Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 576831, 18 pages
doi:10.1155/2010/576831

Research Article

Image Enlargement by Applying Coordinate Rotation and Kernel Stretching to Interpolation Kernels

Sheng-Fu Liang,[1] Hung-Ming Chen,[2] and Yi-Che Liu[2]

[1] *Department of Computer Science and Information Engineering, Institute of Medical Informatics, National Cheng-Kung University, Tainan 701, Taiwan*
[2] *Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan 701, Taiwan*

Correspondence should be addressed to Sheng-Fu Liang, sfliang@mail.ncku.edu.tw

Received 20 July 2009; Revised 27 November 2009; Accepted 25 January 2010

Academic Editor: Dimitrios Tzovaras

Conventional image interpolation methods such as the bilinear and bicubic algorithms determine the weights of the reference pixels based on the special distance between the supplementary pixel and the reference pixels. They suffer from some problems such as blurring and jagging around the edges since the edge structure is not considered in interpolation. In this paper, a coordinate rotation and kernel stretch strategy combined with the bilinear or bicubic algorithm is proposed to tackle these problems. For the edge regions, the spatial coordinate axes are rotated to the edge direction and the edge normal to calculate the distances between the supplementary pixel and the original reference pixels. The kernel function is also stretched along the estimated edge direction so that the weights of the reference pixels along the edge direction would be higher than the others to smooth the edge and prevent the jagging artifacts. An edge enhancement method is also proposed to further reduce the interpolation error and sharpen the edge. The experiment results demonstrate that conventional bilinear and bicubic image interpolation methods combined with the proposed strategy can greatly improve the quantitative and qualitative performance for image enlargement.

1. Introduction

Image interpolation is a technology for image resolution conversion that generates a high-resolution image from its given low-resolution image version. It has been widely applied to various image/video capture/display devices such as digital still cameras, digital video cameras, printers, and multiple-function peripherals. Bilinear and bicubic interpolations are two conventional and widely used methods. However, applying a fixed kernel across the whole image without considering local structural characteristics will suffer from some problems such as blurring and jagging around the edges. Therefore, several methods have been proposed to tackle these problems by analyzing the image first to achieve better interpolation quality.

The adaptive interpolation schemes spatially adapt the interpolators to better match the local structure around the edges [1–6]. The isophote-oriented approach solves the ensuing partial differential equations (PDEs) to minimize the curvature of interpolated isophotes to reduce the zigzagging artifacts [7]. Based on the assumption that the image to be enhanced is the lowpass filtered subband of a wavelet-transformed high-resolution image, wavelet-based approaches have also been proposed for image interpolation [8–10].

Since human eyes are more sensitive to the edge areas than the smooth areas within an image [11], many algorithms have been proposed to improve the subjectively visual quality of edge regions in the images that need interpolation applied to them [12–14]. In [12], the edge-directed method (NEDI) uses the covariance of the original image to estimate the covariance information of the high-resolution image. In [13, 14], the bilinear method was modified by considering the direction information of edges to preserve better smoothing edges than traditional version. In order to improve the quality of interpolated images, neural-network-based schemes [15, 16], the minimum mean square-error estimation [17], and the autoregressive modeling [18] have also been proposed for adaptive filter design. Several techniques that improve original NEDI [12] by reducing computation cost for video applications [19], considering the effects of the curvature continuity by an iterative refinement approach [20] or applying multiple training windows to mitigate the covariance mismatch problem [21], were also developed. Since local adaptive interpolation method with small window still suffers from the blurring problem [22], adaptive learning approaches by applying multiple bicubic processing [23] or the cellular neural networks [24] have also been proposed.

These approaches demonstrate improved visual quality in reducing the zigzagging artifacts compared with conventional methods, but they may have relatively high computational cost or create some extra artifacts such as the unwanted over fittings at texture regions. In this paper, an adaptive image enlargement scheme that combines image analysis, coordinate rotation, and kernel stretching is proposed to balance the computational cost and visual quality. An input image is first classified into edge and nonnedge regions by the Sobel edge detection method. The interpolation kernel (the bilinear and bicubic algorithms were used in this paper) is applied to the nonedge regions, directly. For the edge regions, the spatial coordinates are rotated according to the estimated edge direction before calculating the spatial distances between the supplementary pixels and their neighbor reference pixels for the interpolation kernel. The kernel function will also be stretched along the estimated edge direction to adjust the weights of the reference pixels to smooth the edge and prevent the jagging artifacts. In addition, an edge enhancement method is also proposed to further reduce the interpolation error and sharpen the edge.

This paper is organized as follows. The system architecture is introduced in Section 2. The details of the proposed approach including image analysis, coordinate rotation, and kernel stretching methods are presented in Section 3. Section 4 introduces the proposed edge enhancement method. Experiment results are presented in Sections 5 and 6 for demonstration. Section 7 concludes this paper.

2. System Architecture

The schematic block diagram of the proposed image enlargement system is shown in Figure 1. It consists of a region classification module, an angle evaluation module, a coordination rotation module, an interpolation module, and an enhancement module. When an original image enters the proposed system, it is firstly divided into 4 × 4 sliding blocks that are processed pixel by pixel. The original block is shown in Figure 2(a) as an illustration, where $O(1,1)$ is defined as the reference pixel and $O(i,j)|_{i\neq 1, j\neq 1}$ is the neighborhood of $O(1,1)$. According to Figure 2(b), the three adjacent image pixels $P(1,0)$, $P(0,-1)$, and $P(1,-1)$ are the interpolated supplementary pixels relative to reference image pixel $O(1,1)$ positioned at offset coordinate points $(1,0)$, $(0,-1)$, and $(1,-1)$ in a Cartesian coordinate reference system for the two times interpolation [16].

We firstly classify the sliding block into an edge or nonedge class by Sobel edge detection. For the nonedge blocks, the conventional interpolation kernel such as the bilinear or bicubic is utilized directly. For a classified edge region, the angle evaluation module will compute the dominant orientation of the sliding block and the original coordinates are rotated to the new coordinates according to the estimated edge orientation. The distances between the supplementary pixel and the surrounding reference pixels in the block are calculated according to the new coordinate. Then the interpolation kernel determines the value of the supplementary pixel according to the modified distances between the supplementary pixel and the surrounding pixels as well as the values of the surrounding pixels. Based on the proposed coordinate rotation strategy, the weights of the reference pixels along the edge direction would be higher than the others to preserve the smoothness of the edge and the interpolation kernel for the edge and nonedge regions can be identical to reduce the computational complexity.

In addition, for some supplementary pixels that do not have reference pixels along the edge direction, an enhancement method is also proposed to reduce the error and sharpen the edge. The reference pixels along the normal of the edge are analyzed to estimate if the edge on the supplementary pixel is a roof edge or a ramp edge. For the roof edge, the value of the supplementary pixel is calculated by averaging the extrapolations of the both sides of the roof. For the ramp edge, the value of the supplementary pixel is determined as the extrapolation of the step side to sharpen the edge. The detail discussion is given in Section 4.

3. Image Interpolation Based on Coordinate Rotation

3.1. Edge Detection and Edge Angle Evaluation. As shown in Figure 1, when an original image enters the proposed system, it is firstly divided into 4 × 4 sliding blocks and each sliding block is classified into an edge or nonedge class by Sobel edge detection, since Sobel is simple and easy to be implemented. When an original pixel $O(i,j)$ at a sliding block is computed, the values of the original pixels nearby $O(i,j)$ are used to calculate the gradients in the horizontal and vertical directions as (1):

$$\begin{aligned} G_x &= [O(i-1,j-1)+2O(i-1,j)+O(i-1,j+1)] \\ &\quad -[O(i+1,j-1)+2O(i+1,j)+O(i+1,j+1)], \\ G_y &= [O(i-1,j-1)+2O(i,j-1)+O(i+1,j-1)] \\ &\quad -[O(i-1,j+1)+2O(i,j+1)+O(i+1,j+1)]. \end{aligned} \tag{1}$$

If the G_x or G_y is large than the threshold, we classify the pixel as a portion of an edge. Otherwise the pixel would be regarded as a nonedge pixel. The threshold value is set as 15 in this paper. The orientation angle of an edge pixel $O(i,j)$ denoted as $A(i,j)$ can be determined by

$$A(i,j) = \tan^{-1}\left(\frac{G_y(i,j)}{G_x(i,j)}\right). \tag{2}$$

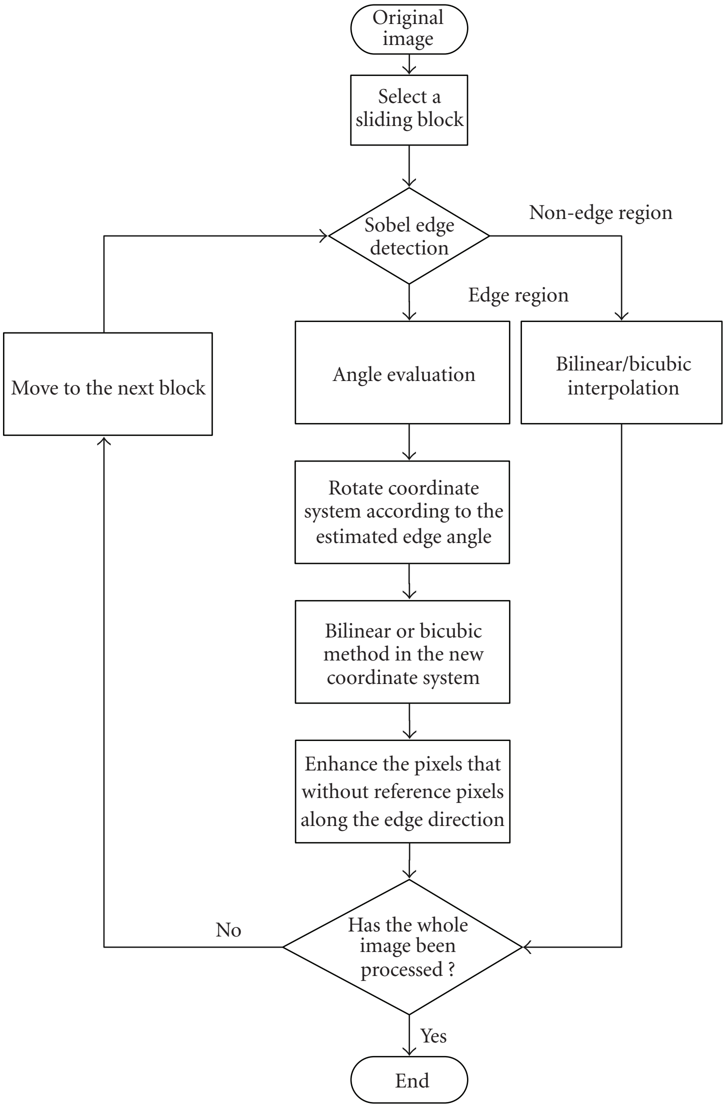

Original image
Select a sliding block
Sobel edge detection
Non-edge region
Edge region
Move to the next block
Angle evaluation
Bilinear/bicubic interpolation
Rotate coordinate system according to the estimated edge angle
Bilinear or bicubic method in the new coordinate system
Enhance the pixels that without reference pixels along the edge direction
No
Has the whole image been processed ?
Yes
End

Figure 1: Schematic block diagram of the proposed image enlargement system.

The obtained orientation angle of each original pixel is quantized into the closest eight quantized angles such as $\theta = 22.5 \times k$, $0 \leq k \leq 7$. In the sliding block that the edge pixel $O(i, j)$ belongs to, the quantized orientation angles of the 16 original pixels are calculated. If the most frequently occurring quantized orientation θ has the supports of 6 or more original pixels, it is regarded as the edge angle of the sliding block for coordinate rotation.

3.2. Image Interpolation Kernels. Before proposing the coordinate rotation and kernel stretching strategy for image interpolation, two conventional interpolation kernels, the bilinear and bicubic methods, utilized in this paper for demonstration, are brief introduced. The concept of one dimensional interpolation is obtaining the signal at the position$x$, denoted as $f(x)$, from the function values of its surrounding points $f(x_i)$, where f represents the estimated function of the signal. The linear interpolation method considers two surrounding points and their distances from the target as the input of the kernel function:

$$K_{\text{linear}}(x) = \begin{cases} 1 - |x|, & 0 \leq |x| < 1, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where x is the interval between the supplementary point and its surrounding point. It means that the weight of the referenced pixel is inversely proportional to the distance between the supplementary point and its referenced point.

The kernel function of nonlinear cubic interpolation [25] is defined as

$$K_{\text{cubic}}(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}|x|^2 + 1, & 0 \leq |x| < 1, \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2, & 1 \leq |x| < 2, \\ 0, & \text{otherwise}. \end{cases} \tag{4}$$

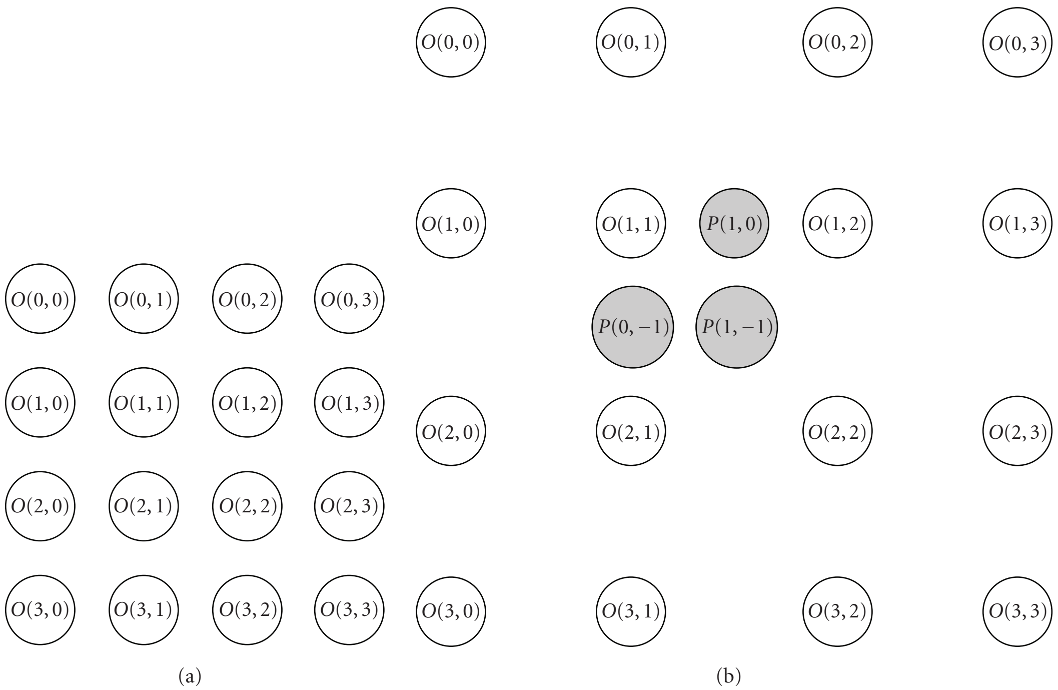


FIGURE 2: (a) A 4 × 4 sliding block window in the original image. (b) The block after two times interpolation.

It utilizes four surrounding points to interpolate the target; so the cubic interpolation needs more referenced points and more computational cost than the linear method to get better precision. Figure 3 shows these two one-dimensional interpolation kernels. They can be extended to two-dimensional interpolation (image interpolation) and called the bilinear and bicubic interpolations, respectively.

The bilinear method uses (3) in the horizontal and vertical directions, respectively, to obtain the value of a supplementary pixel; that is, it is the linear combination of the four surrounding pixels. Similarly, the bicubic method also needs to operate (4) in both horizontal and vertical directions. Since one-dimension cubic interpolation refers to four reference points, 4 × 4 block is required for the bicubic interpolation.

In summary, if we want to use the two-dimensional interpolation kernel to interpolate a supplementary pixel at coordinate (x_p, y_p) in the image I, it can be calculated by

$$I\left(x_p, y_p\right) = \sum_{x}\sum_{y} I(x, y) K\left(x_p - x\right) K\left(y_p - y\right), \quad (5)$$

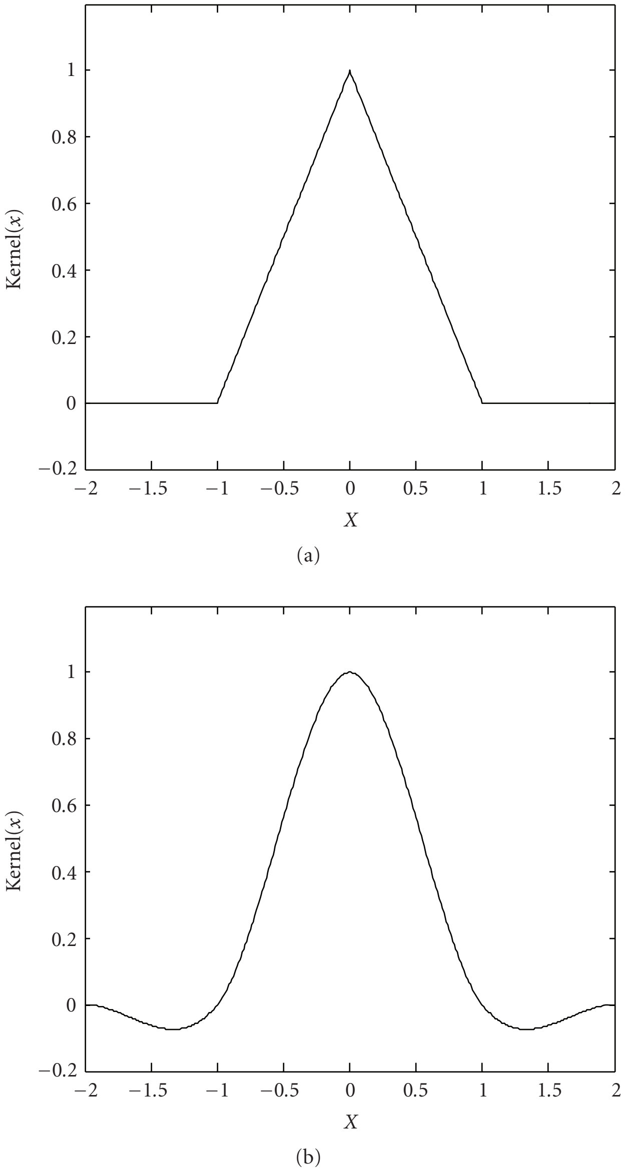
where K is the 1D interpolation kernel ((3) and (4)) and $I(x, y)$ represents the pixel value of the surrounding referenced pixel at position (x, y). As shown in Figure 1, we use (5) for the interpolation of nonedge pixels directly.

3.3. Coordinate Rotation and Kernel Stretching for Interpolation of the Edges. The conventional image interpolation kernels as shown in (5) only consider the spatial distance between the reference pixel and the supplementary pixel to determine the weight of the reference pixel based on the assumption that the supplementary pixel is more similar to the nearer reference pixels due to limited spatial support. However, the supplementary pixel in the edge region should be more similar to the nearer reference pixels along the edge direction. Our previous study that utilized a neural network to obtain weights of reference pixels for image interpolation by automatic training also shows that the resultant weights follow the directional characteristics of the edges [16]. Therefore, in order to adjust the weights of the reference pixels to smooth the edge and prevent the jagging artifacts, in this paper, after determining the dominant edge orientation of the sliding block, the original coordinates of the sliding block are rotated to the new coordinates. Finally, the kernel will operate based on the distances between the supplementary pixel and the surrounding pixels in the new coordinates for interpolation.

Assuming that the original coordinate of a pixel in a sliding block is (x, y) as shown in Figure 4(a) and the estimated edge angle is θ, the coordinate system of the block will be rotated by θ as shown in Figure 4(b) and the resultant coordinates of the pixel denoted as (x', y') can be calculated by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (6)$$

It is noted that after coordinate rotation, the x' axis is parallel to the edge and the y' axis is normal to the edge. In order to enhance the property of the edge, the distance between the supplementary pixel and the surrounding pixel in the x' axis should be shrunk and the distance in the y' axis should



(a)

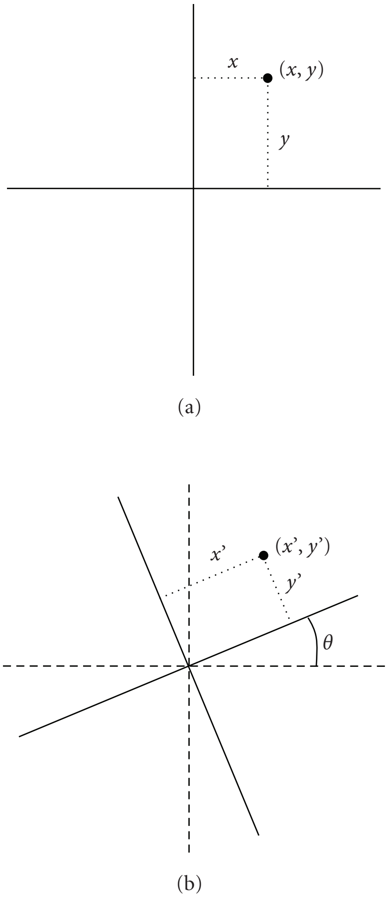
(b)

Figure 3: Two conventional 1D interpolation kernels. (a) The linear interpolation. (b) The cubic interpolation.

be stretched for the kernel function so that the weights of the reference pixels along the edge direction will be increased and the weights of the reference pixels normal to the edge will be weakened. Therefore, (5) is modified as

$$I\left(x_p, y_p\right) = \sum_x \sum_y I(x, y) K\left(ga \cdot \left(x'_p - x'\right)\right) K\left(gb \cdot \left(y'_p - y'\right)\right), \quad (7)$$

where ga and gb in (7) are the distance adjustment parameters in x' and y' axes to adjust the contributions of the neighbor pixels to interpolate the supplementary pixel. We set $gb = 1/ga$ and the parameters are fixed for the whole image in this paper. They can also be adaptive block by block according to the strength of the edge.



(a)

(b)

Figure 4: (a) (x, y), the original coordinates of a pixel. (b) (x', y'), the rotated coordinates of the pixel by the estimated edge angle θ.

In order to determine the distance adjustment parameters, we analyze the average energy of edge blocks of "Lena" by the principle component analysis (PCA) [26]. For each block, the gradient values of each pixel in horizontal and vertical calculated by (1) are grouped into an 16×2 matrix G as

$$G = \left[\overline{G}_x, \overline{G}_y\right]. \quad (8)$$

Through PCA, the gradient matrix G can be decomposed into two eigenvectors. The eigenvector with the larger eigenvalue represents the edge axis and the other eigenvector is normal to the edge. Larger difference between these two eigenvalues means that the edge is sharper. The sharpness of the edge block can be calculated by [26]

$$R = \frac{s_1 - s_2}{s_1 + s_2}, \quad (9)$$

where s_1 and s_2 are the eigenvalues of the two decomposed principle components and s_1 is the larger eigenvalue. Figure 5 shows an example to illustrate the correlation between the R value and the sharpness of the edge. Figures 5(a) and 5(b) show two blocks with different edge strength in "Lena".

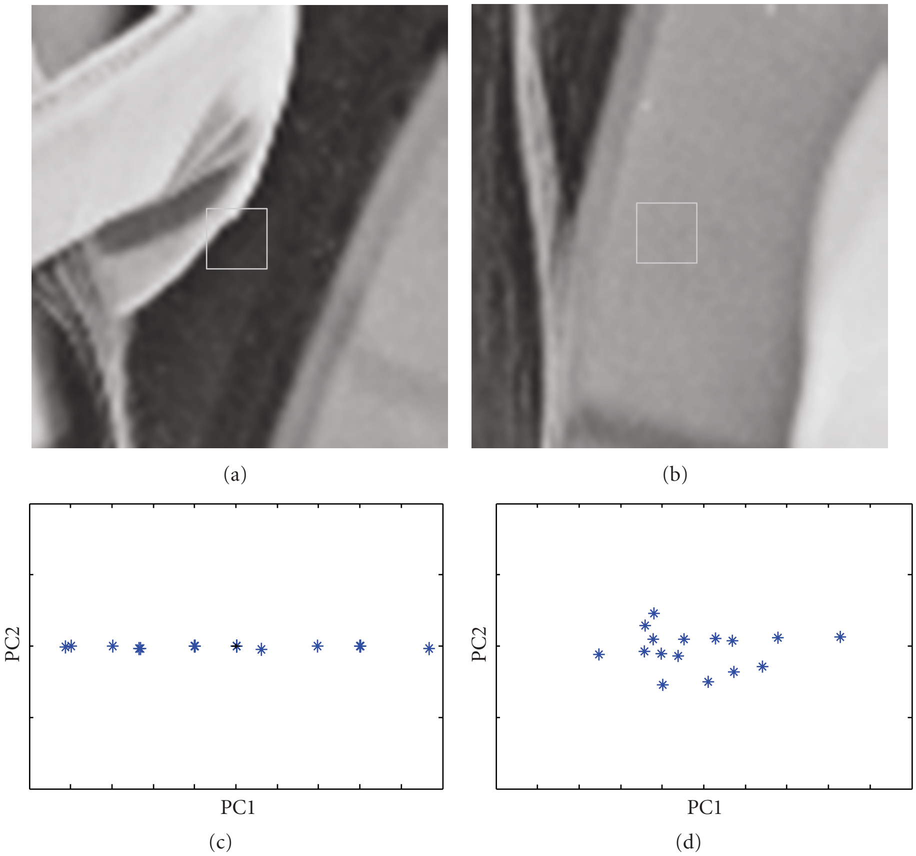


Figure 5: An example to illustrate the correlation between the R value and the strength of the edge. Image blocks with (a) $R = 0.9$ and (b) $R = 0.4$ in "Lena". (c) The distribution of the 16 pixels in block (a) on the two principle components. (d) The distribution of the 16 pixels in block (b) on the two principle components.

Figures 5(c) and 5(d) show the distributions of the block pixels of Figures 5(a) and 5(b) on their corresponding principle components. The R values of Figures 5(a) and 5(b) are 0.9 and 0.4, respectively. It can be observed that the block with larger R value has a stronger edge.

The analysis results of edge blocks in "Lena" are shown in Table 1. The angles of the edge blocks are classified to 8 degrees. The number of blocks for each quantized angle and the corresponding averaged R are presented. The total average of the R value is 0.65. Therefore, we set $ga = 0.65$ and $gb = 1/ga = 1.54$. Applying these two distance adjustment parameters to (7) and utilizing the bicubic method as the kernel, the resultant weights corresponding to the eight different edge orientations for the three supplementary pixels in Figure 2(b) are shown in Table 2 as an example. It could be found that the weights of the reference pixels along the edge direction are larger than the weights of the reference pixels normal to the edge so that the conventional kernels such as the bicubic or bilinear methods integrated with the proposed coordinate rotation and kernel stretching method can keep the directional characteristics of the edges well. By combining (5) for nonedge regions and (7) for edge regions, the proposed method can improve image interpolation results without greatly increase computational cost since the same kernel is utilized for edge and nonedge regions.

4. Image Quality Enhancement

For some supplementary pixels that do not have reference pixels along the edge direction, we propose an image quality enhancement method for the interpolated image to reduce the error and sharpen the edge. The reference pixels along the normal of the edge are analyzed to estimate if the edge on the supplementary pixel is a roof edge or a ramp edge. The enhancement is only applied to the roof and ramp edges since the effects are more observable. For the roof edge, the value of the supplementary pixel is calculated by averaging the extrapolations of the both sides of the roof. For the ramp edge, the value of the supplementary pixel is determined as the extrapolation of the step side to sharpen the edge. Since the enhancement is proved to make the proposed method more complete in the processing procedure, it is simple and only 0, 90, 45, and 135 degrees are considered.

4.1. Enhancement for 0 Degree or 90 Degree Edges.

According to Figure 6(a), for the 0-degree edge, $P(0, -1)$ and $P(1, -1)$ have no original pixels along the edge direction for reference. Similarly, $P(1, 0)$ and $P(1, -1)$ have no original pixels along the edge direction for the 90-degree edge as shown in Figure 6(b). In order to reduce the error and increase the edge contrast of the interpolation results, we detect the roof

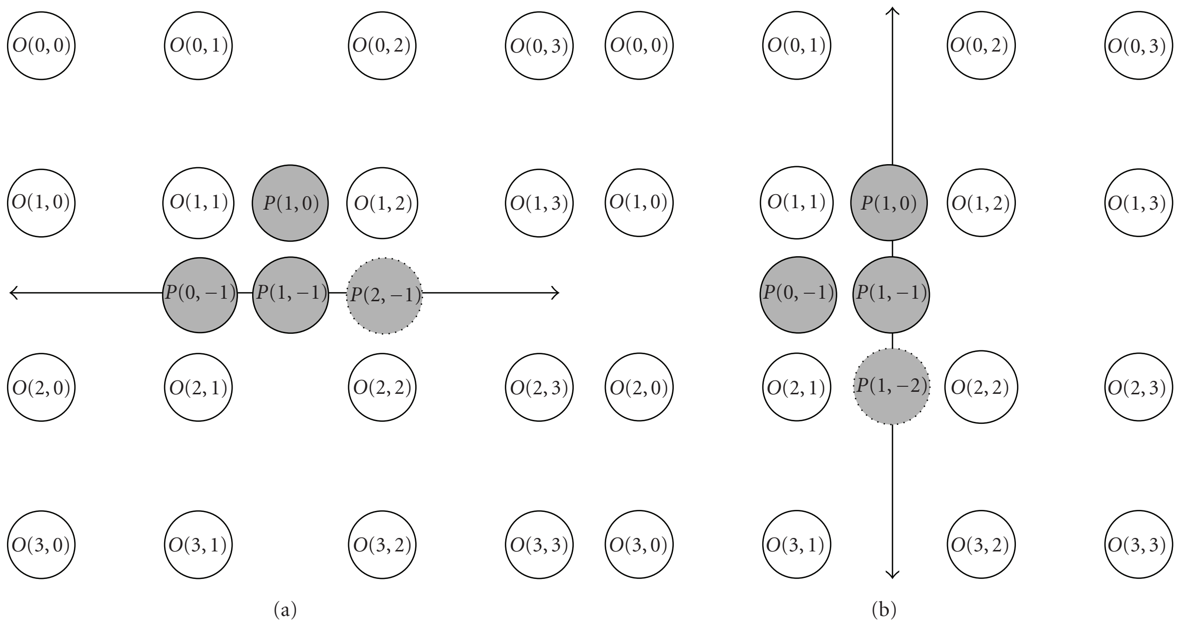


Figure 6: The supplementary pixels without reference pixels along the edge direction for (a) the 0 degree edge and (b) the 90 degree edge.

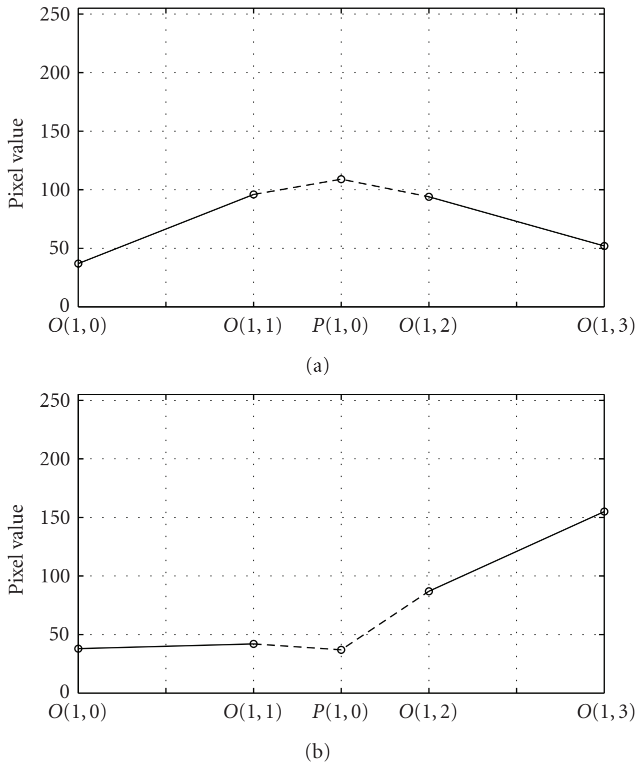


Figure 7: Two edge types for enhancement: (a) the roof edge, and (b) the ramp edge.

and ramp edges [27] based on the intensity profile along the normal direction of the edge.

The interpolation of the supplementary pixel $P(1,0)$ in Figure 6(b) is used as an example for illustration and the edge is 90 degree. As shown in Figure 7, we take four pixels, $(O(1,0), O(1,1))$ and $(O(1,2), O(1,3))$ that are along the normal (0 degree) of the edge direction at the two sides of the supplementary pixel to classify the edge type and to determine the value of $P(1,0)$. Assume $S_L = O(1,1) - O(1,0)$ and $S_R = O(1,3) - O(1,2)$, if

$$S_L \cdot S_R < 0, \quad |S_L| > 35, \; |S_R| > 35, \tag{10}$$

the slopes at the two sides of the edge have opposite slope signs, and it is classified as a roof edge for 8-bit images. The value of $P(1,0)$ is calculated by averaging the linear extrapolations of the both sides of the roof as shown in Figure 7(a). If

$$\operatorname{Max}(|S_L|, |S_R|) > 60, \qquad \operatorname{Min}(|S_L|, |S_R|) < 15, \tag{11}$$

the edge has a plain slope and a sharp slope at the two sides and is classified as a ramp edge. The thresholds used in (10) and (11) are obtained based on the heuristic analysis of the “Lena” image. It does not guarantee to be the optimal values for all cases. The value of $P(1,0)$ is determined as the linear extrapolation of the step side to sharpen the edge as shown in Figure 7(b). After $P(1,0)$ is enhanced, the value of $P(1,-1)$ in Figure 6(b) is determined by

$$P(1,-1) = \frac{1}{2}(P(1,0) + P(1,-2)), \tag{12}$$

where $P(1,-2)$ is the interpolated supplementary pixel in the next lower block.

Similarly, the above mentioned procedure can be utilized for the 0 degree edge shown in Figure 6(a). We first enhance $P(0,-1)$ based on $O(0,1)$, $O(1,1)$, $O(2,1)$, and $O(3,1))$ that are along the normal of the edge direction at the two sides of the $P(0,-1)$. Then we interpolate $P(1,-1)$ by averaging $P(0,-1)$ and $P(2,-1)$ for enhancement.

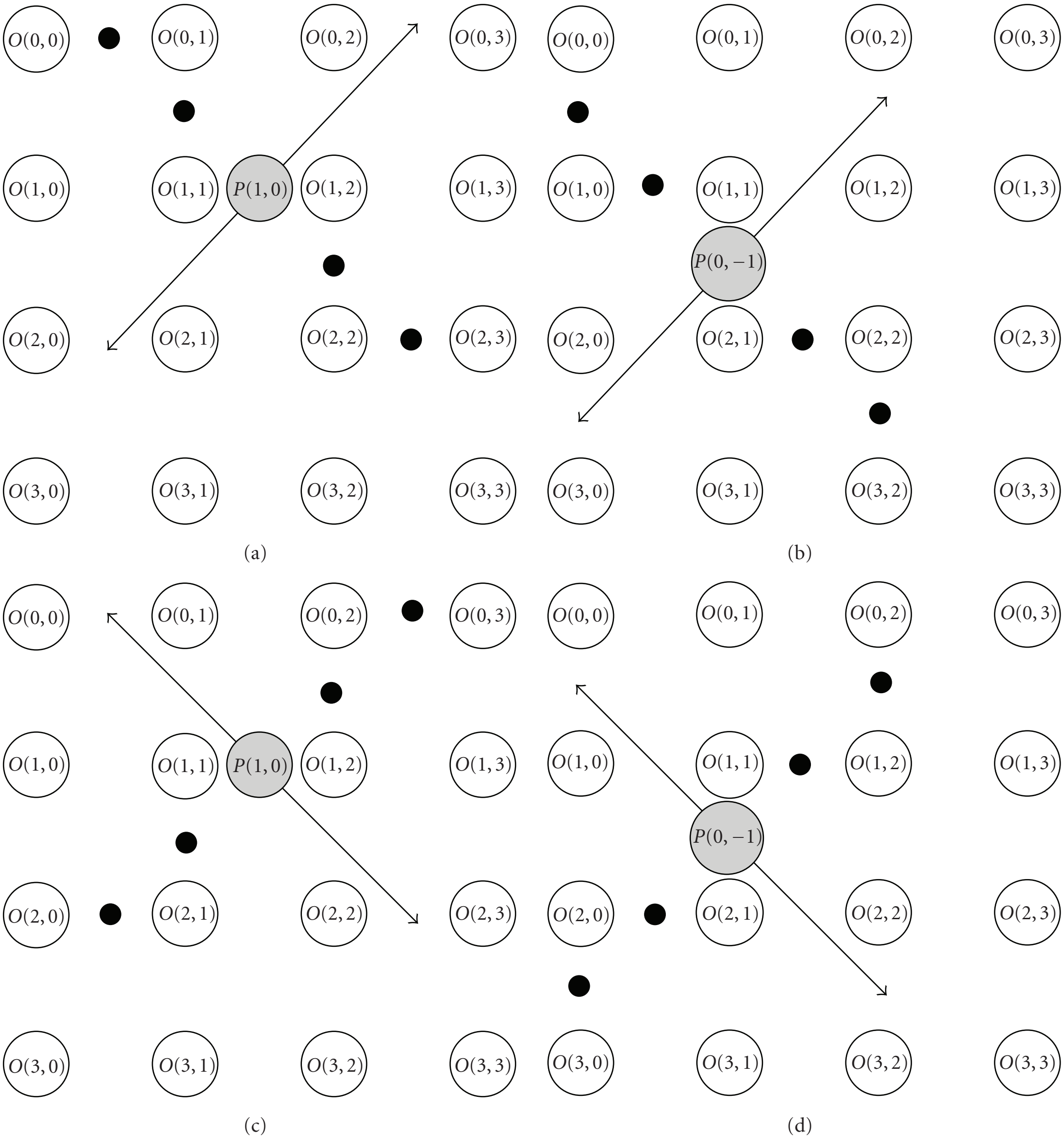

O(0,0) O(0,1) O(0,2) O(0,3)
O(1,0) O(1,1) P(1,0) O(1,2) O(1,3)
O(2,0) O(2,1) O(2,2) O(2,3)
O(3,0) O(3,1) O(3,2) O(3,3)
(a)
O(0,0) O(0,1) O(0,2) O(0,3)
O(1,0) O(1,1) O(1,2) O(1,3)
P(0,−1)
O(2,0) O(2,1) O(2,2) O(2,3)
O(3,0) O(3,1) O(3,2) O(3,3)
(b)
O(0,0) O(0,1) O(0,2) O(0,3)
O(1,0) O(1,1) P(1,0) O(1,2) O(1,3)
O(2,0) O(2,1) O(2,2) O(2,3)
O(3,0) O(3,1) O(3,2) O(3,3)
(c)
O(0,0) O(0,1) O(0,2) O(0,3)
O(1,0) O(1,1) O(1,2) O(1,3)
P(0,−1)
O(2,0) O(2,1) O(2,2) O(2,3)
O(3,0) O(3,1) O(3,2) O(3,3)
(d)

Figure 8: Interpolation for (a) $P(1,0)$ and (b) $P(0,-1)$ in a 45-degree edge block and interpolation for (c) $P(1,0)$ and (d) $P(0,-1)$ in a 135-degree edge block.

4.2. Enhancement for 45-Degree or 135-Degree Edges. According to Figures 2(b) and 8, we can find that for the 45- or 135-degree edge, $P(1,0)$ and $P(0,-1)$ have no original pixels along the edge direction for reference. We can also use the procedure proposed in the above subsection to enhance these two supplementary pixels. The difference is that instead of the original reference pixels, the four supplementary pixels interpolated by the method proposed in Section 3 are utilized as the reference pixels for edge classification and enhancement. For the 45-degree edge, the interpolated supplementary pixels along the edge normal (135-degree orientation) represented as the black rigid circles in Figures 8(a) and 8(b) are utilized for reference. Similarly, the interpolated supplementary pixels along the edge normal (45-degree orientation) in Figures 8(c) and 8(d) are utilized for reference to enhance $P(1,0)$ and $P(0,-1)$ for the 135-degree edge.

5. Experimental Results and Discussions

Six natural photographic images, Parrot, Bike, Aerial, Lena, Airplane, and House as shown in Figure 9, are used as our benchmark images for testing to demonstrate the generalization ability of the proposed approaches. In our experiments, two conventional interpolation kernels, bilinear and bicubic methods combined with the proposed coordinate rotation and image enhancement approach, are compared with the original bilinear and bicubic methods and five modern

(a) (b) (c) (d) (e) (f)

Figure 9: Six natural images in the test set. (a) Parrot; (b) Bike; (c) Aerial; (d) Lena; (e) Airplane; (f) House.

interpolation methods, the edge-directed interpolation [12] and its improvement called ICBI [20], the linear minimum mean square-error estimation- (LMMSE-) based interpolation [17], the soft-decision adaptive interpolation (SAI) [18], and the neural-network-based interpolation [16]. The test images are downsampled by a factor of two with the decimation approach. The interpolated images are compared with the original high resolution images for quantitative and qualitative evaluation.

5.1. Quantitative Comparisons. For quantitative performance evaluation, the peak signal-to-noise ratio (PSNR) defined as

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\left(\sum_{n=1}^{N}\left(I'_n - I_n\right)^2\right)/N}\right) \tag{13}$$

is used as quantitative performance indication, where I_n and I'_n are the gray levels of the original and the interpolated pixels, respectively, and N is the total number of pixels in an image. The PSNR numbers indicate the overall image quality with errors between the original and interpolated results pixel-by-pixel. The figure of merit (FOM) [28] and structural similarity (SSIM) index [11] are also calculated to compare edge preservation and structural similarity performances of different image interpolation methods, respectively. The FOM ranges between 0 and 1 and is defined by

$$\mathrm{FOM} = \frac{1}{\max\left\{\hat{N}, N_{\mathrm{ideal}}\right\}}\sum_{i=1}^{\hat{N}}\frac{1}{1+d_i^2\alpha}, \tag{14}$$

where $\hat{N}$ and N_{ideal} are the number of edge pixels in the interpolated image and the original image, respectively, d_i is the Euclidean distance between the ith edge pixel in the interpolated image and the nearest edge pixel in the original image, and α is a constant typically set to 1/9. FOM value is unity for ideal case and the edges pixels are detected by the Canny edge detector. The Mean SSIM (MSSIM) to evaluate the overall image quality between the original (X) and interpolated image (Y) is calculated by

$$\mathrm{MSSIM}(X, Y) = \frac{1}{M}\sum_{j=1}^{M}\mathrm{SSIM}\left(x_j, y_j\right), \tag{15}$$

where M is the number of local windows in the image, and x_j and y_j are the image contents at the jth local window. In this paper, 8×8 window is used to calculate SSIM. The SSIM between windows x and y is calculated by

$$\mathrm{SSIM}(x, y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\,\mathrm{cov}_{xy} + c_2\right)}{\left({\mu_x}^2 + {\mu_y}^2 + c_1\right)\left({\sigma_x}^2 + {\sigma_y}^2 + c_2\right)}, \tag{16}$$

where μ_x is the average of x, μ_y is the average of y, ${\sigma_x}^2$ is the variance of x, σ_y^2 is the variance of y, and cov_{xy} the covariance of x and y. c_1 and c_2 are two small values included to stabilize the division with weak denominator [11].

Table 3 shows the quantitative comparison of different image interpolation technologies. The best results are denoted in bold. The proposed coordinate rotation and kernel stretching method (Section 3) combined with the bilinear or bicubic interpolation kernel can consistently achieve superior quantitative performance compared with the other

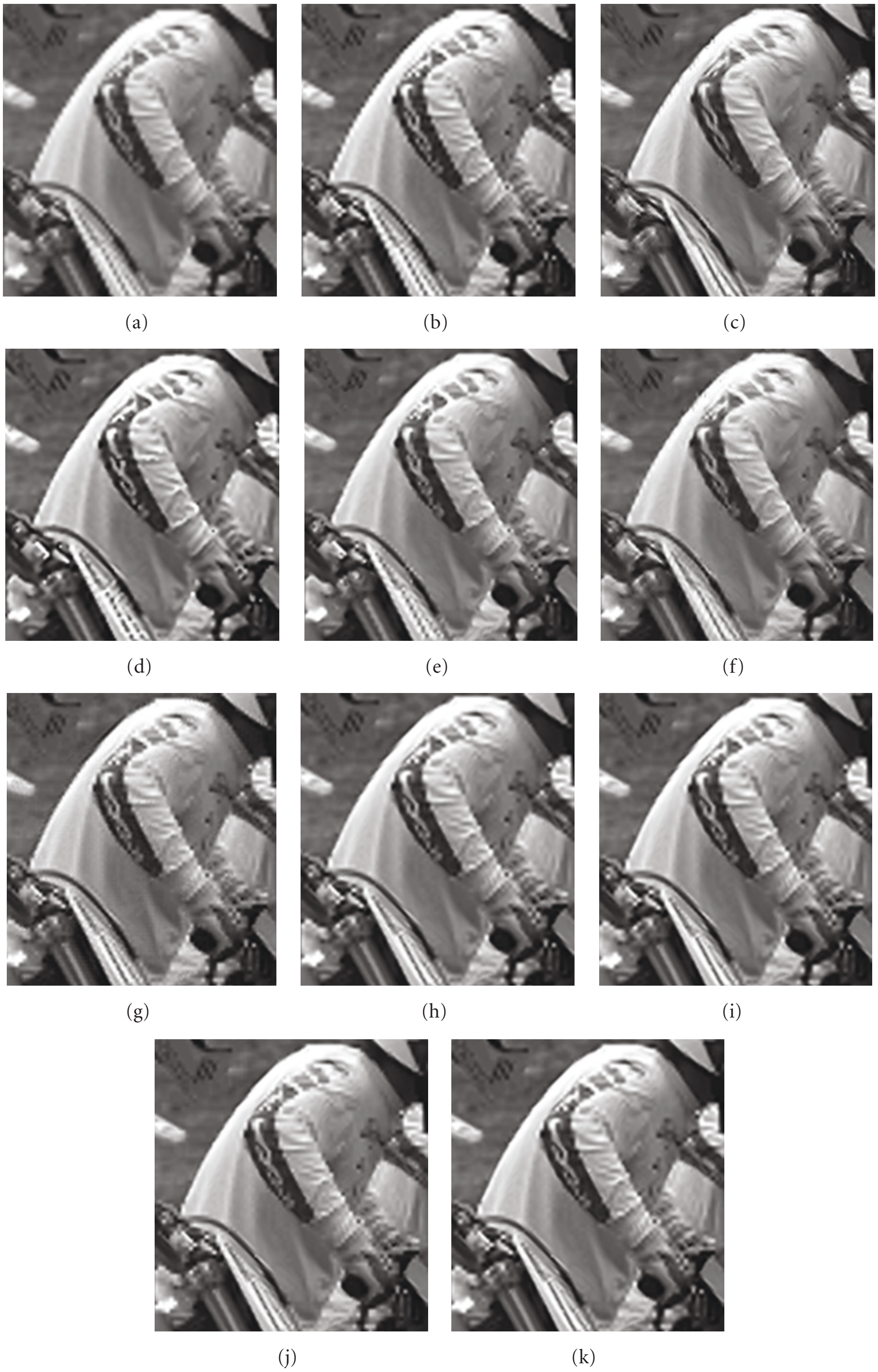


Figure 10: Portions of resolution enhanced "Bike" image by (a) bilinear; (b) bicubic; (c) edge-directed method ($M = 8$) [12]; (d) ICBI [20]; (e) LMMSE-based method ($\zeta = 1$, $\xi = 1.5$) [17]; (f) SAI ($\lambda = 0.5$) [18]; (g) NN-based method [16]; the proposed method utilized (h) bilinear kernel and (i) bicubic kernel without edge enhancement; (j) bilinear kernel and (k) bicubic kernel with edge enhancement.

approaches. In addition, the edge enhancement method proposed in Section 4 has a little average improvement in PSNR, SSIM, and FOM. The bicubic kernel combined with coordinate rotation and edge enhancement can achieve the best average performance.

6. Visual Quality Comparisons

Figures 10 and 11 show the two-time resolution enhanced portions of images "Bike" and "Lena" by using different interpolation methods, respectively. It can be found that

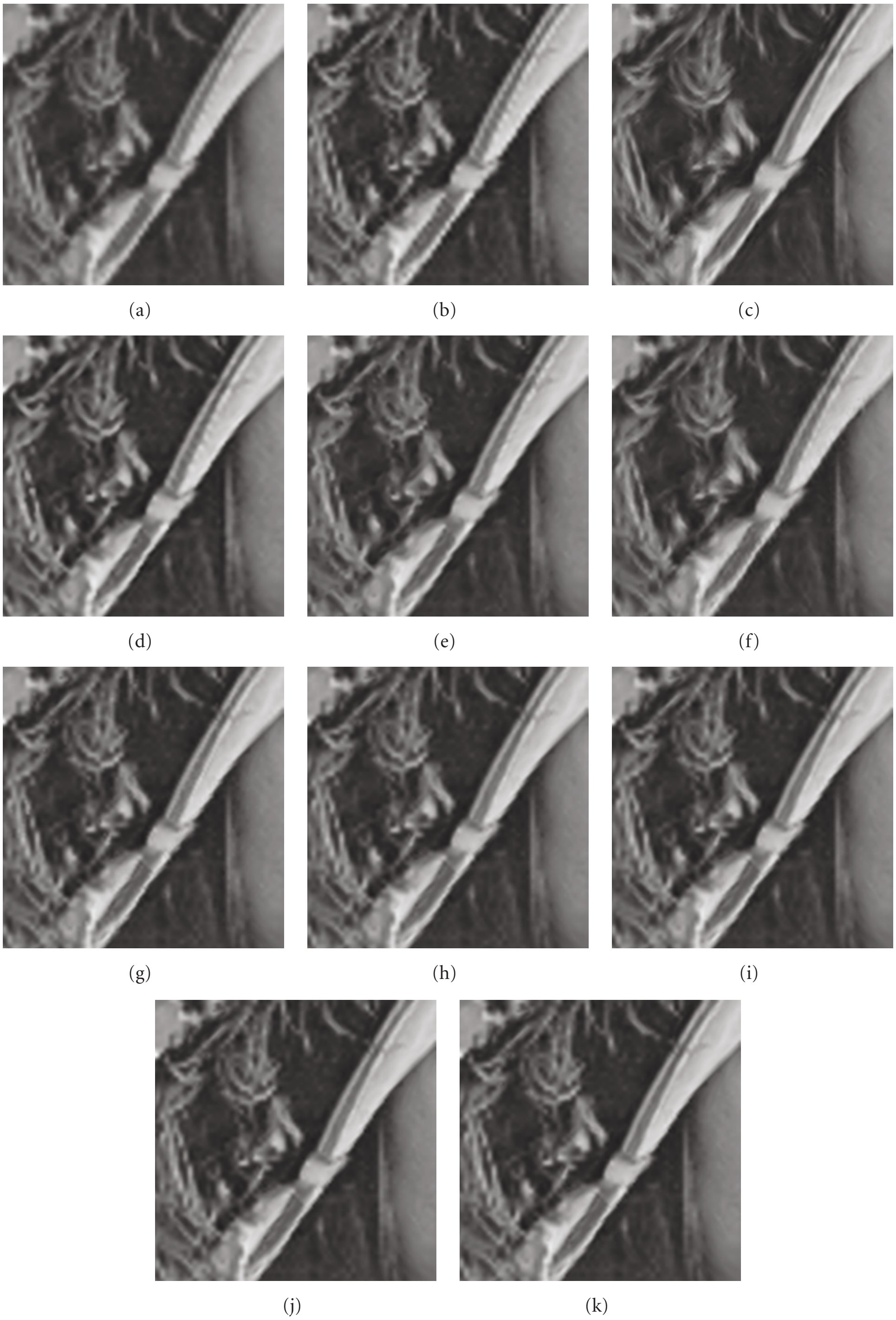


Figure 11: Portions of resolution enhanced "Lena" image by (a) bilinear; (b) bicubic; (c) edge-directed method ($M = 8$) [12]; (d) ICBI [20]; (e) LMMSE-based method ($\zeta = 1$, $\xi = 1.5$) [17]; (f) SAI ($\lambda = 0.5$) [18]; (g) NN-based method [16]; the proposed method utilized (h) bilinear kernel and (i) bicubic kernel without edge enhancement; (j) bilinear kernel and (k) bicubic kernel with edge enhancement.

the bilinear interpolation made some jaggedness along the edges and blurred the image. The bicubic method made less blurring than the bilinear method, but it made more observable jagged edges. All of the modern interpolation methods can achieve better visual quality than the bilinear and bicubic methods. The LMMSE-based interpolation [17] makes some jaggedness along the edges. The edge-directed interpolation [12] and the neural-network-based method [16] can preserve the edge well but the edge-directed approach sometimes produces the speckle noises and the neural-network-based approach makes a little burring. The SAI approach [18] can preserve the local details well but

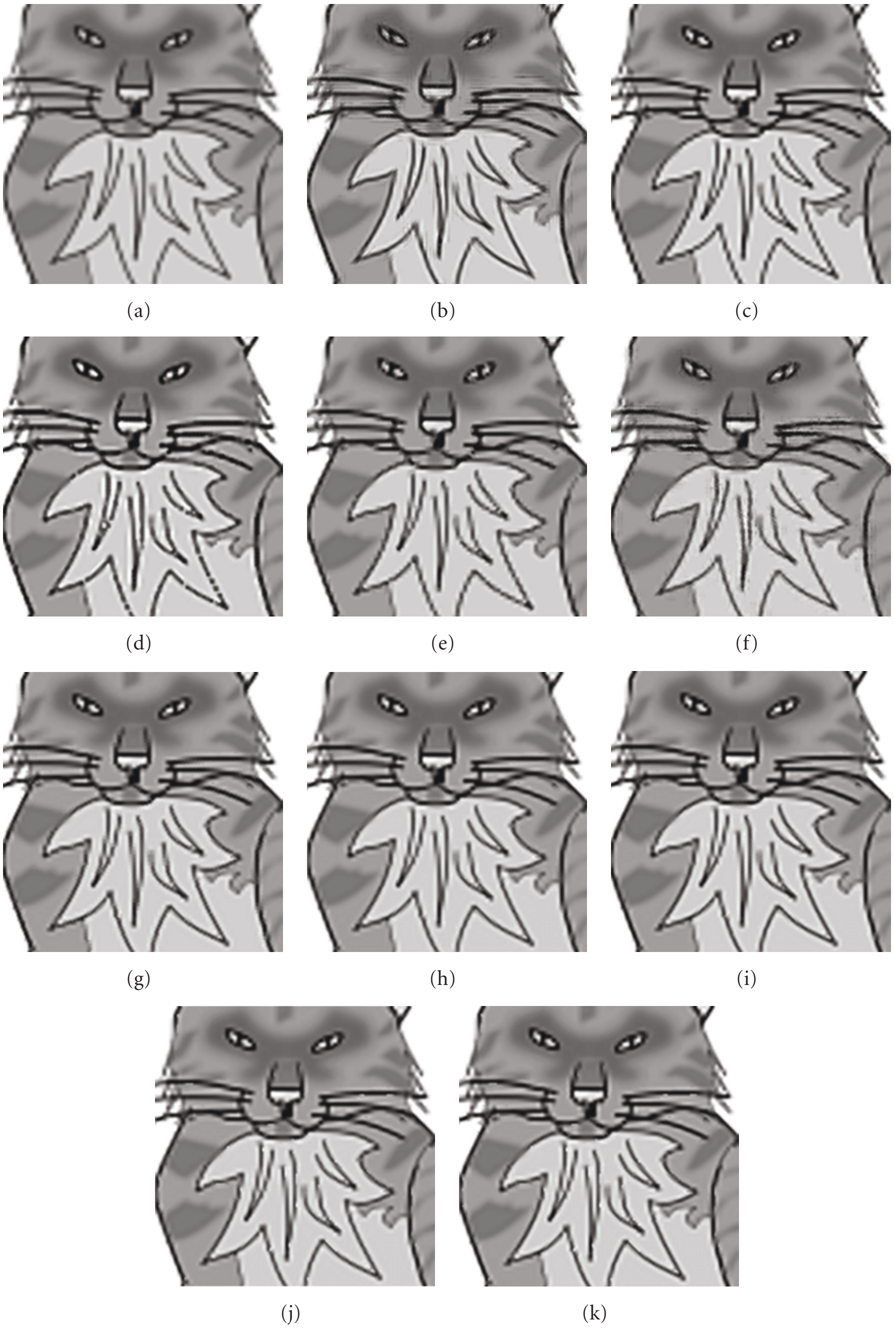
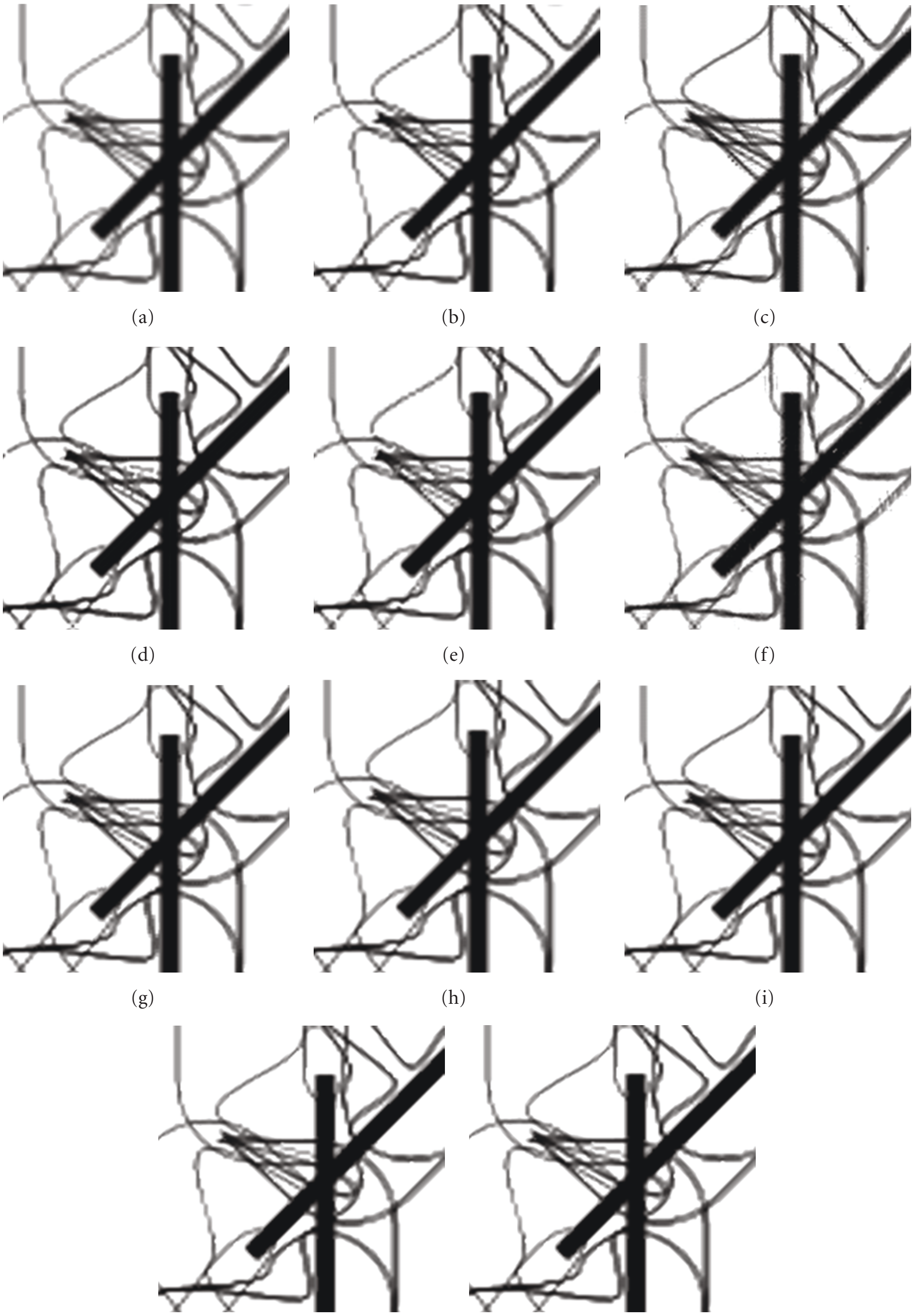


Figure 12: Portions of resolution enhanced "Gray cat" image by (a) bilinear; (b) bicubic; (c) edge-directed method ($M = 8$) [12]; (d) ICBI [20]; (e) LMMSE-based method ($\zeta = 1, \xi = 1.5$) [17]; (f) SAI ($\lambda = 0.5$) [18]; (g) NN-based method [16]; the proposed method utilized (h) bilinear kernel and (i) bicubic kernel without edge enhancement; (j) bilinear kernel and (k) bicubic kernel with edge enhancement.

sometimes produces the slight jaggedness if the parameter λ is fixed. It can also be found that the edge-directed method and the SAI method produce some distortions or over fittings at the texture regions. The ICBI [20] can reduce the speckle noises and over fittings compared with the original edge-directed interpolation [12]. But similar to the LMMSE [17], it also makes some jaggedness. The results of the proposed methods can keep the smoothness of the edges without speckle artifacts and it makes less blurring without jaggedness when the bicubic kernel is utilized. In addition, the proposed enhancement method can further sharpen the edge by analyzing the edge types to reduce the interpolation error.

We also applied different interpolation methods to perform two-time resolution enhancement of two clip arts, Gray Cat [29] and Geometric Motif [30], and the computer



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 13: Portions of resolution enhanced "geometric motif 3 outline" image by (a) bilinear; (b) bicubic; (c) edge-directed method ($M = 8$) [12]; (d) ICBI [20]; (e) LMMSE-based method ($\zeta = 1$, $\xi = 1.5$) [17]; (f) SAI ($\lambda = 0.5$) [18]; (g) NN-based method [16]; the proposed method utilized (h) bilinear kernel and (i) bicubic kernel without edge enhancement; (j) bilinear kernel and (k) bicubic kernel with edge enhancement.

generated 36-point English letters "NK" for comparison. The results are shown in Figures 12–14. In Figure 12, the edge-directed [12] and LMMSE [17] make the jaggedness. SAI [18] produces some noise along the whiskers and the ICBI [20] makes some speckle noise on the edge. The neural-network-based method [16] and the proposed methods can perform better visual quality. In Figure 13, the edge-directed [12] and the SAI [18] cause heavy speckle noise on the edge.

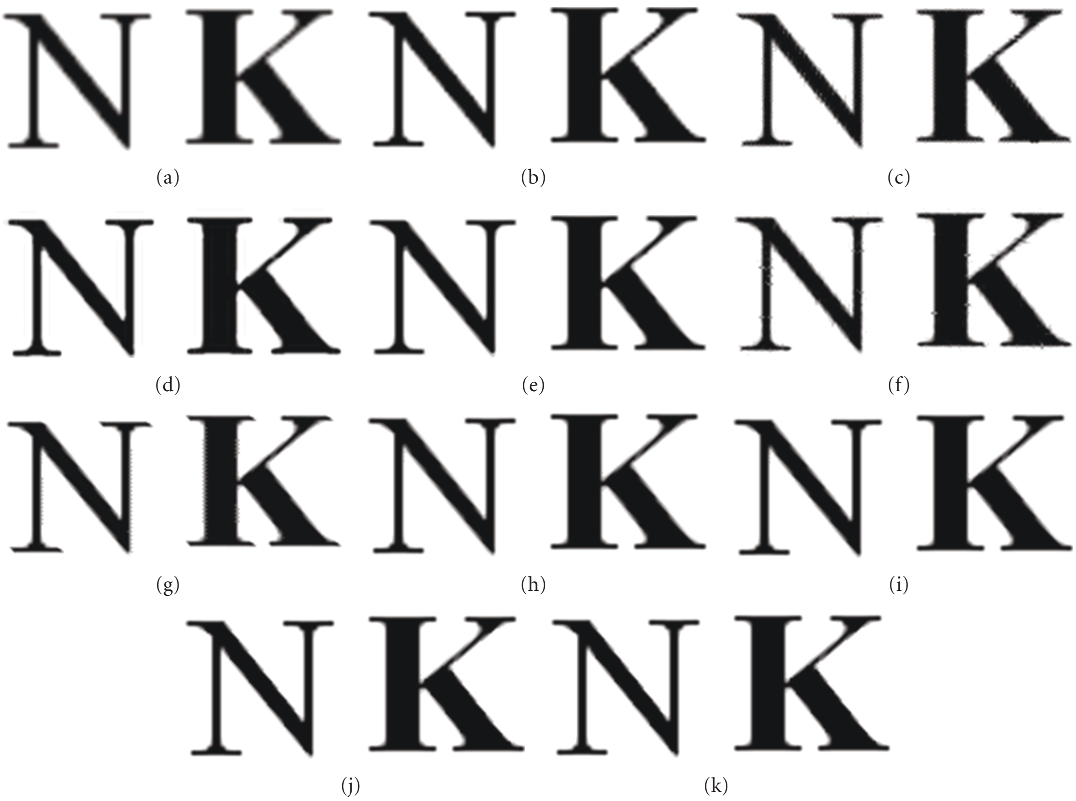

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Figure 14: Resolution enhanced English letters "NK" by (a) bilinear; (b) bicubic; (c) edge-directed method ($M = 8$) [12]; (d) ICBI [20]; (e) LMMSE-based method ($\zeta = 1$, $\xi = 1.5$) [17]; (f) SAI ($\lambda = 0.5$) [18]; (g) NN-based method [16]; the proposed method utilized (h) bilinear kernel and (i) bicubic kernel without edge enhancement; (j) bilinear kernel and (k) bicubic kernel with edge enhancement.

The LMMSE [17] produces slight jaggedness. The neural-network-based method [16], the ICBI [20], and the proposed methods can perform better visual quality. In addition, it can also be found that the proposed method with enhancement (Figures 13(j) and 13(k)) performs sharper results than the proposed method without enhancement (Figures 13(h) and 13(i)) on the two black bars. For the English letters shown in Figure 14, the LMMSE-based interpolation [17], the ICBI [20], and the proposed methods can perform better visual quality and do not generate obvious artifacts compared with the other methods. The above experimental results demonstrate that the proposed methods perform stable and superior visual quality on the clip arts and computer generated letters.

Table 1: The R values of the edge blocks in "Lena". The angles of the edge blocks are classified to 8 degrees.

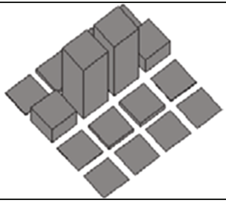
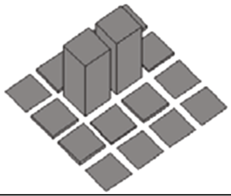
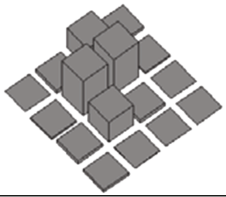
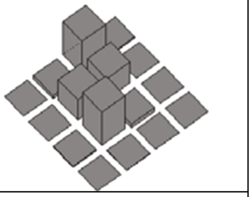
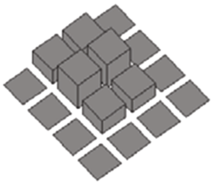
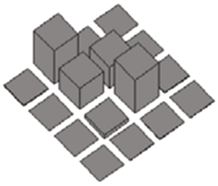
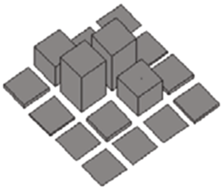
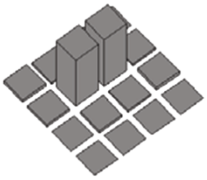
Degree of the edge	0	22.5	45	67.5	90	112.5	135	157.5
Number of blocks	45	888	2162	3312	7318	1916	705	580
R (average edge energy)	0.71	0.57	0.62	0.65	0.68	0.63	0.62	0.60

In addition to interpolation methods, it is noted that the downsampling strategy also affects the interpolation results. Figure 15 and Table 4 show the results and comparison of different algorithms applied to "Lena" downsampled by averaging. Comparing with Figure 11 and Table 3, it can be found that the results of applying the interpolation methods to the image downsampled by averaging are obviously more blurring than the results of the image downsampled by decimation. It is consistent with the FOM performance since the FOMs in Table 4 are much less than the FOMs in Table 3. The jaggedness caused by the bilinear and bicubic methods can be reduced if the image for interpolation is downsampled by averaging. SSIM in Tables 3 and 4 can present this difference since SSIM considers the structural similarity of two images. Although the bicubic and bilinear methods have better performance in PSNR and SSIM than all of the modern interpolation methods in Table 4, they still make more observable jagged edges than the modern interpolation methods.

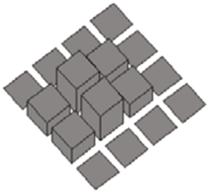
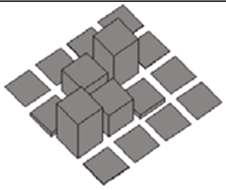
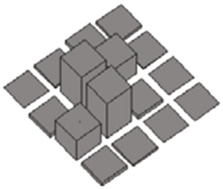
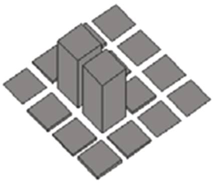
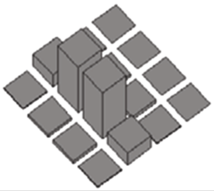
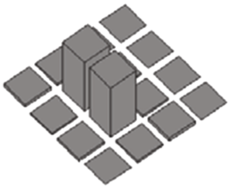
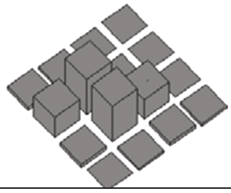
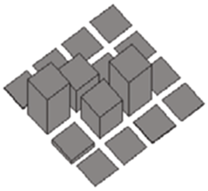
Since the modern interpolation methods interpolate the supplementary pixels based on the analysis results of the image structure characteristics, if the downsmpling method such as averaging modifies values of the reference pixels and their correlations, the interpolation results will be worse than that by decimation. Therefore, in order to perform better interpolation results for evaluation, all of these modern

Table 2: The weighting matrices for image interpolation obtained by applying the proposed coordinate rotation and kernel stretching method to the bicubic kernel.

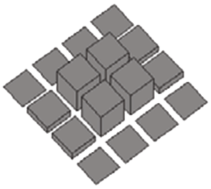
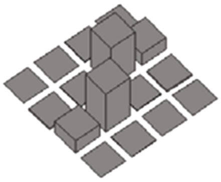
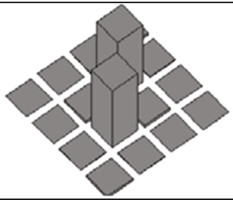
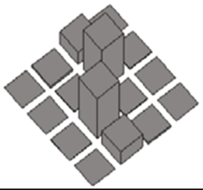
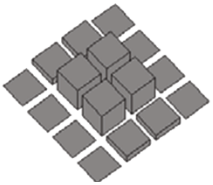
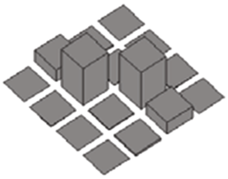
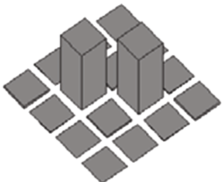
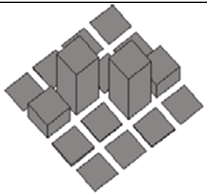
(a) $P(1,0)$.

0°	22.5°	45°	67.5°
			
90°	112.5°	135°	167.5°
			

(b) $P(0,-1)$.

0°	22.5°	45°	67.5°
			
90°	112.5°	135°	167.5°
			

(c) $P(1,-1)$.

0°	22.5°	45°	67.5°
			
90°	112.5°	135°	167.5°
			

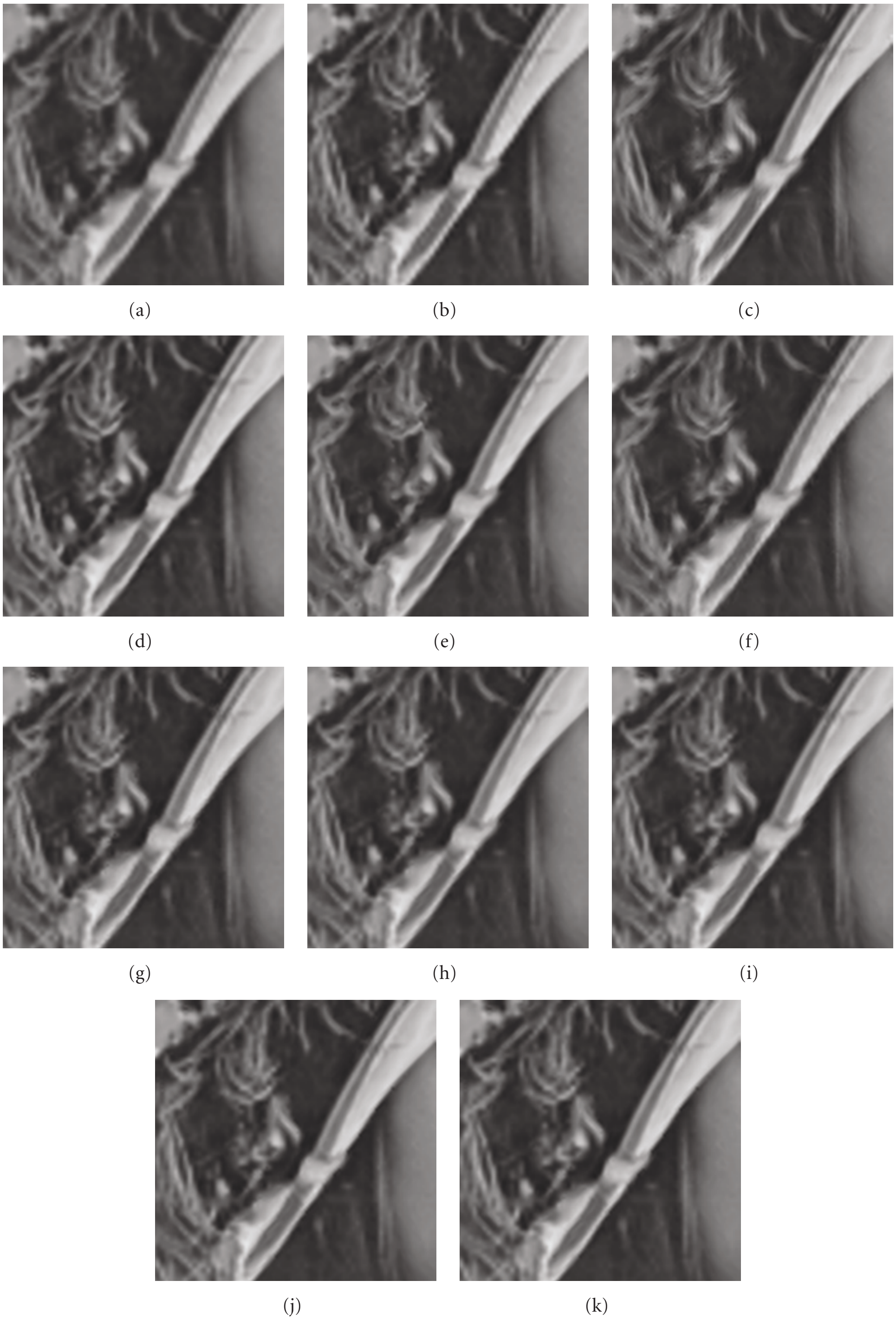

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Figure 15: Portions of resolution enhanced "Lena" image (downsampled by averaging) by (a) bilinear; (b) bicubic; (c) edge-directed method ($M = 8$) [12]; (d) ICBI [20]; (e) LMMSE-based method ($\zeta = 1, \xi = 1.5$) [17]; (f) SAI ($\lambda = 0.5$) [18]; (g) NN-based method [16]; the proposed method utilized (h) bilinear kernel and (i) bicubic kernel without edge enhancement; (j) bilinear kernel and (k) bicubic kernel with edge enhancement.

methods including this paper use decimation to generate the low-resolution images in the experiments.

In addition to better quantitative and visual performance, low computation cost is another advantage of the proposed approaches. Most of these modern interpolation methods need complex computations such as training neural network through multiple images [15, 16] or reducing the cost functions by solving the block-based inverse matrices or the iterative process [12, 17, 18, 20] to reach high visual quality. The computation of the proposed method requires

Table 3: Comparative results in PSNR, SSIM, and FOM of different algorithms applied to various kinds of resolution enhancement images. The best results are denoted in bold.

		Bilinear	Bicubic	Li's [12]	ICBI [20]	LMMSE [17]	SAI [18]	NN-base [16]	Coordinate Rotation (Bilinear/Bicubic)	Coordinate Rotation with Enhancement (Bilinear/Bicubic)
Parrot	PSNR	32.67	32.74	35.56	35.41	35.48	35.54	34.84	35.44/35.84	35.45/**35.85**
	SSIM	0.9207	0.9222	**0.9461**	0.9422	0.9319	0.9028	0.9449	0.9456/0.9453	0.9456/0.9455
	FOM	0.62	0.79	0.66	0.72	0.65	0.69	0.77	0.74/0.75	0.79/**0.82**
Bike	PSNR	23.15	23.11	25.48	25.14	25.74	25.92	25.71	25.81/26.01	25.82/**26.03**
	SSIM	0.7712	0.7902	0.8531	0.8661	0.8631	0.8413	0.8754	0.8762/0.877	0.8766/**0.8777**
	FOM	0.8	0.88	0.91	**0.93**	0.92	0.92	0.92	**0.93/0.93**	**0.93/0.93**
Aerial	PSNR	24.79	24.73	27.22	27.45	27.38	27.41	27.28	27.51/**27.80**	27.52/27.79
	SSIM	0.7928	0.8083	0.8676	0.8865	0.8706	0.8401	0.8868	0.8879/0.8885	0.8883/**0.8888**
	FOM	0.78	0.85	0.91	0.89	0.86	0.88	**0.92**	0.91/0.88	0.91/0.89
Lena	PSNR	29.97	29.89	33.76	33.9	33.79	33.81	33.8	33.70/33.98	33.75/**34.00**
	SSIM	0.8731	0.8731	0.9128	0.9118	0.9011	0.8719	0.914	**0.9154**/0.9152	**0.9154**/0.9153
	FOM	0.7	0.77	0.76	**0.84**	0.75	0.77	0.79	0.83/**0.84**	0.83/**0.84**
Airplane	PSNR	29.1	29.15	32.15	33.15	32.74	32.55	31.92	32.62/33.15	32.67/**33.21**
	SSIM	0.905	0.9087	0.9334	0.9432	0.9241	0.8898	0.9436	0.9439/0.9438	**0.9443**/0.9442
	FOM	0.72	0.79	0.84	**0.87**	0.82	0.84	0.81	**0.87**/0.86	**0.87**/0.86
House	PSNR	31.36	31.3	33.82	33.05	34.16	34.21	33.74	34.19/**34.32**	34.20/**34.32**
	SSIM	0.8782	0.8815	0.907	0.9107	0.7855	0.8758	0.9167	0.9168/0.9168	**0.917/0.917**
	FOM	0.72	0.78	0.85	**0.9**	0.81	0.78	0.82	0.89/0.89	0.89/0.89
Average	PSNR	28.51	28.49	31.33	31.35	31.55	31.57	31.22	31.545/31.85	31.568/**31.866**
	SSIM	0.8568	0.8640	0.9033	0.9101	0.8794	0.8703	0.9136	0.9143/0.9144	0.9145/**0.9148**
	FOM	0.72	0.81	0.82	0.86	0.80	0.81	0.84	0.86/0.86	**0.87/0.87**

Table 4: Comparative results in PSNR, SSIM, and FOM of different algorithms applied to "Lena" downsampled by averaging for interpolation. The best results are denoted in bold.

		Bilinear	Bicubic	Li's [12]	ICBI [20]	LMMSE [17]	SAI [18]	NN-base [16]	Coordinate Rotation (Bilinear/Bicubic)	Coordinate Rotation with Enhancement (Bilinear/Bicubic)
Lena	PSNR	32.5128	**34.082**	30.1348	30.37	30.2383	30.1956	30.3087	30.302/30.3297	30.2835/30.2995
	SSIM	0.9044	**0.9228**	0.8796	0.889	0.8709	0.8386	0.8865	0.8855/0.8861	0.8859/0.8863
	FOM	0.60	0.65	0.69	0.62	0.59	0.60	0.65	0.64/**0.72**	0.60/0.61

about 300 multiplications per pixel only; so the proposed methods can balance the tradeoff between speed and quality.

7. Conclusions

In this paper, a coordinate rotation and kernel stretch strategy was proposed to tackle the problems caused by the conventional bilinear and bicubic algorithms for image interpolation. The spatial coordinate axes of edges detected by the Sobel method were rotated to the edge direction and the edge normal. The kernel function was also stretched along the estimated edge direction to enhance the weights of the reference pixels along the edge direction to smooth the edge and prevent the jagging artifacts. The basic image kernel such as the bilinear or the bicubic methods combined with the proposed method can achieve high quantitative and visual qualitative performance on interpolation of nature and letter images. It can balance the tradeoff between computational cost and interpolation quality since solving the block-based inverse matrices or the iterative process that the other modern interpolation methods utilized is not required for the proposed methods.

Acknowledgment

This work was supported in part by the National Science Council of Taiwan under Grant NSC 98-2221-E-006-161-MY3.

References

[1] K. Xue, A. Winans, and E. Walowit, "An edge-restricted spatial interpolation algorithm," *Journal of Electronic Imaging*, vol. 1, no. 2, pp. 152–161, 1992.

[2] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 247–258, 1995.

[3] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still image," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 285–295, 1995.

[4] H. C. Ting and H. M. Hang, "Spatially adaptive interpolation of digital images using fuzzy inference," in *Visual Communications and Image Processing*, Proceedings of SPIE, pp. 1206–1217, 1996.

[5] S. W. Lee and J. K. Paik, "Image interpolation using adaptive fast b-spline filtering," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 177–180, Berlin, Germany, September 1993.

[6] Y. Cha and S. Kim, "Edge-forming methods for color image zooming," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2315–2323, 2006.

[7] B. S. Morse and D. Schwartzwald, "Isophote-based interpolation," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 227–231, Chicago, Ill, USA, October 1998.

[8] A. Temizel and T. Vlachos, "Wavelet domain image resolution enhancement," *IEEE Proceedings Vision, Image & Signal Processing*, vol. 153, no. 1, pp. 25–30, 2006.

[9] S. G. Chang, Z. Cvetković, and M. Vetterli, "Locally adaptive wavelet-based image interpolation," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 479–482, 2006.

[10] Y. Zhu, S. C. Schwartz, and M. T. Orchard, "Wavelet domain image interpolation via statistical estimation," in *Proceedings of the International Conference on Image Processing*, vol. 3, pp. 840–843, Thessaloniki, Greece, October 2001.

[11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[12] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[13] Y. Cha and S. Kim, "The error-amended sharp edge (EASE) scheme for image," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1496–1606, 2007.

[14] Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 889–900, 2007.

[15] F. M. Candocia and J. C. Principe, "Super-resolution of images based on local correlations," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 372–380, 1999.

[16] C.-T. Lin, K.-W. Fan, H.-C. Pu, S.-M. Lu, and S.-F. Liang, "An HVS-directed neural-network-based image resolution enhancement scheme for image resizing," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 4, pp. 605–615, 2007.

[17] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2226–2238, 2006.

[18] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 887–896, 2008.

[19] M.-J. Chen, C.-H. Huang, and W.-L. Lee, "A fast edge-oriented algorithm for image interpolation," *Image and Vision Computing*, vol. 23, no. 9, pp. 791–798, 2005.

[20] A. Giachetti and N. Asuni, "Fast artifacts-free image interpolation," in *Proceedings of the British Machine Vision Conference*, pp. 123–132, 2008.

[21] W. S. Tam, C. W. Kok, and W. C. Siu, "A modified edge directed interpolation for images," in *Proceedings of the European Signal Processing Conference*, 2009.

[22] S. Battiato, G. Gallo, and F. Stanco, "A locally-adaptive zooming algorithm for digital images," *Image Vision and Computing*, vol. 20, pp. 805–812, 2002.

[23] S. Battiato, F. Rundo, and F. Stanco, "ALZ: adaptive learning for zooming digital images," in *Proceedings of the International Conference on Consumer Electronics*, 2007.

[24] S. Battiato, E. U. Giuffrida, and F. Rundo, "A cellular neural network for zooming digital colour images," in *Proceedings of the International Conference on Consumer Electronics*, 2008.

[25] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[26] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," in *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*, 2002.

[27] X. Jiang and H. Bunke, "Edge detection in range images based on scan line approximation," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 183–199, 1999.

[28] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, NY, USA, 1977.

[29] Davidone, Open clip art library, http://openclipart.org/media/people/davidone.

[30] A. Gemmi, Open clip art library, http://openclipart.org/media/people/Angelo_Gemmi.