Enlargement and reduction of image/video via discrete cosine transform pair, part 2: reduction

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. Based on the novel matrix representations of the forward three-dimensional (3-D) discrete cosine transform (DCT) and the inverse 3-D DCT, applications for reduction of sequence/image/video are proposed. The original sequence/image/video is divided into subunits, and then the reduction is implemented in the frequency-domain by DCT. Two efficient algorithms are presented in accordance with two kinds of reducing multiples, 1/s and o_1/o_2, respectively. The proposed algorithms significantly eliminate ripple and blocky effects. As a result, they can provide numerous applications in data compression as well. Simulation results show that proposed mechanisms enable good performances for reductions of sequences, images, and videos as expected. © 2007 SPIE and IS&T.
used here. Here, we provide the following short summary to facilitate all derivations in this manuscript.

Recall the 1-D DCT pair in a matrix form as follows:

\[ \mathbf{F}_{1D} = \mathbf{C}_N \mathbf{F}_{1D} \]  
(1)

and

\[ \mathbf{C}_N = \left( \frac{2}{N} \right)^{1/2} \begin{bmatrix} 
C(0) \cos \frac{(2 \cdot 0 + 1)0\pi}{2N} & C(0) \cos \frac{(2 \cdot 1 + 1)0\pi}{2N} & \cdots & C(0) \cos \frac{(2(N-1) + 1)0\pi}{2N} \\
C(1) \cos \frac{(2 \cdot 0 + 1)1\pi}{2N} & C(1) \cos \frac{(2 \cdot 1 + 1)1\pi}{2N} & \cdots & C(1) \cos \frac{(2(N-1) + 1)1\pi}{2N} \\
\vdots & \vdots & \ddots & \vdots \\
C(N-1) \cos \frac{(2 \cdot 0 + 1)(N-1)\pi}{2N} & C(N-1) \cos \frac{(2 \cdot 1 + 1)(N-1)\pi}{2N} & \cdots & C(N-1) \cos \frac{(2(N-1) + 1)(N-1)\pi}{2N} 
\end{bmatrix} \]  
(3)

and \( \mathbf{C}_N \in \mathbb{R}^{N \times N} \), \( \mathbf{f}_{1D} = [f(0) f(1) \cdots f(N-1)]^T \in \mathbb{R}^{N \times 1} \), \( \mathbf{F}_{1D} = [F(0) F(1) \cdots F(N-1)]^T \in \mathbb{R}^{N \times 1} \). If one wants to enlarge \( \mathbf{f}_{1D} \) to a size as \( rN \times 1 \), the inverse 1-D DCT should be modified as

\[ \mathbf{f}_{1D-r} = \mathbf{C}_{N,r}^T \mathbf{F}_{1D} \]  
(4)

where \( \mathbf{f}_{1D} \in \mathbb{R}^{N \times 1} \), \( \mathbf{f}_{1D-r} \in \mathbb{R}^{rN \times 1} \), \( \mathbf{C}_{N,r} \in \mathbb{R}^{rN \times N} \), and

\[ \mathbf{C}_{N,r} = \left( \frac{2}{N} \right)^{1/2} \begin{bmatrix} 
C(0) \cos \frac{(2 \cdot 0 + r)0\pi}{2Nr} & C(0) \cos \frac{(2 \cdot 1 + r)0\pi}{2Nr} & \cdots & C(0) \cos \frac{(2(N-1) + r)0\pi}{2Nr} \\
C(1) \cos \frac{(2 \cdot 0 + r)1\pi}{2Nr} & C(1) \cos \frac{(2 \cdot 1 + r)1\pi}{2Nr} & \cdots & C(1) \cos \frac{(2(N-1) + r)1\pi}{2Nr} \\
\vdots & \vdots & \ddots & \vdots \\
C(N-1) \cos \frac{(2 \cdot 0 + r)(N-1)\pi}{2Nr} & C(N-1) \cos \frac{(2 \cdot 1 + r)(N-1)\pi}{2Nr} & \cdots & C(N-1) \cos \frac{(2(N-1) + r)(N-1)\pi}{2Nr} 
\end{bmatrix} \]  
(5)

For the 2-D case, the fast 2-D DCT pair is

\[ \mathbf{F}_{2D} = \mathbf{C}_{n_1} \mathbf{f}_{2D} \mathbf{C}_{n_2}^T \]  
(6)

and

\[ \mathbf{f}_{2D} = \mathbf{C}_{n_1}^T \mathbf{F}_{2D} \mathbf{C}_{n_2} \]  
(7)

where \( \mathbf{f}_{2D} \in \mathbb{R}^{m_1 \times m_2} \) and \( \mathbf{F}_{2D} \in \mathbb{R}^{m_1 \times m_2} \). Based on the same idea of enlargement of a 1-D sequence as addressed in Eq. (4), take the upsampling of the 2-D basis and rewrite Eq. (7) as

\[ \mathbf{f}_{2D-r} = \mathbf{C}_{n_1,r} \mathbf{F}_{2D} \mathbf{C}_{n_2,r}^T \]  
(8)

to reconstruct the \( r \)-multiple 2-D image denoted by \( \mathbf{f}_{2D-r} \), where \( \mathbf{C}_{n_1,r} \in \mathbb{R}^{m_1 \times rN} \), \( \mathbf{C}_{n_2,r} \in \mathbb{R}^{m_2 \times rN} \), \( \mathbf{f}_{2D-r} \in \mathbb{R}^{m_1 \times rNes} \), and \( \mathbf{F}_{2D} \in \mathbb{R}^{m_1 \times m_2} \).

2 Reduction of Sequence/Image/Video Via 1-D/2-D/3-D DCT

Two essential conditions for reduction of a sequence/image/video are (1) to have almost same appearances in a reduced size and (2) to have a small recover error between the recovered sequence/image/video after reduction/enlargement and the original one as possible, so that some applications can be explored, such as the sequence/image/video compression.

2.1 Reduction of a Sequence Via 1-D DCT

To reduce an original sequence \( \mathbf{f}_{1D} \) for \( i=0,1,\ldots,N-1 \) to a desired sequence \( \mathbf{f}_{1D-r} \) for \( i=0,1,\ldots,rN-1 \), where \( r=1/s \), it is preferable to reduce the original sequence to a reduced sequence one half the size at a time, and then repeat it again and again until achieving the desired size, such as \( r=1/8=1/2[1/2(1/2)] \).

To reduce the original sequence \( \mathbf{f}_{1D} \in \mathbb{R}^{N \times 1} \) to the desired sequence \( \mathbf{f}_{1D/2} \in \mathbb{R}^{N/2 \times 1} \), perform the 1-D DCT of Eq. (1) to acquire the frequency coefficient matrix \( \mathbf{F}_{1D} \) first. Next, divide \( \mathbf{F}_{1D} \) into two (\( s \times s \)) parts, i.e., the low-frequency coefficient matrix as \( \mathbf{F}_{1D/2} = [\mathbf{F}_{1D}(0) \cdots \mathbf{F}_{1D}(N/2-1)]^T \) and the high-frequency matrix as \( \mathbf{F}_{1D/2} = [\mathbf{F}_{1D}(N/2) \cdots \mathbf{F}_{1D}(N-1)]^T \), where \( \mathbf{F}_{1D/2} \)
\[ f_{1D}^{-1}(p) = C_{N,1}(p) f_{1D}(p), \] (9)

where

\[ C_{N,1}(p) = \left\{ \frac{1}{N} \right. \right. \left\{ \begin{array}{l}
C(0) \cos \left( \frac{2 \cdot 0 + 1/s}{2(N/s)} \pi \right) \\
C(1) \cos \left( \frac{2 \cdot 1 + 1/s}{2(N/s)} \pi \right) \\
\vdots \\
C(N/s - 1) \cos \left( \frac{2 \cdot (N/s - 1)}{2(N/s)} \pi \right) \\
\left. \end{array} \right. \}
\]

\[ C_{N,1}(2p) = \left\{ \frac{1}{N} \right. \right. \left\{ \begin{array}{l}
C(0) \cos \left( \frac{2 \cdot 0 + 1/(2s)}{2(N/s)} \pi \right) \\
C\left( \frac{N}{s} \right) \cos \left( \frac{2 \cdot 1/(2s) + 1/(2s)}{2(N/s)} \pi \right) \\
\vdots \\
C\left( \frac{N}{s} - 1 \right) \cos \left( \frac{2 \cdot (N/s - 1)/(2s)}{2(N/s)} \pi \right) \\
\left. \end{array} \right. \}
\] (10)

\[ f_{1D}^{-1}(i) \in R^{N \times 1}, \quad C_{N,1}(p) \in R^{N \times N/s}, \quad F_{1D}(p) \in R^{N \times 1}, \quad r = 1/s = 1/2, \quad s = 2, \quad p = 1, 2. \]

Next, to maintain the original characteristic in a N/2 × 1 size, take two reduced sequences as

\[ f_{1D}^{-1}(i) = 2f_{1D}^{-1}(i) \text{ and } f_{1D}^{-1}(i) = 2f_{1D}^{-1}(i), \]

where \( s = 2, \) and \( f_{1D}^{-1}(i) \in R^{N \times 1} \text{ and } f_{1D}^{-1}(i) \in R^{N \times 1}. \) Then, determine the local maximal value \( f_{1D,\text{max}(i)} \) and the local minimal value \( f_{1D,\text{min}(i)} \), where

\[ f_{1D,\text{max}(i)} = \max[f_{1D}(2i), f_{1D}(2i+1)] \quad \text{and} \quad f_{1D,\text{min}(i)} = \min[f_{1D}(2i), f_{1D}(2i+1)] \quad \text{for } i = 0, 1, \ldots, N/2 - 1 \text{ from the original sequence } f_{1D}. \]

It is reasonable to assume the value of the reduced sequence \( f_{1D}^{-1}(i) \) is restricted to the range \([f_{1D,\text{min}(i)}, f_{1D,\text{max}(i)}]. \) So, we modify the reduced sequence as

\[ f_{1D}^{-1}(i) = f_{1D,\text{max}(i)} \quad \text{if } f_{1D}^{-1}(i) > f_{1D,\text{max}(i)}, \] (12)

\[ f_{1D}^{-1}(i) = f_{1D,\text{min}(i)} \quad \text{if } f_{1D}^{-1}(i) < f_{1D,\text{min}(i)}, \] (13)

\[ f_{1D}^{-1}(i) \text{ remains invariant, otherwise,} \] (14)

where \( i = 0, 1, \ldots, N/2 - 1, \) \( r = 1/2, \) and \( p = 1, 2. \) Then, have the recovered \( f_{1D,p} \) in the original size of \( N \times 1 \) from the modified \( f_{1D}^{-1}(i) \) by Eqs. (1) and (4) with \( r = 2, \) where \( p = 1, 2, f_{1D,p} \in R^{N \times 1} \text{ and } f_{1D^{-1}/2} \in R^{N \times 2}. \)

Finally, compare \( f_{1D,p}(i) \) with \( f_{1D}(i) \) only at \( i = 0, 1, \ldots, N-2 \) to avoid the ripple effect and take the corresponding \( f_{1D}^{-1}(i/p) \) with a smaller error (mean square error, MSE) between \( f_{1D,p}(i) \) and \( f_{1D}(i) \) as the desired reduced sequence.

If the size of the sequence cannot be presented by \( r = 1/2 \lfloor 1/2 \rfloor \rfloor \), other reduced factors \( r \) are necessary, i.e., \( r = 1/3, 1/5, \ldots \). Then, the just-mentioned mechanism should be modified appropriately.

### 2.2 Reduction of Image Via 2-D DCT

To extend the 1-D case for a sequence, as mentioned in Sec. 2.1 to the 2-D case for an image, the reduction algorithm of an image is addressed here. To reduce an original image \( f_{2D}(i,j) \) for \( i,j = 0,1,\ldots,N-1 \) to a desired image \( f_{2D}^{-1}(i,j) \) for \( i,j = 0,1,\ldots,N-1, \) it is preferable to reduce the original image to a reduced image at one half the size at a time for \( r = 1/s = 1/2, \) then repeat it again and again until achieving the desired size.

If one wants to reduce the original image \( f_{2D} \in R^{N \times N} \) to the desired image \( f_{2D} \in R^{N \times N} \) where \( r = 1/s = 1/2, \) perform the 2-D DCT of Eq. (6) to have the frequency-domain coefficient \( F_{2D} \) first. Next, divide \( F_{2D} \) into four \((s^2)\) parts as
\[ F_{2D} = \begin{bmatrix} F_{2D(1,1)} & F_{2D(1,2)} \\ F_{2D(2,1)} & F_{2D(2,2)} \end{bmatrix}, \]

where \( F_{2D(1,1)}, F_{2D(1,2)}, F_{2D(2,1)}, F_{2D(2,2)} \in \mathbb{R}^{N/2 \times N/2} \) and \( F_{2D(1,1)} \) is a low-horizontal low-vertical frequency coefficient matrix, \( F_{2D(1,2)} \) is a high-horizontal low-vertical frequency coefficient matrix, \( F_{2D(2,1)} \) is a low-horizontal high-vertical frequency coefficient matrix, \( F_{2D(2,2)} \) is a high-horizontal high-vertical frequency coefficient matrix. Then, take four respectively reduced spatial-domain images with horizontal high-vertical frequency coefficient matrix. Then, vertical frequency coefficient matrix, \( N_{2D} \), restore some high-frequency property in an 2D image. When an image is reduced, its high-frequency property \( f_{2D} \) is shown in Eq. (10), and some other high-frequency property reduced images \( \tilde{f}_{2D(1,1)}, \tilde{f}_{2D(1,2)}, \tilde{f}_{2D(2,1)}, \tilde{f}_{2D(2,2)} \) as \( f_{2D} = \tilde{f}_{2D(1,1)} + \tilde{f}_{2D(1,2)} + \tilde{f}_{2D(2,1)} + \tilde{f}_{2D(2,2)} \), where \( s = 2 \) and \( p_{1}, p_{2} = 1, 2 \), \( s = 2 \), \( C_{N,1}(p_{1}, p_{2}) \) is shown in Eq. (11), and \( C_{N,1}(p_{1}, p_{2}) \) is shown in Eq. (11).

When an image is reduced, its high-frequency property disappears. To maintain the original characteristic and try to restore some high-frequency property in an \( N/2 \times N/2 \) size, we take five reduced images based on the low-frequency-property reduced image \( \tilde{f}_{2D(1,1)} \) and some other high-frequency property reduced images \( \tilde{f}_{2D(1,1),p_{1},p_{2}}, \tilde{f}_{2D(1,2),p_{1},p_{2}}, \tilde{f}_{2D(2,1),p_{1},p_{2}}, \tilde{f}_{2D(2,2),p_{1},p_{2}} \) as \( f_{2D} = \tilde{f}_{2D(1,1)} + \tilde{f}_{2D(1,2)} + \tilde{f}_{2D(2,1)} + \tilde{f}_{2D(2,2)} \) and \( f_{2D} = \tilde{f}_{2D(1,1),p_{1},p_{2}} + \tilde{f}_{2D(1,2),p_{1},p_{2}} + \tilde{f}_{2D(2,1),p_{1},p_{2}} + \tilde{f}_{2D(2,2),p_{1},p_{2}} \) where \( s = 2 \) and \( p_{1}, p_{2} = 1, 2 \), \( i, j = 0, 1, \ldots, N/2 \). Then, determine the local maximal value \( f_{2D,\max(i,j)} \) and the local minimal value \( f_{2D,\min(i,j)} \) where \( f_{2D,\max(i,j)} = \max(\tilde{f}_{2D(1,2),p_{1},p_{2}}, \tilde{f}_{2D(2,1),p_{1},p_{2}}, \tilde{f}_{2D(2,2),p_{1},p_{2}}) \), and \( f_{2D,\min(i,j)} = \min(\tilde{f}_{2D(1,2),p_{1},p_{2}}, \tilde{f}_{2D(2,1),p_{1},p_{2}}, \tilde{f}_{2D(2,2),p_{1},p_{2}}) \) for \( i = 0, 1, \ldots, N/2-1 \) and \( j = 0, 1, \ldots, N/2-1 \), from the original image. It is reasonable to assume the pixel value of the reduced image \( \tilde{f}_{2D(1,2),p_{1},p_{2}}(i,j) \) is restricted to the range \( [f_{2D,\min(i,j)}, f_{2D,\max(i,j)}] \). So, modify the reduced image by

\[ f_{2D(1,2),p_{1},p_{2}}(i,j) = f_{2D,\max(i,j)} \quad \text{if} \quad f_{2D(1,2),p_{1},p_{2}}(i,j) > f_{2D,\max(i,j)}, \]

\[ f_{2D(1,2),p_{1},p_{2}}(i,j) = f_{2D,\min(i,j)} \quad \text{if} \quad f_{2D(1,2),p_{1},p_{2}}(i,j) < f_{2D,\min(i,j)}, \]

\[ f_{2D(1,2),p_{1},p_{2}}(i,j) \quad \text{remains invariant, otherwise}, \]

where \( i = 0, 1, \ldots, N/2-1 \), \( j = 0, 1, \ldots, N/2-1 \), and \( p_{1}, p_{2} = 1, 2, \ldots, S \). Next, compute the recovered \( \tilde{f}_{2D,p} \) in the original size from the modified \( f_{2D(1,2),p_{1},p_{2}} \) by Eqs. (6) and (8) with \( r = 2 \), \( s = 2 \), \( \tilde{f}_{2D,p} = f_{2D(1,2),p_{1},p_{2}} \) in \( \mathbb{R}^{N \times N} \) and \( f_{2D(1,2),p_{1},p_{2}} \) in \( \mathbb{R}^{N/2 \times N/2} \). To confirm the best one of \( \tilde{f}_{2D,p} \) that retains

Digital Image Processing

Three Dimensional Enlargement and Reduction of Image by Fast 3-D DCT

!!!@@@#### $$$$%$

%&%%% **** ~ ~ ++

?? {{ } } [ ] :: :: ...

Fig. 3 "Text."
most original characteristics, compare the recovered image \( \mathbf{f}_{2D}(i,j) \) with the original one \( \mathbf{f}_{2D}(i,j) \) for \( p=1,2,\ldots,5 \). To avoid the edge-ripple effect, compare the region in \( i,j=0,1,\ldots,N-2 \). Then, take the corresponding \( \mathbf{f}_{2D-1/2,p} \) with a smaller error (MSE) between \( \mathbf{f}_{2D,p}(i,j) \) and \( \mathbf{f}_{2D}(i,j) \) as the desired reduced image.

To acquire the \( r \)-multiple \( (r=1/s) \) reduction of a large image, a novel algorithm is presented as follows:

\[
\mathbf{f}_{2D}(g_1N,g_2N) = \begin{pmatrix} \mathbf{f}_{2D}(g_1N,g_2N+1) & \cdots & \mathbf{f}_{2D}(g_1N,(g_2+1)N-1) \\ \mathbf{f}_{2D}(g_1N+1,g_2N) & \cdots & \mathbf{f}_{2D}(g_1N+1,(g_2+1)N-1) \\ \vdots & \ddots & \vdots \\ \mathbf{f}_{2D}(g_1(N-1),g_2N) & \cdots & \mathbf{f}_{2D}(g_1(N-1),(g_2+1)N-1) \end{pmatrix}
\]

where \( g_1=0,1,\ldots,(l_1/N-1) \) and \( g_2=0,1,\ldots,(l_2/N-1) \).

1. Subdivide the original image \( \mathbf{f}_{2D}(i_1,j_1) \in \mathbb{R}^{l_1 \times l_2} \) into nonoverlapped subimages \( \mathbf{f}_{2D}(g_1N,g_2N) \in \mathbb{R}^{N \times N} \) by

\[
\mathbf{f}_{2D-1/2}(g_1N,g_2N) = \begin{pmatrix} \mathbf{f}_{2D-1/2}(g_1N,g_2N) & \cdots & \mathbf{f}_{2D-1/2}(g_1N,(g_2+1)N-1) \\ \vdots & \ddots & \vdots \\ \mathbf{f}_{2D-1/2}(g_1(N-1),g_2N) & \cdots & \mathbf{f}_{2D-1/2}(g_1(N-1),(g_2+1)N-1) \end{pmatrix}
\]

6. Modify the respectively reduced subimage \( \mathbf{f}_{2D-1/2}(g_1N,g_2N) \) of each divided subblock \( \mathbf{f}_{2D}(g_1N,g_2N) \) by Eqs. (17), (18), and (19), with \( p=1,2,\ldots,5 \), \( g_1=0,1,\ldots,(l_1/N-1) \), and \( g_2=0,1,\ldots,(l_2/N-1) \).

3. Divide the frequency-domain coefficients of every subblock \( \mathbf{f}_{2D}(g_1N,g_2N) \) into four \((s^2)\) parts as \( \mathbf{f}_{2D}(g_1N,g_2N)^{(1)} \), \( \mathbf{f}_{2D}(g_1N,g_2N)^{(1,2)} \), \( \mathbf{f}_{2D}(g_1N,g_2N)^{(2,1)} \), and \( \mathbf{f}_{2D}(g_1N,g_2N)^{(3)} \) by Eq. (15), where \( g_1=0,1,\ldots,(l_1/N-1) \) and \( g_2=0,1,\ldots,(l_2/N-1) \).

4. Take the reduced spatial-domain subimages \( \mathbf{f}_{2D}(g_1N,g_2N)^{(p)} \) with \( p=1,2,3 \) by Eqs. (6) and (8) with \( r=2 \), where \( p=1,2,\ldots,5 \), \( g_1=0,1,\ldots,(l_1/N-1) \), and \( g_2=0,1,\ldots,(l_2/N-1) \).

5. Take the reduced subimages \( \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(p)} \) by

\[
\begin{align*}
\mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(1)} &= \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(1)} \\
\mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(2)} &= \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(1)} + \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(1,2)} \\
\mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(3)} &= \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(1)} + \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(2,1)} \\
\mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(4)} &= \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(1)} + \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(1,2)}
\end{align*}
\]

7. Recover the subimage \( \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(p)} \) in the original subblock of size \( N \times N \) from modified reduced images \( \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(p)} \) by Eqs. (6) and (8) with \( r=2 \), where \( r=1,2,\ldots,5 \), \( g_1=0,1,\ldots,(l_1/N-1) \), and \( g_2=0,1,\ldots,(l_2/N-1) \).

8. Compare the recovered subimage \( \mathbf{f}_{2D}(g_1N,g_2N)^{(p)} \) with the original subimage \( \mathbf{f}_{2D}(g_1N,g_2N)^{(p)} \) only at \( i=0,1,\ldots,N-2 \) and \( j=0,1,\ldots,N-2 \).

9. Combine desired local subimages \( \mathbf{f}_{2D-1/2}(g_1N,g_2N)^{(p)} \) to form a new image. If the size of the new image is as

![Fig. 4 Eight 288 × 352-pixel frames from a video sequence.](image-url)
2.3 Reduction of Video Via 3-D DCT

Based on the same idea of reduction of a 2-D image, the desired reduction algorithm with blocky and ripple effect cancellations for a video \( f_{3D} \) in a size of \( l_1 \times l_2 \times l_3 \) can also be derived. To shorten the paper length, the detailed procedures are omitted here.

Even though various combinations of subparts of the frequency-domain coefficient matrix \( F_{3D} \) associated with some necessary processes can yield a better performance of reduction for a video, it will cost a lot of CPU time. So we suggest using the low-frequency case at this point.

2.4 Arbitrary-Ratio Reduction of Image Via 2-D DCT

Similarly, it is preferable for new reduction algorithms with ripple and blocky effect cancellations, as proposed in Secs. 2.1, 2.2, and 2.3 for 1-D, 2-D, and 3-D cases, to have the multiple \( (r=1/s) \) as \((1/2)^n\), where \( n \) is an integer. To handle other cases, a novel algorithm is presented in the following. However, if performing an \((1/2)^n\) multiple enlargement is necessary, so we suggest applying the proposed algorithms in Secs. 2.1, 2.2, and 2.3 due to the consideration of complexity. To shorten the length of the paper, we address only the non-\((1/2)^n\)-multiple reduction for a 2-D case in this paper. The 3-D case can also be extended from the 2-D case.

Consider an original image \( f_{2D} \) with a size of \( n_1 \times n_2 \). It is desired to have the \( o_1/o_2\)-multiple reduced image \( f_{2D-o_1/o_2} \) in a size \([n_1/o_1] \times [n_2/o_2]\), where \( o_1/o_2 \) cannot be expressed in the form \((1/2)^n\), \( o_1/o_2 \ll 1 \), and \( n_1 \), \( o_1 \), and \( o_2 \) are integers. First, compute \( F_{2D} \), the 2-D DCT of \( f_{2D} \), using Eq. (6), where \( F_{2D} \in \mathbb{R}^{n_1 \times n_2} \). Then, take the low-frequency part of \( F_{2D} \), \( F_{2D-o_1/o_2} \), as

\[
F_{2D-o_1/o_2} = \begin{bmatrix}
F_{2D}(0,0) & F_{2D}(0,1) & \cdots & F_{2D}(0, \lfloor n_2/o_2 \rfloor - 1) \\
F_{2D}(1,0) & F_{2D}(1,1) & \cdots & F_{2D}(1, \lfloor n_2/o_2 \rfloor - 1) \\
\vdots & \vdots & \ddots & \vdots \\
F_{2D}(\lfloor n_1/o_1 \rfloor - 1,0) & F_{2D}(\lfloor n_1/o_1 \rfloor - 1,1) & \cdots & F_{2D}(\lfloor n_1/o_1 \rfloor - 1, \lfloor n_2/o_2 \rfloor - 1)
\end{bmatrix},
\]

(20)

where \( F_{2D-o_1/o_2} \in \mathbb{R}^{\lfloor n_1/o_1 \rfloor \times \lfloor n_2/o_2 \rfloor} \). The reduced 2-D image \( f_{2D-o_1/o_2} \) can be constructed by

\[
f_{2D-o_1/o_2} = C_{\lfloor n_1/o_1 \rfloor}^T F_{2D-o_1/o_2} C_{\lfloor n_2/o_2 \rfloor}.
\]

(21)

The detailed non-\((1/2)^n\)-multiple reduction algorithm of a large 2-D image with blocky and ripple effect cancellations is then given as follows.

2.4.1 Algorithm 2

Consider the original image \( f_{2D} \) with a size of \( l_1 \times l_2 \), where pixel elements of \( f_{2D} \) are expressed as \( f_{2D}(i,j) \) for \( 0 \leq i \leq l_1 - 1 \) and \( 0 \leq j \leq l_2 - 1 \). It is desirable to get the \( o_1/o_2\)-multiple reduced image \( f_{2D-o_1/o_2} \) in a size \([l_1/(o_1/o_2)] \times [l_2/(o_1/o_2)]\), where \( o_1/o_2 \ll 1 \).

1. Subdivide \( f_{2D} \) into nonoverlapped subblocks \( f_{2D}(g_1,g_2) \) with a size of \( N \times N \). The relationship between \( f_{2D} \) and \( f_{2D}(g_1,g_2) \) is addressed in step 1 of Algorithm 1.

2. Reduce \( f_{2D}(g_1,g_2) \) by performing Eqs. (6), (20), and (21) for \( n_1=N \) and \( n_2=N \); consequently, also obtain the low frequency-domain coefficient matrix \( F_{2D-o_1/o_2}(g_1,g_2) \) of subblock \( f_{2D}(g_1,g_2) \) and the \( o_1/o_2\)-multiple image \( F_{2D-o_1/o_2}(g_1,g_2) \), respectively, where \( F_{2D-o_1/o_2}(g_1,g_2) = C_N^T F_{2D-o_1/o_2} C_N \).
Guo et al.: Enlargement and reduction of image/video via discrete cosine transform pair

\[
\mathbf{F}_{2D} = \begin{bmatrix}
\mathbf{F}_{2D}(0,0) & \mathbf{F}_{2D}(0,1) & \cdots & \mathbf{F}_{2D}(0,\frac{N_0}{2}-1) \\
\mathbf{F}_{2D}(1,0) & \mathbf{F}_{2D}(1,1) & \cdots & \mathbf{F}_{2D}(1,\frac{N_0}{2}-1) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{F}_{2D}(\frac{N_0}{2}, 0) & \mathbf{F}_{2D}(\frac{N_0}{2}, 1) & \cdots & \mathbf{F}_{2D}(\frac{N_0}{2}, \frac{N_0}{2}-1)
\end{bmatrix},
\]

\[
f_{2D-\alpha_2}(x, y) = c_{\alpha_2}(x) \mathbf{F}_{2D}(x, y) c_{\alpha_2}(y) \mathbf{F}_{2D}^T(\mathbf{F}_{2D}(x, y) c_{\alpha_2}(y)) = \mathbf{F}_{2D}(x, y) c_{\alpha_2}(y) \mathbf{F}_{2D}^T(\mathbf{F}_{2D}(x, y) c_{\alpha_2}(y)).
\]

3. Combine the desired reduced subimages \( f_{2D-\alpha_2}(x, y) \) to achieve a complete reduced image \( f_{2D-\alpha_2}(x, y) \).

4. Set

\[
f_{2D-\alpha_2}(i, j) = \max \left[ f_{2D}(i, j), f_{2D}(i, j+1), f_{2D}(i, j-1), f_{2D}(i, j-1), f_{2D}(i, j-1), f_{2D}(i, j-1) \right],
\]

where \( i = 0, 1, \ldots, [l_1/\alpha_2] - 1 \), \( j = 0, 1, \ldots, [l_2/\alpha_2] - 1 \), and if \( [j/\alpha_2] + 1 \) is larger than \( l_2 - 1 \) or \( l_2 - 1 \), \( [j/\alpha_2] + 1 \) should be set as \( l_2 - 1 \) (or \( l_2 - 1 \)). Modify the points of \( f_{2D-\alpha_2}(x, y) \) to eliminate the ripple effect as

Table 1 CPU time (peak-to-peak signal-to-noise ratio) (PSNR) of various enlargements based on reduced images.

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>Time (s)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Lena&quot;</td>
<td>Bilinear Method</td>
<td>1.3750</td>
<td>32.9014 dB</td>
</tr>
<tr>
<td></td>
<td>Cubic Convolution</td>
<td>4.7500</td>
<td>33.6376 dB</td>
</tr>
<tr>
<td></td>
<td>Parametric Cubic Convolution</td>
<td>22.3400 s</td>
<td>33.6534 dB</td>
</tr>
<tr>
<td></td>
<td>Fast Algorithm 3</td>
<td>1.5310 s</td>
<td>33.8636 dB</td>
</tr>
<tr>
<td>&quot;Lena&quot;</td>
<td>Time</td>
<td></td>
<td>1.3750 s</td>
</tr>
<tr>
<td>&quot;Algorithm 7&quot;</td>
<td>Time</td>
<td></td>
<td>1.3900 s</td>
</tr>
<tr>
<td>&quot;Peppers&quot;</td>
<td>Time</td>
<td></td>
<td>4.6560 s</td>
</tr>
<tr>
<td>&quot;Text&quot;</td>
<td>Time</td>
<td></td>
<td>22.3280 s</td>
</tr>
<tr>
<td></td>
<td>Fast Algorithm 3</td>
<td></td>
<td>1.4680 s</td>
</tr>
<tr>
<td>&quot;Lena&quot;</td>
<td>Time</td>
<td></td>
<td>4.0730 s</td>
</tr>
<tr>
<td>&quot;Algorithm 7&quot;</td>
<td>Time</td>
<td></td>
<td>1.4060 s</td>
</tr>
<tr>
<td>&quot;Peppers&quot;</td>
<td>Time</td>
<td></td>
<td>4.6560 s</td>
</tr>
<tr>
<td>&quot;Text&quot;</td>
<td>Time</td>
<td></td>
<td>22.2500 s</td>
</tr>
<tr>
<td></td>
<td>Fast Algorithm 3</td>
<td></td>
<td>1.4840 s</td>
</tr>
<tr>
<td>&quot;Lena&quot;</td>
<td>Time</td>
<td></td>
<td>4.7500 s</td>
</tr>
<tr>
<td>&quot;Algorithm 7&quot;</td>
<td>Time</td>
<td></td>
<td>1.4690 s</td>
</tr>
<tr>
<td>&quot;Peppers&quot;</td>
<td>Time</td>
<td></td>
<td>4.7340 s</td>
</tr>
<tr>
<td>&quot;Text&quot;</td>
<td>Time</td>
<td></td>
<td>22.1400 s</td>
</tr>
<tr>
<td></td>
<td>Fast Algorithm 3</td>
<td></td>
<td>1.5470 s</td>
</tr>
</tbody>
</table>
3 Simulation Results of 2-D/3-D DCT for Reduction of Image/Video

All the following experiments in this paper were performed on a machine equipped with an Intel Pentium IV 3.2-GHz CPU and 512 MB of RAM with Windows XP Professional and Matlab as the runtime environment. Computer simulations using real images were performed to evaluate the performance of proposed algorithms for reductions of 2-D and 3-D cases as follows.

3.1 Example 1

Consider three 512 \times 512-pixel images (“Lena,” “Peppers,” and “Text”) shown in Figs. 1–3. To shorten the length of the paper, we show these only in a small size, except for Fig. 3. First, we reduce the three original images to ones with a size of 256 \times 256 via Algorithm 1, and the LL Algorithm, which uses only the low-frequency-domain coefficient matrix \( F_{2D(1,1)} \). The reductions of Algorithm 1 and the LL Algorithm use 8,703 and 1,6720 s, respectively. Then, acquire the recovered images via the bilinear method, \(^9\) cubic convolution, \(^9\) parametric cubic convolution, \(^9\) Algorithm 3 in Ref. 6, and fast Algorithm 3 in Ref. 6. Simulation results are shown in Table 1.

3.2 Example 2

Consider an original video sequence with eight 288 \times 352-pixel frames and show the first to eighth frames in Figs. 4(a)–4(h), respectively. It is desired to reduce the number of frames of the original video sequence from eight to four and have each frame of the reduced video sequence in a 144 \times 176 size, as shown in Figs. 5(a)–5(d). Finally, eight 288 \times 352-pixel frames recovered from Fig. 5 are shown in Fig. 6. The PSNRs between each corresponding frames of the original video sequence [Figs. 4(a)–4(h)] and the recovered video sequence [Figs. 6(a)–6(h)] are shown in Table 2. It’s obvious the proposed reduction and the enlargement algorithms have a good performance as expected.

3.3 Example 3

Consider a 512 \times 512-pixel image (“Text”) from Fig. 3. First, apply Algorithm 2 to acquire an 8/10-multiple reduced image, as shown in Fig. 7. Then, we obtain the 10/8-multiple enlarged image of Fig. 7, by Algorithm 3 (in Ref. 6). The PSNR between the original image of Fig. 3 and the recovered image of Fig. 7 is 23.2269 dB. The result shows the proposed Algorithm 3 of Ref. 6 and Algorithm 2 have good performance, as expected.

4 Conclusions

Two efficient algorithms for the reduction of sequence/image/video were proposed for different types of reduced multiples, \( 1/s \) and \( o_1/o_2 \). The algorithms were implemented progressively to avoid losing their characteristics. Moreover, the pixel values of reduced sequence/image/video were restricted in a reasonable region to avoid the ripple and blocky effects. These algorithms were implemented in the frequency domain by DCT. Due to the essentially even-function property of the DCT, the proposed reduction of a sequence/image/video also demonstrates the desired symmetric property. This property is important when reducing a sequence/image/video.

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.

Table 2 PSNRs of the original frames (Fig. 4) and the recovered frames (Fig. 6).

<table>
<thead>
<tr>
<th>Frame indices</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (dB)</td>
<td>29.76</td>
<td>28.64</td>
<td>29.51</td>
<td>28.97</td>
<td>29.65</td>
<td>29.41</td>
<td>29.41</td>
<td>28.99</td>
</tr>
</tbody>
</table>

From Table 1, it is obvious that the reduced images performed by Algorithm 1 have a higher quality than those from the LL Algorithm, but Algorithm 1 spends more CPU time. Also, Algorithm 3 in Ref. 6 with \( p=2 \) has the best quality among all for the enlargements. These observations become obvious for very large size images, such as 1024 \times 1024 images.
Guo et al.: Enlargement and reduction of image/video via discrete cosine transform pair…

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 in 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 of image processing, pattern recognition, and wavelet-based features.

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 deg and PhD degrees in electrical engineering from the Chung-Kung University. He is currently an associate professor of Computer Science and Information Engineering, National Cheng-Kung University, Lujhu Township, Kaohsiung, Taiwan. His research interests include partial differential system control and optimal control.

Yueh-Ching Liao received her MS and PhD degrees in electrical engineering from University of Houston, Texas, in 1985 and 1988, respectively. In August 1988 he became an associate professor with the Department of Electrical Engineering, National Cheng-Kung University, Taiwan, where he has been a full professor since August 1992 and a distinguished professor since August 2002. His research interests include state-space self-tuning control, chaotic system control, partial differential system control, numerical analysis, and robotics. He was an editor for the Journal of the Chinese Institute of Electrical Engineering from 1997 to 2002, he has been the (executive) editor for Science Development, published by National Science Council, Taiwan, since April 2001, an associate editor for the International Journal of Systems Science since July 2003, and an associate editor for the Journal of the Franklin Institute since April 2007.

Journal of Electronic Imaging 043007-9 Oct–Dec 2007/Vol. 16(4)