Enlargement and reduction of image/video via discrete cosine transform pair, part 1: novel three-dimensional discrete cosine transform and enlargement

Shu-Mei Guo
Chen-Bang Li
National Cheng Kung University
Department of Computer Science and Information Engineering
Tainan, Taiwan
E-mail: guosm@mail.ncku.edu.tw

Chia-Wei Chen
Kao Yuan University
Department of Mechanical and Automation Engineering
Lujhu Township, Kaohsiung, Taiwan

Yueh-Ching Liao
National Cheng Kung University
Department of Computer Science and Information Engineering
Tainan, Taiwan

Jason Sheng-Hong Tsai
National Cheng Kung University
Department of Electric Engineering
Tainan, Taiwan
E-mail: shtsai@mail.ncku.edu.tw

Abstract. We propose novel matrix-form representations of the conventional forward three-dimensional (3-D) discrete cosine transform (DCT) and the inverse 3-D DCT. The pair of the computationally efficient 3-D DCTs causes the formulation to be more concise with a systematic structure, which has a high potential for the development of other applications via the 3-D DCT pair. Applications for enlargement of image/video implemented by the DCT pair are also presented. These new algorithms effectively eliminate ripple and blocky effects and maintain the original characteristic. Moreover, due to the essentially even-function property of the DCT, the proposed enlargement of an image demonstrates the desired symmetric property. As a result, they can also find numerous applications in image processing and video coding. Simulation results show that proposed mechanisms enable good performance for scaling algorithms of sequences, images, and videos, as expected. © 2007 SPIE and IS&T.

1 Introduction

Recently, much research effort has been devoted to treating and utilizing the 3-D discrete cosine transform (DCT). The 3-D DCT is very efficient when the varieties between sub-blocks are small, which results in low spectral magnitudes in higher frequency components. The 3-D DCT has been successfully applied in video coding, computerized tomography, high-definition television, motion compensation coding, etc. What’s more it can also be applied for 2-D medical image compression. Moreover, improved methods of the compression ratio of the 3-D DCT have been presented.

Concerning reduction of the computing time for a 3-D DCT pair, some algorithms have been proposed such as the 3-D vector-radix decimation-in-frequency algorithm and the row-column-frame (RCF) algorithm or through mapping it to one dimension and using another transform. To the best of our knowledge, no paper has investigated how to perform the 3-D DCT pair through the matrix operation, which is newly proposed in this paper and shown later in Eqs. (13) and (18), in particular, for the case of the nonsquare transformation matrix C of dimension \( n_1 \times n_2 \) presented later in Eq. (32), where \( n_1 \) and \( n_2 \) may not be a power of 2. Therefore, the application of video enlargement proposed in this paper is not constrained by the special type of the original video size via the novel matrix form of 3-D DCT.

Image interpolation plays a central role in many applications. It is one of the fundamental signal processing operations, and it is required for resolution conversion to adapt to the characteristics of a particular display device. For digital images, interpolation is necessary to change the size of a digital image when viewed on a certain display device, and it is also used in image processing.
such as in any geometric transformation or warping of images. Interpolation is also one of the intermediate operations in the multirate processing of images.

A standard approach for interpolation is to fit the original data with a continuous model and resample this function on a new sampling grid. For ideal linear and stationary systems, an optimal approach for interpolation is sinc function, which would enable exact reconstruction of a band-limited signal. Due to the impossibility of realizing the function physically, different approximations have been devised.

The principle that is common to all interpolation schemes is to determine the parameters of a continuous image representation from a set of discrete data. In the case of nearest neighbor interpolation, the underlying image model is a polynomial spline of zero order. This is extremely simple to implement but tends to produce an image with a blocky appearance. More satisfactory results can be obtained with bilinear interpolation, which uses an implicit first-order spline model, or by using cubic convolution, parametric cubic convolution, and high order spline.

To overcome this difficulty, the novel mechanisms in this paper are proposed using a higher order approximation. The proposed mechanism for the enlargement of a sequence/image significantly improves the blocky and ripple effects produced by the conventional DCT approaches. Notice that the proposed blocky/ripple cancellation mechanism proposed here can also improve the quality of enlargement of an image via the high-order B-spline, cubic convolution, and parametric cubic convolution, since these mechanisms all involve a high-order (at least second-order) interpolation in the spatial domain.

Due to the essentially even-function property of the DCT, the proposed enlargement of a sequence/image at here demonstrates the desired symmetric property. As for other approaches, whether in the frequency domain or the spatial domain, the enlargement of an image induces an asymmetric property, even though the original image is symmetric. This situation becomes obvious for a very large enlargement. Unfortunately, the symmetric property is required for practical use in most cases.

This paper is organized as follows. Section 2 proposes a novel matrix operation for the 3-D DCT pair. The enlargement algorithm with ripple and blocky effect cancellations for 2-D and 3-D cases is proposed in Sec. 3 with an integer enlargement ratio. The arbitrary-ratio enlargement mechanism with ripple and blocky effect cancellations for 2-D and 3-D cases is further proposed in Sec. 3. Section 4 gives a brief conclusion and discussion. Computer simulation results of the proposed algorithms are presented at the end of respective sections.

2 Novel Matrix Form Representation of 3-D DCT Pair

The main idea of the computationally efficient 3-D DCT pair is to reduce the repeated operations (i.e., cosine functions) and segment the original data with smaller cuboids to decrease memory size. Moreover, the potent 3-D DCT must be convenient to use in actual applications. In the following, the matrix form of 3-D DCT can be applied in our proposed video enlargement method without the special media size, making it more suitable to be realized by a microprocessor.

2.1 Forward Novel 3-D DCT

The conventional forward 3-D DCT is defined as

\[
F(u, v, w) = \left( \frac{2}{n_1} \right)^{1/2} \left( \frac{2}{n_2} \right)^{1/2} \left( \frac{2}{n_3} \right)^{1/2} \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} \sum_{k=0}^{n_3-1} C(u)C(v)C(w) \times \sum_{i'=0}^{n_1-1} \sum_{j'=0}^{n_2-1} \sum_{k'=0}^{n_3-1} f(i', j', k') \cos \left( \frac{(2i+1)u\pi}{2n_1} \right) \cos \left( \frac{(2j+1)v\pi}{2n_2} \right) \cos \left( \frac{(2k+1)w\pi}{2n_3} \right),
\]

where

\[
C(u), C(v), C(w) = 1/\sqrt{2} \quad \text{for } u, v, w = 0,
\]

\[
C(u), C(v), C(w) = 1 \quad \text{for } u, v, w = \text{otherwise},
\]

\[
f(i', j', k') \text{ are the data in the spatial domain for } 0 \leq i' \leq n_1 - 1, 0 \leq j' \leq n_2 - 1, \text{ and } 0 \leq k' \leq n_3 - 1; \quad \text{and } F(u, v, w) \text{ are the data in the frequency domain for } 0 \leq u \leq n_1 - 1, 0 \leq v \leq n_2 - 1, \text{ and } 0 \leq w \leq n_3 - 1. \text{ Notice that } n_i \text{ for } i = 1, 2, 3 \text{ may not be power of 2. Then, Eq. (1) can be rewritten as}
\]

\[
F(u, v, w) = \left( \frac{2}{n_1} \right)^{1/2} \left( \frac{2}{n_2} \right)^{1/2} \left( \frac{2}{n_3} \right)^{1/2} C(u) \sum_{i=0}^{n_1-1} \cos \left( \frac{(2i+1)u\pi}{2n_1} \right) C(v) \sum_{j=0}^{n_2-1} \cos \left( \frac{(2j+1)v\pi}{2n_2} \right) C(w) \sum_{k=0}^{n_3-1} \cos \left( \frac{(2k+1)w\pi}{2n_3} \right) \left. \begin{bmatrix}
    f(i, 0, 0) & f(i, 0, 1) & \cdots & f(i, 0, n_3-1) \\
f(i, 1, 0) & f(i, 1, 1) & \cdots & f(i, 1, n_3-1) \\
    \vdots & \vdots & \ddots & \vdots \\
f(i, n_3-1, 0) & f(i, n_3-1, 1) & \cdots & f(i, n_3-1, n_3-1)
\end{bmatrix} \right)
\]

To deal with more \( F(u, v, w) \) at the same time, set
Guo et al.: Enlargement and reduction of image/video via discrete cosine transform...

\[ \hat{F}(u) = \begin{bmatrix}
F(u,0,0) & F(u,0,1) & \cdots & F(u,0,n_3 - 1) \\
F(u,1,0) & F(u,1,1) & \cdots & F(u,1,n_3 - 1) \\
\vdots & \vdots & \ddots & \vdots \\
F(u,n_2 - 1,0) & F(u,n_2 - 1,1) & \cdots & F(u,n_2 - 1,n_3 - 1)
\end{bmatrix}. \tag{3} \]

From Eqs. (2) and (3), one has

\[
\hat{F}(u) = \left( \frac{2}{n_1} \right)^{1/2} C(u) \sum_{i=0}^{n_v - 1} \cos \left( \frac{(2i + 1) \pi}{2n_1} \right) \begin{bmatrix}
C(0) \cos \left( \frac{2 \cdot 0 + 1) \pi}{2n_2} \right) & C(0) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n_2} \right) & \cdots & C(0) \cos \left( \frac{2(n_2 - 1) + 1) \pi}{2n_2} \right) \\
C(0) \cos \left( \frac{2 \cdot 0 + 1) \pi}{2n_2} \right) & C(0) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n_2} \right) & \cdots & C(0) \cos \left( \frac{2(n_2 - 1) + 1) \pi}{2n_2} \right) \\
\vdots & \vdots & \ddots & \vdots \\
C(n_v - 1) \cos \left( \frac{2 \cdot 0 + 1) \pi}{2n_2} \right) & C(n_v - 1) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n_2} \right) & \cdots & C(n_v - 1) \cos \left( \frac{2(n_2 - 1) + 1) \pi}{2n_2} \right)
\end{bmatrix} \times \begin{bmatrix}
f(i,0,0) & f(i,0,1) & \cdots & f(i,0,n_3 - 1) \\
f(i,1,0) & f(i,1,1) & \cdots & f(i,1,n_3 - 1) \\
\vdots & \vdots & \ddots & \vdots \\
f(i,n_v - 1,0) & f(i,n_v - 1,1) & \cdots & f(i,n_v - 1,n_3 - 1)
\end{bmatrix} \times \left( \frac{2}{n_3} \right)^{1/2} \begin{bmatrix}
C(0) \cos \left( \frac{2 \cdot 0 + 1) \pi}{2n_3} \right) & C(0) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n_3} \right) & \cdots & C(0) \cos \left( \frac{2(n_3 - 1) + 1) \pi}{2n_3} \right) \\
C(0) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n_3} \right) & C(0) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n_3} \right) & \cdots & C(0) \cos \left( \frac{2(n_3 - 1) + 1) \pi}{2n_3} \right) \\
\vdots & \vdots & \ddots & \vdots \\
C(n_3 - 1) \cos \left( \frac{2 \cdot 0 + 1) \pi}{2n_3} \right) & C(n_3 - 1) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n_3} \right) & \cdots & C(n_3 - 1) \cos \left( \frac{2(n_3 - 1) + 1) \pi}{2n_3} \right)
\end{bmatrix}. \tag{4}
\]

For the convenience of notation, set the orthogonal matrix \( C_\nu \) as

\[
C_\nu = \left( \frac{2}{n_3} \right)^{1/2} \begin{bmatrix}
C(0) \cos \left( \frac{2 \cdot 0 + 1) \pi}{2n} \right) & C(0) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n} \right) & \cdots & C(0) \cos \left( \frac{2(n - 1) + 1) \pi}{2n} \right) \\
C(1) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n} \right) & C(1) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n} \right) & \cdots & C(1) \cos \left( \frac{2(n - 1) + 1) \pi}{2n} \right) \\
\vdots & \vdots & \ddots & \vdots \\
C(n - 1) \cos \left( \frac{2 \cdot 0 + 1) \pi}{2n} \right) & C(n - 1) \cos \left( \frac{2 \cdot 1 + 1) \pi}{2n} \right) & \cdots & C(n - 1) \cos \left( \frac{2(n - 1) + 1) \pi}{2n} \right)
\end{bmatrix}. \tag{5}
\]

and

\[
\hat{f}(i) = \begin{bmatrix}
f(i,0,0) & f(i,0,1) & \cdots & f(i,0,n_3 - 1) \\
f(i,1,0) & f(i,1,1) & \cdots & f(i,1,n_3 - 1) \\
\vdots & \vdots & \ddots & \vdots \\
f(i,n_v - 1,0) & f(i,n_v - 1,1) & \cdots & f(i,n_v - 1,n_3 - 1)
\end{bmatrix}. \tag{6}
\]

Then, Eq. (4) can be simplified as

\[
\hat{F}(u) = \left( \frac{2}{n_1} \right)^{1/2} C(u) \sum_{i=0}^{n_v - 1} \cos \left( \frac{(2i + 1) \pi}{2n_1} \right) C_\nu \hat{f}(i) C_\nu^T
\]

\[
\hat{F}(u) = \left( \frac{2}{n_1} \right)^{1/2} C(u) \sum_{i=0}^{n_v - 1} \cos \left( \frac{(2i + 1) \pi}{2n_1} \right) C_\nu \hat{f}(i) C_\nu^T
\]

\[
\hat{F}(u) = \left( \frac{2}{n_1} \right)^{1/2} C(u) \sum_{i=0}^{n_v - 1} \cos \left( \frac{(2i + 1) \pi}{2n_1} \right) C_\nu \hat{f}(i) C_\nu^T
\]
\[
\hat{F}(u) = \left( \frac{2}{n_1} \right)^{1/2} \begin{bmatrix}
C(u) \cos \left( \frac{2 \cdot 0 + 1 \pi}{2n_1} \right) I_{n_2} & C(u) \cos \left( \frac{2 \cdot 1 + 1 \pi}{2n_1} \right) I_{n_2} & \cdots & C(u) \cos \left( \frac{2(n_1 - 1) + 1 \pi}{2n_1} \right) I_{n_2} \\
\vdots & \vdots & & \vdots \\
\end{bmatrix} \times \begin{bmatrix}
C_{n_2} \hat{f}(0) C_{n_3}^T \\
C_{n_2} \hat{f}(1) C_{n_3}^T \\
\vdots \\
C_{n_2} \hat{f}(n_1 - 1) C_{n_3}^T \\
\end{bmatrix},
\]

where \( I_{n_2} \) denotes the \( n_2 \times n_2 \) dimensional identity matrix.

To further consider all the data \( F(u,v,w) \) in the frequency domain for \( 0 \leq u \leq n_1 - 1, 0 \leq v \leq n_2 - 1, \) and \( 0 \leq w \leq n_3 - 1 \) at the same time, set

\[
F = \begin{bmatrix}
\hat{F}(0) \\
\hat{F}(1) \\
\vdots \\
\hat{F}(n_1 - 1) \\
\end{bmatrix}
\]

and

\[
F = \left( \frac{2}{n_1} \right)^{1/2} \begin{bmatrix}
C(0) \cos \left( \frac{2 \cdot 0 + 1 \pi}{2n_1} \right) I_{n_2} & C(0) \cos \left( \frac{2 \cdot 1 + 1 \pi}{2n_1} \right) I_{n_2} & \cdots & C(0) \cos \left( \frac{2(n_1 - 1) + 1 \pi}{2n_1} \right) I_{n_2} \\
C(1) \cos \left( \frac{2 \cdot 0 + 1 \pi}{2n_1} \right) I_{n_2} & C(1) \cos \left( \frac{2 \cdot 1 + 1 \pi}{2n_1} \right) I_{n_2} & \cdots & C(1) \cos \left( \frac{2(n_1 - 1) + 1 \pi}{2n_1} \right) I_{n_2} \\
\vdots & \vdots & & \vdots \\
C(n_1 - 1) \cos \left( \frac{2 \cdot 0 + 1 \pi}{2n_1} \right) I_{n_2} & C(n_1 - 1) \cos \left( \frac{2 \cdot 1 + 1 \pi}{2n_1} \right) I_{n_2} & \cdots & C(n_1 - 1) \cos \left( \frac{2(n_1 - 1) + 1 \pi}{2n_1} \right) I_{n_2} \\
\end{bmatrix} \times \begin{bmatrix}
C_{n_2} \hat{f}(0) C_{n_3}^T \\
C_{n_2} \hat{f}(1) C_{n_3}^T \\
\vdots \\
C_{n_2} \hat{f}(n_1 - 1) C_{n_3}^T \\
\end{bmatrix},
\]

Equations (8)–(10) imply

\[
\hat{F}(u) = (C_{n_2} \otimes I_{n_3}) \text{diag}(C_{n_2}, C_{n_2}, \ldots, C_{n_2}) F C_{n_3}^T,
\]

where \( \otimes \) denotes the Kronecker product and \( \text{diag}(C_{n_2}, C_{n_2}, \ldots, C_{n_2}) \) is a block diagonal matrix of dimension \((n_2 n_3) \times (n_2 n_3)\) with block diagonal elements \( C_{n_2} \). The Kronecker product between two matrices \( A \) and \( B \) is defined as follows:

\[
A = \{a_{ij}; i = 1, 2, 3, \ldots; m; j = 1, 2, 3, \ldots, n\}, \quad B = \{b_{ij}\},
\]

where the dimension of \( B \) is arbitrary. Equation (12) transfers the spatial domain \( f(i,j,k) \) into the frequency domain \( F(u,v,w) \) for \( 0 \leq i \leq n_1 - 1, 0 \leq j \leq n_2 - 1, 0 \leq k \leq n_3 - 1, 0 \)
\[ F = \begin{bmatrix} \hat{F}(0) \\ \hat{F}(1) \\ \vdots \\ \hat{F}(n-1) \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n-1} C_n(0,i) \hat{f}(i) C_n^T(i) \\ \sum_{i=0}^{n-1} C_n(1,i) \hat{f}(i) C_n^T(i) \\ \vdots \\ \sum_{i=0}^{n-1} C_n(n-1,i) \hat{f}(i) C_n^T(i) \end{bmatrix} \]

\[ Q(i) = C_n \hat{f}(i) C_n^T(i) \]

Note here that \( n \) may not be a power of 2. The details of the novel forward 3-D DCT are summarized as follows:

### 2.1.1 Algorithm 1

1. Subdivide the original spatial data into nonoverlapping \( n \times n \times n \) subcuboids, \( f(i,j,k) \), where \( 0 \leq i \leq n - 1, 0 \leq j \leq n - 1, \) and \( 0 \leq k \leq n - 1 \).
2. Perform the 3-D DCT on each subcuboid \( f \) using Eq. (15).
3. Other remaining subblocks that can be subdivided into the normal \( n \times n \times n \) form are performed by Eq. (13) appropriately.

Theoretically, the conventional 3-D DCT of Eq. (1) takes \( n^6 \) addition operations, \( 3n^6 + 6n^3 \) multiplication operations, and \( 3n^6 \) operations for cosine functions; however, the novel 3-D DCT of Eq. (15) requires only \( 3n^4(n-1) \) addition operations, \( 3n^4 \) multiplication operations, and \( n^2 \) operations for cosine functions to have \( C_n \). Thus, the proposed approach of Eq. (15) requires fewer operations. The total numbers of arithmetic operations (including trivial operations) for the 3-D DCT-II vector-radix decimation-in-frequency (3-D VR DIF) algorithm are \( 7/8 \cdot n^6 \log_2 n \) multiplications and \( 9/2 \cdot n^4 \log_2 n - 3n^3 + 3n^2 \) additions, which are even less than our proposed algorithm. However, it is required to have \( n \) as power of 2 and \( n_1 = n_2 = n_3 = n \). Similar situations appear in the literature for the case of an inverse 3-D DCT. Indeed, by appropriately taking advantage of Refs. 12–14, we can improve the computation efficiency of our proposed algorithm for the special case \( n_1 = n_2 = n_3 = n \) and \( n \) is power of 2, but this is beyond the scope of this paper.

### 2.2 The Inverse Novel 3-D DCT

The conventional inverse 3-D DCT is defined as

\[ f(i,j,k) = \frac{2}{n_1} \left( \frac{2}{n_2} \right)^{1/2} \left( \frac{2}{n_3} \right)^{1/2} \]

\[ \times \sum_{u=0}^{n_1-1} \sum_{v=0}^{n_2-1} \sum_{w=0}^{n_3-1} C(u)C(v)C(w)F(u,v,w)\cos \left( \frac{(2i+1)u\pi}{2n_1} \right) \]

\[ \times \cos \left( \frac{(2j+1)v\pi}{2n_2} \right) \cos \left( \frac{(2k+1)w\pi}{2n_3} \right), \]

where

\[ C(u), C(v), C(w) = 1/\sqrt{2} \quad \text{for } u,v,w = 0, \]

\[ C(u), C(v), C(w) = 1 \quad \text{for } u,v,w = \text{otherwise}, \]

\( f(i,j,k) \) are the data in the spatial domain for \( 0 \leq i \leq n_1 - 1, 0 \leq j \leq n_2 - 1, \) and \( 0 \leq k \leq n_3 - 1, \) and \( F(u,v,w) \) are the data in the frequency domain for \( 0 \leq u \leq n_1 - 1, 0 \leq v \leq n_2 - 1, \) and \( 0 \leq w \leq n_3 - 1. \) Through the same approach as shown in Sec. 2.1, one has

\[ Q(i) = C_n \hat{f}(i) C_n^T(i). \]
3 Enlargement of Sequence/Image/Video via Fast 1-D/Fast 2-D/Novel 3-D DCT

Based on the preceding 3-D DCT architecture, interpolating some desired pixels into original pixels of the still image and some desired frames into original ones of the video sequence are presented in this section, respectively. Thus, applications of the DCT pair are proposed for the enlargement of a sequence/image/video.

To enlarge a sequence/image/video, it is desired to take the upsampling of the original basis of the inverse DCT to have more sampling points, then reconstruct the sequence/image/video by the resampled basis. To clearly express the idea, we first describe the detailed content for the 1-D DCT as follows.

3.1 Enlargement of Sequence via Fast 1-D DCT

If \( f(t) \) is an even function that satisfies the Dirichlet condition, then the Fourier series of \( f(t) \) is given by

\[
 f(t) = \frac{1}{2} a_0 + \sum_{n=0}^{\infty} \left( a_n \cos \frac{n\pi t}{p} + b_n \sin \frac{n\pi t}{p} \right),
\]

where \( a_n = (2/p) \int_0^p f(t) \cos(n\pi t/p) dt \), \( b_n = 0 \), and \( 2p \) is the period of \( f(t) \). For the actual application of Fourier series in the image processing, the period is generally selected as \( 2p = 2\pi \).

Consider the time function \( f(i) \), which contains \( N \) points for \( i = 0, 1, 2, \ldots, N-1 \). Redefine a new symmetric 2N point \( f_2(i) \), as

\[
 f_2(i) = \begin{cases} 
 f(i), & 0 \leq i \leq N - 1 \\
 f(2N - 1 - i), & N \leq i \leq 2N - 1.
\end{cases}
\]

Then, take the discrete Fourier transform (DFT) of \( f_2(i) \) as

\[
 F_2(u) = \sum_{i=0}^{N-1} f_2(i) \exp \left( -j \frac{2\pi i u}{2N} \right) + \sum_{u=N}^{2N-1} f_2(i) \exp \left( -j \frac{2\pi i u}{2N} \right)
\]

\[
 = \sum_{u=0}^{N-1} f_2(i) \left\{ \exp \left( -j \frac{2\pi i u}{2N} \right) + \exp \left( j \frac{2\pi (i + 1/2) u}{2N} \right) \right\},
\]

for \( u = 0, 1, 2, \ldots, 2N-1 \). Since the frequency domain function \( F_2(u) \) is neither a real function nor a symmetric one for the specified \( u \), premultiply Eq. (24) by \( \exp(-j(\pi u/2N)) \) to yield

\[
 V(u) = \exp \left( -j \frac{\pi u}{2N} \right) F_2(u)
\]

\[
 = \sum_{i=0}^{N-1} f_2(i) \left\{ \exp \left( -j \frac{2\pi (i + 0.5) u}{2N} \right) \right\}
\]

\[
 = 2 \sum_{i=0}^{N-1} f_2(i) \cos \left( \frac{\pi(i + 0.5) u}{N} \right),
\]

for \( u = 0, 1, 2, \ldots, 2N-1 \). Thus, the inverse DFT of \( V(u) \),
Guo et al.: Enlargement and reduction of image/video via discrete cosine transform...

\( f(i) = f_2(i - 0.5) \), is an even sequence with \( f(j) = f(-j) \), as shown in Fig. 1. After some mathematical manipulations, one has

\[
F(u) = \left( \frac{2}{N} \right)^{1/2} N^{-1} \sum_{i=0}^{N-1} C(u) f(i) \cos \left( \frac{(2i + 1) \pi u}{2N} \right),
\]

(26)

where \( u = 0, 1, 2, \ldots, N-1 \), \( C(u) = 1/\sqrt{2} \) for \( u = 0 \), and \( C(u) = 1 \) for \( u \neq 0 \). From Eq. (23), with some observations, one has

\[
f(i) = \left( \frac{2}{N} \right)^{1/2} N^{-1} \sum_{u=0}^{N-1} C(u) F(u) \cos \left( \frac{(2i + 1) \pi u}{2N} \right),
\]

(27)

where \( i = 0, 1, 2, \ldots, N-1 \), \( C(u) = 1/\sqrt{2} \) for \( u = 0 \), and \( C(u) = 1 \) for \( u \neq 0 \). Equations (26) and (27) are called as the forward 1-D DCT and the inverse 1-D DCT, respectively. The 1-D DCT pair, Eqs. (26) and (27), can also be represented in a matrix form as follows:

\[
F_{1D} = C_N f_{1D},
\]

(28)

and

\[
f_{1D} = C_N^T F_{1D},
\]

(29)

where \( C_N \in \mathbb{R}^{N\times N} \) is shown in Eq. (5), \( f_{1D} = [f(0) \; f(1) \; \cdots \; f(N-1)]^T \in \mathbb{R}^{N\times 1} \), and \( F_{1D} = [F(0) \; F(1) \; \cdots \; F(N-1)]^T \in \mathbb{R}^{N\times 1} \). To have an enlargement of the original sequence \( f(i) \), resample the original basis of the inverse 1-D DCT by more sampling points, then reconstruct the time domain sequence to have the new resampled basis. If we desire to enlarge \( f(i) \) in a \( r \)-multiple length, the basis of inverse 1-D DCT of Eq. (27), \( \cos \left( \frac{(2i + 1) \pi u}{2N} \right) \) for \( i = 0, 1, 2, \ldots, N-1 \), should be modified as \( \cos \left( \frac{(2i + r) \pi u}{2Nr} \right) \) for \( i = 0, 1, 2, \ldots,Nr-1 \). Then the inverse 1-D DCT of Eq. (27) should be reformulated as

\[
f(i) = \left( \frac{2}{N} \right)^{1/2} N^{-1} \sum_{u=0}^{N-1} C(u) F(u) \cos \left( \frac{(2i + r) \pi u}{2Nr} \right),
\]

(30)

where \( i = 0, 1, 2, \ldots, Nr-1 \), \( C(u) = 1/\sqrt{2} \) for \( u = 0 \), and \( C(u) = 1 \) for \( u \neq 0 \). An enlargement interpolation of extendible inverse discrete cosine transform \( \tilde{M} \) (EIDCT) is defined as a sampling rate increment from an \( \tilde{N} \)-point signal to another \( \tilde{M} \)-point one. It is easy to prove theory of the modified IDCT (Eq. (30)) with a shifting parameter \( r = \tilde{M} / \tilde{N} \) makes the enlargement to work correctly as the EIDCT. In addition, Eq. (30) can be represented in a matrix form as follows:

\[
f_{1D \sim r} = C_N^T f_{1D},
\]

(31)

where \( f_{1D} \in \mathbb{R}^{N\times 1} \), \( f_{1D \sim r} \in \mathbb{R}^{N_r\times 1} \), \( C_N \in \mathbb{R}^{N\times N} \), and

\[
C_N = \left( \frac{2}{N} \right)^{1/2} N^{-1} \sum_{u=0}^{N-1} C(u) \cos \left( \frac{(2i + r) \pi u}{2Nr} \right)
\]

(28)

\[
\begin{bmatrix}
C(0) \cos \left( \frac{2 \cdot 0 + r \cdot 0 \pi}{2Nr} \right) & C(0) \cos \left( \frac{2 \cdot 1 + r \cdot 1 \pi}{2Nr} \right) & \cdots & C(0) \cos \left( \frac{2(Nr-1) + r \cdot 0 \pi}{2Nr} \right) \\
C(1) \cos \left( \frac{2 \cdot 0 + r \cdot 1 \pi}{2Nr} \right) & C(1) \cos \left( \frac{2 \cdot 1 + r \cdot 1 \pi}{2Nr} \right) & \cdots & C(1) \cos \left( \frac{2(Nr-1) + r \cdot 1 \pi}{2Nr} \right) \\
\vdots & \vdots & \ddots & \vdots \\
C(N-1) \cos \left( \frac{2 \cdot 0 + r(N-1) \pi}{2Nr} \right) & C(N-1) \cos \left( \frac{2 \cdot 1 + r(N-1) \pi}{2Nr} \right) & \cdots & C(N-1) \cos \left( \frac{2(Nr-1) + r(N-1) \pi}{2Nr} \right)
\end{bmatrix}
\]

(32)
When we desire to deal with the DCT of a long sequence, the sequence is usually subdivided into nonoverlapped subsequences, each in an $8 \times 1$ size. Thus, the size $N$ in Eqs. (28) and (29) is usually chosen as 8. Nevertheless, the blocky effect is raised on the edge of every enlarged subsequence, because when two neighboring subsequences have a discontinuous connection, the interpolated point at the edge of the $i$th subsequence would yield a significant difference with the started point of the $(i + 1)$th subsequence. This situation becomes obvious for a large value of $r$.

Moreover, the slope discontinuity at adjacent ends of two subsequences $i$ and $i + 1$ leads to a amplitude ripple in a reconstructed one. These problems can be eliminated by the proposed method, as addressed in the following.

To enlarge a long sequence $f_{\text{ID}}$ to a new one in the $r$-multiple original size, we first divide $f_{\text{ID}}$ into $N$-point subsequences $f_{\text{ID}(g)}$ that have $p$-point overlaps. The relationship between $f_{\text{ID}}$ and $f_{\text{ID}(g)}$ is shown as follows:

$$f_{\text{ID}(g)} = \{f_{\text{ID}[g^* (N - p)]} f_{\text{ID}[g^* (N - p) + 1]} \cdots f_{\text{ID}[g^* (N - p) + N - 1]}\}^T,$$

where

$$f_{\text{ID}} \in \mathbb{R}^{N \times 1}, \quad f_{\text{ID}(g)} \in \mathbb{R}^{N \times 1}, \quad m = (l - p)/(N - p), \quad g = 0, 1, 2, \ldots, (m - 1).$$

Then, one has the $r$-multiple enlargement of $f_{\text{ID}(g)}$, denoted by $f_{\text{ID} \cdots (g)}$ for $g = 0, 1, 2, \ldots, (m - 1)$, by Eqs. (25) and (28).

To eliminate the blocky and ripple effects, take $f_{\text{ID} \cdots (g)}$, a part of $f_{\text{ID} \cdots (g)}$, to have the desired $r$-multiple enlargement of $f_{\text{ID}}$ as follows:

$$f_{\text{ID} \cdots (g)} = \left[ f_{\text{ID} \cdots (g)} \left( \frac{p}{2} \right) f_{\text{ID} \cdots (g)} \left( \frac{p}{2} + 1 \right) \cdots f_{\text{ID} \cdots (g)} \left( N - 1 - \frac{p}{2} \right) \right]^T,$$

for $g = 1, 2, \ldots, (m - 2)$. For $g = 0$ and $m - 1$, take $f_{\text{ID} \cdots (g)}$ and $f_{\text{ID} \cdots (g)}$ as

$$f_{\text{ID} \cdots (g)} = \left[ f_{\text{ID} \cdots (g)} (0) f_{\text{ID} \cdots (g)} (1) \cdots f_{\text{ID} \cdots (g)} (N - 1 - \frac{p}{2}) \right]^T,$$

and

$$f_{\text{ID} \cdots (g)} = \left[ f_{\text{ID} \cdots (g)} \left( \frac{p}{2} \right) f_{\text{ID} \cdots (g)} \left( \frac{p}{2} + 1 \right) \cdots f_{\text{ID} \cdots (g)} (N - 1) \right]^T,$$

respectively. Note that the parameter $p$ must be chosen as an even integer to symmetrically take the central part of $f_{\text{ID} \cdots (g)}$ as $f_{\text{ID} \cdots (g)}$. Then, compound $f_{\text{ID} \cdots (g)}$ to get $f_{\text{ID} \cdots (g)}$, where $f_{\text{ID}} \in \mathbb{R}^{r \times 1}$ and $f_{\text{ID} \cdots (g)} \in \mathbb{R}^{r \times 1}$. An example is shown in Fig. 2 for $l = 20$, $N = 8$, $r = 2$, and $p = 2$. The points of $f_{\text{ID} \cdots (i)}$ at $i = 0, r, 2r, \ldots, (l - 1)r$ are original points of $f_{\text{ID}}$, and others are interpolated points. In Refs. 29 and 30 overlapping points are averaged or filtered to eliminate the blocky effect. Moreover, a rough idea was proposed to simply truncate some overlapping points. Here, Eqs. (33)–(36) provide a specific and adaptive scheme to process various cases by tuning parameter $p$.

To further reduce the other kind of ripple effect induced by the overdamped bases, i.e., $\cos \{[(2 + r) \pi N]/2N\}$ for $u = 0, 1, \ldots, N - 1$, in Eq. (30), we should shape interpolated points in the spatial domain as

$$f_{\text{ID} \cdots (i)} = \max \{f_{\text{ID} \cdots (i)} (p), f_{\text{ID} \cdots (i)} ((p + 1)r)\}$$

if $f_{\text{ID} \cdots (i)} > \max \{f_{\text{ID} \cdots (i)} (p), f_{\text{ID} \cdots (i)} ((p + 1)r)\},$

$$f_{\text{ID} \cdots (i)} = \min \{f_{\text{ID} \cdots (i)} (p), f_{\text{ID} \cdots (i)} ((p + 1)r)\}$$

if $f_{\text{ID} \cdots (i)} < \min \{f_{\text{ID} \cdots (i)} (p), f_{\text{ID} \cdots (i)} ((p + 1)r)\},$

where $i = pr + 1, pr + 2, \ldots, (p + 1)r - 1$, and $p = 0, 1, 2, \ldots, (l - 2)$. Considering the boundary condition, i.e., the last set of interpolated points, let

$$f_{\text{ID} \cdots (i)} = f_{\text{ID} \cdots (i)} ((l - 1)r)$$

if $i = (l - 1)r + 1, (l - 1)r + 2, \ldots, lr - 1.$

The interpretation of the preceding mechanism is obvious, since interpolated points between two same-gray-level pixels in the spatial domain should also have the consistent gray level. As a result, overdamped pixel values of interpolated points should be shaped to a consistent one.

Based on the preceding analyses and some other detailed interpretations given in later sections, note the following:

1. The proposed mechanism for the enlargement of a sequence/image significantly improves the blocky and ripple effects induced via the conventional DCT approaches. Notice that the blocky/ripple effect cancellation mechanism proposed here can also improve the quality of enlargement of an image via literatures, since the literature all involve a high-order (at least second-order) interpolation in the spatial domain.

2. Due to the essentially even-function property of the DCT, where pixel values $f(i)$ are located at bases $\cos \{[(2 + i) \pi N]/2N\}$ for $i = 0, 1, 2, \ldots, N - 1$, given in Eq. (26) and also illustrated in Fig. 1 where $f(i) = 0(2 + i)$, the proposed enlargement of a sequence/image demonstrates the desired symmetric property also, shown in Example 1 (Figs. 5 and 6 in Sec. 4). Other approaches do not have the even-function property, in either the frequency domain or the spatial domain, so the enlargement of an image has a significantly asymmetric property even though
the original image is symmetric. This situation becomes obvious for a very large enlargement. Unfortunately, the symmetric property is required for a practical use in many cases.

3. The application for the enlargement/reduction of a video is implemented by the computationally efficiency 3-D DCT pair, newly presented in this paper, where nonsquare size transformation matrices \( C \) given in Eqs. (32) and (44) are utilized for the desired goal, in which \( N \) may not be power of 2. To the best of our knowledge, no paper presented a computationally fast algorithm for this case, particularly when \( n_i \) for \( i=1,2,3 \) are not power of 2.

### 3.2 Enlargement of Image Via Fast 2-D DCT

The preceding 1-D algorithm can then be extended to the 2-D case. For the 2-D case, the fast 2-D DCT is

\[
F_{2D} = C_{n_1} F_{2D} C_{n_2}^T, \tag{41}
\]

and the inverse 2-D DCT is

\[
f_{2D} = C_{n_1}^T F_{2D} C_{n_2}, \tag{42}
\]

where

\( f_{2D} \in R^{l_1 \times l_2}, \quad F_{2D} \in R^{l_1 \times l_2}, \)

and the coefficient matrices \( (C_{n_1} \) and \( C_{n_2} \) are defined in Eq. (5). Based on the same idea of enlargement as the 1-D sequence addressed previously, take the upsampling of the 2-D basis and rewrite Eq. (42) as

\[
f_{2D \to r} = C_{n_1,r}^T F_{2D} C_{n_2,r} \tag{43}
\]

to reconstruct the \( r \)-multiple 2-D image denoted by \( f_{2D \to r} \), where \( C_{n_1,r} \in R^{l_1 \times n_1}, \quad C_{n_2,r} \in R^{l_2 \times n_2}, \quad f_{2D \to r} \in R^{l_1 \times l_2}, \quad F_{2D} \in R^{l_1 \times l_2}, \) and \( C_{n_i,r} \) (and \( C_{n_2,r} \)) is defined in Eq. (32). To propose the \( r \)-multiple enlargement mechanism of the 2-D image in a very large and eliminate the blocky and ripple effects, some modifications are required. The \( r \)-multiple enlargement algorithm with blocky and ripple effect cancellations for a large 2-D image is then summarized as follows:

#### 3.2.1 Algorithm 3

Consider the original image \( f_{2D} \) of size \( l_1 \times l_2 \). The pixel element of \( f_{2D} \) is expressed as \( f_{2D}(i,j) \) for \( 0 \leq i \leq l_1-1 \) and \( 0 \leq j \leq l_2-1 \).

1. Subdivide \( f_{2D} \) into subblocks \( f_{2D}(g_1,g_2) \) in a size \( N \times N \), with \( p \) points overlapped at every edge. The relationship between \( f_{2D} \) and \( f_{2D}(g_1,g_2) \) is

![Fig. 2 Example for the enlargement of a 1-D sequence.](image)
\[
\begin{align*}
\mathbf{f}_{2D(g_1,g_2)}(\mathbf{r}) &= \left[
\begin{array}{cccc}
\mathbf{f}_{g_1 \ast (N-p), g_2 \ast (N-p)}(\mathbf{r}) & \mathbf{f}_{g_1 \ast (N-p), g_2 \ast (N-p)+1} & \cdots & \mathbf{f}_{g_1 \ast (N-p), g_2 \ast (N-p)+N-1} \\
\mathbf{f}_{g_1 \ast (N-p)+1, g_2 \ast (N-p)}(\mathbf{r}) & \mathbf{f}_{g_1 \ast (N-p)+1, g_2 \ast (N-p)+1} & \cdots & \mathbf{f}_{g_1 \ast (N-p)+1, g_2 \ast (N-p)+N-1} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{f}_{g_1 \ast (N-p)+N-1, g_2 \ast (N-p)}(\mathbf{r}) & \mathbf{f}_{g_1 \ast (N-p)+N-1, g_2 \ast (N-p)+1} & \cdots & \mathbf{f}_{g_1 \ast (N-p)+N-1, g_2 \ast (N-p)+N-1}
\end{array}
\right],
\end{align*}
\]
for
\[
m_1 = (l_1 - p)/(N-p), \quad g_1 = 0, 1, \ldots, (m_1 - 1), \quad m_2 = (l_2 - p)/(N-p),
\]
\[g_2 = 0, 1, \ldots, (m_2 - 1), \quad l_1 > N, \quad l_2 > N, \quad \mathbf{f}_{2D} \in \mathbb{R}^{l_1 \times l_2}, \quad \text{and} \quad \mathbf{f}_{2D(g_1,g_2)} \in \mathbb{R}^{N \times N}.
\]

2. Enlarge \(\mathbf{f}_{2D(g_1,g_2)}\) for \(m_1=\mathbb{N}\) and \(m_2=\mathbb{N}\) by Eqs. (41) and (43) to have the frequency-domain coefficient matrix of subblock \(\mathbf{F}_{2D(g_1,g_2)}\) and the \(r\)-multiple image \(\mathbf{f}_{2D-\hat{r}(g_1,g_2)}\) respectively, where \(\mathbf{F}_{2D(g_1,g_2)} = \mathbf{C}^T_{\alpha} \mathbf{F}_{2D(g_1,g_2)} \mathbf{C}_{\alpha}, \quad \mathbf{f}_{2D-\hat{r}(g_1,g_2)} = \mathbf{C}_{\alpha} \mathbf{F}_{2D(g_1,g_2)} \mathbf{C}_{\alpha} \mathbf{r}, \quad \mathbf{F}_{2D(g_1,g_2)} \in \mathbb{R}^{N \times N}, \quad \text{and} \quad \mathbf{f}_{2D-\hat{r}(g_1,g_2)} \in \mathbb{R}^{N \times N}.
\]

3. Take \(\mathbf{f}_{2D-\hat{r}(g_1,g_2)}\), a part of \(\mathbf{f}_{2D-\hat{r}(g_1,g_2)}\), where
\]
\[
\mathbf{f}_{2D-\hat{r}(g_1,g_2)} = \left[
\begin{array}{cccc}
\mathbf{f}_{2D-\hat{r}(g_1,g_2)}(0, \frac{p_1}{2}) & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(0, \frac{p_1}{2}+1) & \cdots & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(0, N_r - \frac{p_1}{2}) \\
\mathbf{f}_{2D-\hat{r}(g_1,g_2)}(\frac{p_1}{2}, \frac{p_2}{2}) & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(\frac{p_1}{2}, \frac{p_2}{2}+1) & \cdots & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(\frac{p_1}{2}, N_r - \frac{p_2}{2}) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{f}_{2D-\hat{r}(g_1,g_2)}(N_r - \frac{p_1}{2}, \frac{p_2}{2}) & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(N_r - \frac{p_1}{2}, \frac{p_2}{2}+1) & \cdots & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(N_r - \frac{p_1}{2}, N_r - \frac{p_2}{2})
\end{array}
\right],
\]
and \(\mathbf{f}_{2D-\hat{r}(g_1,g_2)}(\alpha, \beta)\) is the \(\alpha\)'th row, \(\beta\)'th column element of \(\mathbf{f}_{2D-\hat{r}(g_1,g_2)}\), except for the boundary conditions at \(g_1=0, \quad g_1=m_1-1, \quad g_2=0, \) or \(g_2=m_2-1. \) For \(g_1=0, \quad g_1=m_1-1, \quad g_2=0, \) and \(g_2=m_2-1, \) \(\mathbf{f}_{2D-\hat{r}(g_1,g_2)}\) are chosen as
\[
\mathbf{f}_{2D-\hat{r}(g_1,g_2)} = \left[
\begin{array}{cccc}
\mathbf{f}_{2D-\hat{r}(g_1,g_2)}(0, \frac{p_1}{2}) & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(0, \frac{p_1}{2}+1) & \cdots & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(0, N_r - \frac{p_1}{2}) \\
\mathbf{f}_{2D-\hat{r}(g_1,g_2)}(\frac{p_1}{2}, \frac{p_2}{2}) & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(\frac{p_1}{2}, \frac{p_2}{2}+1) & \cdots & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(\frac{p_1}{2}, N_r - \frac{p_2}{2}) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{f}_{2D-\hat{r}(g_1,g_2)}(N_r - \frac{p_1}{2}, \frac{p_2}{2}) & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(N_r - \frac{p_1}{2}, \frac{p_2}{2}+1) & \cdots & \mathbf{f}_{2D-\hat{r}(g_1,g_2)}(N_r - \frac{p_1}{2}, N_r - \frac{p_2}{2})
\end{array}
\right].
\]
respectively. For \((g_1, g_2) = (0, 0)\), \((g_1, g_2) = (0, m_2 - 1)\), \((g_1, g_2) = (m_1 - 1, 0)\) and \((g_1, g_2) = (m_1 - 1, m_2 - 1)\), \(\hat{f}_{2D \rightarrow r(g_1, g_2)}\) are naturally chosen as

\[
\begin{bmatrix}
    f_{2D \rightarrow r(g_1, g_2)}(0, 0) & f_{2D \rightarrow r(g_1, g_2)}(0, 1) & \cdots & f_{2D \rightarrow r(g_1, g_2)}(0, N_r - 1) \\
    f_{2D \rightarrow r(g_1, g_2)}(1, 0) & f_{2D \rightarrow r(g_1, g_2)}(1, 1) & \cdots & f_{2D \rightarrow r(g_1, g_2)}(1, N_r - 1) \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{2D \rightarrow r(g_1, g_2)}(N_r - 1, -1, 0) & f_{2D \rightarrow r(g_1, g_2)}(N_r - 1, -1, 1) & \cdots & f_{2D \rightarrow r(g_1, g_2)}(N_r - 1, -1, N_r - 1) \\
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
    f_{2D \rightarrow r(g_1, g_2)}(0, 0) & f_{2D \rightarrow r(g_1, g_2)}(0, 1) & \cdots & f_{2D \rightarrow r(g_1, g_2)}(0, N_r - 1) \\
    f_{2D \rightarrow r(g_1, g_2)}(1, 0) & f_{2D \rightarrow r(g_1, g_2)}(1, 1) & \cdots & f_{2D \rightarrow r(g_1, g_2)}(1, N_r - 1) \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{2D \rightarrow r(g_1, g_2)}(N_r - 1, 0) & f_{2D \rightarrow r(g_1, g_2)}(N_r - 1, 1) & \cdots & f_{2D \rightarrow r(g_1, g_2)}(N_r - 1, 1, N_r - 1) \\
\end{bmatrix}
\]

Note that the parameter \(p\) must be chosen as an even integer to symmetrically take the central part of \(f_{2D \rightarrow r(g_1, g_2)}\) as \(f_{2D \rightarrow r(g_1, g_2)}\).

4. Compound subblocks in step 3, \(\hat{f}_{2D \rightarrow r(g_1, g_2)}\), to have \(f_{2D \rightarrow r}\) which is the \(r\)-multiple enlargement of \(f_{2D}\), where \(f_{2D \rightarrow r} \in R^{r_1 \times r_2}\).
5. Modify horizontal interpolated points of $f_{2D-\rightarrow r}$ as

$$f_{2D-\rightarrow r}(p_1 r, j) = \min\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[p_1 r, (p_2 + 1)r]\} \text{ if } f_{2D-\rightarrow r}(p_1 r, j) < \min\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[p_1 r, (p_2 + 1)r]\} \text{ for } j = p_2 r + 1, \ldots, p_2 r + 2, \ldots, (p_2 + 1)r - 1,$$

$$f_{2D-\rightarrow r}(p_1 r, j) = \max\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[p_1 r, (p_2 + 1)r]\} \text{ if } f_{2D-\rightarrow r}(p_1 r, j) > \max\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[p_1 r, (p_2 + 1)r]\} \text{ for } j = p_2 r + 1, \ldots, p_2 r + 2, \ldots, (p_2 + 1)r - 1,$$

where $p_1 = 0, 1, \ldots, (l_1 - 1)$ and $p_2 = 0, 1, \ldots, (l_2 - 2)$.

6. Modify vertical interpolated points of $f_{2D-\rightarrow r}$ as

$$f_{2D-\rightarrow r}(i, p_2 r) = \min\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r]\} \text{ if } f_{2D-\rightarrow r}(i, p_2 r) < \min\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r]\} \text{ for } i = p_1 r + 1, \ldots, (p_1 + 1)r - 1,$$

$$f_{2D-\rightarrow r}(i, p_2 r) = \max\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r]\} \text{ if } f_{2D-\rightarrow r}(i, p_2 r) > \max\{f_{2D-\rightarrow r}(p_1 r, p_2 r), f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r]\} \text{ for } i = p_1 r + 1, \ldots, (p_1 + 1)r - 1,$$

where $p_1 = 0, 1, \ldots, (l_1 - 2)$ and $p_2 = 0, 1, \ldots, (l_2 - 1)$.

7. Set

$$f_{2D-\rightarrow r, \min}(p_1, p_2) = \min\{f_{2D-\rightarrow r}(p_1 r, p_2 r + 1), f_{2D-\rightarrow r}(p_1 r, p_2 r + 2), \ldots, f_{2D-\rightarrow r}[p_1 r, (p_2 + 1)r - 1], f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r + 1], f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r + 2], \ldots, f_{2D-\rightarrow r}[(p_1 + 1)r, (p_2 + 1)r - 1], f_{2D-\rightarrow r}[(p_1 + 2)r, (p_2 + 1)r - 1], \ldots, f_{2D-\rightarrow r}[(p_1 + 1)r, (p_2 + 1)r] + 1\},$$

and

$$f_{2D-\rightarrow r, \max}(p_1, p_2) = \max\{f_{2D-\rightarrow r}(p_1 r, p_2 r + 1), f_{2D-\rightarrow r}(p_1 r, p_2 r + 2), \ldots, f_{2D-\rightarrow r}[p_1 r, (p_2 + 1)r - 1], f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r + 1], f_{2D-\rightarrow r}[(p_1 + 1)r, p_2 r + 2], \ldots, f_{2D-\rightarrow r}[(p_1 + 1)r, (p_2 + 1)r - 1], f_{2D-\rightarrow r}[(p_1 + 2)r, (p_2 + 1)r - 1], \ldots, f_{2D-\rightarrow r}[(p_1 + 1)r, (p_2 + 1)r] + 1\},$$

where $p_1 = 0, 1, \ldots, (l_1 - 1)$ and $p_2 = 0, 1, \ldots, (l_2 - 1)$.

8. Modify the last group of horizontal interpolated points of $f_{2D-\rightarrow r}$ as

$$f_{2D-\rightarrow r}(i, j) = f_{2D-\rightarrow r}[(i, (l_2 - 1)r)] \text{ for } j = (l_2 - 1)r + 1, \ldots, (l_2 - 1)r + 2, \ldots, l_2 r - 1,$$

where $i = 0, 1, \ldots, (l_1 - 1)r$.

9. Modify the last group of vertical interpolated points of $f_{2D-\rightarrow r}$ as

$$f_{2D-\rightarrow r}(i, j) = f_{2D-\rightarrow r}[(i l_2 - 1)r] \text{ for } i = (l_1 - 1)r + 1, \ldots, (l_1 - 1)r + 2, \ldots, l_1 r - 1,$$

where $j = 0, 1, \ldots, (l_2 - 1)r$.

Notice that if $m_1$ or $m_2$ in step 1 of Algorithm 3 is not integer, it’s desired to replace $m_1$ and $m_2$ by their integral.
parts, respectively. Then, enlarge the new $f_{2D}(g_1,g_2)$ by steps 1 to 4 of Algorithm 3. To deal with those remained pixels of $f_{2D}$, not included in the new $f_{2D}(g_1,g_2)$, take the last $N \times N$ points of $f_{2D}$ as subblocks $f_{2D,\text{last}}$ from top to down and left to right, appropriately. Then, enlarge $f_{2D,\text{last}}$ by Eqs. (41) and (43) to get $r$-multiple subblocks $f_{2D,\text{last}}$ and eliminate subpoints of $f_{2D,\text{last}}$ that overlap with $f_{2D-r}$ to get $f_{2D,\text{last}}$. Finally, compound $f_{2D-r}$ and $f_{2D,\text{last}}$ to have the $r$-multiple enlargement of $f_{2D}$, and appropriately apply steps 5 to 9 of Algorithm 3 to eliminate the ripple effect.

The purposes of steps 1 and 2 of Algorithm 3 are to subdivide the original image to smaller subblocks with $p$

\[
\begin{bmatrix}
    f_{2D-r(g_1,g_2)}(0,0) & f_{2D-r(g_1,g_2)}(0,1) & \cdots & f_{2D-r(g_1,g_2)}(0,Nr-1-pr) \\
    f_{2D-r(g_1,g_2)}(1,0) & f_{2D-r(g_1,g_2)}(1,1) & \cdots & f_{2D-r(g_1,g_2)}(1,Nr-1-pr) \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{2D-r(g_1,g_2)}(Nr-1-pr,0) & f_{2D-r(g_1,g_2)}(Nr-1-pr,1) & \cdots & f_{2D-r(g_1,g_2)}(Nr-1-pr,Nr-1-pr)
\end{bmatrix}
\]

except for the boundary condition at $g_1=m_1-1$, $g_2=m_2-1$, or $(g_1,g_2)=(m_1-1,m_2-1)$. For $g_1=m_1-1$, $g_2=m_2-1$, and $(g_1,g_2)=(m_1-1,m_2-1)$, $f_{2D-r(g_1,g_2)}$ are chosen as

\[
\begin{bmatrix}
    f_{2D-r(g_1,g_2)}(0,0) & f_{2D-r(g_1,g_2)}(0,1) & \cdots & f_{2D-r(g_1,g_2)}(0,Nr-1-pr) \\
    f_{2D-r(g_1,g_2)}(1,0) & f_{2D-r(g_1,g_2)}(1,1) & \cdots & f_{2D-r(g_1,g_2)}(1,Nr-1-pr) \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{2D-r(g_1,g_2)}(Nr-1-pr,0) & f_{2D-r(g_1,g_2)}(Nr-1-pr,1) & \cdots & f_{2D-r(g_1,g_2)}(Nr-1-pr,Nr-1-pr)
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
    f_{2D-r(g_1,g_2)}(0,0) & f_{2D-r(g_1,g_2)}(0,1) & \cdots & f_{2D-r(g_1,g_2)}(0,Nr-1) \\
    f_{2D-r(g_1,g_2)}(1,0) & f_{2D-r(g_1,g_2)}(1,1) & \cdots & f_{2D-r(g_1,g_2)}(1,Nr-1) \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{2D-r(g_1,g_2)}(Nr-1-pr,0) & f_{2D-r(g_1,g_2)}(Nr-1-pr,1) & \cdots & f_{2D-r(g_1,g_2)}(Nr-1-pr,Nr-1)
\end{bmatrix}
\]

respectively.

For the preceding fast Algorithm 3, we can enlarge every subblock from left to right and top to down, and just cover previously enlarged sub-blocks. So, step 3 of the preceding fast Algorithm 3 can actually be neglected. Indeed, the fast Algorithm 3 loses its quality to exchange the operating time.

Indeed, the parameter $p$ in Algorithm 3 dominates the quality and the computational efficiency of the enlargement of a sequence/image/video deeply. A longer $p$ value results in a better quality of the enlargement, but it costs more operating time. By the rule of thumb, it is suggested to choose parameters $N$ and $p$ as 8 and 2, respectively, in general.

### 3.3 Enlargement of Video Via Novel 3-D DCT

For the 3-D case, the novel 3-D DCT is shown in Eqs. (6) and (13) and the novel inverse 3-D DCT are shown in Eqs. (3), (18), and (19). To enlarge the video $f_{3D}$ in a $r$-multiple original size, rewrite Eqs. (18) and (19) as
\[ f_{3D-r} = \begin{bmatrix} f_r(0) \\ f_r(1) \\ \vdots \\ f_r(rn_1 - 1) \end{bmatrix} = \begin{bmatrix} \sum_{n=0}^{n_1-1} c_{n_1,r}(u,0)N_r(u) \\ \sum_{n=0}^{n_1-1} c_{n_1,r}(u,1)N_r(u) \\ \vdots \\ \sum_{n=0}^{n_1-1} c_{n_1,r}(u,rn_1 - 1)N_r(u) \end{bmatrix} \]

where

\[ N_r(u) = C_{r,r}^T \hat{f}(u) c_{n,r} \]

\[ c_{r,r}^T \in R^{n_2 \times n_2}, \quad c_{n,r} \in R^{n_1 \times m_1}, \quad \hat{f}(u) \in R^{n_1 \times n_2}, \quad c_{r,r}(\alpha, \beta) \]
denotes the \( \alpha \)'th row, \( \beta \)'th column element of \( c_{n,r}, N_r \in R^{n_2 \times n_2}, f_r(\bullet) \) is a frame of the enlarged video, \( f_r(\bullet) \in R^{n_2 \times n_2}, f_{3D-r} \) is the desired enlarged video sequence, and \( f_{3D-r} \in R^{n_2 \times n_2 \times m_3} \). The element of video \( f_{3D} \) is expressed as \( f_{3D}(i,j,k) \). Let the first, second, and third indices of the pixel \( f_{3D}(i,j,k) \) denote the time, the horizontal pixel, and the vertical pixel positions, respectively. Due to the same idea of enlargements of 1-D and 2-D cases addressed previously, one could extend it to 3-D case, the enlargement of video with blocky and ripple effect cancellations.

### 3.4 Arbitrary-Ratio Enlargement of Image Via Fast 2-D DCT

New enlargement algorithms with ripple and blocky effect cancellations have been proposed in Secs. 3.1–3.3 for 1-D, 2-D, and 3-D cases, respectively, but the multiple (\( r \)) of proposed algorithms must be an integer. To handle the non-integer case, a new algorithm is proposed in the following. Nevertheless, if an integer case is counted, we suggest applying the proposed algorithms in Secs. 3.1–3.3, since the representation of the noninteger case is much more complicated than the integer case. To shorten the length of the paper, we just address the noninteger enlargement for 2-D case in this paper. The 3-D/1-D case can be extended/simplified from the 2-D case.

Consider an original image \( f_{3D} \) in a size \( n_1 \times n_2 \). To get a \( o_1/o_2 \)-multiple enlarged image \( f_{3D-o_1/o_2} \) in a size \( n_1o_1/n_2o_2 \times [n_2o_2/n_2] \), where \([a]\) denotes the largest integer smaller than \( a, o_1/o_2 > 1, o_1 \) and \( o_2 \) are integers. Thus, \([n_1o_1/n_2o_2]\) and \([n_2o_2/n_2]\) must be matrix, because the size of the enlarged digital image is integer. First, compute \( F_{3D} \), the 2-D DCT of \( f_{3D} \), using Eqs. (41). From Eq. (32), get the coefficient matrix of the \( o_1/o_2 \)-multiple enlargement as

\[
C_{n_1o_1/n_2o_2} = \left( \frac{2}{N} \right)^{1/2} \begin{bmatrix} \cos(2 \cdot 0 \cdot o_2 + o_1)0\pi \sin(0 \cdot o_2 + o_1)0\pi \\ \cos(2 \cdot 0 \cdot o_2 + o_1)1\pi \sin(0 \cdot o_2 + o_1)1\pi \\ \vdots \\ \cos(2 \cdot 0 \cdot o_2 + o_1)(N-1)\pi \sin(0 \cdot o_2 + o_1)(N-1)\pi \end{bmatrix}
\]

Rewrite Eq. (43) as

\[
f_{3D-o_1/o_2} = C_{n_1o_1/n_2o_2}^T F_{3D} C_{n_2o_1/n_2o_2}, \quad (46)
\]

where \( C_{n_1o_1/n_2o_2} \) \( \in R^{n_1 \times (n_1o_1/n_2o_2)}, C_{n_2o_1/n_2o_2} \) \( \in R^{n_2 \times (n_2o_1/n_2o_2)}, f_{3D-o_1/o_2} \) \( \in R^{(n_1o_1/n_2o_2) \times (n_2o_1/n_2o_2)}, C_{n_1o_1/n_2o_2}^T \) and \( C_{n_2o_1/n_2o_2}^T \) are given in Eq. (45).

The detailed noninteger enlargement algorithm of a large 2-D image with blocky and ripple effect cancellations is then proposed as follows:

### 3.4.1 Algorithm 4

Consider the original image \( f_{3D} \) in a size \( l_1 \times l_2 \). The pixel element of \( f_{3D} \) is expressed as \( f_{3D}(i,j) \) for \( 0 \leq i \leq l_1 - 1 \) and \( 0 \leq j \leq l_2 - 1 \). It is desired to get the \( o_1/o_2 \)-multiple enlarged image \( f_{3D-o_1/o_2} \) in a size \([l_1(o_1/o_2)] \times [l_2(o_1/o_2)]\), where \( o_1/o_2 > 1 \).

1. Subdivide \( f_{3D} \) into subblocks \( f_{3D}(s_1,s_2) \) of a size \( N \times N \) with overlapped \( p \) points at every edge. The relationship between \( f_{3D} \) and \( f_{3D}(s_1,s_2) \) is addressed in step 1 of Algorithm 3.
2. Enlarge \( f_{3D}(s_1,s_2) \) by performing Eqs. (41) and (46) for \( n_1 = N \) and \( n_2 = N \); consequently, obtain the frequency-domain coefficient matrix of subblock \( F_{3D}(s_1,s_2) \) and the \( o_1/o_2 \)-multiple image \( f_{3D-o_1/o_2}(s_1,s_2) \), respectively, where \( F_{3D}(s_1,s_2) = C_{n_1o_1/n_2o_2}^T F_{3D}(s_1,s_2) C_{n_1o_1/n_2o_2} \), \( f_{3D-o_1/o_2}(s_1,s_2) = C_{n_1o_1/n_2o_2}^T f_{3D-o_1/o_2}(s_1,s_2) C_{n_1o_1/n_2o_2} \)
3. Take \( f_{3D-o_1/o_2}(s_1,s_2) \), a part of \( f_{3D-o_1/o_2}(s_1,s_2) \), as
where

\[
\tilde{g}_1 = \begin{cases} 
\frac{p_1}{o_2} & \text{if } [\beta_1 (g_1 + 1) - \beta_1 (g_1)] = \frac{o_1}{o_2} \\
\frac{p_1}{o_2} + 1 & \text{if } [\beta_1 (g_1 + 1) - \beta_1 (g_1)] \neq \frac{o_1}{o_2}
\end{cases}
\]

\[
\tilde{g}_2 = \begin{cases} 
\frac{p_1}{o_2} & \text{if } [\beta_2 (g_2 + 1) - \beta_2 (g_2)] = \frac{o_1}{o_2} \\
\frac{p_1}{o_2} + 1 & \text{if } [\beta_2 (g_2 + 1) - \beta_2 (g_2)] \neq \frac{o_1}{o_2}
\end{cases}
\]

\[
\tilde{x}_1 = \begin{cases} 
\frac{p_1}{o_2} & \text{if } [\beta_2 (g_1 + 1) - \beta_2 (g_1)] = \frac{o_1}{o_2} \\
\frac{p_1}{o_2} + 1 & \text{if } [\beta_2 (g_1 + 1) - \beta_2 (g_1)] \neq \frac{o_1}{o_2}
\end{cases}
\]

\[
\tilde{x}_2 = \begin{cases} 
\frac{p_1}{o_2} & \text{if } [\beta_2 (g_2 + 1) - \beta_2 (g_2)] = \frac{o_1}{o_2} \\
\frac{p_1}{o_2} + 1 & \text{if } [\beta_2 (g_2 + 1) - \beta_2 (g_2)] \neq \frac{o_1}{o_2}
\end{cases}
\]

\[
\tilde{\beta}_i = \{ \beta_i (i) | i = 1, 2 \}
\]

\[
\tilde{\beta}_2 = \{ \beta_2 (i) | i = 1, 2 \}
\]

and \( \tilde{f}_{2D-o_1/o_2(g_1, g_2)}(\alpha, \beta) \) is the \( \alpha \)'th row, \( \beta \)'th column element of \( \tilde{f}_{2D-o_1/o_2(g_1, g_2)} \) except for the boundary condition at \( g_1 = m_1 - 1 \) or \( g_2 = m_2 - 1 \).

1. For \( g_1 = m_1 - 1 \) and \( g_2 = m_2 - 1 \), \( \tilde{f}_{2D-o_1/o_2(g_1, g_2)} \) are chosen as

\[
\tilde{f}_{2D-o_1/o_2(g_1, g_2)} = \begin{bmatrix}
0, N_1 & 0 \\
0, N_1 \\
\vdots \\
N_1, 0 \\
N_1, 1 \\
\vdots \\
N_1, N_2
\end{bmatrix}
\]

2. For \( g_1 = (m_1 - 1, m_2 - 1) \), \( \tilde{f}_{2D-o_1/o_2(g_1, g_2)} \) are chosen as

\[
\tilde{f}_{2D-o_1/o_2(g_1, g_2)} = \begin{bmatrix}
0, N_1 & 0 \\
0, N_1 \\
\vdots \\
N_1, 0 \\
N_1, 1 \\
\vdots \\
N_1, N_2
\end{bmatrix}
\]
5. Set
\[
\hat{f}_{2D-o/2}(g_1,g_2) = \begin{bmatrix}
f_{2D-o/2}(g_1,g_2)(0,0) & \cdots & f_{2D-o/2}(g_1,g_2)(0,1) & \cdots & f_{2D-o/2}(g_1,g_2)(N/2-1,N/2-1) \\
f_{2D-o/2}(g_1,g_2)(1,0) & \cdots & f_{2D-o/2}(g_1,g_2)(1,1) & \cdots & f_{2D-o/2}(g_1,g_2)(1,N/2-1) \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
f_{2D-o/2}(g_1,g_2)(N/2-1,0) & \cdots & f_{2D-o/2}(g_1,g_2)(N/2-1,1) & \cdots & f_{2D-o/2}(g_1,g_2)(N/2-1,N/2-1)
\end{bmatrix}
\]

Note that it would be better to select parameters \(N\) and \(p\) so that \(N(o/2)\) and \(p(o/2)\) are integers and \(N\) is near 8.

4. Compound the subblocks of step 3, \(\hat{f}_{2D-o/2}\), to get \(f_{2D-o/2}\), which is the \(o/2\)-multiple enlargement of \(f_{2D}\) where \(f_{2D-o/2} \in R^{(o/2)I \times (o/2)J}\).

5. Set
\[
f_{2D-o/2-min}(i,j) = \min(f_{2D}[i(j/2)],[j(j/2)],f_{2D}[i(j/2)]+1,j(j/2)+1), \quad \text{and} \quad f_{2D-o/2-max}(i,j) = \max(f_{2D}[i(j/2)],[j(j/2)],f_{2D}[i(j/2)]+1,j(j/2)+1),
\]
where \(i = 0,1,\ldots,[l/2(o/2)]-1, j = 0,1,\ldots,[l/2(o/2)]-1, \) and if \([l/2(o/2)]+1 [j/2(o/2)]+1\) is bigger than \(l/2-1\) or \(l/2-1\), \([l/2(o/2)]+1 [j/2(o/2)]+1\) should be set as \(l/2-1\) or \(l/2-1\). Modify points of \(f_{2D-o/2}\) to eliminate the ripple effect as

- \(f_{2D-o/2}(i,j) = f_{2D-o/2-min}(i,j)\) if \(f_{2D-o/2}(i,j) < f_{2D-o/2-min}(i,j)\),
- \(f_{2D-o/2}(i,j) = f_{2D-o/2-max}(i,j)\) if \(f_{2D-o/2}(i,j) > f_{2D-o/2-max}(i,j)\),
- \(f_{2D-o/2}(i,j)\) remains invariant, otherwise,

where \(i = 0,1,\cdots,[l/2(o/2)]-1\) and \(j = 0,1,\cdots,[l/2(o/2)]-1\).

4 Simulation Results of Fast 2-D/New 3-D DCT for Enlargement of Image/Video

4.1 Example 1
Consider the original image with 48 \(\times\) 48 pixels shown in Fig. 3. It is desired to have the 16-multiple enlargement of it. To reduce the length of the paper, we show just the left-half side of the 16-multiple enlarged image, in a 768 \(\times\) 384-pixel size, by various methods.

4.1.1 Case i
Divide the original image into 36 8 \(\times\) 8 subblocks that have no overlapped pixels, and get the enlargement of the original image shown in Fig. 4 with 768 \(\times\) 768 pixels by Eqs. (41) and (43) directly. Some blocky and ripple effects appear in Fig. 4 obviously.

4.1.2 Case ii
In order to eliminate the blocky and ripple effects, apply Algorithm 3 with \(p=4\) and \(N=8\) to have the enlarged image shown in Fig. 5.

4.1.3 Case iii
In order to reduce the operating time, we use the fast mechanism addressed in Sec. 3.2 to have the one shown in Fig. 6 with \(p=2\) and \(N=8\). The methods used for Figs. 5 and 6 spend 12.4220 and 0.6710 s for the full-size image 768 \(\times\) 768 pixels, respectively. All following experiments through this paper were performed on a machine equipped...
with Intel Pentium IV 3.2-GHz CPU and 512MB of RAM with Windows XP Professional and Matlab as the runtime environment.

4.1.4 Case iv
The enlarged image is realized by the bilinear method.\textsuperscript{24}

4.1.5 Case v
The enlarged image is implemented by the cubic method.\textsuperscript{25}

4.1.6 Case vi
The enlarged image in Fig. 7 is achieved by the parametric cubic method.\textsuperscript{23}

4.1.7 Case vii
The enlarged image in Fig. 8 is realized by the Haar wavelet transform method.\textsuperscript{22} Operating times of various enlargement methods are shown in Table 1. From Figs. 2–8, we can see the proposed mechanisms (Figs. 4 and 5) maintain the originally symmetric appearances; however, other methods (parametric cubic method and wavelet method)

\begin{table}[h]
\centering
\begin{tabular}{|l|c|}
\hline
Method & Operating Time (s) \\
\hline
Proposed Algorithm 3 for $p=4$ & 12.4220 \\
Proposed fast Algorithm 3 for $p=2$ & 0.6710 \\
Bilinear method & 0.3280 \\
Cubic convolution & 2.5310 \\
Parametric cubic convolution & 42.1870 \\
Wavelet method & 0.1100 \\
\hline
\end{tabular}
\caption{Comparison of operating times of various enlargement methods.}
\end{table}

Fig. 5 Enlarged image (shown in half size) with blocky and ripple effect cancellations by proposed Algorithm 3—maintaining the symmetric appearance.

Fig. 6 Enlarged image (shown in half size) with blocky and ripple effect cancellations by proposed fast Algorithm 3—maintains the symmetric appearance.

Fig. 7 Enlarged image (shown in half size) by parameter cubic convolution—losing the symmetric appearance.

Fig. 8 Enlarged image (shown in half size) by wavelet method—losing the symmetric appearance.

Fig. 9 Enlarged image (shown in half size) with blocky and ripple effect cancellations by proposed fast Algorithm 3—maintains the symmetric appearance.
result in significantly asymmetric appearances. This situation becomes obvious for a very large enlargement. Unfortunately, the symmetric property is required for practical use in many cases.

4.2 Example 2

Consider an original video sequence that has 8 frames and show specially two frames of this video sequence, the first and second frames in Figs. 9(a) and 9(b), respectively. It is desired to interpolate four frames into two side-by-side frames and each frame has the original size. The first to sixth frames of the interpolated video sequence are shown in Figs. 10(a)–10(f), respectively. The interpolated video sequence can be used for demonstrating the effect of the slow motion.

5 Conclusion

A novel matrix form of the 3-D DCT pair was proposed. The widely used fast 2-D DCT pair is just a special case of the newly proposed algorithm. The computationally efficient 3-D DCT pair significantly avoids redundant computations and shortens the processing time than the computation of 3-D DCT pair through its definition. Due to its systematic structure, the proposed 3-D DCT pair is suitable to be applied for a wide class applications. The mechanisms for enlargement of image/video with blocky and ripple effect cancellations are proposed based on the newly presented DCT pair. The proposed ripple effect cancellation mechanism can also improve the quality of enlargement of an image via other high-order mechanisms. Due to the essentially even-function property of the DCT, the proposed enlargement of a sequence/image/video demonstrates the desired symmetric property. Applications for the enlargement/reduction of a video is implemented by the computationally efficiency 3-D DCT pair, where non-square size transformation matrices $C$ of dimension $n_l \times n_l$ are utilized for the desired goal, where $n_l$ and $n_l$ may not be powers of 2.

Acknowledgments

This work was supported by the National Science Council of Republic of China under contracts NSC-95-2211-E-006-109 and NSC-95-2211-E-006-362.

References


Shu-Mei Guo received her MS degree from the Department of Computer and Information Science, New Jersey Institute of Technology, in 1987 and her PhD degree in computer and systems engineering from University of Houston, Texas, in May 2000. Beginning is June 2000 she was an assistant professor with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, where she has been an associate professor since August 2005. Her research interests include various applications on evolutionary programming, chaos systems, Kalman filtering, fuzzy methodology, sampled-data systems, image processing, and computer and systems engineering.

Chen-Bang Li received his MS degree from the Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, in 2004. His research interests include various applications of image processing.

Chia-Wei Chen received his BS degree in electrical engineering from the Chung-Yuan University and his MS and PhD degrees in electrical engineering from the Cheng-Kung University. He is currently an associate professor with the Department of Mechanical and Automation Engineering of Kao Yuan University, Lujhu Township, Kaohsiung, Taiwan. His research interests include partial differential system control and optimal control.

Yueh-Ching Liao received her MS degree from the Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology, Taiwan, in 2003 and she is currently a PhD student with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. Her research interests include the computer-aided diagnosis of breast lesions, applications of image processing, pattern recognition, and wavelet-based features.