IEIE Transactions on Smart Processing and Computing

Inter-Conversion Matrix for Transcoding Block DCT and DWT-Based Compressed Images

Donggyun Kim, Sanghee Lim, and Joonki Paik

Image Processing and Intelligent Systems Laboratory, Graduate School of Advanced Image Science, Multimedia, and Film, Chung-Ang University / Seoul, Korea paikj@cau.ac.kr

***** Corresponding Author: Joonki Paik

Received November 22, 2013; Revised January 5, 2014, Accepted March 20, 2014; Published June 30, 2014

** Regular Paper*

***Abstract*:** This study derived the inter-conversion matrices, which can be used in heterogeneous image transcoding between the compressed images using different transforms, such as the 8×8 block discrete cosine transform (BDCT) and the one-level discrete wavelet transform (DWT). Basically, to obtain the one-level DWT coefficients from 8×8 BDCT, inverse BDCT should be performed followed by forward DWT, and vice versa. On the other hand, if the proposed inter-conversion approach is used, only one inter-conversion matrix multiplication makes the corresponding transcoding possible. Both theoretical and experimental analyses showed that the amount of computation of the proposed approach decreases over 20% when the inter-conversion matrices are used under specific conditions.

***Keywords*:** Transcoding, Inter-conversion matrix, DCT, DWT

1. Introduction

Various video/image compressions techniques, such as H.26x, MPEG, Motion JPEG, Motion JPEG-2000, JPEG, and JPEG-2000, have been proposed and standardized. Each compression standard has been designed for a specific application. Recently, transcoding has attracted increasing attention for the efficient inter-conversion of differently compressed and coded image data [1-7].

The major objective of transcoding is to minimize the computation for differently coded image data by maximally utilizing the compressed data, such as the coding parameters and statistical information [1]. Although transcoding with the same video/image transformation requires a relatively small amount of computation, the amount of computation is very difficult to reduce by transcoding the compression using different transformations

Block discrete cosine transformation (BDCT) and discrete wavelet transformation (DWT) are used widely for most video/image compression standards. The transcoding between JPEG and JPEG-2000, or between H.26x and Motion JPEG-2000 is the case using different transformations. In this case, the efficient inter-conversion of transform coefficients should be considered. Lan proposed fast BDCT followed by fast DWT for the inter-conversion of the transform coefficients from BDCT to DWT [3]. On the other hand, the fast algorithms cannot be used for BDCT and DWT for hardware realization of the transcoder because of circuit complexity. In this case, it is efficient that perform the inter-conversion of the transform coefficients from one transform instead of the two transforms, which consists of one inverse transform followed by a forward transform. This paper presents a direct approach to the inter-conversion of BDCT and one-level DWT coefficients. Without a loss of generality, an 8×8 BDCT and Daubechies D4 filter were used for DWT. Section 2 presents the matrix representations of 8×8 BDCT and one-level DWT. The mathematical justification for considering only one-level DWT is also shown in this section. Section 3 derives the conversion matrices for both BDCT-to-DWT and DWT-to-BDCT. The amount of computation for a straightforward approach and the proposed approach was compared in section 4, and section 5 concludes the paper with future research issues.

2. Matrix Representation of BDCT and One-Level DWT

This section briefly summarizes the matrix representation of 8×8 BDCT and one-level DWT, and proves that two or higher-level DWT cannot be expressed by a matrix representation, which justifies that this work is focused on the inter-conversion between BDCT and one-level DWT.

2.1 Matrix Representation of 8x8 BDCT

Let $f_{8\times8}$ and $F_{8\times8}$ be an 8×8 array of the image and the corresponding DCT coefficients, respectively. If $C_{8\times8}$ represents the eight-point, one-dimensional (1D) DCT matrix, the forward and inverse DCTs can respectively be expressed in matrix-vector multiplication form as

$$F_{8\times8}=C_{8\times8}f_{8\times8}C_{8\times8}{}^{T} \text{ and } f_{8\times8}=C_{8\times8}{}^{T}F_{8\times8}C_{8\times8} \quad (1)$$

where $C_{8\times8}$ is defined as

$$C_{8\times8} = \begin{array}{cc} & \text{column} \quad \rightarrow n \\ \text{row} & 0 \quad 1 \quad 2 \quad 3 \quad \cdots \quad 7 \\ \begin{array}{c} \\ \downarrow k \end{array} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ 7 \end{array} & \begin{bmatrix} \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{8}} & \cdots & \frac{1}{\sqrt{8}} \\ & \sqrt{\frac{2}{8}} \begin{bmatrix} \cos\left[\frac{(2n+1)k\pi}{16}\right] \\ 0\le n\le 7 \\ 1\le k\le 7 \end{bmatrix} & & & & \end{bmatrix} \end{array} \quad (2)$$

For simplicity, if an input image of size 16×24 is represented as

$$f_{16\times24}=\begin{bmatrix} f_{8\times8}(1) & f_{8\times8}(2) & f_{8\times8}(3) \\ f_{8\times8}(4) & f_{8\times8}(5) & f_{8\times8}(6) \end{bmatrix}, \quad (3)$$

its DCT can be expressed as

$$\begin{aligned} F_{16\times24} &= \begin{bmatrix} F_{8\times8}(1) & F_{8\times8}(2) & F_{8\times8}(3) \\ F_{8\times8}(4) & F_{8\times8}(5) & F_{8\times8}(6) \end{bmatrix} \\ &= \begin{bmatrix} C_{8\times8}f_{8\times8}(1)C_{8\times8}^T & C_{8\times8}f_{8\times8}(2)C_{8\times8}^T & C_{8\times8}f_{8\times8}(3)C_{8\times8}^T \\ C_{8\times8}f_{8\times8}(4)C_{8\times8}^T & C_{8\times8}f_{8\times8}(5)C_{8\times8}^T & C_{8\times8}f_{8\times8}(6)C_{8\times8}^T \end{bmatrix}. \end{aligned} \quad (4)$$

By defining the 16×16 DCT generation matrices as

$$D_{16\times16}=\begin{bmatrix} C_{8\times8} & O_{8\times8} \\ O_{8\times8} & C_{8\times8} \end{bmatrix}, \quad (5)$$

where $O_{8\times8}$ represents the 8×8 zero matrix, 16×24 DCT coefficients in (4) can be further simplified as

$$\begin{aligned} F_{16\times24} &= \begin{bmatrix} C_{8\times8} & O_{8\times8} \\ O_{8\times8} & C_{8\times8} \end{bmatrix} \begin{bmatrix} f_{8\times8}(1) & f_{8\times8}(2) & f_{8\times8}(3) \\ f_{8\times8}(4) & f_{8\times8}(5) & f_{8\times8}(6) \end{bmatrix} \begin{bmatrix} C_{8\times8}^T & O_{8\times8} & O_{8\times8} \\ O_{8\times8} & C_{8\times8}^T & O_{8\times8} \\ O_{8\times8} & O_{8\times8} & C_{8\times8}^T \end{bmatrix}, \\ &= D_{16\times16}f_{16\times24}D_{24\times24}^T \end{aligned} \quad (6)$$

By generalizing the derivation for (6), the 8×8 BDCT of an $N\times M$ image, for both N and M multiples of eight, can be expressed as

$$F_{N\times M}=D_{N\times N}f_{N\times M}D_{M\times M}^T. \quad (7)$$

In a similar manner, the IDCT can be derived and expressed as

$$f_{N\times M}=D_{N\times N}^T F_{N\times M}D_{M\times M}. \quad (8)$$

2.2 Matrix Representation of One-Level DWT

Given a specific filter and a boundary processing method, both the forward and inverse DWTs can be expressed in a matrix-vector multiplication form. Without a loss of generality, by adopting Daubechies DB-4 filter and assuming periodic boundary, the corresponding 6×8 forward and inverse DWTs are derived below.

The DB-4 lowpass filter for analysis and synthesis is defined as

$$h[0]=\frac{1+\sqrt{3}}{4\sqrt{2}},\ h[1]=\frac{3+\sqrt{3}}{4\sqrt{2}},\ h[2]=\frac{3-\sqrt{3}}{4\sqrt{2}},\ \text{and } h[3]=\frac{1-\sqrt{3}}{4\sqrt{2}}, \quad (9)$$

and its highpass version can be expressed as

$$g[0]=h[3],\ g[1]=-h[2],\ g[2]=h[1],\ \text{and } g[3]=-h[0]. \quad (10)$$

Let $U_{6\times6}$ and $U_{8\times8}$ be the 1D forward DWT matrices of the size six and eight, respectively, as

$$\mathbf{U}_{6\times6}=\begin{bmatrix} h[0] & h[1] & h[2] & h[3] & 0 & 0 \\ 0 & 0 & h[0] & h[1] & h[2] & h[3] \\ h[2] & h[3] & 0 & 0 & h[0] & h[1] \\ g[0] & g[1] & g[2] & g[3] & 0 & 0 \\ 0 & 0 & g[0] & g[1] & g[2] & g[3] \\ g[2] & g[3] & 0 & 0 & g[0] & g[1] \end{bmatrix},$$

$$\mathbf{U}_{8\times8} = \begin{bmatrix} h[0] & h[1] & h[2] & h[3] & 0 & 0 & 0 & 0 \\ 0 & 0 & h[0] & h[1] & h[2] & h[3] & 0 & 0 \\ 0 & 0 & 0 & 0 & h[0] & h[1] & h[2] & h[3] \\ h[2] & h[3] & 0 & 0 & 0 & 0 & h[0] & h[1] \\ g[0] & g[1] & g[2] & g[3] & 0 & 0 & 0 & 0 \\ 0 & 0 & g[0] & g[1] & g[2] & g[3] & 0 & 0 \\ 0 & 0 & 0 & 0 & g[0] & g[1] & g[2] & g[3] \\ g[2] & g[3] & 0 & 0 & 0 & 0 & g[0] & g[1] \end{bmatrix}. \quad (11)$$

Using (11), the 6×8 forward DWT, which is denoted as $W_{6\times8}$, is expressed as

$$W_{6\times8} = U_{6\times6}\, f_{6\times8} U_{8\times8}{}^{T}. \quad (12)$$

In a similar manner of the IDCT given in (8), the 6×8 inverse DWT can be given as

$$f_{6\times8} = V_{6\times6} W_{6\times8} V_{8\times8}{}^{T}, \quad (13)$$

where the 1D inverse DWT matrices of size six and eight are defined as

$$\mathbf{V}_{6\times6} = \begin{bmatrix} h[0] & 0 & h[2] & g[0] & 0 & g[2] \\ h[1] & 0 & h[3] & g[1] & 0 & g[3] \\ h[2] & h[0] & 0 & g[2] & g[0] & 0 \\ h[3] & h[1] & 0 & g[3] & g[1] & 0 \\ 0 & h[2] & h[0] & 0 & g[2] & g[0] \\ 0 & h[3] & h[1] & 0 & g[3] & g[1] \end{bmatrix},$$

$$\mathbf{V}_{8\times8} = \begin{bmatrix} h[0] & 0 & 0 & h[2] & g[0] & 0 & 0 & g[2] \\ h[1] & 0 & 0 & h[3] & g[1] & 0 & 0 & g[3] \\ h[2] & h[0] & 0 & 0 & g[2] & g[0] & 0 & 0 \\ h[3] & h[1] & 0 & 0 & g[3] & g[1] & 0 & 0 \\ 0 & h[2] & h[0] & 0 & 0 & g[2] & g[0] & 0 \\ 0 & h[3] & h[1] & 0 & 0 & g[3] & g[1] & 0 \\ 0 & 0 & h[2] & h[0] & 0 & 0 & g[2] & g[0] \\ 0 & 0 & h[3] & h[1] & 0 & 0 & g[3] & g[1] \end{bmatrix}. \quad (14)$$

For a general $N\times M$ image, the forward and inverse DWT can be expressed as

$$W_{N\times M} = U_{N\times N}\, f_{N\times M} U_{M\times M}^{T}, \quad (15)$$

and

$$f_{N\times M} = V_{N\times N} W_{N\times M} V_{M\times M}^{T}, \quad (16)$$

respectively.

2.3 Consideration of the Higher-Level DWT

Let the one-level DWT of a 16×16 image be expressed as

$$W_{16\times16}(1) = U_{16\times16}(1)\, f_{16\times16} U_{16\times16}^{T}(1) = \begin{bmatrix} a_{8\times8} & b_{8\times8} \\ c_{8\times8} & d_{8\times8} \end{bmatrix}, \quad (17)$$

where $U_{16\times16}(1)$ and $U_{16\times16}(1)^T$ represent one-level DWT matrices in the column and row directions, respectively, and $a_{8\times8}$, $b_{8\times8}$, $c_{8\times8}$, and $d_{8\times8}$ represent the LL, HL, LH, and HH subbands, respectively.

To convert the two-level DWT directly into the DCT and vice versa, $W_{16\times16}(2)$ should be expressed as

$$\begin{aligned} W_{16\times16}(2) &= U_{16\times16}(2) W_{16\times16}(1) U_{16\times16}^{T}(2) \\ &= U_{16\times16}(2) \begin{bmatrix} a_{8\times8} & b_{8\times8} \\ c_{8\times8} & d_{8\times8} \end{bmatrix} U_{16\times16}^{T}(2). \end{aligned} \quad (18)$$

Because the second-level DWT is performed only for the LL subband, (18) can be rewritten as

$$W_{16\times16}(2) = \begin{bmatrix} U_{8\times8}(1) a_{8\times8} U_{8\times8}{}^{T}(1) & b_{8\times8} \\ c_{8\times8} & d_{8\times8} \end{bmatrix}. \quad (19)$$

If it is assumed that there are $U_{16}(2)$ and $U_{16}(2)^T$ that satisfy (18) and (19), they can be derived as

$$U_{16\times16}(2) \begin{bmatrix} a_{8\times8} & b_{8\times8} \\ c_{8\times8} & d_{8\times8} \end{bmatrix} U_{16\times16}^{T}(2) = \begin{bmatrix} U_{8\times8}(1) a_{8\times8} U_{8\times8}^{T}(1) & b_{8\times8} \\ c_{8\times8} & d_{8\times8} \end{bmatrix}. \quad (20)$$

Let $U_{16\times16}(2)$ be divided into four sub-matrices as

$$U_{16\times16}(2) = \begin{bmatrix} x_{8\times8} & y_{8\times8} \\ z_{8\times8} & w_{8\times8} \end{bmatrix}. \quad (21)$$

Hence, Eq. (20)(20) can then be rewritten as

$$\begin{bmatrix} x_{8\times8} & y_{8\times8} \\ z_{8\times8} & w_{8\times8} \end{bmatrix} \begin{bmatrix} a_{8\times8} & b_{8\times8} \\ c_{8\times8} & d_{8\times8} \end{bmatrix} \begin{bmatrix} x_{8\times8}^{T} & z_{8\times8}^{T} \\ y_{8\times8}^{T} & w_{8\times8}^{T} \end{bmatrix} = \begin{bmatrix} U_{8\times8}(1) a_{8\times8} U_{8\times8}^{T}(1) & b_{8\times8} \\ c_{8\times8} & d_{8\times8} \end{bmatrix}. \quad (22)$$

By comparing the four submatrices in both sides of (22), the following four relationships can be obtained:

$$\begin{aligned} & x_{8\times8} a_{8\times8} x_{8\times8}^{T} + x_{8\times8} b_{8\times8} y_{8\times8}^{T} + y_{8\times8} c_{8\times8} x_{8\times8}^{T} + y_{8\times8} d_{8\times8} y_{8\times8}^{T} = U_{8\times8}(1) a_{8\times8} U_{8\times8}^{T}(1) \\ & z_{8\times8} a_{8\times8} x_{8\times8}^{T} + z_{8\times8} b_{8\times8} y_{8\times8}^{T} + w_{8\times8} c_{8\times8} x_{8\times8}^{T} + w_{8\times8} d_{8\times8} y_{8\times8}^{T} = c_{8\times8} \\ & x_{8\times8} a_{8\times8} z_{8\times8}^{T} + x_{8\times8} b_{8\times8} w_{8\times8}^{T} + y_{8\times8} c_{8\times8} z_{8\times8}^{T} + y_{8\times8} d_{8\times8} w_{8\times8}^{T} = b_{8\times8} \\ & z_{8\times8} a_{8\times8} z_{8\times8}^{T} + z_{8\times8} b_{8\times8} w_{8\times8}^{T} + w_{8\times8} c_{8\times8} z_{8\times8}^{T} + w_{8\times8} d_{8\times8} w_{8\times8}^{T} = d_{8\times8} \end{aligned} \quad (23)$$

The first equation in (23) results in

$$x_{8\times8} = U_{8\times8}(1), \text{ and } x_{8\times8} b_{8\times8} y_{8\times8}^T + y_{8\times8} c_{8\times8} x_{8\times8}^T + y_{8\times8} d_{8\times8} y_{8\times8}^T = o_{8\times8}, \tag{24}$$

where $o_{8\times8}$ represents the 8×8 zero matrix. The second equation in (23) results in

$$w_{8\times8} = c_{8\times8} U_{8\times8}^{-T}(1) c_{8\times8}^{-1}, \tag{25}$$

and

$$z_{8\times8} a_{8\times8} x_{8\times8}^T + z_{8\times8} b_{8\times8} y_{8\times8}^T + w_{8\times8} d_{8\times8} y_{8\times8}^T = o_{8\times8}. \tag{26}$$

From (26), $w_{8\times8}$, which is a submatrix of $U_{16\times16}(2)$, depends on $c_{8\times8}$, which is the LH subband of the one-level DWT. This means a two-level DWT matrix changes according to the input image. As a result, it is impossible to make a fixed DWT matrix for two or higher levels.

3. Inter-Conversion Matrices

3.1 Matrix Representation of 8x8 BDCT

The straightforward approach to BDCT-to-DWT inter-conversion can be expressed as

$$W_{N\times M} = U_N D_{N\times N}^T F_{N\times M} D_{M\times M} U_M^T. \tag{27}$$

Let $A_{N\times N} = U_N D_{N\times N}^T$, and $A_{M\times M}^T = D_{M\times M} U_M^T$ (27) can then be simplified as

$$W_{N\times M} = A_{N\times N} F_{N\times M} A_{M\times M}^T, \tag{28}$$

where $A_{N\times N}$ and $A_{M\times M}^T$ are computed by the column-wise DWT of $D_{N\times N}^T$ and the row-wise DWT of $D_{M\times M}$, respectively. Given the two matrices, the $A_{N\times N}$ and $A_{M\times M}^T$, 8×8 BDCT coefficients can be converted directly to one-level DWT coefficients. As a result, we can consider these two matrices can be considered to be the BDCT-to-DWT inter-conversion matrices.

3.2 Matrix Representation of 8x8 BDCT

The process of DWT-to-BDCT inter-conversion is similar to BDCT-to-DWT inter-conversion. Using (7) and (16), the straightforward approach to DWT-to-BDCT inter-conversion is expressed as

$$F_{N\times M} = D_{N\times N} V_N W_{N\times M} V_M^T D_{M\times M}^T. \tag{29}$$

Let $B_{N\times N} = D_{N\times N} V_N$, and $B_{M\times M}^T = V_M^T D_{M\times M}^T$, then (29) can be expressed as

$$F_{N\times M} = B_{N\times N} W_{N\times M} B_{M\times M}^T, \tag{30}$$

where these two matrices play the role of DWT-to-BDCT inter-conversion.

4. Comparative analysis of computational Amount

This section compares the amount of computation between the straightforward approach and the proposed approach, which uses a one-step matrix multiplication of the BDCT-to-DWT and DWT-to-BDCT inter-conversion. To count the number of operations in the straightforward approach, the fast algorithms for DWT and DCT were not considered; only the multiplications with nonzero coefficients were considered. On the other hand, in the proposed approach, the size of the 8×8 BDCT matrix and the one-level DWT matrix were fixed to $M\times M$, where M is a multiple of 8 and is greater than or equal to 16. In particular, for the DWT Daubechies DB-4 filter and assume periodic boundary were used.

4.1 Analysis of the Straightforward Approach

To estimate the amount of computation of the straightforward approach, it is important to simply add the amount of computation of 8×8 BDCT and one-level DWT.

The 1D DCT length 8 requires 64 (= 8×8) multiplications and 56 ($=7\times8$) additions. The 2D DCT of size 8×8 requires 1024 (= 64×16) multiplications and 896 ($=56\times16$) additions, because an 8×8 DCT requires 16 times more operations than a 1D DCT of length 8. The 8×8 BDCT of an $N\times M$ image (N and M are multiples of 8) requires $(N/8)\times(M/8)$ times more operations than a 2D 8×8 DCT. Therefore, the number of multiplications needed for 8×8 BDCT of an $N\times M$ image is given as

$$(N/8)\times(M/8)\times1024, \tag{31}$$

and the corresponding number of additions is given as

$$(N/8)\times(M/8)\times896. \tag{32}$$

The amount of computation for an 8×8 inverse BDCT is equal to that of the forward version.

Without including multiplications with zero coefficients, the amount of computation in a matrix-vector multiplication form in (7) and (8) is equal to (31) and (32), respectively.

To examine the amount of computation for a one-level DWT, let the number of analysis lowpass and highpass filter coefficients be *al* and *ah*, respectively, and the numbers of synthesis lowpass and highpass filter coefficients be *sl* and *sh*, respectively. The amount of computations of 1D, one-level DWT of length N is given as

$$
\begin{aligned}
&< \text{Forward 1D 1-level DWT} > \\
&MUL:(al+ah)N/2, \qquad ADD:(al+ah-2)N/2 \\
&< \text{Inverse 1D 1-level DWT} > \\
&MUL:(sl+sh)N/2, \qquad ADD:(sl+sh-4)N/2+N
\end{aligned}
\quad . \tag{33}
$$

(33) is satisfied by not only the Daubechies DB-4 filter but also an arbitrary filter. On the other hand, (33) has a restriction that the volume of data that is transformed must be even. The 2D, one-level DWT of an $N \times M$ image (N and M are multiples of 2) is equal to M times the column-wise 1D DWT of length N followed by N times 1D, and DWT of length M. Therefore, the amount of computation for 2D, one-level DWT of an $N \times M$ image is given as

$$
\begin{aligned}
&< \text{Forward 2D 1-level DWT} > \\
&MUL:(al+ah)MN, \qquad ADD:(al+ah-2)MN \\
&< \text{Inverse 2D 1-level DWT} > \\
&MUL:(sl+sh)MN, \qquad ADD:(sl+sh-4)MN+2MN
\end{aligned}
\quad . \tag{34}
$$

where the amount of computation is equal to the amount of computation for (15) and (16) without including multiplications with zero coefficients.

The straightforward approach for the BDCT-to-DWT inter-conversion performs a forward one-level DWT after an 8×8 inverse BDCT of the given 8×8 BDCT coefficients. Using (31), (32) and (34), the number of computations needed for BDCT-to-DWT of a $N \times M$ data, is given as

$$
\begin{aligned}
&MUL:(N/8)\times(M/8)\times 1024+(al+ah)MN, \\
&ADD:(N/8)\times(M/8)\times 896+(al+ah-2)MN
\end{aligned}
\quad . \tag{35}
$$

In a similar manner, the number of computations for DWT-to-BDCT of a $N \times M$ data using straightforward approach can be derived as

$$
\begin{aligned}
&MUL:(sl+sh)MN+(N/8)\times(M/8)\times 1024, \\
&ADD:(sl+sh-4)MN+2MN+(N/8)\times(M/8)\times 896
\end{aligned}
\quad . \tag{36}
$$

4.2 Number of Computations of the Proposed Approach

The number of computations of straightforward approach in the above subsection is equal to the number of computations needed for matrix-vector multiplication in (7), (8), (15), and (16) without considering the multiplications with zero coefficients. In the other hand, it is reasonable to consider and exclude multiplications with zero coefficients, which does not require any real computations.

To calculate the inter-conversion matrices, A and B, for a special case, it was assumed that the image is square and its height is greater than or equal to 16 and is a multiple of 8 ($M \times M$, $M = 16, 24, 32, \cdots$). In addition, the Daubechies DB-4 filter with periodic boundary was used for convolution-based DWT. In this case, the inter-conversion matrices, A and B, are given as

$$
A_{16\times16} = B^T_{16\times16} = \begin{bmatrix}
a_{4\times8} & O_{3\times8} \\
O_{3\times8} & a_{1\times8} \\
a_{1\times8} & a_{4\times8} \\
b_{4\times8} & O_{3\times8} \\
O_{3\times8} & b_{1\times8} \\
b_{1\times8} & b_{4\times8}
\end{bmatrix}, \tag{37}
$$

$$
A_{24\times24} = B^T_{24\times24} = \begin{bmatrix}
a_{4\times8} & O_{3\times8} & O_{4\times8} \\
O_{4\times8} & a_{1\times8} & O_{3\times8} \\
O_{3\times8} & a_{4\times8} & a_{1\times8} \\
a_{1\times8} & O_{4\times8} & a_{4\times8} \\
b_{4\times8} & O_{3\times8} & O_{4\times8} \\
O_{4\times8} & b_{1\times8} & O_{3\times8} \\
O_{3\times8} & b_{4\times8} & b_{1\times8} \\
b_{1\times8} & O_{4\times8} & b_{4\times8}
\end{bmatrix}, \text{ and} \tag{38}
$$

$$
A_{32\times32} = B^T_{32\times32} = \begin{bmatrix}
a_{4\times8} & O_{3\times8} & O_{4\times8} & O_{4\times8} \\
O_{4\times8} & a_{1\times8} & O_{3\times8} & O_{4\times8} \\
O_{4\times8} & a_{4\times8} & a_{1\times8} & O_{3\times8} \\
O_{3\times8} & O_{4\times8} & a_{4\times8} & a_{1\times8} \\
a_{1\times8} & O_{4\times8} & O_{4\times8} & a_{4\times8} \\
b_{4\times8} & O_{3\times8} & O_{4\times8} & O_{4\times8} \\
O_{4\times8} & b_{4\times8} & O_{3\times8} & O_{4\times8} \\
O_{4\times8} & b_{4\times8} & b_{4\times8} & O_{3\times8} \\
O_{3\times8} & O_{4\times8} & b_{4\times8} & b_{4\times8} \\
b_{4\times8} & O_{4\times8} & O_{4\times8} & b_{4\times8}
\end{bmatrix}. \tag{39}
$$

In (37), (38) and (39), $O_{4\times8}$ and $O_{3\times8}$ represent zero matrices, and $a_{4\times8}$, $a_{1\times8}$, $b_{4\times8}$ and $b_{1\times8}$ are as follows. (Here, all the elements are calculated down to five decimal places and rounded.)

$$
a_{4\times8} = \begin{bmatrix}
0.5 & 0.6343 & 0.4001 & 0.0452 & -0.25 & -0.308 & -0.1657 & -0.0286 \\
0.5 & 0.2298 & -0.5576 & -0.4704 & 0.25 & 0.3143 & -0.0396 & -0.0457 \\
0.5 & -0.3092 & -0.4001 & 0.6200 & -0.25 & 0.3143 & 0.1657 & -0.036 \\
0.4665 & -0.6110 & 0.4788 & -0.3007 & 0.125 & 0.0045 & -0.063 & 0.0526
\end{bmatrix}, \tag{40}
$$

$$
a_{1\times8} = \begin{bmatrix} 0.0335 & 0.0561 & 0.0788 & 0.1058 & 0.125 & 0.1257 & 0.1027 & 0.0578 \end{bmatrix}, \tag{41}
$$

$$
b_{4\times8} = \begin{bmatrix}
0 & 0.0286 & -0.0396 & -0.308 & -0.433 & -0.0452 & 0.5576 & 0.6343 \\
0 & -0.0053 & -0.1657 & 0.1213 & 0.433 & -0.4065 & -0.4001 & 0.6671 \\
0 & -0.036 & 0.0396 & 0.1365 & -0.433 & 0.62 & -0.5576 & 0.3092 \\
-0.125 & 0.1637 & -0.1283 & 0.0806 & -0.0335 & -0.0012 & 0.0169 & -0.0141
\end{bmatrix}, \tag{42}
$$

and

$$
b_{1\times8} = \begin{bmatrix} 0.125 & 0.2094 & 0.294 & 0.3949 & 0.4665 & 0.4692 & 0.3832 & 0.2158 \end{bmatrix} \tag{43}
$$

In (37), (38) and (39), A is equal to B^T. This is because V is equal to U^T in (15) and (16) when the DWT is considered using a Daubechies DB-4 filter with a periodic boundary. In addition, A and B^T are expanded with a specific pattern by looking through (37), (38) and (39). Consequently, the composition of the matrix, A and B^T, can be deduced when the size of the matrix is greater than 32×32, and using (28) and (30), the amount of computation can be determined using the inter-conversion matrix when the matrix size is $M \times M$ $(M = 16,\ 24,\ 32,\ \cdots)$. The result is given as

$$MUL : 20M^2, \qquad ADD : 18M^2. \tag{44}$$

In (44), M is a multiple of 8 and greater than or equal to 16. The amount of computation in (44) was also obtained without considering multiplication with zero coefficients in (28) and (30).

To compare with the amount of computation between the straightforward and proposed approaches, we apply the assumptions given in (35) and (36), which result in

$$MUL : 24M^2, \qquad ADD : 20M^2. \tag{45}$$

A comparison of (44) and (45) showed that the proposed approach reduced amount of computation significantly compared to the straightforward version.

4. Conclusion

In this paper, the matrix representation was derived for the BDCT-to-DWT and the DWT-to-BDCT inter-conversion. Based on the practical consideration the 2D, 8×8 BDCT and one-level DWT, was used to compare the amount of computation between the straightforward and proposed approaches. The proposed approach presents the inter-conversion matrices for BDCT-to-DWT and DWT-to-BDCT. Based on an analysis of both approaches, we confirmed that the proposed approach requires more than 20% less computation. This observation coincides with the main purpose of transcoding that the conversion of compression format should be performed with as small amount of computation as possible. Although the fast algorithms for DCT and DWT, were not considered for the comparison, the fast version of the proposed approach will be developed in future research. The proposed inter-conversion method can be a theoretical background for transcoding the DCT-based H.26x and the DWT-based JPEG-2000 compression standards.

An automatic field-adaptive focal length estimation method was presented for correcting the fisheye lens distorted images. The proposed method divides the corrected crop image of the input image into ten fields, and estimates the focal length for characterizing the orthographic projection model. The proposed algorithm can accurately calibrate the distortion made by the wide angle lens, and can be applied to a range of imaging systems.

Acknowledgement

This work was supported by the Ministry of Science, ICT & Future Planning as Software Grand Challenge Project (grant no. 14-824-09-003) and by the MSIP, Korea, under the ITRC support program (NIPA-2014-CAU) supervised by NIPA.

References

[1] J. Xin, C. W. Lin, and M. T. Sun: ‘Digital Video Transcoding’, Proceedings of IEEE, vol. 93, no. 1, Jan. 2005, pp. 84-97. Article (CrossRef Link)

[2] A. Vetro, C. Christopoulos, and H. Sun: ‘Video Transcoding Architectures and Techniques: An Overview’, IEEE Signal Processing Magazine, vol. 20, no. 2, Mar. 2003, pp. 18-29. Article (CrossRef Link)

[3] M. F. Lan, Y. C. Chen, Y. C. Chuang, and E. His: ‘From DCT-based to Wavelet-based Video Transcoding Algorithm and Architecture Design’, Proceedings of 2002 International Conference on Consumer Electronics, Jun. 2002, pp. 236-237. Article (CrossRef Link)

[4] P. C. Hu, M. Kaveh, and Z. L. Zhang: ‘A Wavelet to DCT Progressive Image Transcoder’, In Proc. of 2000 International Conference on Image Processing, Sep. 2000, pp. 968-971. Article (CrossRef Link)

[5] X. Zhang, T. Huang; Y. Tian, M. Geng, S. Ma, and W. Gao, “Fast and Efficient Transcoding Based on Low-Complexity Background Modeling and Adaptive Block Classification,” IEEE Trans. on Multimedia, vol. 15, no. 8, pp. 1769-1785, 2013. Article (CrossRef Link)

[6] Q. Tang and P. Nasiopoulos, “Efficient Motion Re-Estimation With Rate-Distortion Optimization for MPEG-2 to H.264/AVC Transcoding,” vol. 20, no. 2, pp.262-274, 2010. Article (CrossRef Link)

[7] K. Viswanath, J. Mukherjee, and P. K. Biswas, “Wavelet transcoding in the block discrete cosine transform space,” IET Image Processing, vol. 4, no. 3, pp.143-157, 2010. Article (CrossRef Link)

Donggyun Kim was born in Busan, Korea in 1983. He received his B.S. and M.S. degrees in electronic and electrical engineering from Chung-Ang University, Korea, in 2007 and 2009, respectively. Currently, he is pursuing a Ph.D. degree in image processing at Chung-Ang University. His research interests include image restoration, digital auto-focusing, lens distortion correction, and image super-resolution.

Sanghee Lim was born in Seoul, Korea in 1982. He received his B.S. and M.S. degrees in electronic and electrical engineering from Chung-Ang University, Korea, in 2006 and 2008, respectively. His research interests include image compression, object tracking, and image restoration.

Joonki Paik was born in Seoul, Korea in 1960. He received his B.S. degree in control and instrumentation engineering from Seoul National University in 1984. He received his M.S. and the Ph.D. degrees in electrical engineering and computer science from Northwestern University in 1987 and 1990, respectively. From 1990 to 1993, he joined Samsung Electronics, where he designed the image stabilization chip sets for consumer's camcorders. Since 1993, he has been with the faculty at Chung-Ang University, Seoul, Korea, where he is currently a Professor in the Graduate school of Advanced Imaging Science, Multimedia and Film. From 1999 to 2002, he was a visiting Professor at the Department of Electrical and Computer Engineering at the University of Tennessee, Knoxville. Dr. Paik was a recipient of Chester-Sall Award from IEEE Consumer Electronics Society, Academic Award from the Institute of Electronic Engineers of Korea, and Best Research Professor Award from Chung-Ang University. He has served the Consumer Electronics Society of IEEE as a member of the Editorial Board. Since 2005, he has been head of National Research Laboratory in the field of image processing and intelligent systems. In 2008, he worked as a full-time technical consultant for the System LSI Division in Samsung Electronics, where he developed various computational photographic techniques including an extended depth of field (EDoF) system. From 2005 to 2007 he served as Dean of the Graduate School of Advanced Imaging Science, Multimedia, and Film. From 2005 to 2007 he has been Director of Seoul Future Contents Convergence (SFCC) Cluster established by Seoul Research and Business Development (R&BD) Program. Dr. Paik is currently serving as a member of Presidential Advisory Board for Scientific/Technical policy of the Korean Government and a technical consultant of the Korean Supreme Prosecutor's Office for computational forensics.