Directional Bicubic Interpolation — A New Method of Image Super-Resolution

Liu Jing , Gan Zongliang and Zhu Xiuchang

Abstract Bicubic interpolation is a standard method in image interpolation field because of its low complexity and relatively good results. But as it only interpolates in horizontal and vertical directions, edges easily suffer from artifacts such as blocking, blurring and ringing. This paper proposed a new method of image super-resolution which is named directional bicubic interpolation. According to local strength and directions, different ways are used to interpolate missing pixels. Compared with bicubic interpolation, the proposed method can preserve sharp edges and details better. Experiment results show that the proposed method is better than existing edge-directed interpolations in terms of subjective and objective measures, and its computation complexity is low.

Keywords Bicubic • Local gradients • Direction extraction • Edge strength • Direction parameter

1 Introduction

As high-resolution (HR) images can provide more details and better perceptual quality, image interpolation is required to produce a HR image from its low-resolution (LR) counterpart in many fields, such as medical images, HD videos, satellite imaging, mobile devices (e.g. iphone, ipad, etc) and so on.

Liu Jing (🖂)
Jiangsu Provincial Key Lab of Image Processing and Image Communication, Nanjing University of Posts and Telecommunications, Nanjing, China
e-mail: liujingnj@ gmail.com

Gan Zongliang
Jiangsu Provincial Key Lab of Image Processing and Image Communication, Nanjing University of Posts and Telecommunications, Nanjing, China

Zhu Xiuchang
Jiangsu Provincial Key Lab of Image Processing and Image Communication, Nanjing University of Posts and Telecommunications, Nanjing, China

© 2013. The authors - Published by Atlantis Press

Traditional polynomial-based interpolations include bilinear, bicubic [1] and cubic spline interpolation [2], which have low computation complexity, but easily suffer from artifacts along edges or textures. To remedy this disadvantage, many edge-directed interpolations are proposed. They try to interpolate missing pixels along edges as much as possible. Jensen and Anastassiou [3] detect edges by using an edge fitting operator. Li and Orchard [4] estimate local covariance of HR images with the help of the local LR covariance to match the edge directions. Yang [5] detects local gradient direction with principal components analysis (PCA) as the direction of the edge. Zhou [6] compares gradients of two orthogonal directions to compute local edge strength and direction and use different ways to interpolate missing pixels with different strengths and directions.

Compared with traditional interpolation methods, edge-directed interpolations can preserve shape edges better, but may also generate some artifacts in texture areas. In this paper, we propose a novel method of image super-resolution for 2D images. Simulation results show that compared to existing edge-directed interpolations, our proposed method achieves better results, and at the same time, its computation complexity is low.

2 Proposed Algorithm

1D bicubic only interpolates along horizontal or vertical directions, which results in non-horizontal or non-vertical being smoothed. To solve this problem, we first estimate edge strength and direction through local image gradients. Then for pixels on strong edges or on weak edges and textures, interpolate by different ways.

2.1 Local Edge Direction

Before interpolation, the first step is to detect local edge direction. As gradients across edges are larger than along edges, local gradients are used for detecting edge directions. For a digital image, four directions (0°, 45°, 90°and 135°) are considered in a 7×7 neighborhood as follows:

$$G_{0^\circ} = \sum_{m=\pm 1}\sum_{n=3,\pm 1}\left|I(i+m, j-n) - I(i+m, j-n+2)\right| + \sum_{m=\pm 3}\left|I(i+m, j-1) - I(i+m, j+1)\right|$$

$$G_{90^\circ} = \sum_{m=3,\pm 1}\sum_{n=\pm 1}\left|I(i-m, j+n) - I(i-m+2, j+n)\right| + \sum_{n=\pm 3}\left|I(i-1, j+n) - I(i+1, j+n)\right|$$

$$
\begin{aligned}
G_{45^\circ} = {} & \sum_{n=3,\pm1} \left| I(i+1, j-n) - I(i-1, j-n+2) \right| + \sum_{n=\pm1} \left| I(i+3, j-n) - I(i+1, j-n+2) \right| \\
& + \sum_{n=3,1} \left| I(i-1, j-n) - I(i-3, j-n+2) \right| \\
& + \frac{1}{2} \sum_{m=3,-1} \left| I(i+m, j-m) - I(i+m-2, j-m+2) \right| \\
G_{135^\circ} = {} & \sum_{n=3,\pm1} \left| I(i-1, j-n) - I(i+1, j-n+2) \right| + \sum_{n=\pm1} \left| I(i-3, j-n) - I(i-1, j-n+2) \right| \\
& + \sum_{n=3,1} \left| I(i+1, j-n) - I(i+3, j-n+2) \right| \\
& + \frac{1}{2} \sum_{m=3,-1} \left| I(i-m, j-m) - I(i-m+2, j-m+2) \right|
\end{aligned}
\tag{1}
$$

Suppose the direction across edge is between the biggest two gradients $G_{\max 1}$, $G_{\max 2}$ of $G_{0^\circ}, G_{45^\circ}, G_{90^\circ}, G_{135^\circ}$. Compute the ratio of $G_{\max 1}$, $G_{\max 2}$, where 1 is added to avoid division by zero. If $(1+G_{\max 1})/(1+G_{\max 2}) > T$, pixel (i, j) is on a strong edge, the direction of the edge is across $G_{\max 1}$; otherwise, pixel (i, j) is on a weak edge, the direction of the edge is across a direction between the two gradients.

2.2 Algorithm Description

In the paper, we divide pixels into two classes: pixels on strong edges and pixels on weak edges or textures. Take the interpolation of pixel $I_{HR}(2i,2j)$ for example.

For pixels on strong edges, 1D bicubic interpolation is used along edge directions directly. As shown is (2) [5], if edge direction is 45° or 135°, we choose four nearest pixels along the direction and the coefficients are[-1, 9, 9, -1]/16. While if edge direction is 0° or 90°, we choose eight nearest pixels perpendicular to the direction, and coefficients are [-1, 9, 9, -1, -1, 9 , 9 ,-1]/32.

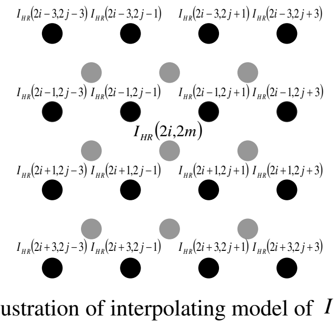

Fig. 1 Illustration of interpolating model of $I_{HR}(2i,2j)$

$$I_{HR(2i,2j)} = \begin{cases} -\frac{1}{32} I_{HR(2i-1,2j-3)} + \frac{9}{32} I_{HR(2i-1,2j-1)} + \frac{9}{32} I_{HR(2i-1,2j+1)} - \frac{1}{32} I_{HR(2i-1,2j+3)} \\ -\frac{1}{32} I_{HR(2i+1,2j-3)} + \frac{9}{32} I_{HR(2i+1,2j-1)} + \frac{9}{32} I_{HR(2i+1,2j+1)} - \frac{1}{32} I_{HR(2i+1,2j+3)} \\ \hfill \theta = 0^\circ \\ -\frac{1}{16} I_{HR(2i+3,2j-3)} + \frac{9}{16} I_{HR(2i+1,2j-1)} + \frac{9}{16} I_{HR(2i-1,2j+1)} - \frac{1}{16} I_{HR(2i-3,2j+3)} \\ \hfill \theta = 45^\circ \\ -\frac{1}{32} I_{HR(2i-3,2j-1)} + \frac{9}{32} I_{HR(2i-1,2j-1)} + \frac{9}{32} I_{HR(2i+1,2j-1)} - \frac{1}{32} I_{HR(2i+3,2j-1)} \\ -\frac{1}{32} I_{HR(2i-3,2j+1)} + \frac{9}{32} I_{HR(2i-1,2j+1)} + \frac{9}{32} I_{HR(2i+1,2j+1)} - \frac{1}{32} I_{HR(2i+3,2j+1)} \\ \hfill \theta = 90^\circ \\ -\frac{1}{16} I_{HR(2i-3,2j-3)} + \frac{9}{16} I_{HR(2i-1,2j-1)} + \frac{9}{16} I_{HR(2i+1,2j+1)} - \frac{1}{16} I_{HR(2i+3,2j+3)} \\ \hfill \theta = 135^\circ \end{cases} \tag{2}$$

For pixels on weak edges or textures. Firstly, interpolate along directions across the two biggest gradients $G_{\max 1}$, $G_{\max 2}$ as (2) respectively. Then merge them with proper weights. Suppose p_1, p_2 are bicubic results corresponding to $G_{\max 1}$ and $G_{\max 2}$, weights can be computed as

$$\begin{cases} w_1 = \dfrac{1}{1 + G_{\max 2}^k} \\ w_2 = \dfrac{1}{1 + G_{\max 1}^k} \end{cases} \tag{3}$$

where k is an exponent parameter, and 1 is added to avoid division by zero. The interpolation result p of pixels on weak edges or textures can be estimate as

$$p = (w_1 p_1 + w_2 p_2)/(w_1 + w_2) \tag{4}$$

To summarize the method, an informative description of our method is given as follows:

Input: low resolution image I_{LR}.

Output: high resolution image I_{HR}.

Begin

Initialization. Enlarge I_{LR} to I_{HR} through $I_{HR}(2i-1, 2j-1) = I_{LR}(i, j)$, $1 \le i \le M, 1 \le j \le N$, where *M*, *N* is the size of LR image.

Interpolating pixels $I_{HR}(2i,2j)$. Before interpolating, first compute its local variance σ . If $\sigma < T_{\sigma}$, use bilinear interpolation directly. Otherwise:

Compute gradients of four directions as (1).

Extract the two biggest gradients $G_{\max 1}$, $G_{\max 2}$, and interpolate the pixel p as follows:

$$\begin{cases} (1+G_{\max 1})/(1+G_{\max 2}) > T \\ p = p_1 \\ (1+G_{\max 1})/(1+G_{\max 2}) < T \\ p = (w_1 p_1 + w_2 p_2)/(w_1 + w_2) \end{cases} \tag{5}$$

where p_1 , p_2 are the bicubic interpolation results corresponding to $G_{\max 1}$ and $G_{\max 2}$. Here threshold T_{σ} , parameter k in (3) and threshold T in (5) are predefined as 10, 5 and 1.15 respectively, in which T_{σ} is chosen based on experience and the others are chosen by training 20 images.

Interpolating pixels $I_{HR}(2i-1,2j)$ and $I_{HR}(2i,2j-1)$. Interpolation is conducted in the same way as step 2 except rotating the axis by 45°.

End

3 Experiments and Analyses

In this section, we compare our results with four methods: Bicubic interpolation [1], NEDI [4], PCAI [5] and DCCI [6]. Fig.2 displays the test images.

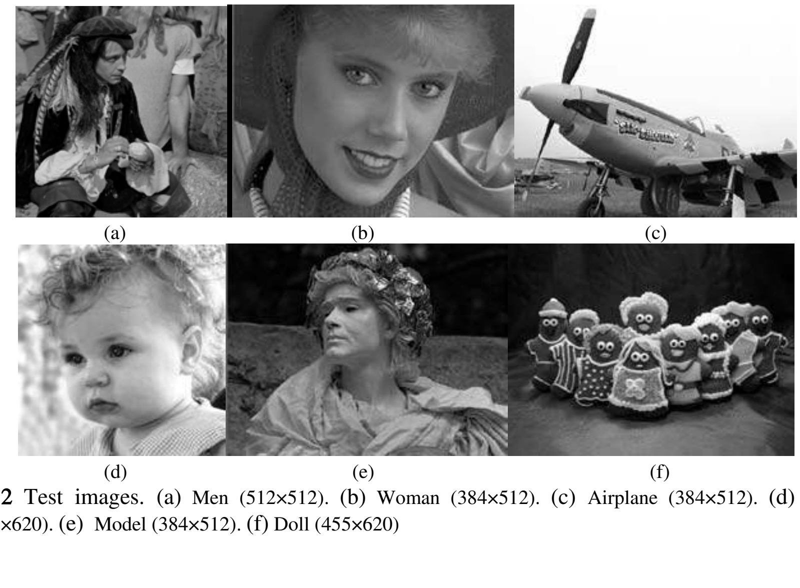

Fig.2 Test images. (a) Men (512×512). (b) Woman (384×512). (c) Airplane (384×512). (d) Baby (602×620). (e) Model (384×512). (f) Doll (455×620)

3.1 Experiment Results

Table 1 reports the PSNR results under five methods for two times zoom. It can be observed that the proposed method has the best result, which is 0.46dB, 0.44dB, 0.42dB, 0.15dB higher than Bicubic [1], NEDI [4], PCAI [5] and DCCI [6] respectively. Table 2 and Table 3 compare structural similarity (SSIM) [7] and feature similarity (FSIM) [8] results, which also show that our method is better.

Table 1 Comparison of five different methods in PSNR (dB)

Images	Bicubic[1]	NEDI[4]	PCAI[5]	NCCI[6]	Proposed
Men	26.44	26.70	26.43	26.70	**26.79**
Woman	35.43	35.30	35.35	35.57	**35.79**
Airplane	29.32	29.74	29.66	30.14	**30.22**
Baby	32.08	32.18	32.13	32.42	**32.52**
Model	31.63	31.63	31.45	31.74	**31.82**
Doll	28.32	27.76	28.43	28.50	**28.83**
Average	30.54	30.55	30.58	30.85	**31.00**

Table 2 Comparison of five different methods in SSIM [7]

Images	Bicubic[1]	NEDI[4]	PCAI[5]	NCCI[6]	Proposed
Men	0.7983	0.8014	0.7963	0.8032	**0.8059**
Woman	0.9029	0.9014	0.8982	0.9007	**0.9051**
Airplane	0.9261	0.931	0.9291	0.9348	**0.9364**
Baby	0.8968	0.8969	0.8924	0.8948	**0.8989**
Model	0.9246	0.9217	0.9230	0.9250	**0.9253**
Doll	0.8542	0.8487	0.8543	0.8551	**0.8617**
Average	0.8838	0.8835	0.8822	0.8856	**0.8889**

Table 3 Comparison of five different methods in FSIM [8]

Images	Bicubic[1]	NEDI[4]	PCAI[5]	NCCI[6]	Proposed
Men	0.9619	0.9621	0.9595	0.9627	**0.9634**
Woman	0.9564	0.9540	0.9573	0.9579	**0.9581**
Airplane	0.9429	0.9506	0.9485	0.9538	**0.9540**
Baby	0.9468	0.945	0.9472	0.9481	**0.9485**
Model	0.9865	0.9856	0.9853	0.9864	**0.9866**
Doll	0.9733	0.9701	0.9713	0.9731	**0.9746**
Average	0.9613	0.9612	0.9615	0.9637	**0.9642**

Fig.3 shows partition results of 'Airplane'. It can be seen that the proposed method has the best results. Bicubic interpolation [1] has the most severe blurring (Fig. 3(b)). NEDI [4] breaks the geometric duality between the LR covariance and HR covariance, and generates speckle noise (Fig. 3(c)). PCAI [5] simply divides directions into four classes which doesn't match the structure of real images, and has obvious blurring artifacts (Fig. 3(d)). DCCI [6] generates shape edges but has weak subjective results (Fig. 3(e)). Fig.4 and Fig.5 are partition results of 'Airplane' and 'Doll', which also show similar results.

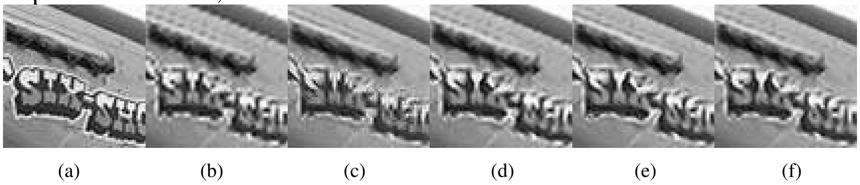

Fig. 3 The portion result of 'Airplane' image under five methods. (a)Original image. (b) Bicubic [1]. (c) NEDI [4]. (d) PCAI [5]. (e) DCCI [6]. (f) Proposed

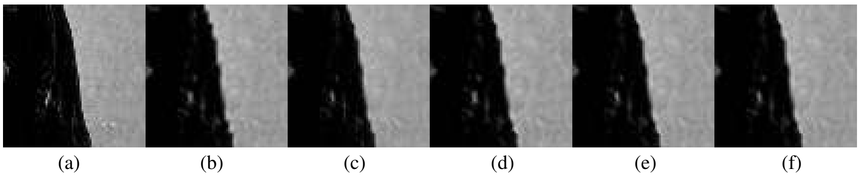

Fig. 4 The portion result of 'Men' image under five methods. (a) Original image (b) Bicubic [1]. (c) NEDI [4]. (d) PCAI [5]. (e) DCCI [6]. (f) Proposed

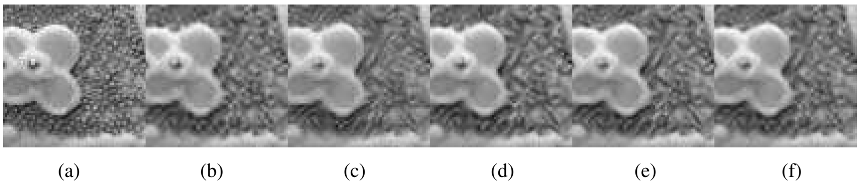

Fig. 5 The portion result of 'Doll' image under five methods. (a) Original image (b) Bicubic [1]. (c) NEDI [4]. (d) PCAI [5]. (e) DCCI [6]. (f) Proposed

3.2 Computation Complexity

To decrease computation complexity, we dramatically use bilinear interpolation based on variance σ. Table 4 is the average percentage of pixels interpolated by bilinear and the proposed method about 6 test images in this paper. It can be seen that about 33.50% of the pixels can use bilinear interpolation directly.

Table 4 Average percentage of pixels interpolated by bilinear interpolation and the proposed method

Percentage	Bilinear	Proposed
Average	33.50%	66.50%

4 Conclusion

In this paper, directional bicubic interpolation — a novel method of image super-resolution is proposed. Unlike 1D bicubic interpolation [1] which only interpolates along horizontal and vertical directions, the proposed method increases flexibility to interpolation along arbitrary directions. Experiments show that the proposed method is better than modern edge-directed interpolations in both subjective and objective measures. Moreover, as we dramatically use bilinear interpolation based on local image variance, the computation complexity is low and this method can be used in real time.

Acknowledgements This study is funded by the National Nature Science Foundation, P. R. China (No. 61071091, No. 61071166, No.61172118) and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (Information and Communication Engineering).

References

1. Keys, R. (1981). Cubic convolution interpolation for digital image processing. Acoustics, Speech and Signal Processing, IEEE Transactions on, 29(6), 1153-1160.
2. Hou, H., & Andrews, H. (1978). Cubic splines for image interpolation and digital filtering. Acoustics, Speech and Signal Processing, IEEE Transactions on, 26(6), 508-517.
3. Jensen, K., & Anastassiou, D. (1995). Subpixel edge localization and the interpolation of still images. Image Processing, IEEE Transactions on, 4(3), 285-295.
4. Li, X., & Orchard, M. T. (2001). New edge-directed interpolation. Image Processing, IEEE Transactions on, 10(10), 1521-1527.
5. Yang, B., Gao, Z., & Zhang, X. (2012, July). Principal Components Analysis-Based Edge-Directed Image Interpolation. In Multimedia and Expo (ICME), 2012 IEEE International Conference on (pp. 580-585). IEEE.
6. Zhou, D., Shen, X., & Dong, W. (2012). Image zooming using directional cubic convolution interpolation. Image Processing, IET, 6(6), 627-634.
7. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. Image Processing, IEEE Transactions on, 13(4), 600-612.
8. Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011). FSIM: a feature similarity index for image quality assessment. Image Processing, IEEE Transactions on, 20(8), 2378-2386.