Published in IET Image Processing
Received on 17th May 2010
Revised on 18th February 2012
doi: 10.1049/iet-ipr.2011.0534

ISSN 1751-9659

Image zooming using directional cubic convolution interpolation

D. Zhou[1] *X. Shen*[1] *W. Dong*[2]

[1]*School of Control and Computer Engineering, North China Electric Power University, Beijing, People's Republic of China*
[2]*LIAMA-NLPR, CAS Institute of Automation, Beijing, People's Republic of China*
E-mail: zdw@ncepu.edu.cn

Abstract: Image-zooming is a technique of producing a high-resolution image from its low-resolution counterpart. It is also called image interpolation because it is usually implemented by interpolation. Keys' cubic convolution (CC) interpolation method has become a standard in the image interpolation field, but CC interpolates indiscriminately the missing pixels in the horizontal or vertical direction and typically incurs blurring, blocking, ringing or other artefacts. In this study, the authors propose a novel edge-directed CC interpolation scheme which can adapt to the varying edge structures of images. The authors also give an estimation method of the strong edge for a missing pixel location, which guides the interpolation for the missing pixel. The authors' method can preserve the sharp edges and details of images with notable suppression of the artefacts that usually occur with CC interpolation. The experiment results demonstrate that the authors'method outperforms significantly CC interpolation in terms of both subjective and objective measures.

1 Introduction

Image-zooming is a technique of producing a high-resolution (HR) image from its low-resolution (LR) counterpart. It is also called image interpolation because it is usually implemented by interpolation. Image-zooming is often required in many image processing applications because an HR image can offer more details and a better view. For example, an HR medical image may be very helpful for a doctor to make a correct diagnosis. For image coding, an HR image can be effectively compressed by using its LR counterpart. Classical image-zooming methods include nearest neighbour (NN) interpolation, bilinear (BI) interpolation, cubic convolution (CC) interpolation [1] and cubic spline interpolation [2, 3]. Classical methods attempt to turn discrete data into a continuous function and then resample it. The interpolated continuous function g in one dimension (1D) can be written as [2]

$$g(x) = \sum_{k=1}^{K} c_k u_k(x) \tag{1}$$

where c_k's are the coefficients to be determined from the input data, $u_k(x)$'s are the chosen basis functions or kernels and K is the number of the given data points. The coefficients c_k's are simply sample values for NN, BI and CC interpolations. Equation (1) can be extended to two-dimensional (2D) images by forming the tensor product of the basis functions. Classical methods have a very good interpolation effect on the smooth images. In fact, the intensity surfaces of almost all natural images are highly irregular. They are only smooth along the edge directions, whereas abrupt discontinuity might happen across edge directions. Fig. 1 is an example of the intensity surface of 'Lena' image (Fig. 6*g*). The intensity surfaces of the other images are also similar. Therefore resampled images obtained by classical methods typically incur common artefacts [4] such as blurring, blocking and ringing and so on in the edge regions.

Many edge-directed interpolation methods [5–11] have been proposed in order to remedy the disadvantages of the classical interpolation methods. They try to interpolate a missing pixel along instead of across the edge direction. A crucial step of those methods is to explicitly or implicitly estimate the edge directions in an HR image from its LR counterpart. Jensen and Anastassiou [5] propose to detect the edges in an image using projections onto an orthonormal basis. Li and Orchard [6] use the local covariances of the LR image to estimate the local HR image covariances, which implicitly match the edge directions. This method has been extended by Muresan and Parks [7] based on the optimal recovery principle. Zhang and Wu [8] interpolate the missing samples in two mutually orthogonal directions. Then the two directional estimates are fused through linear minimum mean-square-error (MMSE) principle. Cha and Kim [9] use bilinear interpolation and then try to correct the errors utilising the interpolation error theorem in an edge-adaptive way. Li and Nguyen [10] model an image as the Markov random field, which indicates the edge direction information by an energy function. Recently Giachetti and Asuni [11] propose an iterative curvature-based interpolation (ICBI), which interpolates locally a pixel value along the direction where

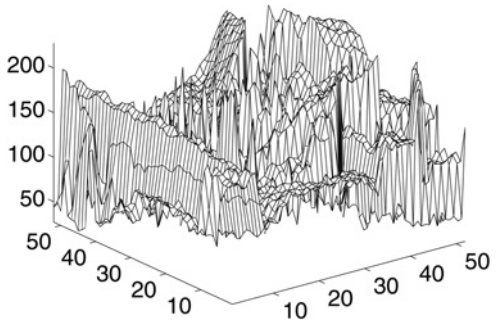

Fig. 1 *Pixel intensity surface of Lena image*

second-order image derivative is lower. Then the interpolated value is modified using an iterative refinement minimising differences in second-order derivatives, maximising second-order derivative values and smoothing isolevel curves.

Another class of the edge-directed interpolation technique is based on the wavelet analysis [12–15]. Those methods try to apply the wavelet transform to an LR image, in order to extract information about the edges of the LR image. Then the detail subbands of the HR image are predicted from those of its LR counterpart. The interpolated HR image can be reconstructed by the inverse wavelet transform. Chang *et al.* [12] and Carey *et al.* [13] extract the edge information by the local maxima in the wavelet domain. Muresan and Parks [14] improve the strategy using the behaviour of edges across scales (i.e. cone influence) in the wavelet domain. Zhu *et al.* [15] characterise edge behaviour by a parameterised discrete time signal and extract the edge information by linear composite MMSE.

In general, the edge-directed image interpolation methods can improve the subjective quality of the interpolated images at the expense of higher computational complexity. The HR images produced using them have the sharper edges than using classical methods. However, they are often apt to cause artefacts in complex edges such as textures.

In this paper, we aim at 2D images and propose a new and simple edge-directed CC interpolation scheme. Our method can preserve both sharp edges and texture details. The experimental results demonstrate that our method not only significantly outperforms CC interpolation but also does several recently published interpolation methods in terms of both subjective and objective measures.

The remainder of this paper is organised as follows. In Section 2, Keys' CC interpolation is briefly introduced. In Section 3, the details of the proposed directional CC interpolation method are described. Section 4 discusses the problem of the parameter choice. Section 5 is the experimental results of the methods under comparison. Finally, conclusion is given in Section 6.

2 CC interpolation

Keys' CC interpolation method [1] performs better than linear interpolation and has become a standard in the interpolation field [16]. If f is a sampled function, then $g(x_k) = f(x_k)$, $k = 1, 2, \ldots, K$. For equally spaced data and 1D signal, the convolution-based interpolation function g can be written in the form (1). The coefficient c_k is simply a sample point $f(x_k)$ (denoted by f_k) and $u_k(x)$ can be represented as $u(x - x_k/h)$ in which h denotes the sampling increment and the x_k's are the interpolation nodes, $x_k = kh$. Without loss of generality, we will use $h = 1$ and (1) can be simplified as

$$g(x) = \sum_k f_k u(x - k) \tag{2}$$

The CC interpolation kernel is composed of piecewise cubic polynomials defined on the subintervals $(-2, -1)$, $(-1, 0)$, $(0, 1)$ and $(1, 2)$. Outside the interval $(-2, 2)$, the interpolation kernel is zero. The interpolation kernel $u(s)$ is defined as

$$u(s) = \begin{cases} (a+2)|s|^3 - (a+3)|s|^2 + 1, & 0 \le |s| < 1 \\ a|s|^3 - 5a|s|^2 + 8a|s| - 4a, & 1 \le |s| < 2 \\ 0, & 2 \le |s| \end{cases} \tag{3}$$

Keys derives the optimal constant $a = -1/2$ by forcing the Taylor series expansion of the sampled function f to agree in as many terms as possible with the CC interpolation function g. For this choice of a, CC interpolation yields a third-order approximation of the original function. For all other choices of a, it yields only a first-order approximation, just like NN interpolation. Keys also derives the CC kernel with fourth-order approximation at the cost of a larger spatial support.

Using the constant $a = -1/2$, the CC interpolation kernel is

$$u(s) = \begin{cases} \frac{3}{2}|s|^3 - \frac{5}{2}|s|^2 + 1, & 0 \le |s| < 1 \\ -\frac{1}{2}|s|^3 + \frac{5}{2}|s|^2 - 4|s| + 2, & 1 \le |s| < 2 \\ 0, & 2 \le |s| \end{cases} \tag{4}$$

The CC method uses four sample points for each interpolation point. For two times interpolation, $s = 1/2$ at every interpolation point, the CC interpolation filter is $[-1, 9, 9, -1]/16$. The 2D CC interpolation function is a separable extension of the 1D interpolation function. 2D interpolation can be accomplished by 1D interpolation with respect to each coordinate. Readers interested in more details about the CC interpolation are referred to [1].

3 Proposed algorithm

CC interpolates indiscriminately the missing pixels in the same direction (horizontal or vertical) and so non-horizontal or non-vertical edges are smoothed. Edge-directed interpolation methods try to first detect edge directions, and then interpolate along the detected edge directions. The common problem is that the detected edges are usually inaccurate, especially in the regions with complex edges such as textures. We propose an estimation method of the strong edge for a missing pixel location, which guides the interpolation for the missing pixel. In order to avoid the difficulties detecting weak edges, we propose a novel interpolation scheme. Interpolating a missing pixel on a strong edge applies CC interpolation along the estimated strong edge direction, while interpolating a missing pixel on a weak edge by fusing the two orthogonal directional CC interpolation results.

An LR image I_l can be considered to be a directly downsampled version of the HR image I_h corresponding to I_l as shown in Fig. 2, where the downsampling factor is 2 and the grey and white dots are missing pixels while the black dots are the pixels in the known LR image I_l. That is, the HR image I_h is restored by copying the LR image I_l's pixels into an enlarged grid and then filling with the

doi: 10.1049/iet-ipr.2011.0534

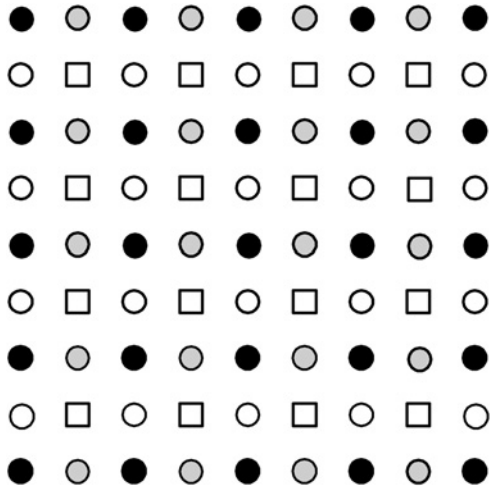

Fig. 2 *Formation of an LR image I_l from an HR image I_h by downsampling*

Black dots represent the LR image pixels and the grey and white dots represent the missing HR pixels. Interpolation is performed in two passes. The first pass interpolates the missing pixels marked by white squares and the second pass interpolates the pixels marked by grey and white circles

missing pixels. We can see from Fig. 2 that, for two times zoom, three quarters of the pixels of the HR image I_h are missed. The missing pixels are interpolated in two passes. The pixels marked by white squares are interpolated in the first pass. The other missing pixels are interpolated in the second pass which includes two cases: interpolating the pixels marked by grey circles and interpolating the pixels marked by white circles.

3.1 Edge detection

To interpolate the missing pixels, the first step is to detect the edge directions. The gradients of images are usually used to predict edge directions because the gradient values across the edges are greater than the ones along them. For an analogue image, the edge direction at a point is arbitrary, but for a digital image, there are only the four potential edge directions at a pixel in a 3×3 neighbourhood: horizontal, vertical, 45° diagonal and 135° diagonal. Thus, here we only consider the edges of the four directions. Furthermore, we estimate the edge direction at a pixel in a larger neighbourhood than 3×3, that is, 7×7 or 5×5 neighbourhood. In the following, we estimate the direction of location (i, j) in an HR image by computing the two orthogonal directional gradients using the known pixels in the neighbourhood of (i, j), and assume that the pixel intensity value at location (x, y) is represented as $I(x, y)$.

1. Computing the gradients in the first pass: For every pixel to be estimated in the first pass, there are the four immediate diagonal pixels to be known as illustrated by Fig. 2. Thus, we estimate the strength of the edges in 45° and 135° diagonal directions in terms of the gradients in the two directions, respectively. According to the known pixels in the 7×7 neighbourhood referring to Fig. 3, the gradients at the central location (i, j) are computed as

$$G_1 = \sum_{m=3,\pm 1} \sum_{n=3,\pm 1} |I(i+m, j-n) - I(i+m-2, j-n+2)|$$

(45° diagonal)

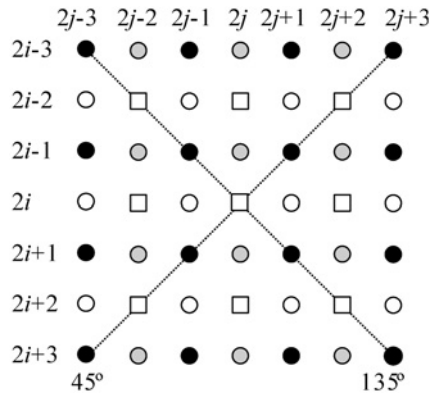

Fig. 3 *Interpolation in the first pass*

For each pixel to be estimated (i.e. the central pixel in the figure), the gradients in the 45° and 135° diagonal directions are computed in the 7×7 neighbourhood. The black circles are the known LR image pixels, whereas the other circles (grey and white) and squares are the missing HR image pixels

$$G_2 = \sum_{m=3,\pm 1} \sum_{n=3,\pm 1} |I(i+m, j+n) - I(i+m-2, j+n-2)|$$

(135° diagonal)

(5)

2. Computing the gradients in the second pass: The interpolated pixels in the first pass can be considered to be known. Interpolation in the second pass includes two cases as illustrated by Fig. 4. Here for every pixel to be estimated, the four immediate pixels in the horizontal and vertical directions are known. Fig. 4*a* represents that two immediate pixels in the vertical direction are the estimated values in the first pass and that two immediate pixels in the horizontal direction are from the known LR image. Similarly, Fig. 4*b* represents that two immediate pixels in the horizontal direction are the estimated values in the first pass and that two immediate pixels in the vertical direction are from the known LR image. Thus, we estimate the strength of the edges in the horizontal and vertical directions in terms of the gradients in the two directions. The computational procedure of the two cases in Figs. 4*a* and *b* is exactly identical. We only require considering either case. Now we consider the 5×5 neighbourhood around the central pixel (i, j) in Fig. 4*a*. With the aid of the estimated pixels in the first pass, the gradients at the central

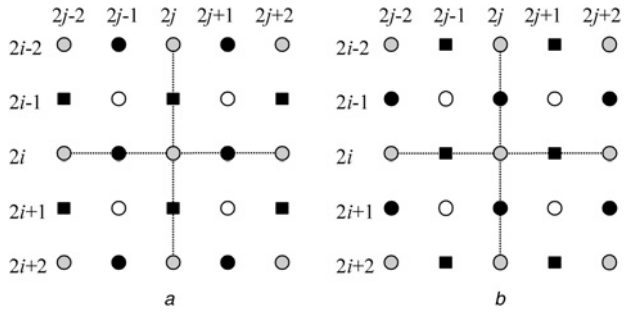

Fig. 4 *Interpolation in the second pass*

For each pixel to be estimated (i.e. the central pixels in the figure), the gradients in the horizontal and vertical directions are computed in the 5×5 neighbourhood. The black circles are the known LR image pixels, the black squares are the estimated pixels in the first pass and the other circles (grey and white) are the missing HR image pixels

location (i, j) are computed as

$$G_1 = \sum_{m=\pm 1}\sum_{n=0,2} |I(i+m, j-n) - I(i+m, j-n+2)| + \sum_{m=0,\pm 2} |I(i+m, j-1) - I(i+m, j+1)| \text{ (horizontal)}$$

$$G_2 = \sum_{m=0,2}\sum_{n=\pm 1} |I(i-m, j+n) - I(i-m+2, j+n)| + \sum_{n=0,\pm 2} |I(i-1, j+n) - I(i+1, j+n)| \text{ (vertical)} \quad (6)$$

With the two orthogonal directional gradients G_1 and G_2 for a pixel location (x, y), the edge direction at the pixel can be estimated. Since the gradient is small when along an edge, we thus estimate the edge direction using the ratio of the two orthogonal directional gradients

$$\begin{cases} \text{if}(1+G_1)/(1+G_2) > T \\ \quad \text{the pixel } (x, y) \text{ is on a } 135^\circ \text{ or vertical strong edge;} \\ \text{else if}(1+G_2)/(1+G_1) > T \\ \quad \text{the pixel } (x, y) \text{ is on a } 45^\circ \text{ or horizontal strong edge;} \\ \text{else} \\ \quad \text{the pixel } (x, y) \text{ is in a weak or textured region;} \\ \text{end} \end{cases} \quad (7)$$

where 1 is added to the gradients to avoid division by zero and T is a threshold parameter, which is a key factor for the decision whether a pixel is on a strong edge. The choice on T will be discussed in Section 4.

3.2 Algorithm description

The estimated edge direction at a missing pixel position can be used to estimate the pixel intensity value. For a missing pixel on a strong edge, we estimate the pixel value by CC interpolation along the strong edge. Since a local gradient is greater across an edge than along it, the gradient can indicate the strength of a local edge. Therefore for a missing pixel in a weak edge or textured region, the pixel value is estimated by combining the two orthogonal directional CC interpolation results, whose contributions to the missing pixel value should be in inverse proportion to the gradients corresponding to their interpolation directions. Assuming that the two orthogonal directional CC interpolation values at location (x, y) are p_1 (45° diagonal or horizontal directional interpolation value) and p_2 (135° diagonal or vertical directional interpolation value), the weights combining p_1 and p_2 are computed as

$$\begin{cases} w_1 = \dfrac{1}{1+G_1^k} \\ w_2 = \dfrac{1}{1+G_2^k} \end{cases} \quad (8)$$

where k is an exponent parameter adjusting the weighting effect, whose choice will be discussed in Section 4. As in (7), 1 is added to the denominators to avoid division by zero. The interpolation value p of the missing pixel at location (x, y) in a weak or textured region is estimated as

$$p = (w_1 p_1 + w_2 p_2)/(w_1 + w_2) \quad (9)$$

We will give a detailed description of the interpolation process of the proposed method in the following. The zooming factor applied here is two in both horizontal and vertical directions. The interpolation for other factors which are powers of two can be achieved by iteratively performing the following algorithm.

1. Initialisation: this step expands the known $n \times n$ LR image I_l onto the grid of the $(2n-1) \times (2n-1)$ HR image I_h as shown in Fig. 2. The mapping relation is $I_h(2x-1, 2y-1) = I_l(x, y)$, $x, y = 1, 2, \ldots, n$ where (x, y) is the pixel location.
2. Interpolation in the first pass: the pixels represented by white squares in Fig. 2 are interpolated. The filled pixels are $I_h(2x, 2y)$, $x, y = 1, 2, 3, \ldots, n$ (including border pixels). For every pixel to be interpolated, compute the 45° and 135° diagonal directional CC interpolation values p_1 and p_2 and compute the two diagonal directional gradients G_1 and G_2 using (5). Then the edge direction at the pixel is estimated by (7) and the pixel value p is estimated as

$$\begin{cases} \text{if } (1+G_1)/(1+G_2) > T \\ \quad p = p_2; \\ \text{else if } (1+G_2)/(1+G_1) > T \\ \quad p = p_1; \\ \text{else} \\ \quad \text{Compute weights using (8) and } p \text{ using (9);} \\ \text{end} \end{cases} \quad (10)$$

3. Interpolation of Case 1 of the second pass: the pixels represented by grey circles in Fig. 2 are interpolated. The filled pixels are $I_h(2x-1, 2y)$, $x, y = 1, 2, 3, \ldots, n$ (including border pixels). For every grey pixel, compute the horizontal and vertical CC interpolation values p_1 and p_2 and compute the horizontal and vertical gradients G_1 and G_2 using (6). Then the missing pixel p is estimated using (10).
4. Interpolation of Case 2 of the second pass: this step is exactly identical to step 3. The pixels represented by white circles in Fig. 2 are interpolated. The filled pixels are $I_h(2x, 2y-1)$, $x, y = 1, 2, 3, \ldots, n$ (including border pixels).

4 Selection of threshold T and exponent k

The results of the proposed method depend on the choice of the threshold T in (10) and the exponent k in (8). Unfortunately, the two parameters cannot be easily determined for a given LR image because they depend on the varying scene structures of images. We derive T and k through training. Twenty-four 512×768 Kodak colour images [17] were used as training samples, which were first converted into grey images and then downsampled the grey images to obtain their LR counterparts. The HR images were reconstructed from the LR ones using the proposed method with the different T and k. k was separately set from 1 to 6 by step length 1, whereas T varied from 1 to 1.5 by step length 0.05 for every k. The average peak signal-to-noise ratio (PSNR) curves are shown for different cases in Fig. 5. It can be seen that the higher PSNR values can be achieved when the greater exponent k is used. However, the curves for $k = 5$ and $k = 6$ are nearly the same, which implies that the PSNR values cannot be further improved when $k > 5$; so we set $k = 5$ in all our experiments. On the other hand, the PSNR values increase gradually when $T \in [1, 1.25]$, while decreasing slowly when $T > 1.25$. $T = 1.25$ seems to be optimal. Considering that a greater T

doi: 10.1049/iet-ipr.2011.0534

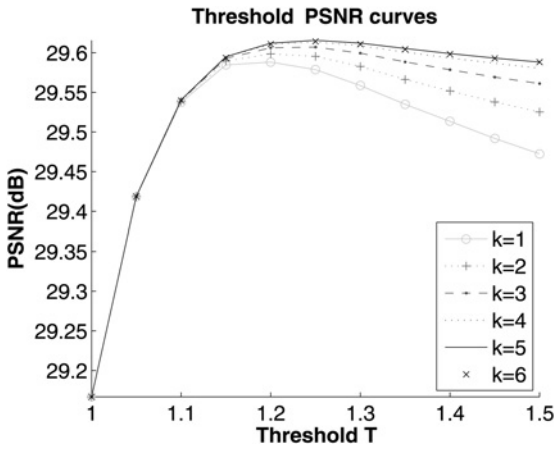

Fig. 5 *Training results of the parameters T and k in (10) and (8) using the grey versions of 24 Kodak colour images [17]*

Threshold-PSNR curves represent the average PSNR values under different threshold T and exponent k values

probably results in losing some strong edges, $T = 1.15$ may be appropriate although the average PSNR values are slightly lower comparing with the results when $T = 1.25$. Throughout our experiments, the threshold T was set to 1.15.

The two parameters T and k have been obtained by training 24 samples. To test how the number and contents of training samples affect the parameters, we repeat the above process using the grey versions of two hundred sample images selected from Caltech 101 datasets [18], which are widely used in object recognition community. The average PSNR curves are relatively similar to Fig. 5, which indicates that the selection for the threshold T and exponent k is rather robust in terms of average PSNR performance.

5 Experimental results

The proposed method was compared with the other four representative interpolation methods, which are classical CC [1], directional filtering and data fusion (DFDF) [8], error-amended sharp edge (EASE) [9], ICBI [11]. CC interpolation was implemented by Matlab's 'INTERP2' function. The Matlab codes of DFDF and ICBI methods were available from the original authors. We used default settings for ICBI code. The EASE algorithm was based on our own implementation. For thoroughness and fairness of our comparison study, we selected the eight representative testing images (Fig. 6), including a synthetic image 'Rings' (Fig. 6*a*) which was first used by Keys [1], a cartoon image 'Flinstones' (Fig. 6*b*) and six natural images (Figs. 6*c*–*f*). Those testing images represent different image types, which are significantly different in contents. We downsampled the original HR grey images by a factor of two in both row and column dimensions to obtain the LR images, from which the zoomed HR images were reconstructed by different methods. Since the original HR images are known, we can measure the PSNRs of the zoomed HR images. All errors were calculated excluding the border of twelve pixels around the images to reduce the boundary effects. The exponent k in (8) was set to 5 and the threshold T in (10) was set to 1.15. Table 1 tabulates the PSNR results of the five different methods for two times zoom. The highest PSNR value of each row is shown in bold. The proposed method consistently outperforms the other methods for all

Fig. 6 *Set of testing images*

a Rings
b Flinstones
c Bike
d Airplane
e Flower
f Parrot
g Lena
h Monarch

testing images, and exceeds the average PSNR value of the second best method by 1.36 dB. The greatest PSNR improvement was 4.79 dB over the second best method for the 'Rings' image. For the image 'Parrot' with rich textures, the proposed method also has a PSNR improvement of approximately 1 dB over the second best method.

Table 2 tabulates the structural similarity (SSIM) results of the five different methods for two times zoom. SSIM is a method that provides a quality measurement of images

Table 1 PSNR (dB) results of the reconstructed HR images by different methods for 2 times zoom

Image	CC [1]	DFDF [8]	EASE [9]	ICBI [11]	Proposed
Rings	49.32	46.54	41.65	41.69	**54.11**
Flinstones	26.95	27.05	26.53	27.03	**27.79**
bike	25.56	25.65	25.55	25.20	**26.39**
airplane	29.76	30.25	29.94	29.69	**30.68**
Flower	31.69	31.84	31.66	31.74	**32.54**
Parrot	33.30	33.13	33.06	33.12	**34.26**
Lena	33.80	33.77	33.59	33.88	**34.39**
monarch	30.14	30.75	30.09	30.78	**31.30**
average	32.57	32.37	31.51	31.64	**33.93**

Highest PSNR of each row is shown in bold

Table 2 SSIM results of the reconstructed HR images by different methods for 2 times zoom

Image	CC [1]	DFDF [8]	EASE [9]	ICBI [11]	Proposed
Rings	**0.9999**	0.9998	0.9993	0.9995	**0.9999**
Flinstones	0.9633	0.9643	0.9627	0.9621	**0.9682**
bike	0.9492	0.9457	0.9474	0.9447	**0.9537**
airplane	0.9680	0.9686	0.9692	0.9654	**0.9712**
Flower	0.9773	0.9768	0.9769	0.9769	**0.9801**
Parrot	0.9809	0.9798	0.9808	0.9795	**0.9831**
Lena	0.9720	0.9708	0.9720	0.9701	**0.9742**
monarch	0.9857	0.9864	0.9859	0.9860	**0.9885**
average	0.9745	0.9740	0.9743	0.9730	**0.9774**

Highest SSIM of each row is shown in bold

based on structural content, which measures the similarity between two images (the original and the reconstructed images) [19]. It gives results between 0 and 1, where 1 means the best quality and 0 means the worst quality. The Matlab source code of the SSIM indices is available online at [20]. The proposed method still consistently outperforms the other methods for all testing images. It can be observed that SSIM measure has a good correlation with PSNR measure but there exist exceptions. EASE falls behind DFDF and ICBI on average PSNR but outperforms ICBI and DFDF in terms of average SSIM metric, although DFDF has 0.86 dB improvement on average PSNR over EASE.

It is well known that objective metric is not completely in accordance with subjective tests, the evaluation of the visual appearance of the interpolated images is required for different methods. The first image to be inspected is the synthetic image 'Rings'. The original image was downsampled by a factor of four. Then the downsampled image was magnified four times. The interpolated results are shown in Fig. 7. In this example, CC and DFDF introduce the prominent aliasing artefacts near the 45° direction looking at the lower-right corners of Figs. 7*b* and *c*. It is interesting that EASE and ICBI do significantly better than CC and DFDF near the 45° direction, but they introduce the obvious aliasing artefacts near the horizontal and vertical directions looking at the lower-left and upper-right corners of Figs. 7*d* and *e*. The interpolated image produced using the proposed

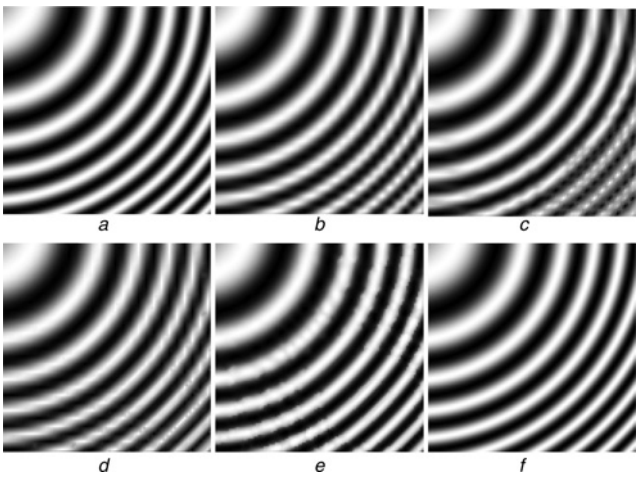

Fig. 7 *4 times zoom of the synthetic image 'Rings'*
a Original
b CC
c DFDF
d EASE
e ICBI
f Proposed

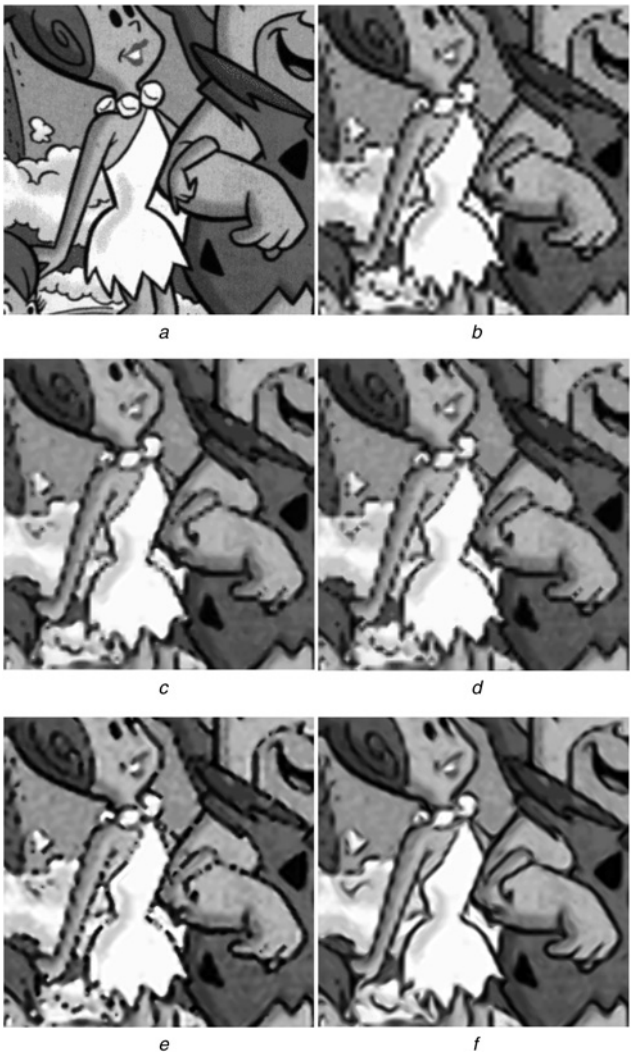

Fig. 8 *Part of the 4 times zoomed image 'Flinstones'*
a Original
b CC
c DFDF
d EASE
e ICBI
f Proposed

doi: 10.1049/iet-ipr.2011.0534

method is far superior to the others and has hardly any visual artefacts looking at Fig. 7*f*.

Fig. 8 exhibits part of the four times zoomed results of the cartoon image 'Flinstones'. This image has exaggerated contour borders and is a good example for testing the edge recovery ability of the different methods. CC (Fig. 8*b*) introduces the most severe blocking and blurring artefacts. Since EASE is only to amend the bilinear interpolation error, its capability to recover edges is inferior to the other methods except CC, but its computational cost is very low compared with the others except CC. ICBI (Fig. 8*e*) can retain sharper edges than DFDF (Fig. 8*c*) and EASE (Fig. 8*d*), but it also tends to result in strong discontinuities on a long edge, that is, without natural and smooth transitions, paying attention to the man's hand and the rim of the girl's skirt. The proposed methods best recovers the edge structures (Fig. 8*f*).

Figs. 9 and 10 are parts of two times zoomed natural image Lena and Flower. The proposed method performs best and eliminates many of the ringing, aliasing and other visual artefacts of the other methods, which is exemplified by the reconstructed rims of the hat and cheek in Lena (Fig. 9*f*) and by the flower petals and left pedicel in Flower (Fig. 10*f*). The visual quality of the reconstructed HR images produced by CC (Figs. 9*b* and 10*b*) is the worst although its average PSNR measure is the second highest. It can also be seen again that DFDF (Figs. 9*c* and 10*c*) is superior to EASE (Figs. 9*d* and 10*d*), whereas ICBI (Figs. 9*e* and 10*e*) is superior to DFDF and EASE for the reconstruction of edge structures, next only to the proposed method. It should also be noted that the proposed method can reconstruct the fine texture details. Our method reconstructs satisfactorily the flower petals and has the least aliasing as shown in Fig. 10*f*, whereas the other methods are, to some extent, blurring and noisy.

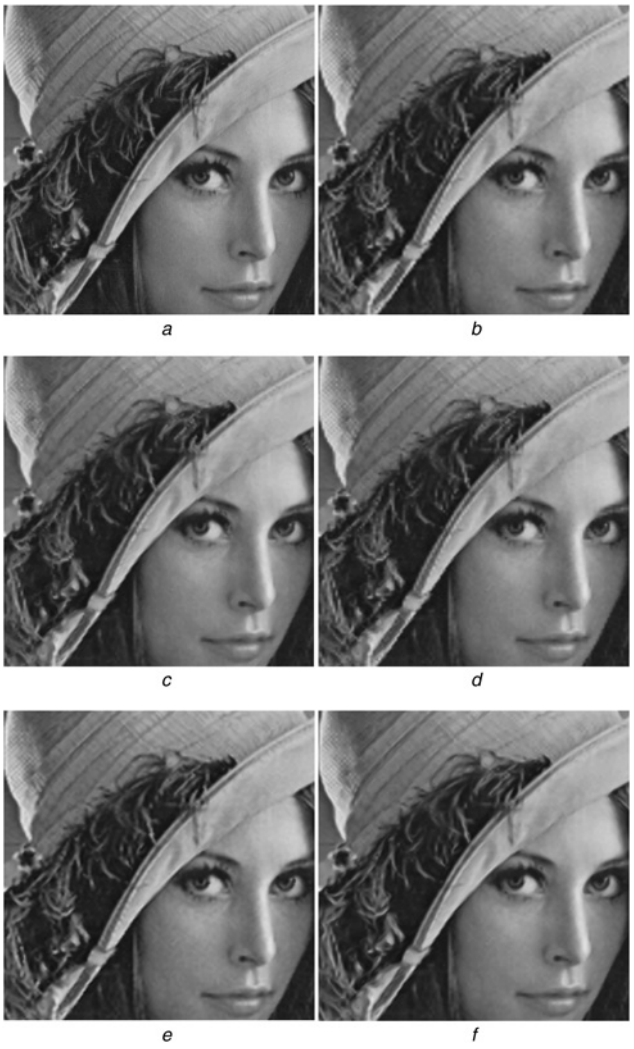

Fig. 9 *Part of the 2 times zoomed image Lena*

a Original
b CC
c DFDF
d EASE
e ICBI
f Proposed

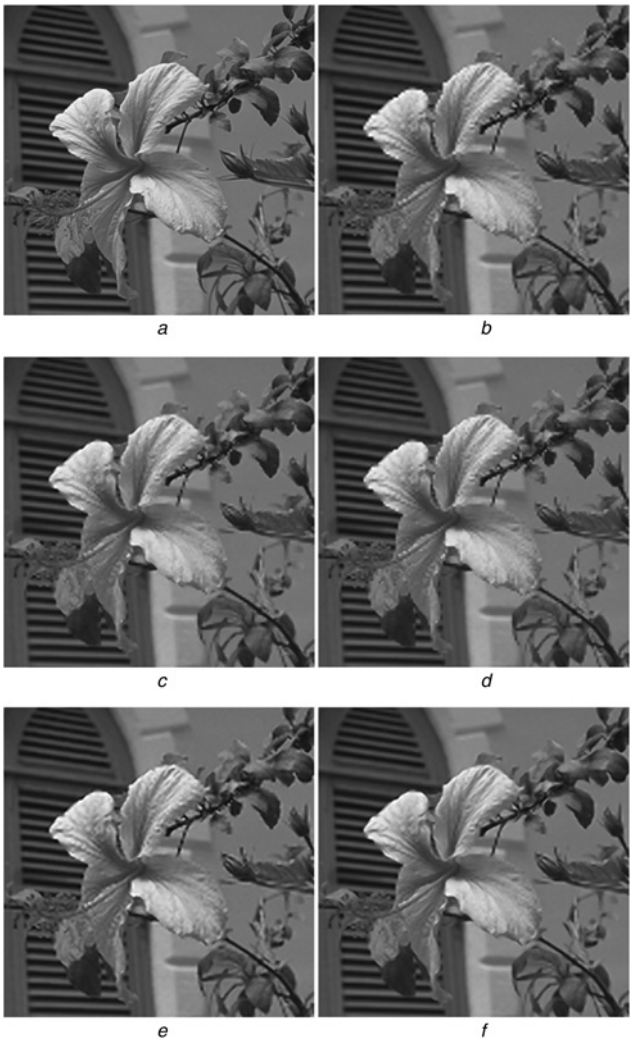

Fig. 10 *Part of the 2 times zoomed image Flower*

a Original
b CC
c DFDF
d EASE
e ICBI
f Proposed

6 Conclusion

We propose a new edge-directed image-zooming method. This method is an extension of the classical CC interpolation proposed by Keys [1]. CC interpolates indiscriminately the missing pixels in the same direction

(horizontal or vertical) and typically results in blurring, blocking, ringing or other artefacts in zoomed images. Our method first explicitly decides the local edge direction in terms of the radio of the two orthogonal directional gradients for a missing pixel position. If the pixel is on a strong edge, it will be estimated using CC interpolation value along the edge direction; if the pixel is in a weak edge or textured region, it will be estimated by combining the two orthogonal directional CC interpolation values in terms of the gradients in the two directions. The experiment results show that our method can effectively suppress the common artefacts in zoomed images. Our method not only outperforms significantly the CC interpolation but also does several recently published interpolation methods in terms of both subjective and objective measures. However, the results of our method are dependent on the selection of T and k. Although the two parameters have been obtained by training, it is still needed for further research to find a better solution to the two key parameters.

7 Acknowledgments

The authors thank Dr. L. Zhang of the Hong Kong Polytechnic University and Dr. Giachetti of the University of Verona for making their Matlab codes available online. This work was supported by the Beijing Natural Science Foundation under grant no. 4112061, the National Natural Science Foundation of China under grant no. 61172104 and Beijing-funded joint development program of the central universities in Beijing (the sharing management information system of the large instruments of the key scientific research platforms of the capital universities).

8 References

1 Keys, R.G.: 'Cubic convolution interpolation for digital image processing', *IEEE Trans. Acoust. Speech Signal Process.*, 1981, **ASSP-29**, (6), pp. 1153–1160
2 Hou, H.S.: 'Cubic splines for image interpolation and digital filtering', *IEEE Trans. Acoust. Speech Signal Process.*, 1978, **ASSP-26**, (6), pp. 508–517
3 Unser, M., Aldroubi, A., Eden, M.: 'Fast B-spline transforms for continuous image representation and interpolation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1991, **13**, (3), pp. 277–285
4 Thevenaz, P., Blu, T., Unser, M.: 'Image interpolation and resampling', in Bankman, I.N. (Ed.): 'Handbook of medical imaging, processing and analysis, analysis' (Academic, New York, 2000), pp. 393–420
5 Jensen, K., Anastassiou, D.: 'Subpixel edge localization and the interpolation of still images', *IEEE Trans. Image Process.*, 1995, **4**, (3), pp. 285–295
6 Li, X., Orchard, M.T.: 'New edge-directed interpolation', *IEEE Trans. Image Process.*, 2001, **10**, (10), pp. 1521–1527
7 Muresan, D.D., Parks, T.W.: 'Adaptively quadratic (aqua) image interpolation', *IEEE Trans. Image Process.*, 2004, **13**, (5), pp. 690–698
8 Zhang, L., Wu, X.: 'An edge-guided image interpolation algorithm via directional filtering and data fusion', *IEEE Trans. Image Process.*, 2006, **15**, (8), pp. 2226–2238
9 Cha, Y., Kim, S.: 'The error-amended sharp edge (ease) scheme for image zooming', *IEEE Trans. Image Process.*, 2007, **16**, (6), pp. 1496–1505
10 Li, M., Nguyen, T.: 'Markov random field model-based edge-directed image interpolation', *IEEE Trans. Image Process.*, 2008, **17**, (7), pp. 1121–1128
11 Giachetti, A., Asuni, N.: 'Fast artifacts-free image interpolation'. Proc. British Machine Vision Conf., Leeds, UK, 2008, pp. 123–132
12 Chang, S.G., Cvetkovic, Z., Vetterli, M.: 'Locally adaptive wavelet-based image interpolation', *IEEE Trans. Image Process.*, 2006, **15**, (6), pp. 1471–1485
13 Carey, W.K., Chuang, D.B., Hemami, S.S.: 'Regularity-preserving image interpolation', *IEEE Trans. Image Process.*, 1999, **8**, (9), pp. 1293–1297
14 Muresan, D.D., Parks, T.W.: 'Prediction of image detail'. Proc. IEEE Int. Conf. on Image Processing, 2000, vol. 2, pp. 323–326
15 Zhu, Y., Schwartz, S.C., Orchard, M.T.: 'Wavelet domain image interpolation via statistical estimation'. Proc. IEEE Int. Conf. on Image Processing, 2001, vol. 3, pp. 840–843
16 Blu, T., Thevenaz, P., Unser, M.: 'Linear interpolation revitalized', *IEEE Trans. Image Process.*, 2004, **13**, (5), pp. 710–719
17 Available at http://www.site.uottawa.ca/126edubois/demosaicking/
18 Available at http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html
19 Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: 'Image quality assessment: from error visibility to structural similarity', *IEEE Trans. Image Process*, 2004, **13**, (4), pp. 600–612
20 Wang, Z.: 'SSIM Index for Image Quality Assessment', http://www.ece.uwaterloo.ca/126z70wang/research/ssim/

doi: 10.1049/iet-ipr.2011.0534