Modified edge-directed interpolation for images

Wing-Shan Tam
Chi-Wah Kok
Wan-Chi Siu
The Hong Kong Polytechnic University
Department of Electronic and Information Engineering
Hung Hom, Kowloon, Hong Kong
E-mail: wstam@ieee.org

Abstract. *We present a modification of the new edge-directed interpolation method that eliminates the prediction error accumulation problem by adopting a modified training window structure, and extending the covariance matching into multiple directions to suppress the covariance mismatch problem. Simulation results show that the proposed method achieves remarkable subjective performance in preserving the edge smoothness and sharpness among other methods in the literaturé. It also demonstrates consistent objective performance among a variety of images. © 2010 SPIE and IS&T.* [DOI: 10.1117/1.3358372]

1 Introduction

Image interpolation is a process that estimates a set of unknown pixels from a set of known pixels in an image. It has been widely adopted in a variety of applications, such as resolution enhancement, image demosaicing,[1,2] and unwrapping omni-images.[3] The kinds of distortion and levels of degradation imposed on the interpolated image depend on the interpolation algorithm, as well as the prior knowledge of the original image. Two of the most common types of degradation are the zigzag errors (also known as the jaggies), and the blurring effects.[4] As a result, high quality interpolated images are obtained when the pixel values are interpolated according to the edges of the original images. A number of edge-directed interpolation (EDI) methods have been presented in the literature. Some of them match the local geometrical properties of the image with predefined templates in an attempt to obtain an accurate model and thus estimate the unknown pixel values.[5–8] However, these algorithms suffer from the inherent problem with the use of edge maps or other image feature maps, where the edges and other image features are difficult if not impossible to be accurately located. The poor edge estimation limits the visual quality of the interpolated images. Other EDI methods make use of the isophote-based methods to direct the edge interpolation to conform the pixel intensity contours.[7,8] These algorithms are highly efficient in interpolating sharp edges (with significant intensity changes across edges). However, the interpolation performance is degraded with blurred edges, which are commonly observed in natural images. To cater this problem, edge enhancement or sharpening techniques are proposed.[9] However, the use of an edge map is indispensable and noise amplification is aroused with the application of postprocessing techniques. Besides using edge maps, some EDI methods direct the interpolation by further locating the edge orientation with the use of a gradient operator.[10–12] These methods are effective in eliminating the blurring and staircase problems by detecting the edge orientation adaptively. However, they suffer from the inherent problem of using an edge map, and the gradient operator is not fully adaptive to the image structure. Other EDI methods make use of local statistical and geometrical properties to interpolate the unknown pixel values, and are shown to be able to obtain high visual quality interpolated images without the use of edge maps.[13–18] The new edge-directed interpolation (NEDI) method[13] models the natural image as a second-order locally stationary Gaussian process, and estimates the unknown pixels using simple linear prediction. The covariance of the image pixels in a local block (also known as a training window) is required for the computation of the prediction coefficients. Compared to conventional methods such as the bilinear or bicubic methods, the NEDI method preserves the sharpness and continuity of the interpolated edges. However, this method considers only the four nearest neighboring pixels along the diagonal edges. As a result, not all the unknown pixels are estimated from the original image, which degrades the quality of the interpolated image. Moreover, the NEDI method has a large interpolation kernel size, which reduces the visual quality and the peak signal-to-noise ratio (PSNR) of the interpolated texture image. The Markov random field (MRF) model-based method[14] models the image with MRF and extends the edge estimation in a number of possible directions by increasing the number of neighboring pixels in the kernel. The MRF model-based method is able to preserve the visual quality of the interpolated edges and also maintain the fidelity of the interpolated image, thus enhancing the PSNR level. The more accurate the MRF model, the better the efficiency of the MRF model-based method. However, the computational complexity is inevitably increased. Though both the NEDI and MRF model-based methods are statistically optimal, the NEDI method adopts a relatively simple model and is thus less computationally expensive. Therefore, a lot of research has been performed to enhance the performance of the NEDI method. The improved new edge-directed interpolation (iNEDI) method[16]

Paper 09115R received Jul. 8, 2009; revised manuscript received Dec. 25, 2009; accepted for publication Jan. 25, 2010; published online Mar. 22, 2010.

1017-9909/2010/19(1)/013011/20/$25.00 © 2010 SPIE and IS&T.

modifies the NEDI method by varying the size of the training window according to the edge size and achieves better PSNR performance. However, the computational cost is high and the performance is highly dependent on the chosen parameters, which are also image dependent. Regarding the computational cost, there are fast algorithms that integrate the advantages of the isophote-based methods and edge enhancement techniques, which can achieve high quality interpolated images.[17,18] However, not all these methods are statistically optimized, thus they degrade the continuity and sharpness of the interpolated edges. The iterative curvature-based interpolation (ICBI) method[18] considers the effects of the curvature continuity, curvature enhancement, and isophote contour. By properly weighting these three effects, the ICBI method produces perceptually pleasant images and significantly reduces the computational cost. However, similar to the iNEDI method, the performance depends on the chosen parameters.

This work presents an improved NEDI method, namely modified edge-directed interpolation (MEDI), which is an extension of our work in Ref. 19. In Ref. 19, we proposed a different training window to mitigate the interpolation error propagation problem. A similar training window was later found to be presented in improved edge-directed interpolation (IEDI)[15] independently. While the enlarged training window eliminates the error propagation problem, it also inevitability increases the interpolation error due to the worsened covariance mismatch problem. As a result, the interpolation results obtained by IEDI are shown to be worse than that of NEDI in most cases. To mitigate the covariance mismatch problem, we propose to apply multiple training windows. A brief and rapid report of the proposed method has been presented in Ref. 20. In the brief report, only the framework of the proposed method and the grayscale image interpolation performance have been presented. In this work, a detailed analysis and elaboration of the proposed method is presented with the assistance of a pseudocode and extensive simulations. The performance of the proposed method applied to color image interpolation is also investigated. The performance and computational complexity of the proposed method is examined, with comprehensive simulations and comparisons with other EDI-based interpolation methods (including NEDI, IEDI, iNEDI, and ICBI methods) and filtering approaches (including Lanczos filtering and B-spline filtering). The simulation results show that the proposed method generates high visual quality images and demonstrates a highly consistent objective performance over a wide variety of images.

2 Algorithm

Consider the interpolation of a low-resolution image X (with size $H\times W$) to a high-resolution image Y (with size $2H\times 2W$), such that $Y_{2i,2j}=X_{i,j}$. This is graphically shown in Fig. 1, where the white dots denote the pixels from X. The NEDI method is a two-step interpolation process that first estimates the unknown pixels $Y_{2i+1,2j+1}$ [gray dot in Fig. 1(a)], then the pixel $Y_{2i+1,2j}$ [black dot in Fig. 1(b)]. Note that the pixel $Y_{2i,2j+1}$ [not shown in Fig. 1(b)] can also be estimated similar to that of pixel $Y_{2i+1,2j}$. The NEDI method makes use of a fourth-order linear prediction to interpolate an unknown pixel from the four neighboring pixels, e.g., $Y_{2i+1,2j+1}$ is estimated from $\{Y_{2i,2j}, Y_{2i+2,2j}, Y_{2i+2,2j+2}, Y_{2i,2j+2}\}$ as

$$Y_{2i+1,2j+1}=\sum_{k=0}^{1}\sum_{\ell=0}^{1}\alpha_{2k+\ell}Y_{2(i+k),2(j+\ell)}. \tag{1}$$

To simplify the notations, and without ambiguity, the 16 covariance values and four cross-covariance values obtained by the four pixels in Eq. (1) are enumerated to be $R_{k\ell}$ and r_k, with $0\leqslant k,\ell\leqslant 3$, respectively, as shown by the labels next to the arrows in Fig. 1(a). For example, $R_{03}=E[Y_{2i,2j}Y_{2i,2j+2}]$ and $r_0=E[Y_{2i,2j}Y_{2i+1,2j+1}]$. The optimal prediction coefficients set α can be obtained as[13]

$$\boldsymbol{\alpha}=\mathbf{R}_{yy}^{-1}\mathbf{r}_y, \tag{2}$$

where $\alpha=[\alpha_0,\cdots,\alpha_3]$, $\mathbf{R}_{yy}=[R_{k\ell}]$ and $\mathbf{r}_y=[r_0,\cdots,r_3]$. The interpolation is therefore locally adapted to $\mathbf{R}_{yy}$ and $\mathbf{r}_y$. However, the computation of $R_{k\ell}$ and r_k would require the knowledge of $Y_{2i+1,2j+1}$, which is not available before the interpolation. This difficulty is overcome by the geometric duality property, where the covariance $\hat{r}_0$ [circled in Fig. 1(a)] estimated from the low-resolution training window is applied to replace the high-resolution covariance r_0, as indicated by the arrow. In a similar manner, the covariance r_k is replaced by $\hat{r}_k$ with $0\leqslant k\leqslant 3$. The unknown pixel $Y_{2i+1,2j+1}$ is therefore estimated by Eq. (1) with $\hat{R}_{k\ell}$ and $\hat{r}_k$. The remaining pixels $Y_{2i,2j+1}$ and $Y_{2i+1,2j}$ can be obtained by the same method with a scaling of $2^{1/2}$ and a rotation of $\pi/4$, as shown in Fig. 1(b). To better handle the texture interpolation, a hybrid approach is adopted, where covariance-based interpolation is applied to edge pixels (pixels near an edge) when the covariance matrix has full rank, and the variance of the pixels in the local block is higher than a predefined threshold ϵ; otherwise, bilinear interpolation is applied to nonedge pixels (pixels in smooth regions). However, prediction error is unavoidable in the interpolated pixels. The NEDI method propagates the errors from the first step to the second step, because the estimation in the second step depends on the result of the first step [the black dot is estimated from the gray dots, as shown in Fig. 1(b)]. To cater this problem, a modified training window structure has been developed independently in Refs. 15 and 19. The training window in the second step of the NEDI method for the interpolation of $Y_{2i+1,2j}$ and $Y_{2i,2j+1}$ is modified to form a sixth-order linear prediction with a 5×9 training window, as illustrated in Fig. 2, where

$$Y_{2i+1,2j}=\sum_{k=0}^{1}\sum_{\ell=-1}^{1}\alpha_{2k+\ell}Y_{2(i+k),2(j+\ell)}. \tag{3}$$

The coefficients $\alpha_{2k+\ell}$ can be estimated from Eq. (2) with the autocovariance matrix $\mathbf{R}_{yy}$ that contains 36 $R_{k\ell}$, and cross-covariance vector $\mathbf{r}_y$ with six elements of r_k with $0\leqslant k,\ell\leqslant 5$. The high-resolution covariances are then replaced by the low-resolution covariances of $\hat{\mathbf{R}}_{yy}$ and $\hat{\mathbf{r}}_y$ using the geometric duality property. The rest of the unknown pixels $Y_{2i,2j+1}$ can be estimated in a similar manner with a sixth-order linear prediction as that for pixels $Y_{2i+1,2j}$, but with the training window rotated by $\pi/2$.

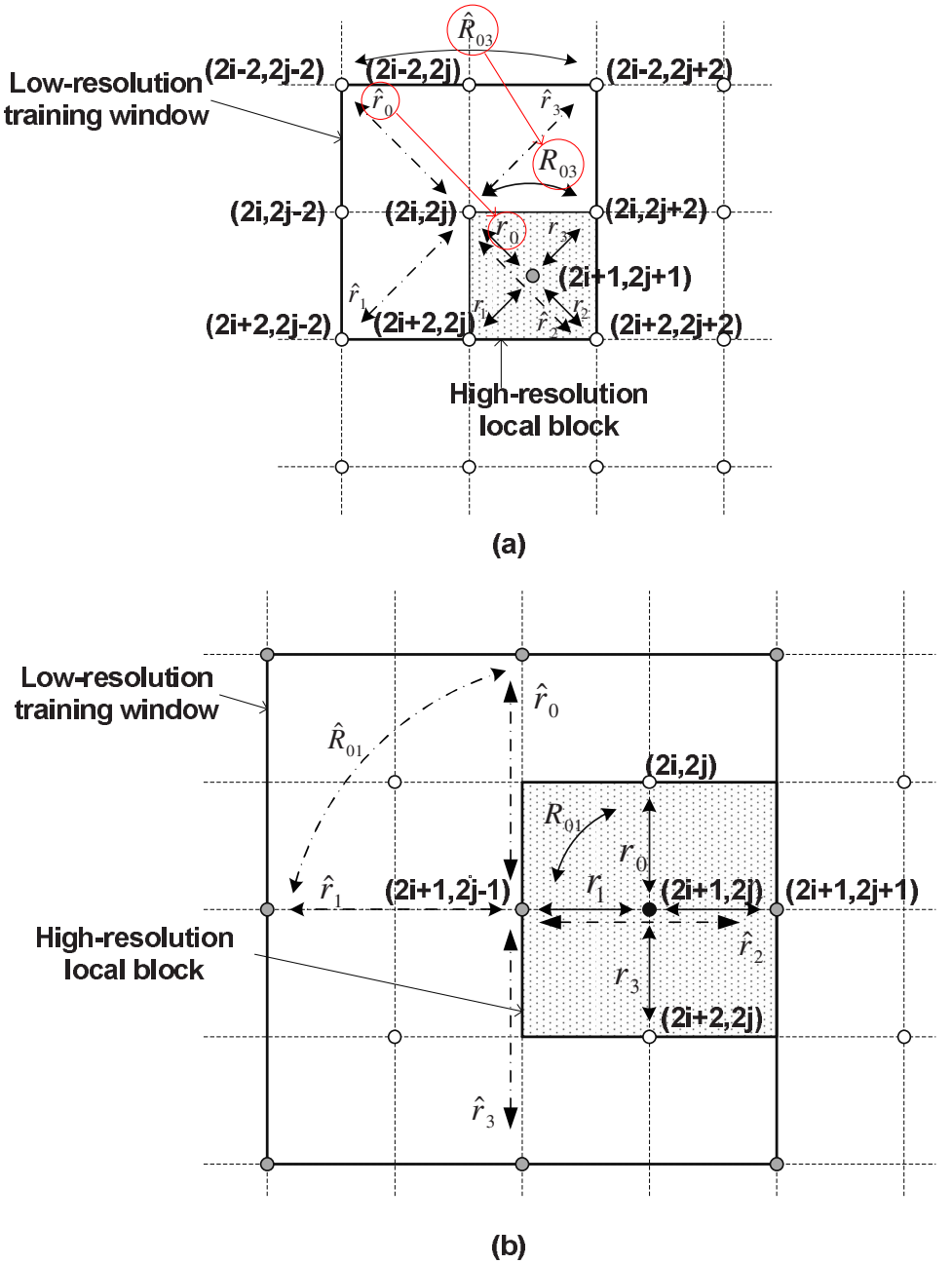

Fig. 1 Illustration of the training windows and local blocks of (a) the first step and (b) the second step of the NEDI method.

Although the interpolation error propagation problem can be rendered by the enlarged training window, both the methods presented in Refs. 15 and 19 still suffer from the covariance structure mismatch problem, as illustrated in Fig. 3, where the white box is the geometric low-resolution training window, the gray box is the corresponding high-resolution local block, and the dash lines "AB" and "CD" indicate the image edges in the local block. Figures 3(a) and 3(b) show the training windows adopted in the NEDI and IEDI methods. Clearly, the geometric duality property is satisfied for the edge AB, as shown in Fig. 3(a). However, it is apparent that the geometric duality property is not satisfied for the edge CD, as shown in Fig. 3(b), and thus causes covariance mismatch. To cater this problem, the consideration of all four locations of the low-resolution training window and the high-resolution local block, as shown in Figs. 3(b)–3(e), is proposed.

2.1 *Proposed Method: Modified Edge-Directed Interpolation*

To reduce the covariance mismatch problem, multiple low-resolution training window candidates are used. Figures 3(b)–3(e) illustrate the four training windows applied in the first step of the proposed method. The NEDI and IEDI methods consider the training window shown in Fig. 3(b) only, and the training window is centered at pixel $Y_{2i,2j}$ (see Fig. 1 for the pixel location) in the first step. Compared

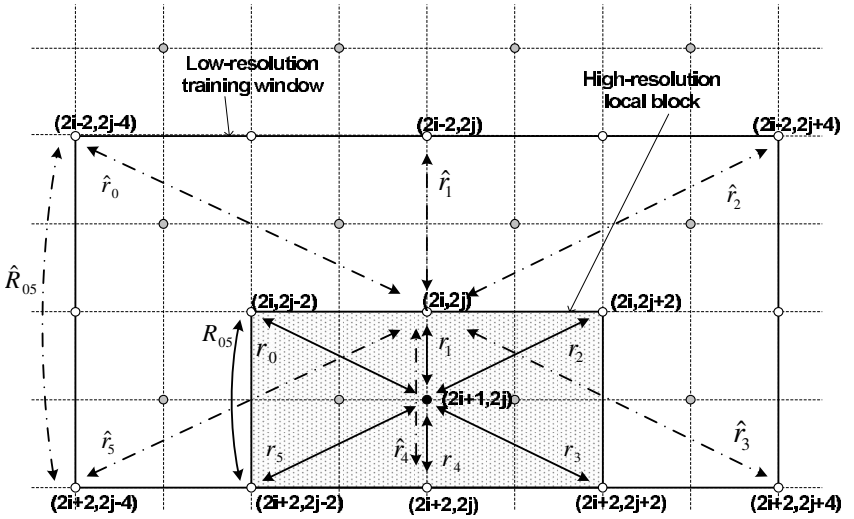

Fig. 2 Illustration of the training window and local block of the second step of the MEDI method.

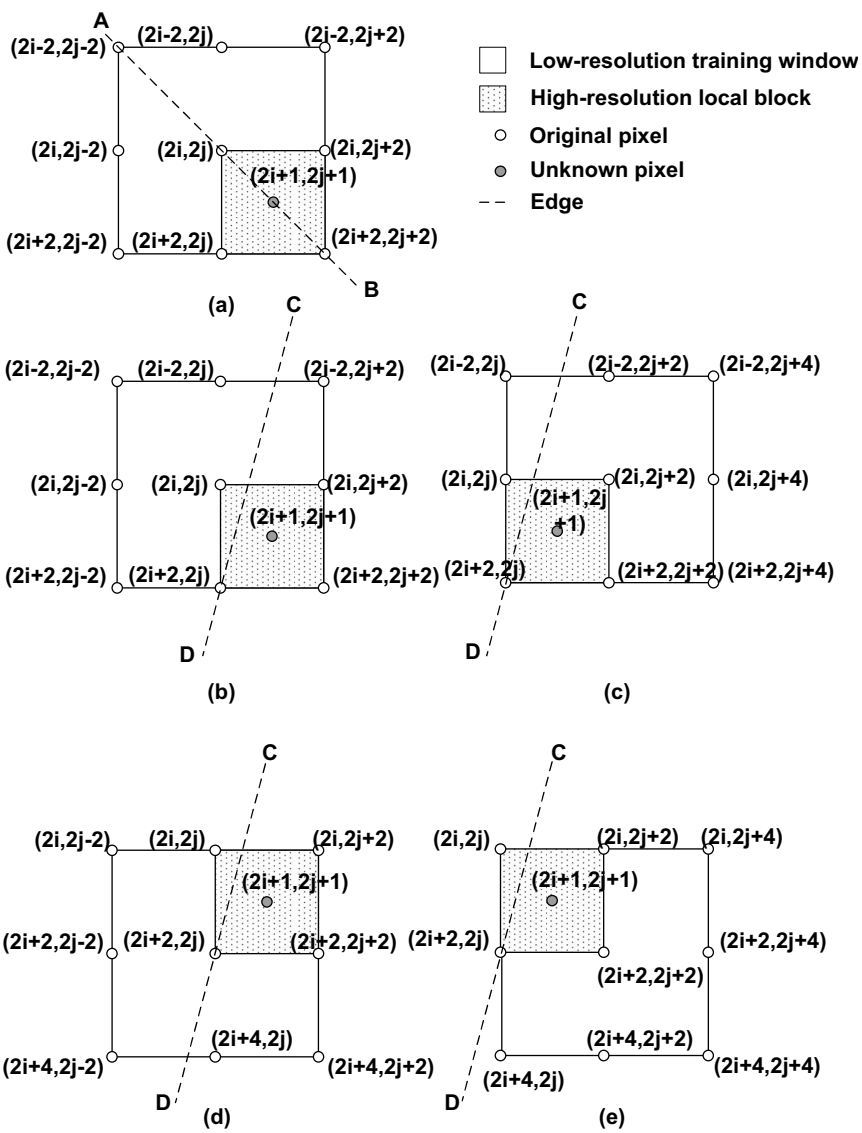

Fig. 3 Illustration of (b) through (e) the four training window candidates in the MEDI method for the estimation of high resolution block in (a).

with the NEDI method, the proposed MEDI method considers three more training windows centered at $Y_{2i,2j+2}$, $Y_{2i+2,2j}$, and $Y_{2i+2,2j+2}$, as illustrated in Figs. 3(c)–3(e), respectively. The covariance signal energy obtained from all training windows is compared. The higher the energy in the training window, the more likely the edge exists. The one that contains the highest energy will be applied to the linear prediction in Eq. (1). In this example, the training window in Fig. 3(c) is applied for the prediction. Similarly, the MEDI method considers six training window candidates in the second step, with such windows centered at $Y_{2i,2j-2}$, $Y_{2i,2j}$, $Y_{2i+2j+2}$, $Y_{2i+2,2j-2}$, $Y_{2i+2,2j}$, and $Y_{2i+2,2j+2}$ (see Fig. 2 for the pixel locations). Hence, the covariance mismatch problem can be mitigated at the cost of computational complexity. The pseudocode of MEDI is shown in Sec. 2.1.1. Similar to the NEDI method, the hybrid framework is applied in the proposed method, where the pixels at edge regions are interpolated by the covariance-based method, and the pixels at smooth regions are interpolated by bilinear interpolation. If the variance of the pixels in the local block is larger than ϵ, the unknown pixel is regarded to be part of an edge, thus the covariance-based method is applied.

2.1.1 *Algorithm 2.1: MEDI* (X)

set

$Y_{2i,2j} = X_{i,j}$

comment: Begin of the first step of the MEDI method, which is identical to that of the NEDI method.

for $i = 1;2;2H$

for $i = 1{:}2{:}2W$
- **comment**: The energy of four 5×5 training windows are computed.
- **comment**: All the training windows have the structure as shown in Fig. 1(a)
- C = the training window with the maximum energy
- $R = C^T C$;
- $r = [r_0; r_1; r_2; r_3]$
- **if rank**$(R) = = 4$ and var$(r) > \epsilon$
  - **then** $\alpha = R^{-1} r$;
  - **else** $\alpha = [1/4; 1/4; 1/4; 1/4]$;
- $y = [Y_{2i,2j}; Y_{2i,2j+2}; Y_{2i+2,2j+2}; Y_{2i+2,2j}]$;
- $Y_{2i+1,2j+1} = \alpha^T y$

comment: End of the first step.

comment: The second step of the MEDI method.

for $j = 1{:}2{:}2H$

for $i = 1{:}2{:}2W$
- **comment**: The energy of six 5×9 training windows are computed.
- **comment**: All the training windows have the structure as shown in Fig. 2.
- C = the training window with the maximum energy
- $R = C^T C$;
- $r = [r_0; r_1; r_2; r_3; r_4; r_5]$;
- if rank$(R) = = 6$ and var$(r) > \epsilon$
  - **then**
    - $\alpha = R^{-1} r$;
    - $y = [Y_{2i-2,2j-2}; Y_{2i,2j}; Y_{2i,2j+2}; \cdots; Y_{2i+2,2j-2}]$;
  - **else**
    - $\alpha = [1/4; 1/4; 1/4; 1/4]$;
    - $y = [Y_{2i,2j}; Y_{2i+1,2j+1}; Y_{2i+2,2j}; Y_{2i+2,2j-1}]$;
- $Y_{2i+1,2j} = \alpha^T y$;

comment: End of the second step.

comment: Repeat the second for updating $Y_{2i,2j+1}$.

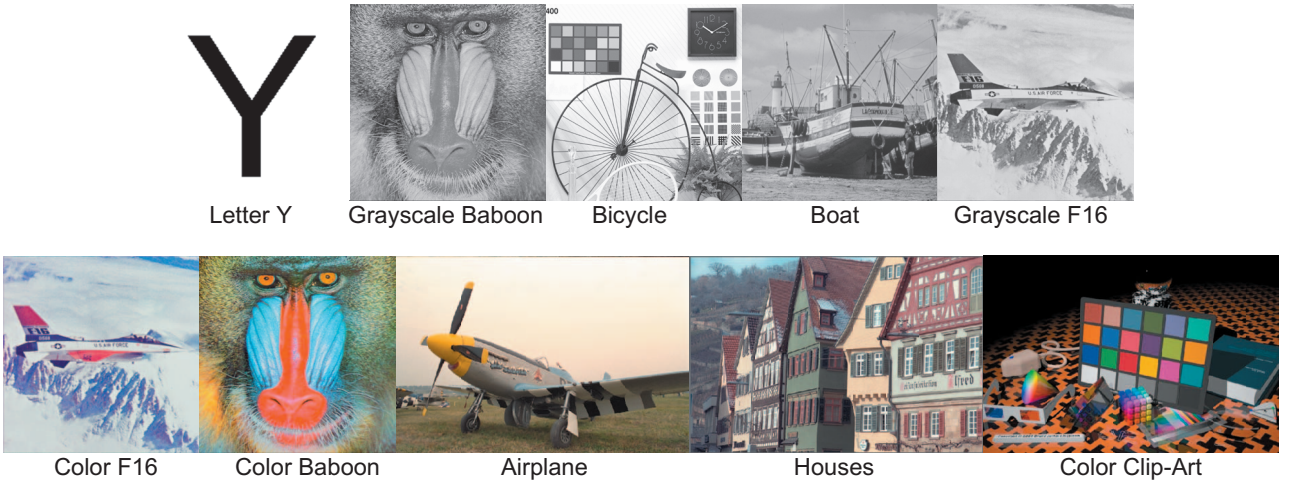

Fig. 4 Test images.

3 Results and Discussion

The proposed algorithm has been compared with other interpolation algorithms in the literature, including bilinear interpolation, the NEDI method,[13] the IEDI method,[15] the iNEDI method,[16] the ICBI method,[18] and the Lanczos and B-spline methods.[21] Subjective and objective comparisons have been performed. The proposed algorithm was implemented in Matlab running on a PC with Intel Pentinum(R) Duo Core 3-GHz CPU and 1-GB DDR RAM. For comparison purposes, the IEDI method is implemented in Matlab without heat diffusion refinement. This is because our investigation mainly focused on the covariance mismatch problem, while heat diffusion refinement is a postprocessing step that does not affect the performance of the covariance-based interpolation method. For bilinear interpolation and Lanczos interpolation, the built-in functions in Matlab were applied in our simulations. For the rest of the interpolation methods, a Matlab source code available on other websites were used.[22–25] The default function parameters of iNEDI and ICBI were applied. The threshold was selected to be ϵ=48 for the MEDI, NEDI, and IEDI methods. The interpolation of the image boundaries was achieved by zero extension. Both synthetic and natural images were tested with different methods. The complete simulation results can be found at http://sites.google.com/site/medidemosite/.

3.1 *Objective Test*

Figure 4 shows the original test images used in the simulations that include both synthetic and natural images. The original test image was first downsampled by a factor of two, that is, from $2H\times 2W$ to $H\times W$. The downsampled images were then expanded to their original sizes by using different interpolation methods. Both direct and average downsampling images were tested. The interpolated images were compared with the original images objectively by measuring the PSNR and the structural similarity index (SSIM).[26] To characterize the error aroused along the image

Table 1 The PSNR, SSIM, and EPSNR of the interpolated images of Letter Y by different interpolation methods.

	Direct downsampling			Average downsampling		
Method	PSNR	SSIM	EPSNR	PSNR	SSIM	EPSNR
MEDI	**22.3807**	0.93271	23.9527	21.8508	0.93221	23.6096
Bilinear	19.3352	0.8745	21.1535	21.939	0.93188	23.7631
NEDI[13]	22.1079	**0.93532**	23.7954	**22.52**	**0.94337**	23.9569
IEDI[15]	20.172	0.88642	**24.1085**	19.9498	0.88269	23.7207
iNEDI[16]	21.2478	0.89537	23.7814	19.9005	0.87583	23.9314
ICBI[18]	19.9623	0.88219	23.7499	20.3081	0.89943	24.1703
Lanczos	19.3242	0.88019	20.9153	19.2655	0.86445	20.8724
B-spline	20.7921	0.83192	23.3555	19.8705	0.81623	**24.2178**

Table 2 The PSNR of interpolated grayscale images by different interpolation methods.

Direct downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Grayscale Baboon	256×256⇒512×512	22.4659	22.2674	23.2121	22.9574	**23.6442**	22.7152	21.8805	23.1127
Bicycle	256×256⇒512×512	18.9029	18.5628	**20.3339**	19.2916	20.0165	19.2561	18.2438	19.4875
Boat	256×256⇒512×512	29.2052	27.0571	**29.6856**	27.5121	29.1492	27.2931	26.8398	29.4465
Grayscale F16	256×256⇒512×512	**32.4444**	28.3414	31.4642	28.769	30.7141	28.2912	28.3929	32.1827
	Sum	103.0184	96.2287	**104.6958**	98.5301	103.524	97.5556	95.357	104.2294
	Average	25.7546	24.057175	**26.17395**	24.632525	25.881	24.3889	23.83925	26.05735

Average downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Grayscale Baboon	256×256⇒512×512	23.2391	**23.5774**	22.8932	22.745	22.8876	22.9102	21.5768	22.9735
Bicycle	256×256⇒512×512	20.4133	**20.4369**	20.0786	19.3137	19.3955	19.5172	17.9836	19.2229
Boat	256×256⇒512×512	29.7456	**29.8099**	29.697	27.4173	27.3921	27.4613	26.5783	27.4958
Grayscale F16	256×256⇒512×512	31.4558	31.4026	**31.958**	28.3813	28.2886	28.4674	28.2627	28.6384
	Sum	104.8538	**105.2268**	104.6268	97.8573	97.9638	98.3561	94.4014	98.3306
	Average	26.21345	**26.3067**	26.1567	24.464325	24.49095	24.589025	23.60035	24.58265

Table 3 The SSIM of interpolated grayscale images by different interpolation methods.

Direct downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Grayscale Baboon	256×256⇒512×512	**0.71384**	0.63208	0.71231	0.67782	0.68594	0.64392	0.64818	0.71652
Bicycle	256×256⇒512×512	0.72795	0.68452	**0.77898**	0.72698	0.72942	0.72134	0.69109	0.72736
Boat	256×256⇒512×512	0.88275	0.83565	**0.89106**	0.85665	0.87552	0.84746	0.83658	0.88271
Grayscale F16	256×256⇒512×512	**0.9411**	0.89548	0.9326	0.90851	0.92332	0.89706	0.90016	0.93956
	Sum	3.26564	3.04773	**3.31495**	3.16996	3.2142	3.10978	3.07601	3.26615
	Average	0.81641	0.7619325	**0.8287375**	0.79249	0.80355	0.777445	0.7690025	0.8165375

Average downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Grayscale Baboon	256×256⇒512×512	0.71344	**0.73605**	0.72802	0.64444	0.65415	0.67009	0.64264	0.66459
Bicycle	256×256⇒512×512	0.77684	0.77263	**0.78123**	0.72326	0.72538	0.73778	0.67892	0.70738
Boat	256×256⇒512×512	0.89151	0.89095	**0.89193**	0.84946	0.84797	0.85421	0.8305	0.85194
Grayscale F16	256×256⇒512×512	0.93265	0.93084	**0.93752**	0.89727	0.89386	0.90276	0.89782	0.90509
	Sum	3.31444	**3.33047**	3.3387	3.11443	3.12136	3.16484	3.04988	3.129
	Average	0.82861	0.8326175	**0.834675**	0.7786075	0.78034	0.79121	0.76247	0.78225

Table 4 The EPSNR of interpolated grayscale images by different interpolation methods.

Direct downsampling									
Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Grayscale Baboon	256×256⇒512×512	29.0107	29.3487	30.9658	29.4053	**31.2072**	29.5201	28.2969	30.6022
Bicycle	256×256⇒512×512	23.9567	23.8848	**26.0887**	24.1098	26.2706	24.2362	22.788	25.0678
Boat	256×256⇒512×512	35.6502	33.8593	**37.3344**	33.1877	37.7065	33.3654	32.7446	37.4664
Grayscale F16	256×256⇒512×512	38.403	34.4961	38.5836	33.8989	38.7091	34.2064	33.6057	**39.7807**
	Sum	127.0206	121.5889	**132.9725**	120.6017	133.8934	121.3281	117.4352	132.9171
	Average	31.75515	30.397225	33.243125	30.150425	**33.47335**	30.332025	29.3588	33.229275
Average downsampling									
Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Grayscale Baboon	256×256⇒512×512	30.9356	**31.1533**	30.4391	29.5384	29.6373	29.5284	27.9232	29.8853
Bicycle	256×256⇒512×512	**26.1789**	25.8765	25.9948	24.2816	24.2618	24.3602	22.4432	24.9405
Boat	256×256⇒512×512	37.5061	**37.7681**	37.3929	33.5122	33.6456	33.2453	32.4332	34.1252
Grayscale F16	256×256⇒512×512	38.5663	**38.7733**	38.6291	34.3699	34.5264	33.7316	33.331	34.9557
	Sum	133.1869	**133.5712**	132.4559	121.7021	122.0711	120.8655	116.1306	123.9067
	Average	33.296725	**33.3928**	33.113975	30.425525	30.517775	30.216375	29.03265	30.976675

edges, the PSNR focused on image edges was measured, and this figure is denoted as edge PSNR (EPSNR). Numerous research focused on the metrics to characterize the error aroused along image edges.[27,28] In our study, the Sobel edge filter is used to locate the edge in the original image, and the PSNR of the pixels on the edge were used to generate the EPSNR. The PSNR, SSIM, and EPSNR of all the test images are summarized in Tables 1–7. PSNR has been widely used to measure the distortion of the grayscale images after processing and is given by

$$\mathrm{PSNR} = 20 \log_{10}\left(\frac{255}{\sqrt{\mathrm{MSE}}}\right), \tag{4}$$

$$\mathrm{MSE} = \frac{1}{2H \times 2W} \sum_{i=0}^{2H-1} \sum_{j=0}^{2W-1} Z_{i,j}^2, \tag{5}$$

$$Z_{i,j} = |L_{i,j} - Y_{i,j}|, \tag{6}$$

where $L_{i,j}$ and $Y_{i,j}$ are the pixels in the original image and the interpolated image at location (i,j), respectively. For color images in RGB representation, each channel is treated independently as a grayscale image. The interpolated images of the three channels are then recombined to give the final image for comparison. Thus, the PSNR is computed as

$$\overline{\mathrm{PSNR}} = (\mathrm{PSNR}_{\mathrm{red}} + \mathrm{PSNR}_{\mathrm{green}} + \mathrm{PSNR}_{\mathrm{blue}})/3, \tag{7}$$

where $\mathrm{PSNR}_{\mathrm{red}}$, $\mathrm{PSNR}_{\mathrm{green}}$, and $\mathrm{PSNR}_{\mathrm{blue}}$ are the PSNR values for the red, green, and blue channels of the color images computed with Eq. (4), respectively. In the following discussion, we abuse the notation PSNR to imply both PSNR and $\overline{\mathrm{PSNR}}$ with respect to the grayscale and color images in concern. High PSNR value of the interpolated images is more favorable, because this implies less distortion. Similar to the computation of PSNR, the EPSNR can be computed using Eqs. (4)–(7). However, only the edge pixels are computed. The edge pixels are located by using the edge map extracted from the original image by Sobel filtering, in which the filter was implemented by the built-in Matlab function. Similarly, the higher the EPSNR, the less distortion is observed on the image edges. Another objective measurement is the SSIM. SSIM is an index characterized by the structural similarity of the original image with the consideration of human visual perception. A SSIM Matlab program downloaded from Ref. 29 was used for SSIM computation. The higher SSIM value indicates that there is greater structural similarity between the original and interpolated images.

The PSNR, SSIM, and EPSNR of the synthetic image "letter Y" are summarized in Table 1. The objective performance of different methods is subject to the downsampling methods. It can be observed that none of the methods show consistently good performance for both downsampling

Table 5 The SSIM of interpolated color images by different interpolation methods.

Direct downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Color Baboon	256×256⇒512×512	21.7184	21.5875	22.4909	22.2508	**22.9795**	22.0173	21.1844	22.3983
Color F16	256×256⇒512×512	**32.1732**	28.4878	31.3585	28.9347	30.7862	28.486	28.5039	31.9925
Houses	256×384⇒512×768	21.9021	21.2115	22.1569	21.5191	**22.9097**	21.3791	20.9029	N/A
Airplane	256×384⇒512×768	30.993	29.0335	**31.3551**	29.4564	30.6967	29.2942	28.8167	N/A
Clip-art	350×233⇒700×466	30.3354	27.543	**30.4736**	28.0549	29.6804	27.8683	27.4148	N/A
	Sum	137.1221	127.8633	**137.835**	130.2159	137.0525	129.0449	126.8227	N/A
	Average	27.42442	25.57266	**27.567**	26.04318	27.4105	25.80898	25.36454	27.1954

Average downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Color Baboon	256×256⇒512×512	22.517	**22.8642**	22.2073	22.0409	22.1847	22.2107	20.8752	22.2748
Color F16	256×256⇒512×512	31.3667	31.3507	**31.7637**	28.5696	28.4844	28.653	28.3593	28.8168
Houses	256×384⇒512×768	22.1414	**22.5025**	22.3424	21.3886	21.498	21.5681	20.662	N/A
Airplane	256×384⇒512×768	31.5177	**31.8161**	31.6655	29.3877	29.4334	29.4888	28.5579	N/A
Clip-art	350×233⇒700×466	30.4358	30.3967	**30.5432**	27.8841	27.8393	27.9774	27.2433	N/A
	Sum	137.9786	**138.9302**	138.5221	129.2709	129.4398	129.898	125.6977	N/A
	Average	27.59572	**27.78604**	27.70442	25.85418	25.88796	25.9796	25.13954	25.5458

cases. The proposed method achieves the highest PSNR and the third highest EPSNR in the direct downsampling case, but it only achieves the third highest PSNR and the sixth highest EPSNR in the average downsampling case. However, the proposed method is able to achieve the second highest SSIM in both cases. Moreover, it can be observed that the optimal statistical methods, including the NEDI and our proposed method, preserve the image structure well in both cases, thus leading to the first two highest SSIM.

Besides the synthetic image, the performance of different interpolation methods was compared with the use of natural grayscale and color images. The results are summarized in Tables 2–7. Interpolation is a reverse process of downsampling. A good match of the interpolation method to the downsampling method would bring the image distortion to minimum, thus leading to a better objective performance. Therefore, the methods that perform well in the direct downsampling case would not present the same performance in the average downsampling case. Shown in Tables 2, 4, 5, and 7, though the bilinear method shows comparatively worse PSNR and EPSNR in the direct downsampling case, it achieves the best PSNR and EPSNR for almost all average downsampled test images. Moreover, though the statistical optimal methods (the NEDI, MEDI, and IEDI methods) are not able to achieve consistent performance in both downsampling cases, they always result in higher SSIM values. This is because these statistical optimal methods predict the unknown pixel adapting to the image covariance structure. Besides the SSIM performance, the NEDI and MEDI methods result in the highest PSNR values for direct downsampled images. However, due to the high contrast of the edges, the iNEDI method shows the best EPSNR performance for the direct downsampled images. Interestingly, the objective performance is highly correlated to the image structure. For example, for images rich in texture, including Grayscale Baboon, Color Baboon, and Houses, the iNEDI method results in better PSNR and EPSNR. Nevertheless, for images containing mainly long edges with low contrast, e.g., Grayscale F16 and Color F16, the statistical optimal methods result in better performance in PSNR, SSIM, and EPSNR, no matter which downsampling method has been adopted. Therefore, it is difficult to tell which one is the winner. However, it can be concluded that the proposed method shows fair objective performance among all methods.

Edge information is image specific, and the EDI methods under test do not compute the missing pixels in the smooth regions and those along the edges in the same manner all the time. Moreover, each EDI method adopts a dif-

Table 6 The EPSNR of interpolated color images by different interpolation methods.

Direct downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Color Baboon	256×256⇒512×512	0.69537	0.61765	0.6949	0.66423	0.67229	0.62949	0.63301	**0.69889**
Color F16	256×256⇒512×512	**0.9239**	0.88084	0.91791	0.89579	0.91004	0.88444	0.8839	0.92273
Houses	256×384⇒512×768	**0.74894**	0.67964	0.73554	0.70215	0.73661	0.67728	0.68957	N/A
Airplane	256×384⇒512×768	0.90415	0.87839	**0.91047**	0.8913	0.89889	0.88651	0.8771	N/A
Clip-art	350×233⇒700×466	**0.92483**	0.8771	0.92448	0.89035	0.90674	0.88225	0.88144	N/A
	Sum	**4.19719**	3.93362	4.1833	4.04382	4.12457	3.95997	3.96502	N/A
	Average								

Average downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Color Baboon	256×256⇒512×512	0.69614	**0.71993**	0.71221	0.62969	0.64023	0.65669	0.62698	0.65069
Color F16	256×256⇒512×512	0.91794	0.91673	**0.92163**	0.8845	0.88137	0.88975	0.88058	0.89235
Houses	256×384⇒512×768	0.73435	0.75591	**0.75973**	0.67686	0.68303	0.69902	0.68169	N/A
Airplane	256×384⇒512×768	0.91157	**0.91534**	0.91285	0.88746	0.88841	0.89115	0.87053	N/A
Clip-art	350×233⇒700×466	0.92393	0.92277	**0.92747**	0.88199	0.88089	0.88801	0.87764	N/A
	Sum	4.18393	4.23068	**4.23389**	3.9605	3.97393	4.02462	3.93742	N/A
	Average	0.836786	0.846136	**0.846778**	0.7921	0.794786	0.804924	0.787484	0.77152

ferent method to identify edge pixels. The iNEDI method determines the edge pixel similar to other covariance-based methods (NEDI, MEDI, and IEDI); however, variable-sized training windows are adopted, which depend on the edge structures, and thus the number of operations vary among different pixels. The ICBI method does not identify the edge pixels explicitly, but directs the interpolations of the missing pixel according to the edge structure. As a result, the required number of operations will still vary from pixel to pixel, and it is difficult if not impossible to distinguish the computational effort for edge detection and interpolation. Hence, it is difficult to compare computational complexity in terms of number of operations per pixel for each interpolation method. Instead, the total computational time for each image interpolation experiment can be used to correlate the computational complexity of different methods, as all the simulation is performed on the same platform. Comparison has been focused on the EDI methods. The number of edge pixels identified by each EDI method for each image, and the computational time used by each method to interpolate each image in both downsampling cases, are summarized in Tables 8 and 9, respectively. It can be observed from Table 8 that the number of edge pixels from an average downsampled image identified by each EDI method is always smaller than that from direct downsampled images. However, longer time is required to interpolate the images obtained from average downsampling than that for the direct downsampled counterparts for each EDI method. As a result, it can be concluded that the computational complexity of EDI methods does depend on both the number of edge pixels in an image and also the correlation structure. Therefore, simply comparing the computational time for edge pixels for EDI methods is misleading, and it is more suitable to compare the computational complexity in terms of average computational time per pixel. Table 8 shows that the average computational time for different EDI methods follows the consistent trend for both downsampling cases. The proposed methods always achieve the second fastest computational time among all EDI methods, and are also the fastest methods when compared to the optimal statistical methods. The computational time of the proposed method can be further reduced by optimizing the source code.

3.2 *Subjective Test*

Besides the objective measurement, a subjective test was performed to evaluate the visual perception of the interpolated images. Error images [i.e., $Z_{i,j}$ in Eq. (6)] are used as an evaluation tool. To obtain a fair comparison, the magni-

Table 7 The EPSNR of interpolated color images by different interpolation methods.

Direct downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Color Baboon	256×256⇒512×512	28.0284	28.4336	29.9507	28.5202	**30.3744**	28.6316	27.3802	29.6541
Color F16	256×256⇒512×512	38.6446	34.9111	38.8584	34.3417	**39.0994**	34.6619	34.0178	40.0112
Houses	256×384⇒512×768	27.5147	27.2156	28.736	26.8039	**29.7968**	27.0301	26.2002	N/A
Airplane	256×384⇒512×768	37.1493	35.4617	**38.5651**	34.8022	38.4843	34.9867	34.4304	N/A
Clip-art	350×233⇒700×466	35.9035	33.3133	**36.9225**	33.1733	36.8886	33.4708	32.4743	N/A
	Sum	167.2405	159.3353	173.0327	157.6413	**174.6435**	158.7811	154.5029	N/A
	Average	33.4481	31.86706	34.60654	31.52826	**34.9287**	31.75622	30.90058	34.83265

Average downsampling

Image	Resolution	MEDI	Bilinear	NEDI[13]	IEDI[15]	iNEDI[16]	ICBI[18]	Lanczos	B-spline
Color Baboon	256×256⇒512×512	29.9595	**30.1837**	29.4468	28.652	28.7599	28.6314	27.0022	29.0262
Color F16	256×256⇒512×512	38.9	**39.1254**	38.941	34.8138	34.9609	34.2051	33.7451	35.3802
Houses	256×384⇒512×768	28.6934	**29.0178**	28.7507	27.0528	27.2206	27.044	25.9211	N/A
Airplane	256×384⇒512×768	38.9656	**39.1958**	38.9874	35.1066	35.2085	34.9665	34.1216	N/A
Clip-art	350×233⇒700×466	36.9597	**37.0404**	36.7639	33.5427	33.6125	33.2301	32.2431	N/A
	Sum	173.4782	**174.5631**	172.8898	159.1679	159.7624	158.0771	153.0331	N/A
	Average	34.69564	**34.91262**	34.57796	31.83358	31.95248	31.61542	30.60662	32.2032

tude of the pixels of the error images are normalized with the same normalization factor among all the interpolation methods, and thus not all error images have their pixel values span from 0 to 255. The normalization performed on the differences among the error images has made it more vivid. For the color image case, the error image of each channel is recombined to give the final error images. Therefore, the distortion on each channel is represented by the corresponding color in the final images. Figure 5 shows the original image, interpolated images, and the error images of test image Letter Y for both downsampling methods. We first consider the direct downsampling case. It is observed that the MEDI interpolated image is perceptually more pleasant among all the interpolated images because of the continuous and smooth diagonal edges. It is more vivid by observing the error images. The white area in the error images indicates the distortion. The brighter the white region, the more the distortion is concentrated. It is observed that the white region in the bilinear, the Lancozs, and the B-spline interpolated images are concentrated along the edges, which is the consequence of blurring after interpolation. The white region is comparatively less obvious in the error images of the iNEDI and ICBI methods. The white region is dispersed in the NEDI case because the edges are interpolated by covariance matching, thus minimizing the error along edges. The white region is even more dispersed in the IEDI case, especially along the diagonal edges, because the IEDI method fully utilizes the low-resolution pixels with an enlarged training window. For the MEDI case, the white region is observed to be even dimmer and segmented along the diagonal edges, because the proposed method accurately adapts the edge orientation by covariance matching in multiple directions. A similar observation is obtained from the average downsampling case, but the error is more significant.

Figures 6 and 7 show the pixel intensity maps of the original and interpolated images of region A in Fig. 5 for direct downsampling and average downsampling cases, respectively. We first consider the direct downsampling case. There is a sharp transition from 0 to 255 across the vertical edge of the original image in region A, as shown in Fig. 6. All vertical edges are blurred after interpolation, and the effect is the least significant for the iNEDI interpolated image, where the transition spanned three columns only. The blurring effect is the most vivid for the bilinear, Lanczos, and B-spline interpolated images. The halo effect is observed in the ICBI interpolated image. The interpolation performance observed from the proposed method, the NEDI method, and the IEDI method are compatible because these methods use the same training window struc-

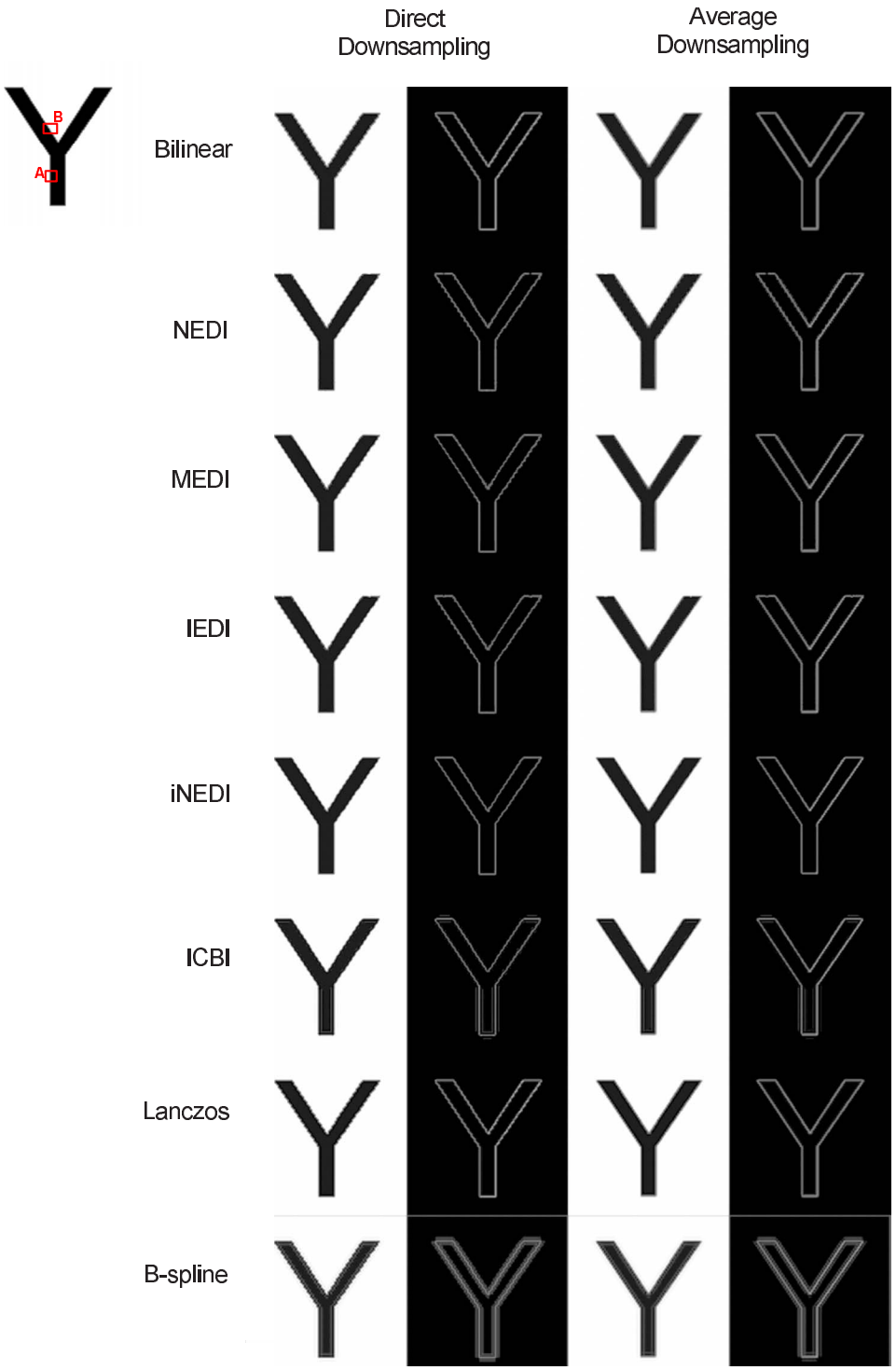

Fig. 5 Original image, interpolated images, and error images of Letter Y (resolution enhancement from 100×100 to 200×200).

ture. Furthermore, the covariance structure is identical in all cases, because it is a perfect vertical edge in the synthetic image. A consistent result can be observed in the average downsampling case, as shown in Fig. 7. The outstanding performance of the proposed method is emphasized in the study of the intensity maps for region B, as shown in Figs. 8 and 9 which contain a diagonal edge, for direct downsampling and average downsampling cases, respectively. The interpolated edge obtained from the bilinear, Lanczos, and B-spline methods are the most blurred.

The halo effect is observed in the ICBI interpolated image. It is observed that the IEDI method achieves sharper diagonal edges than that of the NEDI method, because a modified training window is applied in the second step of

Table 8 The number of edge pixels considered in different edge-directed interpolation methods.

Direct downsampling					
	MEDI	NEDI	IEDI	iNEDI	ICBI
Letter Y	923	1017	984	1312	40,000
Grayscale Baboon	101,337	115,783	101,337	172,136	262,144
Bicycle	69,992	88,165	70,036	113,990	262,144
Boat	56,717	72,527	56,717	95,471	262,144
Grayscale F16	46,963	63,968	46,963	72,163	262,144
Average downsampling					
	MEDI	NEDI	IEDI	iNEDI	ICBI
Letter Y	612	566	716	954	40,000
Grayscale Baboon	90,051	107,577	90,051	156,006	262,144
Bicycle	57,640	71,594	57,665	90,465	262,144
Boat	52,338	64,847	52,338	86,929	262,144
Grayscale F16	44,570	56,925	44,570	66,344	262,144

Table 9 The computation time per pixel of different edge-directed interpolation methods.

Direct downsampling						
Image	Total number of interpolated pixels	MEDI (sec)	NEDI[13] (sec)	IEDI[15] (sec)	iNEDI[16] (sec)	ICBI[18] (sec)
Letter Y	$3\times100\times100$	1.33E−04	2.60E−03	1.33E−04	<1.00E−6	1.17E−03
Grayscale Baboon	$3\times256\times256$	8.65E−05	1.81E−03	1.93E−04	<1.00E−6	1.91E−03
Bicycle	$3\times256\times256$	2.29E−04	4.77E−03	2.19E−04	<1.00E−6	4.57E−03
Boat	$3\times256\times256$	1.98E−04	4.57E−03	1.48E−04	5.09E−06	4.57E−03
Grayscale F16	$3\times256\times256$	1.73E−04	3.89E−03	1.58E−04	<1.00E−6	3.87E−03
	Average	1.64E−04	3.53E−03	1.70E−04	1.02E−06	3.22E−03
Average downsampling						
Image	Total number of interpolated pixels	MEDI (sec)	NEDI[13] (sec)	IEDI[15] (sec)	iNEDI[16] (sec)	ICBI[18] (sec)
Letter Y	$3\times100\times100$	2.33E−04	6.00E−04	2.33E−04	2.00E−04	7.00E−04
Grayscale Baboon	$3\times256\times256$	3.46E−04	6.97E−03	9.05E−04	2.85E−04	9.49E−03
Bicycle	$3\times256\times256$	8.39E−04	7.12E−03	8.29E−04	7.17E−04	5.45E−03
Boat	$3\times256\times256$	8.49E−04	6.01E−03	7.07E−04	7.38E−04	4.83E−03
Grayscale F16	$3\times256\times256$	7.02E−04	3.80E−03	6.97E−04	6.00E−04	3.48E−03
	Average	5.94E−04	4.90E−03	6.74E−04	5.08E−04	4.79E−03

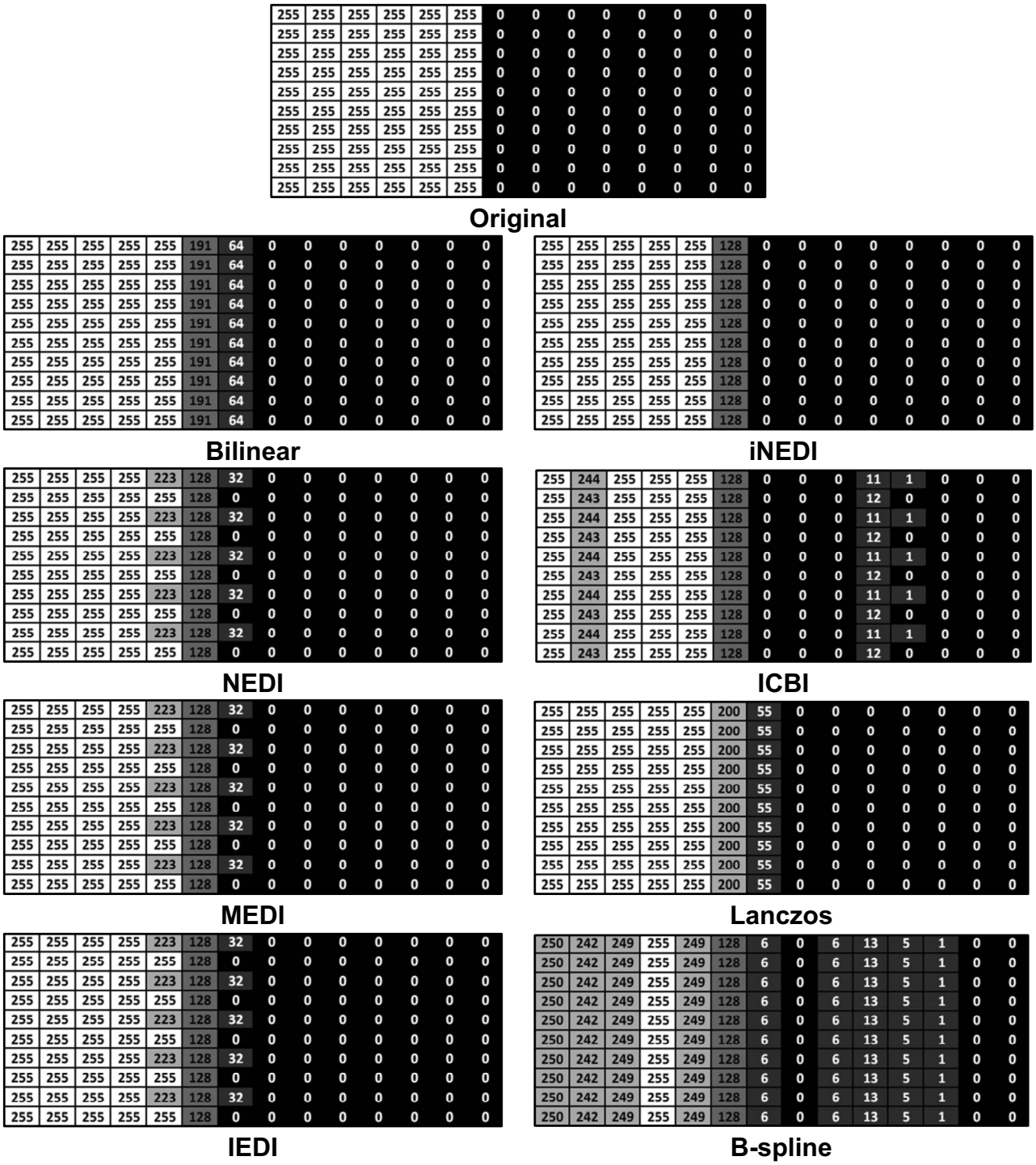

Fig. 6 Pixel intensity maps of the original image and interpolated images of Letter Y in region A for the direct downsampling case.

the IEDI method, which fully utilizes information from the original image. The iNEDI method results in sharp and smooth edges, but the edge continuity is not close to that of the original image. The proposed method does not only form sharp and smooth edges, the interpolated edge structure is highly close to the original edge. The outstanding performance is due to the termination of prediction error propagation and the elimination of covariance mismatch. Average downsampling is able to preserve the visual quality of the downsampled image; however, the image edges are smoothed out by the averaging filter. The filtering approaches, e.g., the bilinear, Lanczos, and B-spline methods, are favorable to the reconstruction of the smoothed image. However, the computational complexity of EDI methods is inevitably increased due to the difficulty in locating the image edges. As shown in Fig. 9 the distortion is more server in restored average downsampling images, no matter which interpolation methods are adopted. Furthermore, the

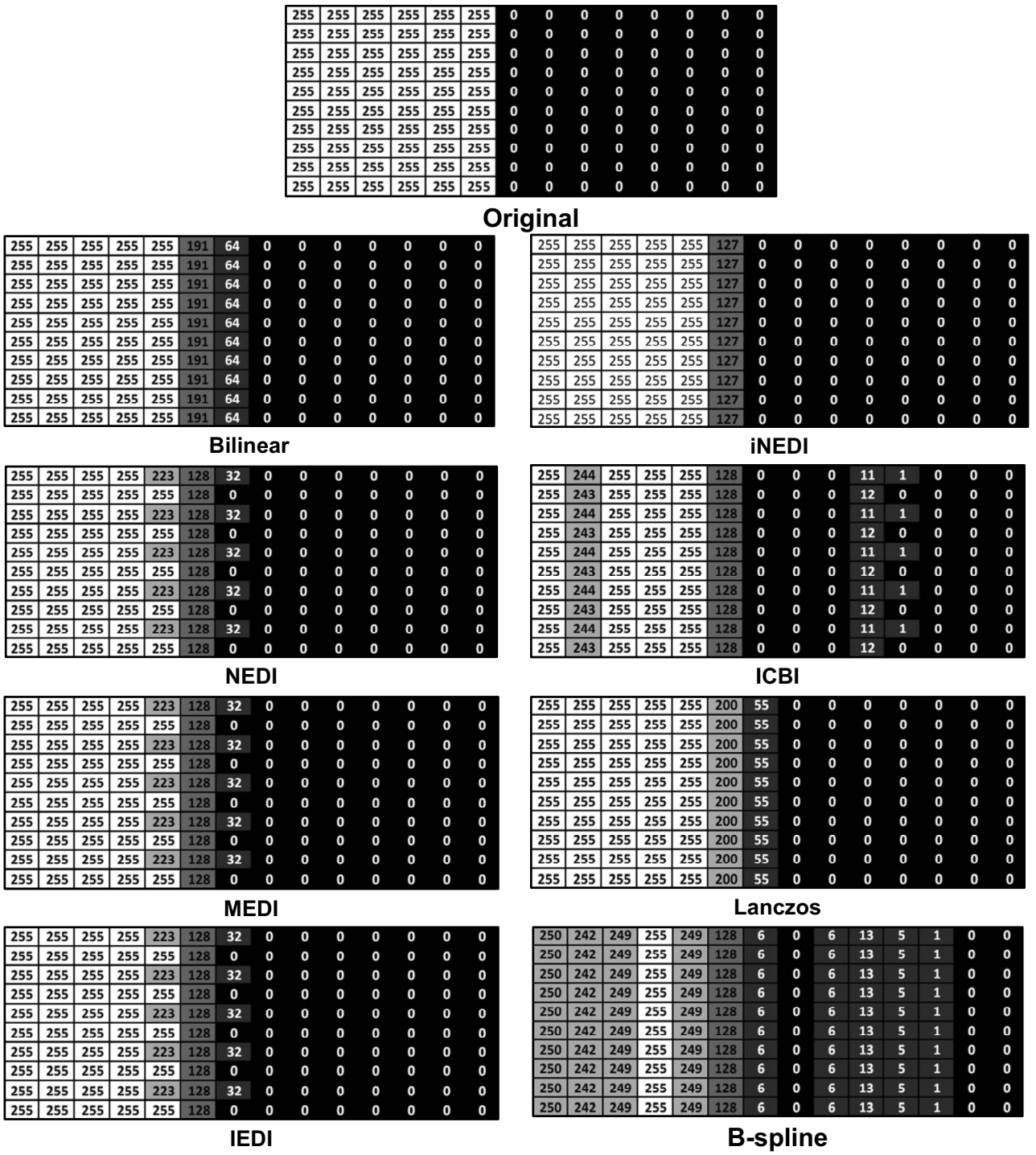

Fig. 7 Pixel intensity maps of the original and interpolated of Letter Y in region A for the average downsampling case.

original pixel intensity cannot be reverted after average downsampling. Therefore, the objective comparison, including PSNR, may be misleading. As a result, it is more efficient to compare the performance of different methods by using the direct downsampled images.

Figure 10 shows the simulation results for the test image Bicycle. Part of the original image is zoomed-in and the corresponding portions of the interpolated images are also shown. Considering the circled beam on the bicycle wheel, the proposed method and the IEDI method show the most outstanding performance in preserving the continuity, smoothness, and sharpness of the interpolated edge. In particular, the proposed method further preserves the image structure, even at edge termination (enclosed with rectangular boxes in the IEDI and MEDI images in Fig. 10), where the IEDI interpolated image shows discontinuity at the end of the beam that should connect to the wheel, while the interpolated image of the proposed method shows al-

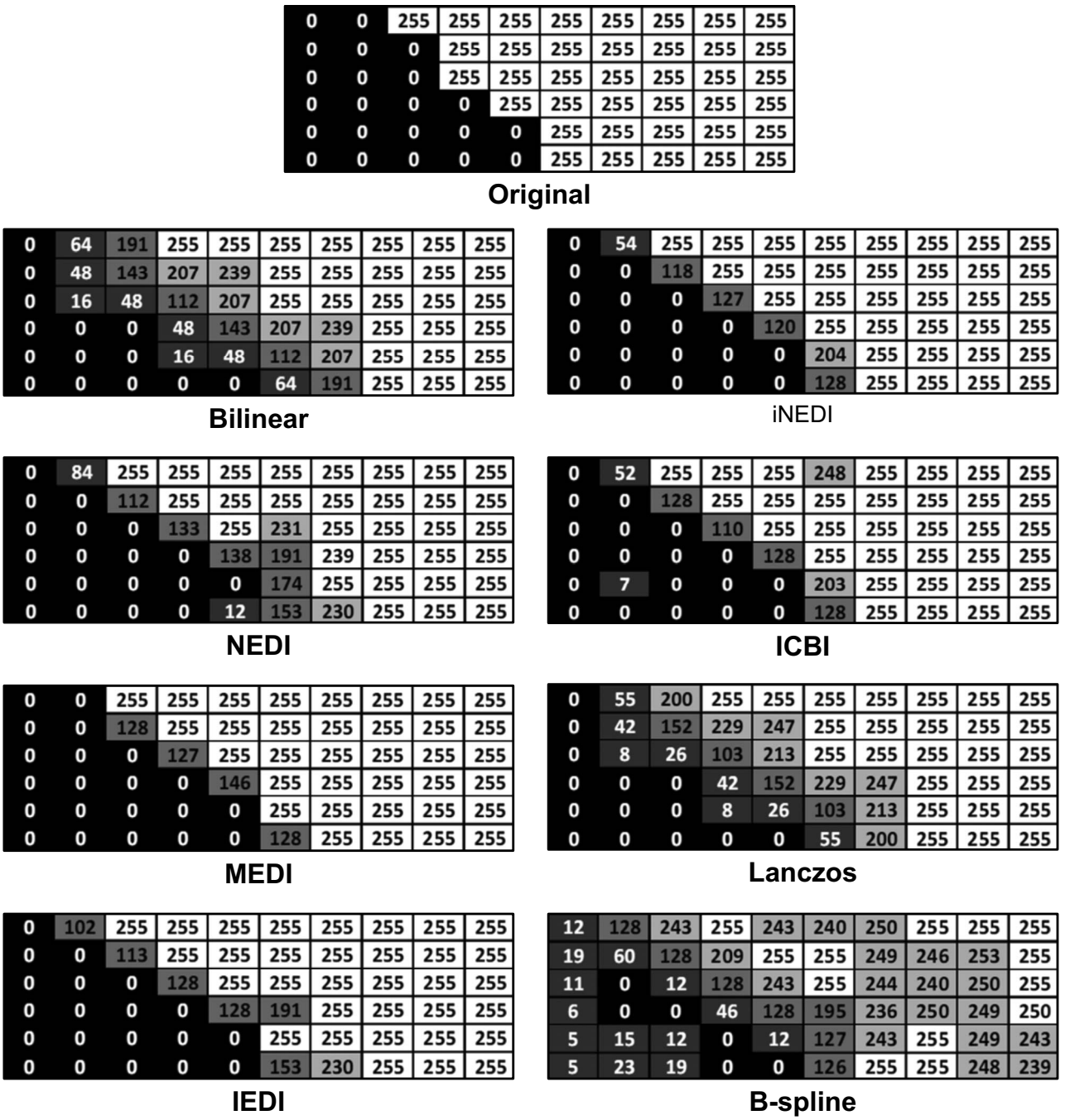

Fig. 8 Pixel intensity maps of the original and interpolated images of Letter Y in region B for the direct downsampling case.

most the same image quality as the original image. This verifies that the proposed method is effective in eliminating the covariance mismatch problem. Therefore, though both the proposed method and the IEDI method show average objective performance in different images, the proposed method outperforms the IEDI method in preserving image structure. Hence, the following comparison focuses on the NEDI method and also the iNEDI method because of its outstanding performance in the synthetic image case.

Figure 11 shows the simulation results for the test image Grayscale Baboon. Grayscale Baboon is rich in texture (the hairs near the nose) and contains lots of low contrast edges (the whiskers). It is observed that the MEDI method outperforms the other methods in preserving the edge continuity and sharpness of the whiskers, independent to the pixel intensity level. It is due to the suppression of covariance mismatch and the termination of prediction error propagation with the enlarged training windows in the second step. The MEDI method preserves the continuity of the whiskers when compared to those of the NEDI and iNEDI methods. The enlarged training window in the second step of the MEDI method reduces the efficiency in detecting short edges or texture. However, the hairs interpolated by the MEDI method are perceptually comparable with those of

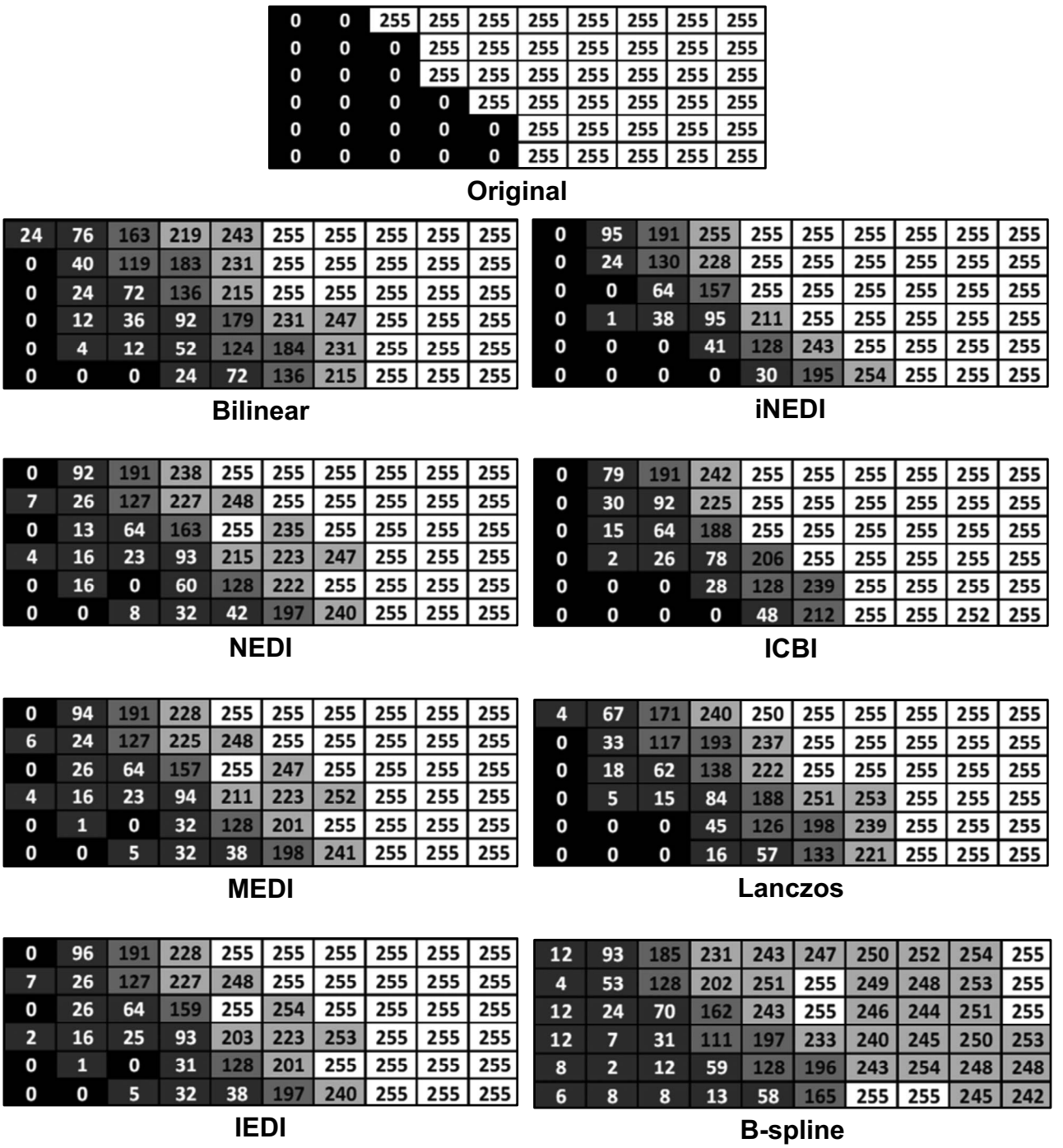

Fig. 9 Pixel intensity maps of the original and interpolated images of Letter Y in region B for the average downsampling case.

the NEDI and iNEDI methods. A consistent performance is observed from color images. Figure 12 shows the propeller of the original color image Airplane and the corresponding portions of interpolated images. The highlighted edge of the MEDI case is the smoothest and sharpest among that of the shown cases due to the elimination of prediction error propagation and suppression of covariance mismatch. It is more apparent in the comparison of the error images, as shown in Fig. 13. The error is the most dispersed in the MEDI interpolated images. Figure 14 shows zoomed-in portions of the interpolated images Grayscale F16 obtained by the NEDI and MEDI methods. The objective performance of the interpolated image obtained by the MEDI method is better than that of the NEDI method depicted in Tables 2 and 3. A consistent subjective performance is also observed. Consider the enclosed edges of the empennage; the MEDI method preserves the edge smoothness and sharpness. The error images further show that MEDI imposes less error along the highlighted edge when compared to that of the NEDI method, where a thinner and dimmer

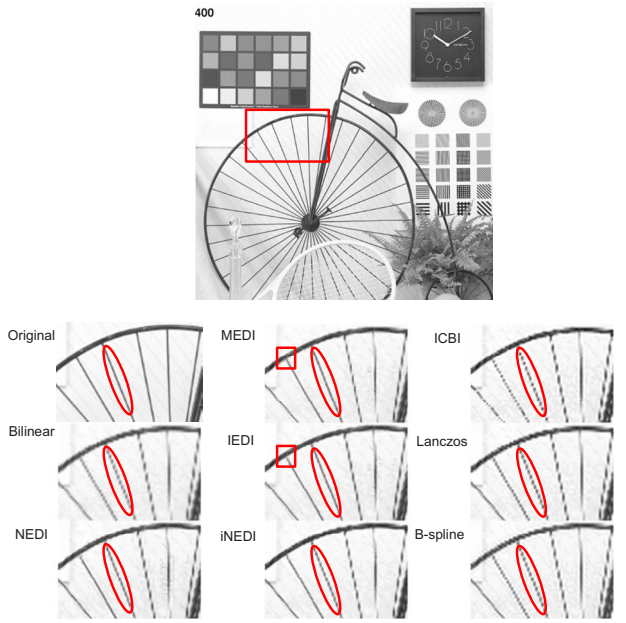

Fig. 10 Original test image Bicycle and zoomed-in portions of the original and interpolated images.

white region is observed in the MEDI error image. This observation shows that the proposed method can achieve comparable objective performance with high visual quality interpolated images, especially in preserving the edge sharpness and continuity, and also the quality of the interpolated image texture.

4 Conclusion

An improved statistical optimized interpolation method, modified edge-directed interpolation, is presented. The proposed method overcomes the existing problems of new edge-directed interpolation by considering multiple training windows and modified training window structure. The covariance mismatch problem is mitigated and the prediction error accumulation problem is eliminated. The performance of the proposed method is verified with extensive simulations and comparisons with other benchmark interpolation methods. Simulation results show that the presented method achieves outstanding perceptual performance with consistent objective performance independent of the image structure. The proposed method can be integrated to differ-

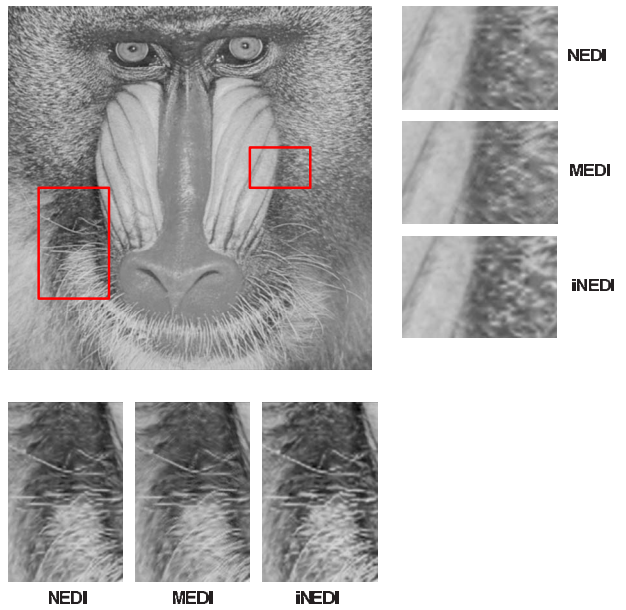

Fig. 11 Original test image Grayscale Baboon and zoomed-in portions of the original and interpolated images.

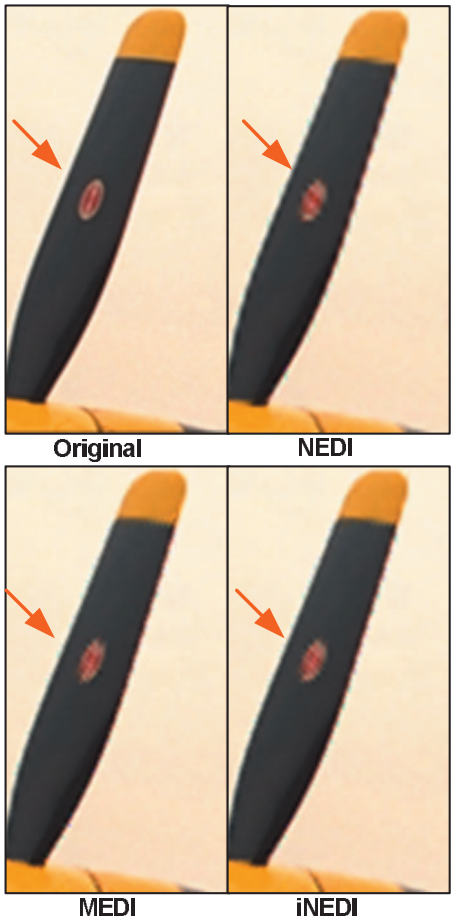

Fig. 12 Portions of the original test image Airplane and corresponding portions of the interpolated images.

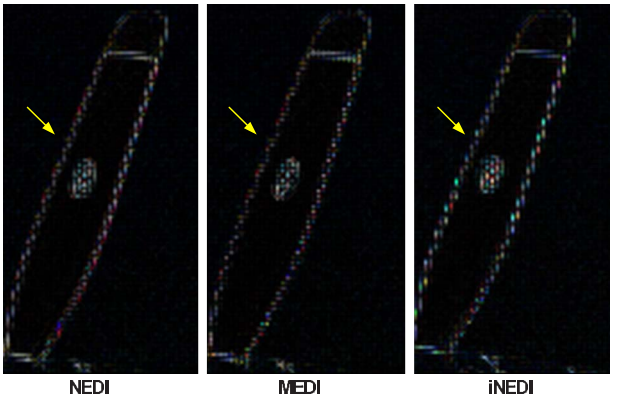

Fig. 13 The difference images of the test image Airplane for the portions shown in Fig. 12.

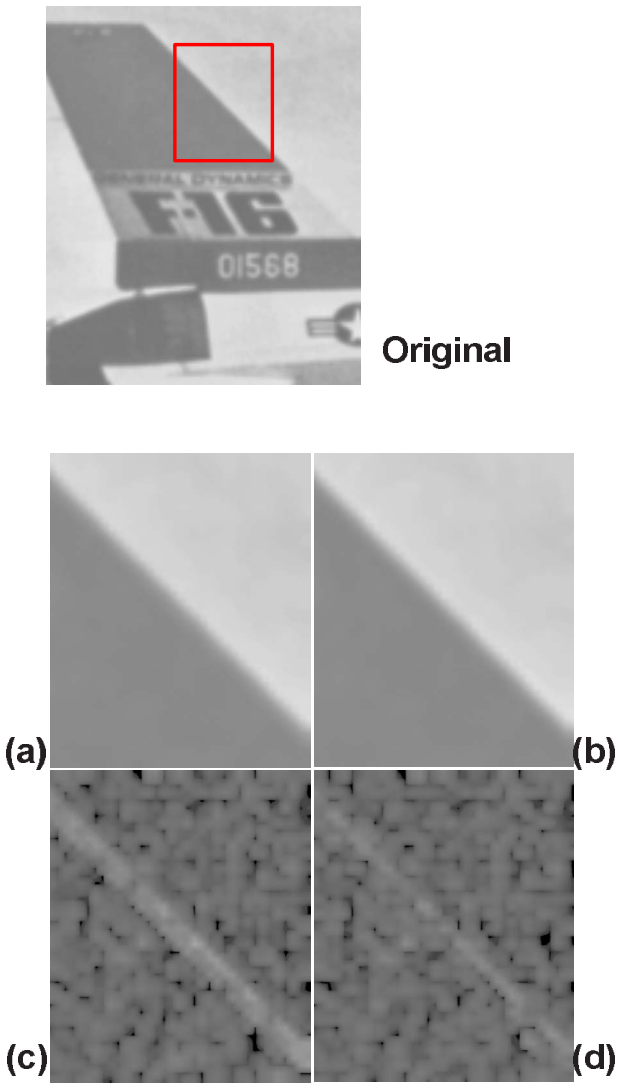

Fig. 14 Test image Grayscale F16. Zoomed-in portions of (a) the NEDI interpolated image, (b) the MEDI interpolated image, (c) the NEDI error image, and (d) the MEDI error image.

ent industrial applications, such as the presented resolution enhancement application or color CCD demosaicing.

Acknowledgments

This work was supported by the Research Grants Council of Hong Kong SAR Government under the CERG grant number PolyU5278/8E(BQ14-F).

References

1. R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander III, "Demosaicking methods for Bayer color arrays," *J. Electron. Imaging* **11**(3), 306–315 (2002).
2. J. Mukherjee, R. Parthasarathi, and S. Goyal, "Markov random field processing for color demosaicing," *Pattern Recogn. Lett.* **22**(3–4), 339–351 (2001).
3. S. W. Jeng and W. H. Tsai, "Improving quality of unwraped omni-images with irregularly-distributed unfilled pixels by a new edge-preserving interpolation technique," *Pattern Recogn. Lett.* **28**(15), 1926–1936 (2007).
4. J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolating methods for image resampling," *IEEE Trans. Image Process.* **2**, 31–39 (1983).
5. K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Process.* **4**, 285–295 (1995).
6. J. Allebach and P. W. Wong, "Edge-directed interpolation," *Proc. Intl. Conf. on Image Processing*, Vol. 2, pp. 707–710, IEEE, Piscataway, NJ (1996).
7. B. S. Morse and D. Schwartzwald, "Isophote-based interpolation," *Proc. Intl. Conf. on Image Processing*, Vol. 3, pp. 227–231, IEEE, Piscataway, NJ (1998).
8. V. R. Alagazi, G. E. Ford, and R. Potharlanka, "Directional interpolation of images based on visual properties and rank order filtering," *Proc. Intl. Conf. on Image Processing*, Vol. 4, pp. 3005–3008, IEEE, Piscataway, NJ (1991).
9. Q. Wang and R. Ward, "A new edge-directed image expansion scheme," *Proc. Intl. Conf. on Image Processing*, Vol. 1, pp. 899–902, IEEE, Piscataway, NJ (2001).
10. S. Battiato, G. Gallo, and F. Stanco, "A locally adaptive zooming algorithm for digital images," *Image Vis. Comput.* **20**, 805–812 (2002).
11. S. Battiato, F. Rundo, and F. Stanco, "ALZ: adaptive learning for zooming digital images," *Int. Conf. on Consumer Electronics (ICCE) Tech. Dig.*, IEEE, Piscataway, NJ (2007).
12. S. Battiato, E. U. Giuffrida, and F. Rundo, "A cellular nerual network for zooming digital colour images," *Int. Conf. on Consumer Electronics (ICCE) Tech. Dig.*, IEEE, Piscataway, NJ (2008).
13. X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.* **10**, 1521–1527 (2001).
14. M. Li and T. Q. Nguyen, "Markov random field model-based edge-directed image interpolation," *IEEE Trans. Image Process.* **17**, 1121–1128 (2008).
15. X. Q. Chen, J. Zhang, and L. N. Wu, "Improvement of a nonlinear image interpolation method based on heat diffusion equation," *Proc. Intl. Conf. Mach. Learn Cybernet.*, pp. 2911–2914 (2003).
16. N. Asuni and A. Giachetti, "Accuracy improvements and artifacts removal in edge based image interpolation," *Proc. 3rd Intl. Conf. on Computer Vision Theory and Applications*, Vol. 1, pp. 58–65 (2008).
17. M. J. Chen, C. H. Huang, and W. L. Lee, "A fast edge-oriented algorithm for image interpolation," *Image Vis. Comput.* **23**, 791–798 (2005).
18. A. Giachetti and N. Asuni, "Fast artifacts-free image interpolation," *Proc. British Mach. Vision Conf.*, Vol. 1, pp. 123–132 (2008).
19. W. S. Tam, "Modified edge-directed interpolation," MSc Thesis, Hong Kong Polytechnic Univ., Hong Kong (2007), see http://library.polyu.edu.hk/record=b2080904~S6.
20. W. S. Tam, C. W. Kok, and W. C. Siu, "A modified edge directed interpolation for images," *Proc. Europ. Signal Process. Conf.* (EUSIPCO), pp. 283–287 (2009).
21. B. Vrecelj and P. P. Vaidyanathan, "Efficient implementation of all-digital interpolation," *IEEE Trans. Image Process.* **10**, 1639–1646 (2001).
22. B. Vrecelj and P. P. Vaidyanathan, "Efficient implementation of all-digital interpolation," see http://www.systems.caltech.edu/bojan/splines/mar00.html.
23. X. Li, "New edge-directed interpolation," see http://www.csee.wvu.edu/~xinl/source.html.
24. N. Asuni, "iNEDI (improved New Edge-Directed Interpolation)," see http://www.mathworks.co.uk/matlabcentral/fileexchange/13470.
25. A. Giachetti and N. Asuni, "ICBI download page," http:www.andreagiachetti.iticbi/.
26. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**, 600–612 (2004).
27. C. Lee, S. Cho, T. Jeong, W. Ahn, and E. Lee, "Objective video quality assessment," *Opt. Eng.* **45**, 017004 (2006).
28. Z. Wang, H. R. Sheikh, and A. C. Bovik, "Objective video quality assessment," in *Handbook of Video Databases: Design and Applications*, Chap. 41, pp. 1041–1078 (2003).
29. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment," http://www.ece.uwaterloo.ca/~z70wang/research/ssim/ (2009).

Wing-Shan Tam received her BEng degree in electronic engineering from the Chinese University of Hong Kong, and her MSc degree in electronic and information engineering from the Hong Kong Polytechnic University, in 2004 and 2007, respectively. Currently, she is pursuing her PhD degree in the Department of Electronic Engineering at the City University of Hong Kong. She has been working in the telecommunication and semiconductor industries. Her research interests include image processing and mixed-signal integrated circuit design for data conversion and power management.

Chi-Wah Kok earned his PhD degree from the University of Wisconsin Madison. Since 1992, he has been working with various semiconductor companies, research institutions, and universities, including AT&T Laboratories Research, Holmdel, Sony United States Research Laboratories, Stanford University, Hong Kong University of Science and Technology, Hong Kong Polytechnic University, City University of Hong Kong, Lattice Semiconductor, etc. In 2006, he founded Canaan Microelectronics Corporation, Limited. His research interests include multimedia signal processing, wavelet and filter banks, and digital communications.

Wan-Chi Siu received the Associateship from Hong Kong Polytechnic University, the MPhil degree from the Chinese University of Hong Kong in 1975 and 1977, respectively, and the PhD degree from the Imperial College of Science, Technology, and Medicine, London, in October 1984. He was with the Chinese University of Hong Kong as a tutor and later as an engineer between 1975 and 1980. He then joined Hong Kong Polytechnic University has been a chair professor of the Department of Electronic and Information Engineering since 1992. He is now the director of Centre for Signal Processing of the same university. He is an expert in digital signal processing, specializing in fast algorithms and video coding. His research interests also include transforms, image processing, and the computational aspects of pattern recognition and wavelets. He has published 380 research papers, more than 150 of which appeared in international journals. He is an editor of the book *Multimedia Information Retrieval and Management* (Springer, Berlin Heidelberg, 2003).