Fast Image Resizing in Discrete Cosine Transform Domain with Spatial Relationship between DCT Block and its Sub-Blocks

Ee-Leng Tan¹, Woon-Seng Gan¹ and Meng-Tong Wong²
¹School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
²Texas Instruments Singapore, Singapore
¹{etanel, ewsgan}@ntu.edu.sg
²wmt8@ti.com

Abstract

This paper presents a computational scalable approach to resize images by factors of \( P/Q \times R/S \) in the discrete cosine transform (DCT) domain. Our resizing approach is based on the framework proposed by Mukherjee and Mitra which employs the spatial relationship between a DCT block and its sub-blocks. To simplify integration with block based coders such as JPEG and MPEG, \( N \times N \) DCT blocks are discussed throughout this work. Our observation of the spatial relationship between a DCT block and its sub-blocks resulted in reduction of the computational cost to resize images. Also, we have suggested using up-sampled DCT blocks smaller than \( N \times N \) to further reduce computational cost.

1. Introduction

Users now have easy access to high definition image or video content due to the emergence of mobile multimedia capable devices and availability of broadband networks. Usually, these devices have smaller displays and limited processing capabilities. Hence, an efficient method to reduce the spatial resolution of image and video contents before delivery to these devices is desired [1].

To be able to directly manipulate compressed image and video contents eliminate the need to decompress and compress such contents which are mandatory with spatial techniques. Moreover, images resized in the discrete cosine transform (DCT) domain [2]-[5] are visually better than those obtained with spatial domain techniques.

Resizing images by factors of \( P/Q \times R/S \) are discussed in [2] and [4]-[5]. Salazar and Tran [2] proposed an arbitrary resizing algorithm in the DCT domain which is a generalization of the approach proposed by Dugad and Ahuja [3]. However, authors in [2] did not suggest the optimal length for DCT and IDCT for high PSNR. Park et al. [4] recently proposed a technique using fast inverse and forward DCT of composite length. It is found that this approach has one of the lowest computational cost and producing visually good images. However, this approach requires various radix DCTs and may not yield efficient computation for all resizing factors. In [5], Mukherjee and Mitra have proposed to resize images with the spatial relationship of the DCT coefficients between a block and its sub-blocks [6]. Mukherjee and Mitra have avoided high computational cost by performing down-sampling by factors of \( Q \times S \) followed by up-sampling by factors of \( P \times R \) to resize images. However, performing down-sampling first instead of up-sampling introduces significant degradation [7].

Our work is based on the framework proposed in [5]. To avoid significant degradation in resized images, we choose to perform up-sampling first followed by down-sampling to resize images. In addition, optimization is introduced to reduce the computational cost.

The rest of this paper is structured as follows. In Section 2, the spatial relationship of the DCT coefficients between a block and its sub-blocks is reviewed. Section 3 presents the framework in [5] to up-sample and down-sample images by factors of \( P \times R \) and \( Q \times S \), respectively. Subsequently, the proposed approach for resizing images by factors of \( P/Q \times R/S \) is discussed in Section 4. Finally, our conclusions are given in Section 5.

2. Spatial relationship of DCT coefficients between DCT block and its sub-blocks

The \( N \)-point Type 2 DCT of a length-\( N \) sequence, \( e(x) \), is stated as follows:
Let \( E(u) = \alpha(u) \sum_{x=0}^{N-1} e(x) \cos \left( \frac{(2x+1)u\pi}{2N} \right) \),

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

for \( 0 \leq u < N \),

where

\[
\alpha(k) = \begin{cases} \sqrt{1/N}, & k = 0, \\ \sqrt{2/N}, & k = 1, 2, ..., N - 1. \end{cases}
\]

Let \( E^{LN} \) and \( E_i \) denote a \( LN \)-point DCT block and \( N \)-point DCT block, respectively, where \( i = 0, 1, ..., L - 1 \). Based on the spatial relationship of the DCT coefficients between a block and its sub-blocks, a \( LN \)-point DCT block can be composed from \( L \) neighboring \( N \)-point DCT blocks. The relationship for the one-dimensional case is

\[
E^{LN} = A_{(L,N)} \left[ E_0 \ E_1 \ ... \ E_{L-1} \right]^T,
\]

where \( A_{(L,N)} \) is a \( LN \times LN \) matrix. The elements of \( A_{(L,N)} \) are computed as

\[
a(j,k) = a(qN + u, pN + i) = \alpha^2(qN + u) \sum_{o=0}^{N-1} \cos \left( \frac{(2o+1)u\pi}{2N} \right) \cos \left( \frac{(2pN + o + 1)(qN + u)\pi}{2LN} \right),
\]

for \( 0 \leq j, k < LN, 0 \leq p, q < L, 0 \leq i, u < N \).

Let \( E_{i,j} \) denote an \( N \times N \) DCT block where \( i = 0, 1, ..., L - 1 \) and \( j = 0, 1, ..., M - 1 \). Let \( E^{LN \times MN} \) denote a \( LN \times MN \) DCT block. The relationship for the two-dimensional case is

\[
E^{LN \times MN} = A_{(L,N)} \left[ E_{0,0} \ E_{0,1} \ ... \ E_{0,M-1} \\
E_{1,0} \ E_{1,1} \ ... \ E_{1,M-1} \\
... \ ... \ ...
\right] A^{(M,N)^\top},
\]

where \( T \) is the transpose operator. Likewise, \( L \times M \) neighboring \( N \times N \) DCT blocks can be decomposed from a \( LN \times MN \) DCT block with the following equation:

\[
\begin{bmatrix}
E_{0,0} & E_{0,1} & \cdots & E_{0,M-1} \\
E_{1,0} & E_{1,1} & \cdots & E_{1,M-1} \\
\vdots & \vdots & \ddots & \vdots \\
E_{L-1,0} & E_{L-1,1} & \cdots & E_{L-1,M-1}
\end{bmatrix} = A_{(L,N)}^{-1} E^{LN \times MN} A^{(M,N)^\top}.
\]

In this paper, (5) and (6) are referred as composition and decomposition, respectively. Let \( D_{i,j}(L) \) and \( U_{i,j}(L) \) denote \( N \times N \) sub-matrices of \( A_{(L,N)} \) and \( A_{(L,N)}^{-1} \), respectively, where \( i, j = 0, 1, ..., L - 1 \). We can re-express \( A_{(L,N)} \) and \( A_{(L,N)}^{-1} \) as

\[
A_{(L,N)} = \frac{1}{\sqrt{L}}
\begin{bmatrix}
D_{0,0}(L) & D_{0,1}(L) & \cdots & D_{0,L-1}(L) \\
D_{1,0}(L) & D_{1,1}(L) & \cdots & D_{1,L-1}(L) \\
\vdots & \vdots & \ddots & \vdots \\
D_{L-1,0}(L) & D_{L-1,1}(L) & \cdots & D_{L-1,L-1}(L)
\end{bmatrix},
\]

and

\[
A_{(L,N)}^{-1} = \sqrt{L}
\begin{bmatrix}
U_{0,0}(L) & U_{0,1}(L) & \cdots & U_{0,L-1}(L) \\
U_{1,0}(L) & U_{1,1}(L) & \cdots & U_{1,L-1}(L) \\
\vdots & \vdots & \ddots & \vdots \\
U_{L-1,0}(L) & U_{L-1,1}(L) & \cdots & U_{L-1,L-1}(L)
\end{bmatrix}.
\]

Spariness of \( A_{(L,N)} \) and \( A_{(L,N)}^{-1} \) are exploited for fast computation in [5] and [6]. In [11], we have generalized a relationship between the \( N \times N \) sub-matrices of \( A_{(L,N)} \) for all cases of \( L \) and \( N \). These findings are illustrated in Fig. 1. This observation not only reduces the memory used to store \( A_{(L,N)} \), but also...
forms the basis of our fast computation scheme. Using the same approach shown here, similar observations are found in $A_{(l,N)}^{-1}$.

3. Image resizing framework

Since composition and decomposition are separable operations, resizing of images can be accomplished by applying one-dimensional operations consecutively in horizontal and vertical directions or vice versa. Hence, we will consider resizing one-dimensional sequence by factors of $P/Q$ in the following.

With our earlier observations of $A_{(l,N)}$ and $A_{(l,N)}^{-1}$, we are able to reduce the computational cost to down-sample and up-sample images with the image resizing framework in [5]. Let $E_{i}$ denote an $N$-point DCT block of an image where $i = 0, 1, \ldots, P - 1$. Down-sampling by a factor of $Q$ is given as

$$G_{u} = \frac{1}{\sqrt{Q}} \hat{A}_{(Q,Q)} \left[ E_{u,0} \ E_{u,1} \ldots E_{u(Q-1)} \right]^T,$$

for $0 \leq u < P$, where $G_{u}$ is the down-sampled $N$-point DCT block and

$$\hat{A}_{(Q,Q)} = \frac{1}{\sqrt{Q}} \left[ D_{0,0} (Q) \ D_{0,1} (Q) \ldots D_{0,(Q-1)} (Q) \right].$$

Equation (9) can be compactly expressed as

$$G_{u} = \frac{1}{\sqrt{Q}} \sum_{i=0}^{Q-1} D_{0,i} (Q) E_{u,i}',$$

for $0 \leq u < P$. (11)

Next, we consider up-sampling by a factor of $P$. Let $F_{(l)}$ denote a $P$-point up-sampled DCT block where $l = 0, 1, \ldots, Q - 1$. Let $\hat{E}_{i}^{PN}$ denote the zero padded $PN$-point DCT block obtained from $E_{i}$. Up-sampling by a factor of $P$ is given as

$$F_{l} = \sqrt{P} A_{(P,N)}^{-1} \hat{E}_{i}^{PN},$$

for $0 \leq l < Q$. (12)

Let $F_{m}^{(l)}$ denote an up-sampled $N$-point DCT block in $F_{(l)}$ where $m = 0, 1, \ldots, P - 1$. These up-sampled $N$-point DCT blocks are computed as

$$F_{m}^{(l)} = P U_{m,0} (P) E_{i},$$

for $0 \leq m < P$. (13)

Using the relationship depicted in Fig. 1, it can be shown that matrix multiplications required by (11) and (13) can be reduced up to a factor of two.

4. Proposed image resizing framework

The results in Section 3 are used to derive a fast scheme to resize images by factors of $P/Q$. We assume the resizing factors, $P/Q$, is an irreducible fractional number where $P$ and $Q$ are integers greater than one to avoid trivial cases.

To resize images by factors of $P/Q$, neighboring $N$-point DCT blocks are first up-sampled by a factor of $P$. These up-sampled DCT blocks are subsequently down-sampled by a factor of $Q$. Let $G_{u}'$ be the resized $N$-point DCT block. From (9) and (12), resizing images by factors of $P/Q$ is generalized as

$$G_{u}' = \frac{P}{Q} \sum_{i=0}^{Q-1} D_{0,i}(P) U_{mod_{a}(u,Q),0} (P) F_{(u(i)/P)},$$

for $0 \leq u < P$, where $mod_{a}(a,b)$ produces the reminder of $a$ divided by $b$. In prior work [8], computational cost is reduced by combining $D_{0,i}(P) U_{mod_{a}(u,Q),0} (P)$. In this paper, reduction of computational cost is achieved by using $\bar{N}$-point up-sampled DCT blocks where $\bar{N} \leq N$.

From our experiments, we have observed that $\bar{N}$-point up-sampled DCT blocks can be used without significant degradation on the quality of the resized images. Two test cases in Table I are used to illustrate the degradation associated with $\bar{N}$-point up-sampled DCT blocks. The PSNR values in Table I have been obtained from original and resized images that are down-sampled and subsequently up-sampled by factors of 3/4 and 4/3, respectively. For case I, up-sampled DCT blocks of $\bar{N} \times \bar{N}$ are used for down-sampling and up-sampling by factors of 3/4 and 4/3, respectively. For case II, up-sampled DCT block of 8 × 8 and $\bar{N} \times \bar{N}$ are used for down-sampling by factors of 3/4 and up-sampling by factors of 4/3, respectively.

The computational cost per pixel is given as $(c_{ma}, c_{a})$ where $c_{ma}$ and $c_{a}$ denote the number of multiplication and addition, respectively. For Table I, the computational cost is derived with $\bar{N} \times \bar{N}$ DCT block. From Table I, we observed that DCT blocks of $4 \times 4$ introduce no degradation in the resized images. This
suggest that resizing of images by factors of $P/Q$ can be carried out with

$$\hat{N} = \left\lfloor \frac{N}{\min(P, Q)} \right\rfloor + 1. \quad (15)$$

Also, by varying $\hat{N}$, a computational scalable with different PSNR scheme is obtained.

The resizing factors in Table II are derived from resizing high-definition (1920 × 1080) to resolutions that are commonly found in mobile devices. The resizing factors for 720 × 480, 640 × 480 and 320 × 240 are considered for this paper. The resizing factors found to be 4/9 × 3/8, 2/9 × 1/3 and 2/9 × 1/6, respectively. Fig. 2 shows the test image used in this paper.

From Table II, significant reduction of computational cost is found for the proposed approach with $\hat{N} \times \hat{N}$ DCT block as compared to 8 × 8 DCT block. For all the resizing factors in Table II, the proposed approach with $\hat{N} \times \hat{N}$ DCT block requires the lowest number of additions. From Table III, it is also observed that there is marginal PSNR improvement with the proposed approach as compared to [4] for all resizing factors.

5. Conclusions

An approach to arbitrarily resize images in the DCT domain is presented in this paper. With our observations of the spatial relationship of DCT coefficients between a block and its sub-block, resizing images in the DCT domain is achieved with low computational cost. Further reduction of computational cost is achieved by using up-sampled DCT blocks smaller than $N \times N$. The block size for the up-sampled DCT block for high PSNR with low computational cost is also suggested. Also, a computational scalable scheme with different PSNR is obtained by varying the size of the up-sampled DCT block.

6. Acknowledgements

This work was supported in part by Texas Instruments Singapore.

7. References


