# **BLOCK-ADAPTIVE DCT-WIENER IMAGE UP-SAMPLING**

Kwok-Wai Hung and Wan-Chi Siu

Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong

## ABSTRACT

DCT-Wiener image up-sampling scheme is highly desirable since it makes use of the advantages of the information in both spatial and DCT domains. The idea is to combine the observed low-frequency DCT coefficients with the estimated high-frequency DCT coefficients obtained by the Wiener filters in the spatial domain. However, the available 1-D and 2-D Wiener filters that were proposed for high-frequency DCT coefficients estimation are block non-adaptive, mainly due to the limited information from the observed image. In this paper, we propose a block-adaptive Wiener filter by utilizing the information from external training data. During the online estimation, for each image block, the k-nearest relevant DCT LR-HR block pairs are searched from the training data, in order to estimate the coefficients of the Wiener filter. Experimental results show that the proposed block-adaptive Wiener filter improves the PSNR value of the DCT-Wiener scheme by 1.5 dB compared with that using non-adaptive 1-D Wiener filter.

Index Terms- DCT, Wiener filter, up-sampling

## **1. INTRODUCTION**

Since the launch of HDTV formats 720p and 1080p, the old video formats, such as VCD and DVD are required to be up-sampled for displaying on the HDTV and 4k TV. Fortunately, there are high correlations between the high-resolution (HR) and low-resolution (LR) videos, such that many up-sampling algorithm were developed to up-sample the LR video by exploiting the correlations [1-4].

Digital videos are mostly coded using the block-based Discrete Cosine Transform (DCT) for high compression efficiency [5-6]. Due to the convenience, some approaches have been proposed to develop re-sizing [7-20] (up-sample [7-11] and down-sample [12]) methods directly in the block DCT domain. Since the introduction of the DCT-Wiener schemes [9-10], the performance (in terms of PSNR) of the up-sample methods have been significantly improved. The idea is to preserve the observed low-frequency DCT coefficients in the down-sampled low-resolution image, while estimating the high-frequency DCT coefficients (that were truncated during the down-sampling process) by the 1-D and 2-D non-adaptive Wiener filters [10, 21]. Hung and Siu [10] discovered that the 1-D 6-tap Wiener filter for motion compensation is not appropriate for estimating the high-frequency DCT coefficients due to the phase-shift issue and non-optimized filter coefficients. Hence, they proposed to train a 2-D non-separable  $6\times 6$ Wiener filter using the Lena image. This is to minimize the mean squares error of the Wiener filter that estimates the high-frequency DCT blocks from the low-frequency DCT blocks.

In this paper, we address the major weakness of previous block non-adaptive Wiener filters [9-10] by proposing a block adaptive Wiener filter to estimate the high-frequency DCT coefficients. Different from adaptive Wiener filters used in the video coding applications [22-24] that requires the reference picture (original HR image) to estimate the block-adaptive filter coefficients, our approach makes use of the learning-based minimum mean squares error estimation [27] to search for the relevant information from training data to estimate the filter coefficients for each image block. As a result, the proposed *block-adaptive* Wiener filter does not require the original image and significantly improves the available *block non-adaptive* Wiener filters [10, 21].

The rest of organization of this paper is as follows. Section 2 describes the basic structure of the proposed block-adaptive DCT-Wiener scheme. Section 3 explains the proposed block-adaptive Wiener filter. Experimental results are given in Section 4, and Section 5 concludes the paper.

### 2. BLOCK-ADAPTIVE DCT-WIENER SCHEME

#### 2.1. Structure of the proposed scheme

Figure 1 illustrates the basic structure of the proposed DCT-Wiener scheme, which is essentially the same as the hybrid schemes in the literature [9-10], except that we propose a block-adaptive Wiener filter for up-sampling a LR block in the spatial domain. The up-sampled HR block is then DCT transformed for retaining the high-frequency DCT coefficients, which is the most important information contributed to the sharpness and fidelity of the up-sampled image, as justified in the experimental Section. Finally, the LF and HF coefficients are combined to form the HR block.



Fig. 1 The basic structure of the proposed block-adaptive DCT-Wiener scheme.

### 2.2. Why is block-adaptive Wiener filter?

The major reason for developing a block-adaptive Wiener filter is to accommodate different DCT block characteristics by tuning the filter coefficients accordingly. The key to tune the filter coefficients is to carry out a searching process to identify the *k*-nearest neighboring DCT LR-HR block pairs from the external training data. The *k*-nearest block pairs are used as training samples of the Wiener filter for the linear weighted minimal mean squares error estimation. Given a sufficiently good searching criterion and a representative dictionary, the reconstruction error of the block-adaptive Wiener filter estimated from the training data is much lower than that of the fixed and non-adaptive Wiener filters.

From the point of view in the classification theory [27], the non-adaptive Wiener filter assumes that there is only 1 class of data, where the filter coefficients were optimized for this unique class. The block-adaptive Wiener filter suggested in this paper further interprets the data into an infinite number of classes by searching k-nearest block pairs which constitute and define a new class for the optimization in each image block. Specifically, the k-nearest block pairs that constitute the new class are used to optimize the filter coefficients. The closest approach [1] to our proposed work is to search for k-nearest block pairs for estimating the HR blocks in the spatial domain; however, this approach is nontrivial and is not designed to estimate the high-frequency DCT coefficients.



Dictionaries  $\mathbf{B}_{\mathbf{y}'} = [\mathbf{y}'_1 \ \mathbf{y}'_2 \ \dots \ \mathbf{y}'_Q] \quad \mathbf{B}_{\mathbf{u}'} = [\mathbf{u}'_1 \ \mathbf{u}'_2 \ \dots \ \mathbf{u}'_Q] \quad \mathbf{B}_{\mathbf{x}'} = [\mathbf{x}'_1 \ \mathbf{x}'_2 \ \dots \ \mathbf{x}'_Q]$ 



Fig. 2 Training blocks: HR block (blue), larger LR block (orange) and LR block (green).



Fig. 3 The estimation process of one observed LR block using proposed learning-based Wiener filter

## **3. PROPOSED BLOCK-ADAPTIVE WIENER FILTER**

#### 3.1. Building the external training data

Given a training HR image  $\mathbf{Y} \in \mathfrak{R}^{2n \times 2m}$ , where 2n and 2mare dimensions of the HR image, we divide the HR image into Q non-overlapping blocks, represented by vectors,  $\{\mathbf{y}'_i\} \in \Re^{4p \cdot p}$ , where  $4p^2$  is the total number of pixels in each block, as illustrated in Figure 2. Moreover, we downsample the training HR image in the DCT domain by truncating the high-frequency DCT coefficients [11], and denote the training LR image as  $\mathbf{X} \in \Re^{n \times m}$ . Similarly, we divide the LR image into non-overlapping blocks,  $\{\mathbf{x}'_i\} \in \mathfrak{R}^{p \cdot p}$ , and overlapping blocks,  $\{\mathbf{u}'_i\} \in \mathfrak{R}^{(p+1) \cdot (p+1)}$ . The overlapping blocks which have a slightly larger block size can accommodate the neighborhood area during the searching process, such that the searching accuracy can be improved [26]. Finally, we concatenate all vectors into matrices by grouping the vectors in an consecutive order. Note that the vectors with variances less than 48 are discarded. The formed the training data are as follows

$$\mathbf{B}_{\mathbf{y}'} = [\mathbf{y}'_1 \ \mathbf{y}'_2 \ \dots \ \mathbf{y}'_Q] \tag{1}$$

$$\mathbf{B}_{\mathbf{n}'} = [\mathbf{u}'_1 \ \mathbf{u}'_2 \ \dots \ \mathbf{u}'_{\mathcal{O}}] \tag{2}$$

$$\mathbf{B}_{\mathbf{x}'} = [\mathbf{x}'_1 \ \mathbf{x}'_2 \ \dots \ \mathbf{x}'_O] \tag{3}$$

where Q is the total number of vectors (blocks). In this paper, we use 24 training images as shown in Figure 4 to constitute around 220,000 blocks in matrices of training data Eq. (1)-(3).

### 3.2. Online estimation

Given an observed LR image  $\mathbf{X} \in \mathfrak{R}^{n \times m}$ , we divide the image into non-overlapping blocks, represented by vectors,  $\{\mathbf{x}_i\} \in \mathfrak{R}^{p \cdot p}$ , and overlapping blocks  $\{\mathbf{u}_i\} \in \mathfrak{R}^{(p+1) \cdot (p+1)}$ , as illustrated in Figure 3. Let us illustrate the estimation process for this observed LR block  $\mathbf{x}_i$  as an example. As shown in Figure 3, we first compute the normalized correlation coefficients [25] between the observed block  $\mathbf{u}_i$  and all training blocks  $\{\mathbf{u}_i\}$  from the training data  $\mathbf{B}_{\mathbf{u}'}$ . The *k* 

nearest training blocks from the training data  $\mathbf{B}_{\mathbf{u}'}$ , which have the highest correlations with the observed block  $\mathbf{u}_i$ , are selected by the following adaptive *k*-nearest neighbor (*k*-NN) criterion [1],

$$\hat{k} = \arg\min_{k} |k|$$
 subject to  $\sum_{j \in [1,k]} Corr(\mathbf{u}_i, \mathbf{u'}_j) \ge T_1$  (4)

where  $\{\mathbf{u}_{j}^{*}\}$  represents the blocks sorted from the highest correlations to the lowest correlations, such that

$$Corr(\mathbf{u}_i, \mathbf{u'}_j) \ge Corr(\mathbf{u}_i, \mathbf{u'}_{j+1}) \quad \text{for } \forall j \quad (5)$$

and  $T_1$  is a threshold to be determined by the crossvalidation [27]. Having obtained the index of the *k*-nearest training blocks using Eq. (4) and Eq. (5), we use the index to extract other *k*-nearest training blocks from the rest training data  $\mathbf{B}_{\mathbf{x}'}$  and  $\mathbf{B}_{\mathbf{y}'}$ , and group them into matrices, as follows

$$\mathbf{T}_{\mathbf{x}'} = \begin{bmatrix} \mathbf{x}'_1 & \mathbf{x}'_1 & \dots & \mathbf{x}'_k \end{bmatrix}$$
(6)

$$\mathbf{T}_{\mathbf{y}'} = \begin{bmatrix} \mathbf{y}'_1 & \mathbf{y}'_1 & \dots & \mathbf{y}'_k \end{bmatrix}$$
(7)

where  $\{\mathbf{x}'_{j}\}\$  and  $\{\mathbf{y}'_{j}\}\$  represent other *k*-nearest training blocks from the training data  $\mathbf{B}_{\mathbf{x}'}\$  and  $\mathbf{B}_{\mathbf{y}'}$ . Let us model the weight of each pair of *k*-nearest LR-HR training data, i.e.  $\mathbf{x}'_{j}$  and  $\mathbf{y}'_{j}$ , using the exponential function to decay the weight of that pair by the computed correlation value, as follows

$$W_{i} = \exp(Corr(\mathbf{u}_{i}, \mathbf{u'}_{i}) \cdot f)$$
(8)

where  $W_j$  represents a diagonal element of the weighting matrix  $\mathbf{W} \in \mathfrak{R}^{k \times k}$ , which concatenates the weights of all training pairs,  $\{\mathbf{x}'_j\}$  and  $\{\mathbf{y}'_j\}$ , and *f* is a parameter to be determined using cross-validation. Hence, the linear weighted MMSE estimation is performed as follows [1, 27]

$$\hat{\mathbf{H}}_{i} = \mathbf{T}_{\mathbf{v}'} \mathbf{W} \mathbf{T}_{\mathbf{x}'}^{T} (\mathbf{T}_{\mathbf{x}'} \mathbf{W} \mathbf{T}_{\mathbf{x}'}^{T})^{-1}$$
(9)

where  $\mathbf{H}_i \in \Re^{4p \cdot p \times p \cdot p}$  are filter coefficients of the Wiener filter, i.e., the linear MMSE estimator. Eventually, the observed vector  $\mathbf{x}_i$  is multiplied with the filter coefficients obtained from Eq. (9) to give the estimated HR block

$$\hat{\mathbf{y}}_i = \hat{\mathbf{H}}_i \mathbf{x}_i \tag{10}$$

For reducing blocking artifacts, in our implementation, we artificially shift the observed LR image by [-1, 2] in both dimensions, i.e., shift for 16 times, in order to obtain 16 different estimates of each high-resolution pixel. Then, we average the 16 estimates to give the final results.

### 4. EXPERIMENTAL RESULTS

The experimental works were done on an Intel i7 3GHz system. As we have explained, several parameters of our learning-based Wiener filter were optimized empirically through cross-validation, where the values are shown in Table 1. Eight CIF  $(352\times288)$  and two 720p  $(1280\times720)$  video sequences, as shown in Figure 5, were down-sampled by two times by truncating the high-frequency DCT coefficients in every image blocks of size  $8\times8$  [11].

Different state-of-the-art methods, including the bicubic interpolation, zero padding [14], overlapping zero padding [7], 1-D non-adaptive Wiener filter [21], hybrid DCT-Wiener scheme [9], and the proposed block-adaptive DCT-Wiener scheme, were applied to up-sample the Y-components of all video frames. The UV components of video frames were up-sampled by the bicubic interpolation. PSNR-Y and SSIM-Y [28] values were measured, as shown in Table 2.

The processing time for up-sampling a frame from QCIF (174×144) to CIF (352×288) is around 2 minutes using our non-optimized MATLAB codes. On the contrary, the 720p sequences require much longer processing time. The major computation comes from searching the k similar training pairs from the training data consisting of around 220,000 training pairs. Subjective comparisons are shown in Figure 6. Sample locations to give notice are indicated by red arrows in this Figure. In general, the proposed up-sampling scheme produces pictures with the highest fidelity to the original high-resolution image, especially for the explicit object boundaries in the Foreman (Figure 6) sequences.

The PSNR and SSIM measurements in Table 2 verify the subjective improvement observed in Figure 6. Specifically, the proposed scheme has the advantage of more than 1 dB in PSNR and around 0.03 in SSIM improvement comparing with the state-of-the-art methods in the literature. In other words, the proposed scheme significantly improves the available methods measured by subjective and objective evaluations.



Fig. 4 The 24 training images (768×512) used in this paper.



Fig. 5 Eight CIF (352×288) and two 720p (1280×720) video sequences used in our experiments.

Parameters	Explanation	Values
р	Block size for down-sampling and up-sampling	4
$T_1$	Threshold for adaptive k-NN in Eq. (4)	350
	Maximum value of $k$ in Eq. (4)	600
f	Weighting function parameter in Eq. (8)	40

 
 Table 1
 The parameter settings of the proposed learningbased Wiener filter

Table 2 PSNR-Y (dB) and M-SSIM of (a) Zero padding[14], (b) overlapping zero padding [7], (c) bicubicinterpolation, (d) 1-D non-adaptive Wiener filter [21], (e)DCT-Wiener scheme [9], and (f) proposed scheme

Videog	PSNR-Y (dB)						
videos	(a)	(b)	(c)	(d)	(e)	(f)	
Akiyo (CIF)	34.778	35.208	34.701	31.628	34.954	37.905	
Bus (CIF)	25.518	25.916	25.435	23.136	25.570	26.897	
Container (CIF)	27.331	27.635	27.110	24.983	27.276	29.004	
Football (CIF)	32.231	32.748	32.184	28.865	32.141	34.452	
Foreman (CIF)	31.588	31.478	31.135	29.394	31.938	32.738	
Mobile (CIF)	23.040	23.397	22.931	21.036	23.137	24.460	
News (CIF)	29.807	30.197	29.610	26.542	30.163	32.354	
Paris (CIF)	23.547	23.562	23.364	21.906	23.306	24.298	
Crew (720p)	38.588	39.037	38.461	35.107	38.757	40.120	
BigShips (720p)	33.391	33.572	33.206	30.882	33.481	34.727	
Average	29.982	30.275	29.814	27.348	30.072	31.695	
¥7+ 1	M-SSIM-Y						
videos	(a)	(b)	(c)	(d)	(e)	(f)	
	()	(-)	(0)	(u)	$(\mathbf{c})$	(1)	
Akiyo (CIF)	0.9588	0.9640	0.9619	0.9396	0.9597	0.9768	
Akiyo (CIF) Bus (CIF)	0.9588 0.8465	0.9640 0.8581	0.9619	0.9396 0.7762	0.9597 0.8489	0.9768 0.8847	
Akiyo (CIF) Bus (CIF) Container (CIF)	0.9588 0.8465 0.8738	0.9640 0.8581 0.8845	0.9619 0.8400 0.8731	0.9396 0.7762 0.8286	0.9597 0.8489 0.8743	(1) 0.9768 0.8847 0.9056	
Akiyo (CIF) Bus (CIF) Container (CIF) Football (CIF)	0.9588 0.8465 0.8738 0.9168	0.9640 0.8581 0.8845 0.9262	0.9619 0.8400 0.8731 0.9139	0.9396 0.7762 0.8286 0.8690	0.9597 0.8489 0.8743 0.9204	0.9768 0.8847 0.9056 0.9381	
Akiyo (CIF) Bus (CIF) Container (CIF) Football (CIF) Foreman (CIF)	0.9588 0.8465 0.8738 0.9168 0.9148	0.9640 0.8581 0.8845 0.9262 0.9183	0.9619 0.8400 0.8731 0.9139 0.9123	0.9396 0.7762 0.8286 0.8690 0.8843	0.9597 0.8489 0.8743 0.9204 0.9167	0.9768 0.8847 0.9056 0.9381 0.9308	
Akiyo (CIF) Bus (CIF) Container (CIF) Football (CIF) Foreman (CIF) Mobile (CIF)	0.9588 0.8465 0.8738 0.9168 0.9148 0.7930	0.9640 0.8581 0.8845 0.9262 0.9183 0.8072	0.9619 0.8400 0.8731 0.9139 0.9123 0.7858	(d)           0.9396           0.7762           0.8286           0.8690           0.8843           0.7181	0.9597 0.8489 0.8743 0.9204 0.9167 0.7948	(1) 0.9768 0.8847 0.9056 0.9381 0.9308 0.8534	
Akiyo (CIF) Bus (CIF) Container (CIF) Football (CIF) Foreman (CIF) Mobile (CIF) News (CIF)	0.9588 0.8465 0.8738 0.9168 0.9148 0.7930 0.9354	0.9640 0.8581 0.8845 0.9262 0.9183 0.8072 0.9440	(c)           0.9619           0.8400           0.8731           0.9139           0.9123           0.7858           0.9392	(d)           0.9396           0.7762           0.8286           0.8690           0.8843           0.7181           0.9016	0.9597 0.8489 0.8743 0.9204 0.9167 0.7948 0.9357	0.9768 0.8847 0.9056 0.9381 0.9308 0.8534 0.9653	
Akiyo (CIF) Bus (CIF) Container (CIF) Football (CIF) Foreman (CIF) Mobile (CIF) News (CIF) Paris (CIF)	0.9588 0.8465 0.8738 0.9168 0.9148 0.7930 0.9354 0.8125	0.9640 0.8581 0.8845 0.9262 0.9183 0.8072 0.9440 0.8160	(c)           0.9619           0.8400           0.8731           0.9139           0.9123           0.7858           0.9392           0.8066	(d)           0.9396           0.7762           0.8286           0.8690           0.8843           0.7181           0.9016           0.7524	0.9597 0.8489 0.8743 0.9204 0.9167 0.7948 0.9357 0.8011	0.9768 0.8847 0.9056 0.9381 0.9308 0.8534 0.9653 0.8521	
Akiyo (CIF) Bus (CIF) Container (CIF) Football (CIF) Foreman (CIF) Mobile (CIF) News (CIF) Paris (CIF) Crew (720p)	0.9588 0.8465 0.8738 0.9168 0.9148 0.7930 0.9354 0.8125 0.9592	0.9640 0.8581 0.8845 0.9262 0.9183 0.8072 0.9440 0.8160 0.9611	(c)           0.9619           0.8400           0.8731           0.9139           0.9123           0.7858           0.9392           0.8066           0.9571	(d)           0.9396           0.7762           0.8286           0.8690           0.8843           0.7181           0.9016           0.7524           0.9339	0.9597 0.8489 0.8743 0.9204 0.9167 0.7948 0.9357 0.8011 0.9585	0.9768 0.8847 0.9056 0.9381 0.9308 0.8534 0.9653 0.8521 0.9658	
Akiyo (CIF) Bus (CIF) Container (CIF) Football (CIF) Foreman (CIF) Mobile (CIF) News (CIF) Paris (CIF) Crew (720p) BigShips (720p)	0.9588 0.8465 0.8738 0.9168 0.9148 0.7930 0.9354 0.8125 0.9592 0.9151	0.9640 0.8581 0.8845 0.9262 0.9183 0.8072 0.9440 0.8160 0.9611 0.9189	0.9619 0.8400 0.8731 0.9139 0.9123 0.7858 0.9392 0.8066 0.9571 0.9121	(d)           0.9396           0.7762           0.8286           0.8690           0.8843           0.7181           0.9016           0.7524           0.9339           0.8777	0.9597 0.8489 0.8743 0.9204 0.9167 0.7948 0.9357 0.8011 0.9585 0.9152	0.9768 0.8847 0.9056 0.9381 0.9308 0.8534 0.9653 0.8521 0.9658 0.9300	

Figure 6 shows that the proposed block-adaptive DCT-Wiener up-sampling scheme preserves the sharpness across edge orientation and the smoothness along edge orientation, which are the fundamental characteristics of sharp edges without artifacts. It further justified the reconstruction ability of high-frequency DCT coefficients of the proposed scheme, owning to the results that small details (such as teeth, eye, nose) are well reconstructed. Note that these tiny details may not have similar counterparts in our training images (Figure 4); however, the proposed scheme are very strong in deringing and avoiding artifacts by adaptively tuning the filter coefficients and averaging 16 estimates.

## 5. CONCLUSION

In this paper, we propose a learning-based Wiener filter which is block-adaptive according to the input LR image block, to estimate the HR block, in order to extract the highfrequency DCT coefficients. The proposed block-adaptive Wiener filter is incorporated into the DCT-Wiener scheme that combines the information from both spatial and DCT domain to up-sample an image which is assumed to be down-sampled in the DCT domain in various applications. Subjective and objective evaluations verify the highly competitive performance of the proposed Wiener filter for the DCT-Wiener scheme.

One of the future directions is to decrease the complexity of the proposed Wiener filter by pre-computing a fixed number of sets of filter coefficients for various image contents. During the online estimation, the search for *k*nearest training pairs is avoided, in order to dramatically decrease the overall computation. It is anticipated that there will be a substantial performance drop by using fixed sets of filter coefficients; however, depending on the number of precomputed filters, the overall performance should still be higher than the non-adaptive Wiener filter using one set of filter coefficients in previous approaches.

Acknowledgement: This work is supported by the Center for Signal Processing, the Hong Kong Polytechnic University and the Research Grant Council of the Hong Kong SAR Government (GRF: B-Q38S).



(e) Proposed up-sampling scheme (f) Original high-resolution image

**Fig. 6** First frame of the reconstructed Foreman sequence for two times up-sampling. (Please look for the electronic version for a better perception.)

### 6. REFERENCES

- [1] K.S. Ni and T.Q. Nguyen, "An AdapTable k-Nearest Neighbors Algorithm for MMSE Image Interpolation," *IEEE Transactions* on Image Processing, vol.18, no.9, pp.1976-1987, Sept. 2009
- [2] Kwok-Wai Hung and Wan-Chi Siu, "A Computationally Scalable Adaptive Image Interpolation Algorithm using Maximum-likelihood De-noising for Real-time Applications", *Journal of Electronic Imaging*, vol. 22, no. 4, p. 043006, Oct.-Dec. 2013.
- [3] Kwok-Wai Hung and Wan-Chi Siu, "Robust soft-decision interpolation using weighted least squares," *IEEE Transactions* on *Image Processing*, vol.21, no.3, pp.1061-1069, March 2012.
- [4] Kwok-Wai Hung and Wan-Chi Siu, "Fast image interpolation using bilateral filter", *IET Image Processing*, vol. 6, no. 7, pp. 877-890, October 2012.
- [5] Z. Shi, X. Sun, F. Wu, "Spatially Scalable Video Coding For HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.22, no.12, pp.1813-1826, Dec. 2012
- [6] C.A. Segall and G.J. Sullivan, "Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.9, pp.1121-1135, Sept. 2007
- [7] IlHong Shin and Hyun Wook Park, "Adaptive Up-Sampling Method Using DCT for Spatial Scalability of Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.19, no.2, pp.206-214, Feb. 2009
- [8] Z. Alkachouh and M.G. Bellanger, "Fast DCT-based spatial domain interpolation of blocks in images," *IEEE Transactions* on *Image Processing*, vol.9, no.4, pp.729-732, Apr 2000
- [9] Zhenyu Wu, Hongyang Yu and Chang Wen Chen, "A New Hybrid DCT-Wiener-Based Interpolation Scheme for Video Intra Frame Up-Sampling," *IEEE Signal Processing Letters*, vol.17, no.10, pp.827-830, Oct. 2010
- [10] Kwok-Wai Hung and Wan-Chi Siu, "Hybrid DCT-Wiener-Based Interpolation via Learnt Wiener Filter", *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (ICASSP 2013), pp. 1419-1423, 26-31 May 2013.
- [11] Haiyan Shu and Lap-Pui Chau, "An efficient arbitrary downsizing algorithm for video transcoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.14, no.6, pp. 887- 891, June 2004
- [12] Haiyan Shu and Lap-Pui Chau, "A Resizing Algorithm With Two-Stage Realization for DCT-Based Transcoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.17, no.2, pp.248-253, Feb. 2007
- [13] YoungSeo Park and HyunWook Park, "Design and analysis of an image resizing filter in the block-DCT domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.14, no.2, pp. 274- 279, Feb. 2004
- [14] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain," *IEEE Transactions on Circuits* and Systems for Video Technology, vol. 11, no. 4, pp. 461–474, Apr. 2001.

- [15] C. Salazar and T.D. Tran, "A Complexity Scalable Universal DCT Domain Image Resizing Algorithm," *IEEE Transactions* on Circuits and Systems for Video Technology, vol.17, no.4, pp.495-499, April 2007
- [16] Young Seo Park and Hyun Wook Park, "Arbitrary-ratio image resizing using fast DCT of composite length for DCT-based transcoder," *IEEE Transactions on Image Processing*, vol.15, no.2, pp.494-500, Feb. 2006
- [17] V. Patil, R. Kumar and J. Mukherjee, "A Fast Arbitrary Factor Video Resizing Algorithm," *IEEE Transactions on Circuits* and Systems for Video Technology, vol.16, no.9, pp.1164-1171, Sept. 2006
- [18] H. W. Park, Y. S. Park and S. K. Oh, "L/M-fold image resizing in block-DCT domain using symmetric convolution," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1016–1034, Sep. 2003.
- [19] J. Mukherjee and S.K. Mitra, "Image resizing in the compressed domain using subband DCT," *IEEE Transactions* on Circuits and Systems for Video Technology, vol.12, no.7, pp.620-627, Jul 2002
- [20] Hyungjun Lim and H. Park, "A Ringing-Artifact Reduction Method for Block-DCT-Based Image Resizing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.21, no.7, pp.879-889, July 2011
- [21] Advanced Video Coding for Generic Audiovisual Services ITU-T and ISO/IEC JTC1, Rec. H.264-ISO/IEC 14496-10 AVC, 2003.
- [22] Jie Dong and King Ngi Ngan, "Parametric interpolation filter for HD video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.20, No.12, pp.1892-1897, Dec. 2010.
- [23] Y. Vatis and J. Ostermann, "Adaptive interpolation filter for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 179–192, Feb. 2009.
- [24] D. Rusanovskyy, K. Ugur, A. Hallapuro, J. Lainema and M. Gabbouj, "Video coding with low-complexity directional adaptive interpolation filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 8, pp 1239–1243, Aug. 2009.
- [25] A. Goshtasby, S. H. Gage and J. F. Bartholic, "A Two-Stage Cross-Correlation Approach to Template Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 374-378, 1984.
- [26] W. T. Freeman, T. R. Jones and E. C. Pasztor, "Examplebased superresolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.
- [27] R. C. Gonzalez and R. E. Woods, *Digital Image Processing* (third edition), Reading, Massachusetts: Addison-Wesley, 1992.
- [28] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error measurement to structural similarity," *IEEE Transactions on Image Processing*, vol. 3, no. 4, pp. 600-612, Apr. 2004.