposed [11].

Having introduced DCT-JPEG and APBT-JPEG in brief, in this paper, we bring forward a unique insight into the relation between them. Relation between APBT-JPEG and DCT-JPEG provides another perspective to cognize APBT-JPEG, and interestingly, this is a better way to illustrate why APBT-JPEG produces better reconstructed image. However, readers may be confused about where the APBT comes from. Therefore, we give the answer in our own way before depicting the proposed relation.

The rest of this paper is organized as follows. Section II tells readers where the APBT comes from. Then in Section III, the relation between DCT-JPEG and APBT-JPEG is first revealed. In order to verify our proposed conclusion, simulation experiments are done and the

Manuscript received July 13, 2015; revised January 6, 2016.
This work was supported by the Natural Science Foundation of Shandong Province, China under Grant No. ZR2015PF004, the National Natural Science Foundation of China under Grant No. 61201371, and the promotive research fund for excellent young and middle-aged scientists of Shandong Province, China under Grant No. BS2013DX022.
Corresponding author email: zhouxiao@sdu.edu.cn.
doi:10.12720/jcm.11.1.84-92

corresponding experiment results are presented in Section IV. Finally, we summarize the paper in Section V.

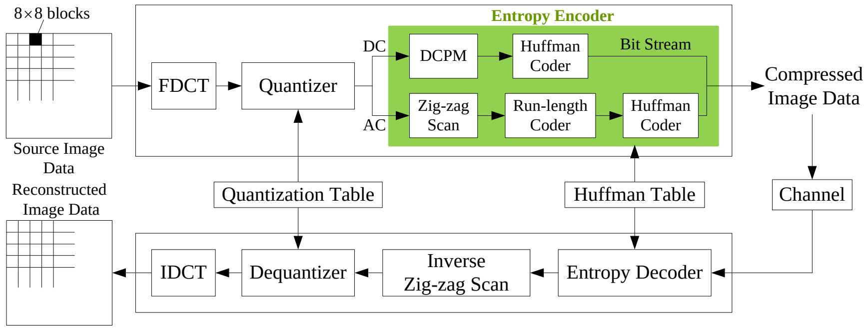


Fig. 1. The scheme of DCT-JPEG (baseline JPEG) image codec. The compression itself is performed in four sequential steps: DCT computation, quantization, zig-zag scan and entropy coding.

II. Where is the APBT from?

A. All Phase Philosophy

All phase philosophy can be considered to an application of overlap philosophy. For a digital sequence $\{x(n)\}$, there are N vectors which are N-dimensional. Each vector contains $x(n)$ and has different intercept phases:

$$\begin{aligned}
\boldsymbol{X}_0 &= [x(n), x(n+1), \cdots, x(n+N-1)]^{\mathrm{T}}, \\
\boldsymbol{X}_1 &= z^{-1}\boldsymbol{X}_0 = [x(n-1), x(n), \cdots, x(n+N-2)]^{\mathrm{T}}, \\
&\vdots \\
\boldsymbol{X}_{N-1} &= z^{-(N-1)}\boldsymbol{X}_0 = [x(n-N+1), x(n-N+2), \cdots, x(n)]^{\mathrm{T}},
\end{aligned} \tag{1}$$

where z^{-j} ($j = 0,1,\cdots,N-1$) is the delay operator. Obviously, $x(n)$ is the intersection of $\boldsymbol{X}_i (i = 0,1,\cdots,N-1)$, that is $x(n) = \boldsymbol{X}_0 \cap \boldsymbol{X}_1 \cap \cdots \cap \boldsymbol{X}_{N-1}$. According to the conventional representation of data matrices, the all phase data matrix of $x(n)$ is defined as $\boldsymbol{A}_N(n) = [\boldsymbol{X}_0, \boldsymbol{X}_1, \cdots, \boldsymbol{X}_{N-1}]$, and it is spanned by the column vectors $\boldsymbol{X}_i (i = 0,1,\cdots,N-1)$.

B. Principle of All Phase Digital Filter (APDF) Based on IDCT

APDF proposed in [12] is a new scheme for 1-dimensional digital FIR filter. Its theory and application were researched further in recent years [13], [14]. APDF is designed on the basis of all phase philosophy, and it is superior to conventional filter in the overall performance.

Note that we denote DCT matrix as $\boldsymbol{C}$ in this paper and it is defined in Eq. (2). Since DCT is an orthogonal transform, the IDCT matrix is $\boldsymbol{C}^{\mathrm{T}}$.

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

To help readers understand the APDF better, the process of 1-dimensional signal passing through the APDF based on IDCT [15] is shown in Fig. 2.

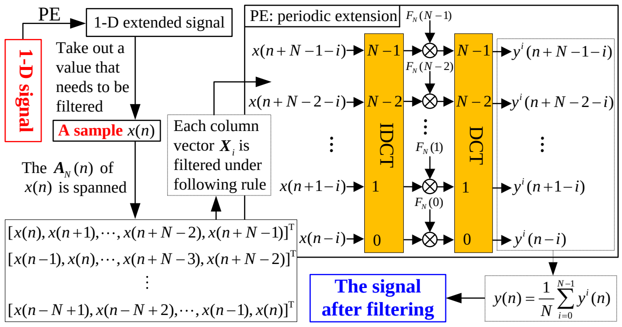


Fig. 2. 1-dimensional signal flowing through the APDF based on IDCT.

In APDF based on IDCT, $\boldsymbol{F}$ that is defined by user is N-dimensional expected sequency response vector and sequency is also named as 'extensive frequency' [5].

$$\boldsymbol{F} = [F_N(0), F_N(1), \cdots, F_N(N-1)]^{\mathrm{T}} \tag{3}$$

The relation between input signal $x(n)$ and output signal $y(n)$ in APDF based on IDCT is shown in Fig. 3.

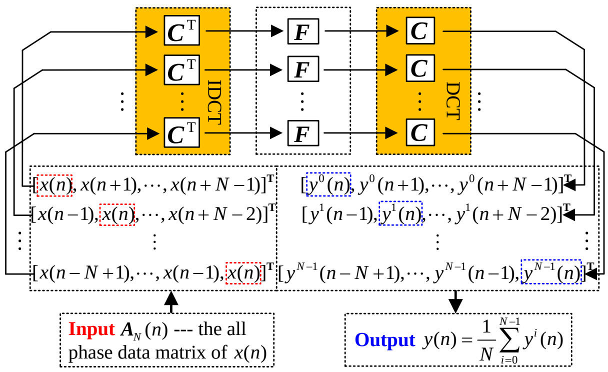


Fig. 3. Relation between input signal $x(n)$ and output signal $y(n)$ in APDF based on IDCT.

In order to avoid this tedious filtering process depicted in Fig. 2 and Fig. 3, in the following, we will introduce a very convenient and simple way to design APDF based on IDCT. Suppose $\boldsymbol{h}$ be the unit sample response of APDF based on IDCT. If we get the $\boldsymbol{h}$, the signal filtering using APDF based on IDCT can be implemented in the convolution form. Thus, how to get the $\boldsymbol{h}$ will be discussed in detail below.

We denote the $N \times N$ matrix $\boldsymbol{S}$ as the extraction operator, where $\boldsymbol{s}_i (i = 0,1,\cdots, N-1)$ is the i-th N-dimensional column vector. The i-th element in vector $\boldsymbol{s}_i$ is the value 1 and the rest elements are the value 0.

$$\boldsymbol{S} = [\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_{N-1}] \tag{4}$$

$\boldsymbol{X}_i$ $(i = 0,1,\cdots, N-1)$ is the i-th column vector of the all phase data matrix of $x(n)$. We can get a value $y^i(n)$ after $\boldsymbol{X}_i$ is filtered. That is:

$$y^i(n) = \boldsymbol{s}_i^{\mathrm{T}} \{\boldsymbol{C}[\boldsymbol{F} \cdot (\boldsymbol{C}^{\mathrm{T}} \boldsymbol{X}_i)]\} \tag{5}$$

where the mark "$\cdot$" represents dot product operation. According to Eq. (1) ~ Eq. (5), output signal can be expressed as:

$$\begin{aligned} y(n) &= \frac{1}{N}\sum_{i=0}^{N-1} y^i(n) = \frac{1}{N}\sum_{i=0}^{N-1}\{\boldsymbol{s}_i^{\mathrm{T}}\{\boldsymbol{C}[\boldsymbol{F}\cdot(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{X}_i)]\}\} \\ &= \frac{1}{N}\sum_{i=0}^{N-1}\{\sum_{k=0}^{N-1} C(i,k)\{F_N(k)[\sum_{j=0}^{N-1} C^{\mathrm{T}}(k,j)x(n-i+j)]\}\} \\ &= \frac{1}{N}\sum_{i=0}^{N-1}[\sum_{j=0}^{N-1}(\sum_{k=0}^{N-1} F_N(k)C(i,k)C^{\mathrm{T}}(k,j))]x(n-i+j) \\ &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[H(i,j)x(n-i+j)] \end{aligned} \tag{6}$$

where

$$\begin{aligned} H(i,j) &= \frac{1}{N}\sum_{m=0}^{N-1} F_N(m)C(i,m)C^{\mathrm{T}}(m,j) \\ &= \frac{1}{N}\sum_{m=0}^{N-1} F_N(m)C(i,m)C(j,m) \end{aligned} \tag{7}$$

Then Eq. (6) can be rewritten as:

$$\begin{aligned} y(n) &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[H(i,j)x(n-i+j)] \\ &= \sum_{i=\tau}^{\tau+N-1}\sum_{\tau=-(N-1)}^{N-1}[H(i,i-\tau)x(n-\tau)] \\ &= \sum_{\tau=-(N-1)}^{N-1}[\sum_{i=\tau}^{\tau+N-1} H(i,i-\tau)]x(n-\tau) \\ &= \sum_{\tau=-(N-1)}^{N-1} h(\tau)x(n-\tau) = h(n) * x(n) \end{aligned} \tag{8}$$

Eq. (8) is the convolution form of signal filtering. From Eq. (8), the unit sample response is obtained, which can be calculated by:

$$h(\tau) = \begin{cases} \sum_{i=\tau}^{N-1} H(i,i-\tau), & \tau = 0,1,\cdots,N-1, \\ \sum_{i=0}^{\tau+N-1} H(i,i-\tau), & \tau = -1,-2,\cdots,-N+1. \end{cases} \tag{9}$$

The relation between $H(i,j)$ and $h(\tau)$ is shown in Fig. 4. Due to $H(i,j) = H(j,i)$, We have $h(\tau) = h(-\tau)$, $(\tau = 0,1,\cdots,N-1)$. Therefore, APDF based on IDCT is a zero-phase FIR filter. Note that the N-order APDF based on IDCT is equivalent to a FIR digital filter with the length of $2N-1$.

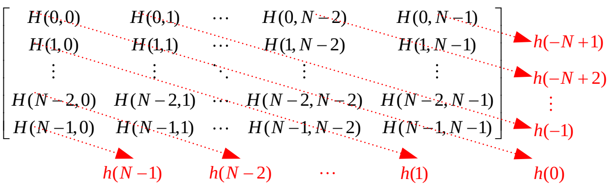


Fig. 4. Relation between $H(i,j)$ and $h(\tau)$. The value $h(\tau)$ that arrow points to is equivalent to the sum of $H(i,j)$ on the dashed line.

C. Derivation of APBT

Suppose $\boldsymbol{h}_{1/2} = [h(0), h(1), \cdots, h(N-1)]^{\mathrm{T}}$. Obviously, if we know $\boldsymbol{h}_{1/2}$, we can get $\boldsymbol{h}$ by using the symmetric property. There is an easy way to calculate the $\boldsymbol{h}_{1/2}$, which will be illustrated in the following.

We denote $\boldsymbol{f}$ as the result of DCT to sequency response vector $\boldsymbol{F}$, and that is:

$$\boldsymbol{f} = \boldsymbol{CF} = [f(0), f(1), \cdots, f(N-1)]^{\mathrm{T}} \tag{10}$$

In view of Eq. (2) which indicates the properties of DCT matrix $\boldsymbol{C}$, we have:

$$\begin{cases} f(-n) = f(n) \\ f(N+n) = -f(N-n) \end{cases}, \quad n = 0,1,\cdots,N-1 \tag{11}$$

Substituting Eq. (2) into Eq. (7), after deduction, we directly present the results as follows:

$$H(i,j) = \begin{cases} \frac{1}{\sqrt{N}} f(j), & i = 0, j \neq 0, \\ \frac{1}{\sqrt{N}} f(i), & i \neq 0, j = 0, \\ \frac{f(i+j)+f(i-j)}{2\sqrt{N}}, & i \neq 0, j \neq 0, i \neq j, \\ \frac{f(2i)+\sqrt{2}f(0)}{2\sqrt{N}}, & i \neq 0, j \neq 0, i = j. \end{cases} \tag{12}$$

Then according to Eq. (9), Eq. (11), and Eq. (12), the Eq. (9) can be rewritten as:

$$h(\tau) = \begin{cases} \frac{1}{N} f(0), & \tau = 0, \\ \frac{1}{N\sqrt{N}}(1+\frac{N-\tau-1}{\sqrt{2}}) f(\tau), & \tau = 1,2,\cdots,N-1. \end{cases} \tag{13}$$

From Eq. (13), $\boldsymbol{h}$ can be expressed in the form of matrix as:

$$\boldsymbol{h}_{1/2} = \boldsymbol{Df} = \boldsymbol{DCF} = \boldsymbol{VF} \tag{14}$$

where $\boldsymbol{V}$ is called transition matrix, which size is $N \times N$, and $\boldsymbol{V} = \boldsymbol{DC}$. **Eq. (14) shows a very effective method to design APDF based on IDCT.** In the light of Eq. (14), we can quickly obtain the unit sample response $\boldsymbol{h}$ from $\boldsymbol{F}$ defined by user.

In Eq. (14), $\boldsymbol{D}$ is a diagonal weighting matrix, and it is given by:

$$\boldsymbol{D}=\frac{1}{N\sqrt{N}}\begin{bmatrix} N & 0 & 0 & \cdots & 0 \\ 0 & 1+\frac{N-2}{\sqrt{2}} & 0 & \cdots & 0 \\ 0 & 0 & 1+\frac{N-3}{\sqrt{2}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (15)$$

On the other hand, when the unit sample response $\boldsymbol{h}$ is provided for us, $\boldsymbol{F}$ can be obtained according to Eq. (16):

$$\boldsymbol{F}=\boldsymbol{V}^{-1}\boldsymbol{h}_{1/2} \quad (16)$$

where $\boldsymbol{V}^{-1}=\boldsymbol{C}^{\mathrm{T}}\boldsymbol{D}^{-1}$. $\boldsymbol{V}^{-1}$ is the inverse matrix of $\boldsymbol{V}$.

In Eq. (14) and Eq. (16), we can see that a conversion between unit sample response $\boldsymbol{h}$ and N-dimensional filter sequency response vector $\boldsymbol{F}$ has been established by transition matrix $\boldsymbol{V}$ and inverse transition matrix $\boldsymbol{V}^{-1}$.

Because the transition matrix $\boldsymbol{V}$ connects IDCT domain and time domain, we call it **all phase biorthogonal transform** (**APBT**), and the elements in the transform matrix $\boldsymbol{V}$ are:

$$V(i,j)=\begin{cases} \frac{1}{N}, & i=0, j=0,1,\cdots,N-1, \\ \frac{N-i+\sqrt{2}-1}{N^2}\cos\frac{i(2j+1)\pi}{2N}, & i=1,2,\cdots,N-1, \; j=0,1,\cdots,N-1. \end{cases} \quad (17)$$

APBT, which is considered to be a DCT-like transform, is non-orthogonal, but it has been applied to image compression successfully instead of the conventional DCT, and better coding performance is achieved.

III. Relation between APBT-JPEG and DCT-JPEG

A. APBT-JPEG and DCT-JPEG Image Compression Schemes

The scheme of APBT-JPEG image codec is shown in Fig. 5. 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 table). Other steps of APBT-JPEG algorithm are identical to DCT-JPEG.

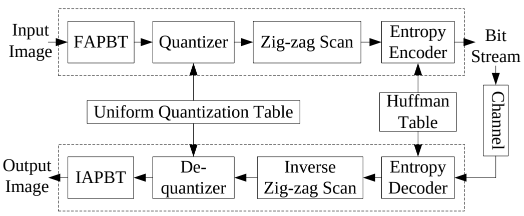


Fig. 5. The scheme of APBT-JPEG image codec. The differences between DCT-JPEG and APBT-JPEG are the transform and quantization table.

As Fig. 6 shows, we need to be clear about transform and quantization step in APBT-JPEG and DCT-JPEG before illustrating their relation.

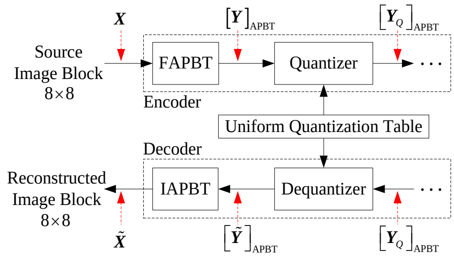


Fig. 6. Transform (inverse transform) and quantization (dequantization) in APBT-JPEG.

We denote the 2-dimensional APBT and the inverse APBT (IAPBT) as follows:

$$\begin{cases} [\boldsymbol{Y}]_{\mathrm{APBT}}=\boldsymbol{VXV}^{\mathrm{T}} \\ \tilde{\boldsymbol{X}}=(\boldsymbol{V}^{-1})[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}(\boldsymbol{V}^{-1})^{\mathrm{T}} \end{cases} \quad (18)$$

where $\boldsymbol{X}$ denotes an 8×8 image block; $[\boldsymbol{Y}]_{\mathrm{APBT}}$ denotes the corresponding block matrix in the APBT transform domain before being quantized; $\tilde{\boldsymbol{X}}$ denotes an 8×8 reconstructed image block by APBT-JPEG; $[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}$ denotes the corresponding block matrix in the APBT transform domain after being dequantized.

Different from APBT-JPEG, $\boldsymbol{C}$ is used as the transform matrix in the transform step of DCT-JPEG:

$$\begin{cases} [\boldsymbol{Y}]_{\mathrm{DCT}}=\boldsymbol{CXC}^{\mathrm{T}} \\ \tilde{\boldsymbol{X}}=\boldsymbol{C}^{\mathrm{T}}[\tilde{\boldsymbol{Y}}]_{\mathrm{DCT}}\boldsymbol{C} \end{cases} \quad (19)$$

where $[\boldsymbol{Y}]_{\mathrm{DCT}}$ denotes the corresponding block matrix in the DCT transform domain before being quantized; $[\tilde{\boldsymbol{Y}}]_{\mathrm{DCT}}$ denotes the corresponding block matrix in the DCT transform domain after being dequantized; $\tilde{\boldsymbol{X}}$ denotes an 8×8 reconstructed image block by DCT-JPEG.

Quantizer in APBT-JPEG is formulated by the following equation. Rounding is to the nearest integer, thus it will result in loss of data in quantization step:

$$[Y_Q(u,v)]_{\mathrm{APBT}}=\mathrm{round}\left(\frac{[Y(u,v)]_{\mathrm{APBT}}}{c\times Q(u,v)}\right) \quad (20)$$

where $[Y(u,v)]_{\mathrm{APBT}}$ is an element of $[\boldsymbol{Y}]_{\mathrm{APBT}}$; $Q(u,v)$, an element of quantization table $\boldsymbol{Q}$, is the corresponding quantization step size, and $[Y_Q(u,v)]_{\mathrm{APBT}}$ is the quantized transform coefficient, normalized by the quantization step size; c is a constant factor called quantization factor here. Quantization factor, which is chosen to satisfy rate and quality control criteria, is an important parameter in bit rate control here.

In the dequantizer of decoder, this normalization is removed by the following equation, which defines dequantization:

$$[\tilde{Y}(u,v)]_{\mathrm{APBT}}=[Y_Q(u,v)]_{\mathrm{APBT}}\times[c\times Q(u,v)] \quad (21)$$

where $[\tilde{Y}(u,v)]_{\mathrm{APBT}}$ is the output of dequantizer, and it is an element of dequantized transform coefficient block $[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}$.

Between the quantization step of APBT-JPEG and DCT-JPEG, the difference is only the quantization table. The quantization tables used in APBT-JPEG and DCT-JPEG are shown in Fig. 7(a) and Fig. (b), respectively.

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

(a)

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

(b)

Fig. 7. Quantization tables: (a) uniform quantization table $\boldsymbol{Q}$ used in APBT-JPEG, and (b) default quantization table suggested by JPEG used in DCT-JPEG.

B. *Relation between APBT-JPEG and DCT-JPEG*

As we mentioned above, **the relation between APBT and DCT can be summarized as:**

$$\begin{cases} \boldsymbol{V} = \boldsymbol{DC} \\ \boldsymbol{V}^{-1} = \boldsymbol{C}^{\mathrm{T}}\boldsymbol{D}^{-1} \end{cases} \tag{22}$$

where $\boldsymbol{D}^{-1}$, the inverse matrix of $\boldsymbol{D}$, is also a diagonal matrix. In accordance with Eq. (15), the diagonal matrix $\boldsymbol{D}^{-1}$ can be formulated as:

$$D^{-1}(i,i) = \begin{cases} \sqrt{N}, & i = 0, \\ \dfrac{N\sqrt{2N}}{N-i+\sqrt{2}-1}, & i = 1,2,\cdots,N-1. \end{cases} \tag{23}$$

Then, the forward and inverse transformation operations can be performed by using the APBT as:

$$\begin{aligned} [\boldsymbol{Y}]_{\mathrm{APBT}} &= \boldsymbol{VXV}^{\mathrm{T}} = (\boldsymbol{DC})\boldsymbol{X}(\boldsymbol{DC})^{\mathrm{T}} \\ &= \boldsymbol{DCXC}^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}} = \boldsymbol{D}(\boldsymbol{CXC}^{\mathrm{T}})\boldsymbol{D} \\ &= \boldsymbol{D}[\boldsymbol{Y}]_{\mathrm{DCT}}\boldsymbol{D} \end{aligned} \tag{24}$$

and

$$\begin{aligned} \tilde{\boldsymbol{X}} &= (\boldsymbol{V}^{-1})[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}(\boldsymbol{V}^{-1})^{\mathrm{T}} \\ &= (\boldsymbol{C}^{\mathrm{T}}\boldsymbol{D}^{-1})[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{D}^{-1})^{\mathrm{T}} \\ &= \boldsymbol{C}^{\mathrm{T}}\boldsymbol{D}^{-1}[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}(\boldsymbol{D}^{-1})^{\mathrm{T}}\boldsymbol{C} \\ &= \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{D}^{-1}[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}\boldsymbol{D}^{-1})\boldsymbol{C} \end{aligned} \tag{25}$$

On the basis of Eq. (19), Eq. (24), and Eq. (25), we have the following relation between the transform coefficients in APBT and DCT domain.

$$\begin{cases} [\boldsymbol{Y}]_{\mathrm{APBT}} = \boldsymbol{D}[\boldsymbol{Y}]_{\mathrm{DCT}}\boldsymbol{D}, \\ [\tilde{\boldsymbol{Y}}]_{\mathrm{DCT}} = \boldsymbol{D}^{-1}[\tilde{\boldsymbol{Y}}]_{\mathrm{APBT}}\boldsymbol{D}^{-1}. \end{cases} \tag{26}$$

To illustrate Eq. (26) in a more understandable way, we draw the picture as Fig. 8 shown. Due to $D(i,i) = 1/D^{-1}(i,i)$, the Eq. (26) can be rewritten in an intuitive way as Eq. (27) and Eq. (28).

$$[\boldsymbol{Y}]_{\mathrm{APBT}} = \begin{bmatrix} \dfrac{[Y(0,0)]_{\mathrm{DCT}}}{D^{-1}(0,0)D^{-1}(0,0)} & \cdots & \dfrac{[Y(0,N-1)]_{\mathrm{DCT}}}{D^{-1}(0,0)D^{-1}(N-1,N-1)} \\ \vdots & \ddots & \vdots \\ \dfrac{[Y(N-1,0)]_{\mathrm{DCT}}}{D^{-1}(N-1,N-1)D^{-1}(0,0)} & \cdots & \dfrac{[Y(N-1,N-1)]_{\mathrm{DCT}}}{D^{-1}(N-1,N-1)D^{-1}(N-1,N-1)} \end{bmatrix} \tag{27}$$

$$\tilde{\boldsymbol{Y}}_{\mathrm{DCT}} = \begin{bmatrix} [\tilde{Y}(0,0)]_{\mathrm{APBT}}D^{-1}(0,0)D^{-1}(0,0) & \cdots & [\tilde{Y}(0,N-1)]_{\mathrm{APBT}}D^{-1}(0,0)D^{-1}(N-1,N-1) \\ \vdots & \ddots & \vdots \\ [\tilde{Y}(N-1,0)]_{\mathrm{APBT}}D^{-1}(N-1,N-1)D^{-1}(0,0) & \cdots & [\tilde{Y}(N-1,N-1)]_{\mathrm{APBT}}D^{-1}(N-1,N-1)D^{-1}(N-1,N-1) \end{bmatrix} \tag{28}$$

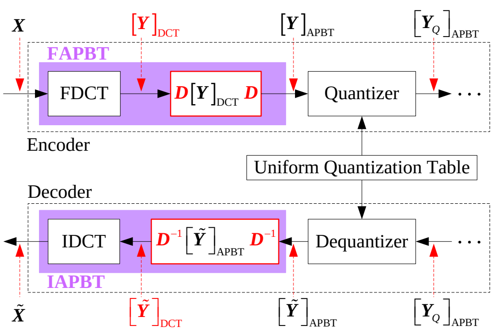


Fig. 8. The APBT-JPEG depicted in a much more essential way.

Since the quantization (dequantization) is applied on the block matrix in the transform domain, from Eq. (27) and Eq. (28), it can be seen that the factor $D^{-1}(u,u)D^{-1}(v,v)$ ($u = 0,1,\cdots N-1$; $v = 0,1,\cdots,N-1$) can be merged into the quantization table. In APBT-JPEG, with the factor $D^{-1}(u,u)D^{-1}(v,v)$ merged into the uniform quantization table, original uniform quantization table is changed into a new quantization table $\boldsymbol{Q}^{*}$ and the APBT degenerates into DCT. The idea of such a merging is also suggested in other work [16].

As a result, DCT-JPEG is generated from APBT-JPEG through the way mentioned above, which is shown in Fig. 9 where $[\boldsymbol{Y}_{Q^{*}}]_{\mathrm{DCT}}$ denotes the transform coefficient matrix quantized by quantization table $\boldsymbol{Q}^{*}$ in DCT-JPEG. In generated DCT-JPEG, the new quantization table $\boldsymbol{Q}^{*}$ is used instead of default one suggested by JPEG.

Therefore, we can conclude that the APBT-JPEG can be implemented by DCT-JPEG with a different quantization table $\boldsymbol{Q}^{*}$. The relation between APBT-JPEG and DCT-JPEG is briefly described in Fig. 10. According to Eq. (20), Eq. (21), Eq. (27), and Eq. (28), the quantization table $\boldsymbol{Q}^{*}$ can be obtained, and the elements in $\boldsymbol{Q}^{*}$ are formulated as:

$$Q^{*}(u,v)=D^{-1}(u,u)D^{-1}(v,v), \qquad (29)$$

where $u=0,1,\cdots N-1$, and $v=0,1,\cdots,N-1$.

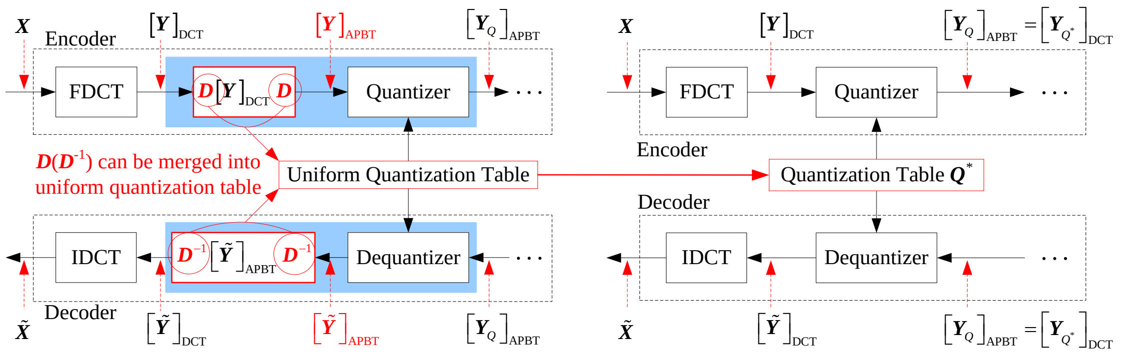


Fig. 9. The procedure to generate DCT-JPEG from APBT-JPEG.

As source image in baseline JPEG is divided into 8×8 block matrices, the variable N in Eq. (29) is equal to the value 8, and corresponding quantization table $\boldsymbol{Q}^*$ is shown in Fig. 11. In comparison to the default quantization table shown in Fig. 7(b), $\boldsymbol{Q}^*$ has a bigger quantization step size for high-frequency DCT coefficients and smaller quantization step size for low-frequency DCT coefficients. **Compared with APBT-JPEG, an identical coding performance would be achieved by using DCT-JPEG with the quantization table $\boldsymbol{Q}^*$**. Corresponding verification experiment results have been presented in Section IV.

The default quantization table is not a part of the JPEG standard. JPEG baseline coder allows users to redefine the quantization table to control the compression ratio and the quality of the reconstructed image. However, the precision of the quantization table values is specified to 8-bit, which indicates the range of quantization step size is 1~255. For this reason, $\boldsymbol{Q}^*$ cannot be used directly in actual JPEG-aware applications. In fact, this compatibility problem does not prevent us from doing corresponding experiments simulated in MATLAB software environment to verify the proposed relation between APBT-JPEG and DCT-JPEG.

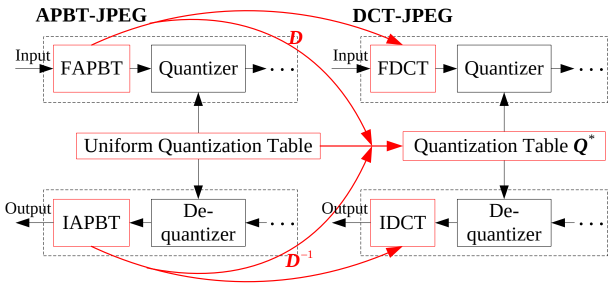


Fig. 10. The APBT-JPEG can be implemented by DCT-JPEG with quantization table $\boldsymbol{Q}^*$.

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	29.4864	36.1662	46.7591	66.1273	112.8864
16.7170	25.5094	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	55.3954	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

Fig. 11. Quantization table $\boldsymbol{Q}^*$ calculated by Eq. (29) ($N=8$), which acts as a bridge between APBT-JPEG and DCT-JPEG.

IV. Verification Experimental Results

A. PSNR Comparison of APBT-JPEG and DCT-JPEG

In order to verify the conclusion proposed in Section III, in this section, we will present simulation results obtained by applying both APBT-JPEG and DCT-JPEG algorithms to test typical images Lena, Barbara, and Zoneplate, all of size 512×512, 8bpp, monochrome images. In APBT-JPEG, $\boldsymbol{V}$ (when N=8) is used as the transform matrix. In the quantization part, the uniform quantization table (see Fig. 7(a)) is adopted, while in DCT-JPEG, $\boldsymbol{C}$ is used in transform step with the quantization tables (see Fig. 7(b) and Fig. 11) adopted in quantization procedure respectively. All these JPEG algorithms use the typical Huffman tables (see [3]) 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, which is the most widely used image quality metrics:

$$\mathrm{PSNR}=10\log_{10}\left(\frac{255^2}{MSE}\right)(\mathrm{dB}) \qquad (30)$$

where *MSE* denotes the mean squared error between the original and reconstructed images.

Table I and Table II show the experimental results with APBT-JPEG and DCT-JPEG in terms of PSNR at different bit rates, applied to images Lena and Barbara, respectively. The coding performance of APBT-JPEG exceeds DCT-JPEG with default quantization table at various bit rates. Moreover, what impresses us is the experimental data produced by DCT-JPEG using quantization table $\boldsymbol{Q}^*$. Experimental data say that the reconstructed images of APBT-JPEG and DCT-JPEG using quantization table $\boldsymbol{Q}^*$ have identical PSNR at all different bit rates. That is to say that our proposed conclusion is verified in the sense of objective quality.

TABLE I: PSNR COMPARISON OF DCT-JPEG AND APBT-JPEG APPLIED TO IMAGE LENA.

Bit rate (bpp)	PSNR (dB)		
	DCT-JPEG		**APBT-JPEG [5]**
	Default quantization table	Quantization table $\boldsymbol{Q}^*$	Uniform quantization table
0.15	26.62	**27.23**	**27.23**
0.20	29.23	**29.44**	**29.44**
0.25	30.83	**30.95**	**30.95**
0.30	32.00	**32.05**	**32.05**
0.40	33.66	**33.71**	**33.71**
0.50	34.77	**34.87**	**34.87**
0.60	35.62	**35.79**	**35.79**
0.75	36.63	**36.88**	**36.88**
1.00	37.93	**38.22**	**38.22**
1.25	39.00	**39.27**	**39.27**
1.50	39.93	**40.17**	**40.17**

TABLE II: PSNR COMPARISON OF DCT-JPEG AND APBT-JPEG APPLIED TO IMAGE BARBARA.

Bit rate (bpp)	PSNR (dB)		
	DCT-JPEG		**APBT-JPEG [5]**
	Default quantization table	Quantization table $\boldsymbol{Q}^*$	Uniform quantization table
0.15	22.41	**22.80**	**22.80**
0.20	23.59	**23.86**	**23.86**
0.25	24.44	**24.69**	**24.69**
0.30	25.19	**25.51**	**25.51**
0.40	26.61	**27.07**	**27.07**
0.50	27.95	**28.42**	**28.42**
0.60	29.23	**29.68**	**29.68**
0.75	30.90	**31.39**	**31.39**
1.00	33.23	**33.59**	**33.59**
1.25	35.10	**35.38**	**35.38**
1.50	36.71	**36.83**	**36.83**

B. Visual Quality Comparison of APBT-JPEG and DCT-JPEG

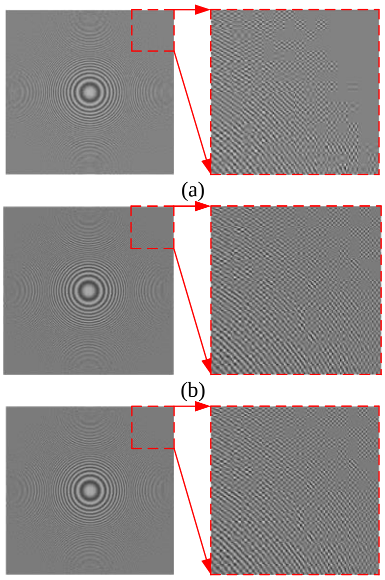
In order to compare the compression performance subjectively, Fig. 12-Fig. 14 show the reconstructed image Lena, Barbara, and Zoneplate, all of size 512×512 obtained by using APBT-JPEG and DCT-JPEG with two different quantization tables at some representative bit rates. From Figs. 12~13, it is clear that, as compared with the DCT-JPEG version with default quantization table, better visual quality of APBT-JPEG version is achieved and the blocking artifacts have been reduced significantly. No obvious blocking artifacts can be perceived. Then, as compared with APBT-JPEG, a really very approximate visual quality of reconstructed image is achieved by using DCT-JPEG with quantization table $\boldsymbol{Q}^*$, and the difference between them cannot be perceived by eyes at all. Besides, in order to verify our proposed conclusion, the difference of visual quality between the APBT-JPEG and DCT-JPEG version has shown in Fig. 14 in detail.

(a) (b) (c)

Fig. 12. Barbara obtained at 0.20bpp: (a) DCT-JPEG version with default quantization table, PSNR=23.59dB, (b) APBT-JPEG version with uniform quantization table, PSNR=23.86dB, (c) DCT-JPEG version with quantization table $\boldsymbol{Q}^*$, PSNR=23.86dB.

(a) (b) (c)

Fig. 13. Lena obtained at 0.20bpp: (a) DCT-JPEG version with default quantization table, PSNR=29.23dB, (b) APBT-JPEG version with uniform quantization table, PSNR=29.44dB, (c) DCT-JPEG version with quantization table $\boldsymbol{Q}^*$, PSNR=29.44dB.



(a) (b) (c)

Fig. 14. Zoneplate obtained at 0.60bpp: (a) DCT-JPEG version with default quantization table, PSNR=13.73dB, (b) APBT-JPEG version, with PSNR=14.14dB, (c) DCT-JPEG version with $\boldsymbol{Q}^*$, PSNR=14.14dB.

To keep the paper reasonably concise, finally, we would like to point out that the identical experiment simulation is also applied to other test images, and similar results can be obtained.

Throughout this paper, all experiments are conducted with MATLAB 7.14 on the desktop computer (3.10GHz Intel Core i3-2100 CPU, 2GB RAM, 5400-7200 rpm HD) installed with 64-bit Windows 7 operating system.

Now, experimental results both in terms of the objective quality and subjective effect have indicated that the APBT-JPEG can be implemented by DCT-JPEG with quantization table $\boldsymbol{Q}^*$. Although $\boldsymbol{Q}^*$ cannot be used directly in the actual applications of baseline JPEG. However, our purpose is not to propose a novel quantization table. With the help of $\boldsymbol{Q}^*$, we can see the relation between APBT-JPEG and DCT-JPEG more clearly. What's more, this relation between two kinds of JPEG algorithms is also a good way to answer why APBT-JPEG yields better coding performance. From another point of view, with quantization table $\boldsymbol{Q}^*$ instead of default one, DCT-JPEG produces better results. We can see that there still exists room to further optimize the quantization table in DCT-JPEG.

V. Conclusions

With in-depth study on APBT-JPEG and DCT-JPEG, two questions are answered in detail in this paper. They are: 1) where does the APBT come from; 2) what is the relation between APBT-JPEG and DCT-JPEG. Both theoretical analysis and experimental results have verified our proposed relation between APBT-JPEG and DCT-JPEG. It is also a good way to explain why APBT-JPEG yields better coding performance.

In addition, we believe that the work done in this paper offers a good inspiration for future work of APBT-JPEG and DCT-JPEG. In DCT-JPEG, it is likely that future research will yield better quantization tables that provide more compression for the same perceived image quality. In APBT-JPEG, with relation between APBT and DCT, referring to the well-developed DCT fast algorithm, some contributions can be made to the fast computation of APBT in the further research work.

Acknowledgment

This work was supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2015PF004), the National Natural Science Foundation of China (Grant No. 61201371), and the promotive research fund for excellent young and middle-aged scientists of Shandong Province, China (Grant No. BS2013DX022). The authors would like to thank Fanfan Yang, Liping Wang, Heng Zhang, and Yunpeng Zhang for their kind help and valuable suggestions. The authors would like to thank the anonymous reviewers and the editor 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, Sep. 2001.

[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, 204, Nov. 2009.

[3] 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 | ITU-T Rec. T. 81, 1st ed., Feb. 17, 1994.

[4] A. Beghdadi, M. C. Larabi, A. Bouzerdoum, and K. M. Iftekharuddin, "A survey of perceptual image processing methods," *Signal Processing: Image Communication*, vol. 28, no. 8, pp. 811-831, Sep. 2013.

[5] 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.

[6] 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, Sep. 2013.

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

[8] 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.

[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] C. Y. Wang, B. C. Jiang, and H. Yuan, "Comparison of interpolation methods in Bayer CFA image compression based on structure separation and APBT-JPEG," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 1, pp. 87-98, Feb. 2014.

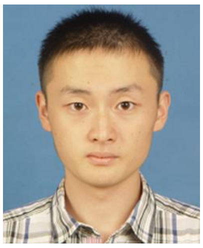
[11] C. Y. Wang, R. Y. Shan, and X. Zhou, "APBT-JPEG image coding based on GPU," *KSII Trans. on Internet and Information Systems*, vol. 9, no. 4, pp. 1457-1470, Apr. 2015.

[12] Z. X. Hou, Z. H. Wang, and X. Yang, "Design and implementation of all phase DFT digital filter," *Acta Electronica Sinica*, vol. 31, no. 4, pp. 539-543, Apr. 2003.

[13] X. Zhou, X. Y. Wang, B. C. Jiang, and C. Y. Wang, "Adaptive threshold edge detection algorithm based on APDCSF and anti-symmetrical biorthogonal wavelet transform," *Journal of Communications*, vol. 9, no. 6, pp. 515-520, Jun. 2014.

[14] X. Zhou, Q. M. Fu, F. F. Yang, and C. Y. Wang, "Implementation of biorthogonal wavelet transform using

windowed APDF based on DCT," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 6, pp. 1-16, Dec. 2014.
[15] L. L. Zhao, "All phase digital filtering and image interpolation based on IDCT/DCT," Ph.D. thesis, Tianjin University, China, Jun. 2006.
[16] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Low-complexity 8×8 transform for image compression," *Electronics Letters*, vol. 44, no. 21, pp. 1249-1250, Oct. 2008.



Qiming Fu was born in Jiangxi province, China in 1992. He received his B.S. degree in electronic information science and technology from Shandong University, Weihai, China in 2013. He is currently pursuing his M.E. degree in electronics and communication engineering at Shandong University, China. His current research interests include digital image processing and analysis techniques.

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.

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, pattern recognition and machine learning.

Baochen Jiang was born in Shandong province, China in 1962. He received his B.S. degree in radio electronics from Shandong University, China in 1983 and his M.E. degree in communication and electronic systems from Tsinghua University, China in 1990. Now he is a professor in the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China. His current research interests include signal and information processing, image/video processing and analysis, and smart grid technology.