A new directional image interpolation based on Laplacian operator

SAID OUSGUINE[1], FEDWA ESSANNOUNI[1], LEILA ESSANNOUNI[1], MOHAMMED ABBAD[1], DRISS ABOUTAJDINE[1]

[1] LRIT Research Laboratory (associated unit to CNRST, URAC n° 29),
Mohammed V University, Faculty of Sciences
Rabat, MAROCCO

lamak07@hotmail.fr

Abstract: The interpolation task plays a key role in the reconstruction of high-resolution image quality in super-resolution algorithms. In fact, the foremost shortcoming encountered in the classical interpolation algorithms, is that they often work poorly when used to eliminate blur and noise in the input image. In this sense, the aim of this work is to develop an interpolation scheme for the purpose of reducing these artifacts in the input image, and consequently preserve the sharpness of the edges. The proposed method is based on the image interpolation, and it is started by the estimation of the edges directions using the Laplacian operator, and then interpolated the missing pixels from the strong edge by using the cubic convolution interpolation. We begin from a gray high-resolution image that is down-sampled by a factor of two, to obtain the low-resolution image, and then reconstructed using the proposed interpolation algorithm. The method is implemented and tested using several gray images and compared to other interpolation methods. Simulation results show the performance of the proposed method over the other methods of image interpolation in both PSNR, and two perceptual quality metrics SSIM, FSIM in addition to visual quality of the reconstructed images results.

Key–Words: Image interpolation, Cubic convolution, Laplacian operator, Edge detection

1 Introduction

In this paper, we address the problem of image interpolation, to reconstruct a high-resolution image from a low-resolution one. The image interpolation is an important task for image resizing, up-scaling and image enhancement. In addition, the image interpolation is widely applied in a variety of applications, such as remote sensing, surveillance, medical imaging, computer vision, high-resolution television and consumer electronics. Although the sensor technology of digital cameras, camcorders, and scanners have attained great developments in the past decades, image interpolation using software methodology is still wanted as a useful and effective solution to the hardware system for the extension of spatial resolution.

The principle process of image interpolation is to transform an image from one resolution to another resolution without losing its visual content, to obtain as an output a high-resolution image with good visual appearance and quality.

Interpolation algorithms can be classified in two categories: adaptive class and non-adaptive class or by combining the both classes.

Adaptive classes are methods that use the analysis of the local structure of image, such as edge information and intensity variation. Their principle is to restore sharp edges in the high-resolution image and texture. Thus, to guarantee the success of these methods, the interpolation method must be done along the edges. In this class, we can find color filter array, edge directions and threshold based interpolation methods. In contrast, non-adaptive methods treat the full image in a uniform way. These algorithms use mathematical interpolation methods, which include cubic interpolation, the nearest neighbor, etc. In fact, this class of methods fails to keep the integrity of edge structures and suffers from the artifact (jagged edges in the up-scaling) around edge areas.

Several approaches based on models that are more sophisticated have been suggested in the literature. Ramponi [1] proposes to use the warped distance to the interpolated pixel instead of a regular one, and so changing the one-dimensional kernel of a separable interpolation filter. However, the visual quality of the interpolated edges remains reduced. An improvement of this technique was made by Huang and Lee [2]. They change the two-dimension interpolation kernel by the characteristics of the local gradient. However, the necessity to adjust a sharpness parameter for each image and the absence of a special textured area processing, are the main disadvantages of this algorithm. Another methods of interpolation based on edge detection was introduced in [3, 4, 14]. In [3], a projection on an orthonormal basis to detect the edge di-

rection has been applied. Another approach, NEDI, was proposed by Li and Orchard in [4]. The optimal interpolation in terms of the mean squared error was suggested under the assumption that the local image covariance is constant in a large window and at different scales. The method succeeds in upscaling edges of different orientations, but in the areas of texture and clutter, the assumption made becomes false, leading to unnatural images and even strong artifacts. Moreover, the computational complexity of NEDI makes it is inappropriate for the real-time processing.
Using some tricks to adapt window size and to handle matrix conditioning , we obtained a modified (iNEDI) method see [14], providing sharp images although with a high computational cost and a rather unnatural texture in high frequency regions. Li and Zhang [6] propose an edge-guided algorithm via directional filtering and data fusion. For each unknown pixel, its neighbors are divided into two subsets. Using the subsets, two estimate pixel values are generated. Then a more robust estimate pixel value is obtained by a data fusion method.
The authors in [9] note that the assumptions about image continuity lead to over smoothed edges in common image interpolation algorithms and suggest a wavelet-based interpolation method with no continuity limitations. The algorithm estimates the regularity of edges by computing the decay of wavelet transform coefficients across scales and conserves the fundamental regularity by extrapolating a new subband to be used in image resizing. Then the HR image is constructed by reverse WT. Muresan and Parks [5] extended this strategy through the influence of a full cone sharp edge in the wavelet scale space, rather than just the top module, for estimation of the best coefficients of scale through an optimal recovery theory. An interpolation scheme was proposed by Giachetti and Asuni [8], its procedure is to interpolate locally the missing pixels in the two diagonal directions, while the second order image derivative is lower, these interpolated pixels values are revised using an iterative refinement to minimize the differences in second order image derivative. Cha and Kim [7] describe an interpolation method by using a bilinear interpolation and adjusting the errors by adopting the interpolation error theorem in an edge-adaptive manner. In method [25], another type of interpolation methods which use the kernel regression was presented. This latter determines an adaptive steering kernel regression along the edge direction using covariance matrices, and it can cover a broad range of edge angle. Zhou and Shen [10] propose an image zooming using cubic convolution interpolation with detecting the edge-direction, this method is based on the detection of the edge-direction of the missing pixels. In [11], the authors develop an interpolation method based on the analysis of the local structure on the images. They classified the image into two partitions: homogenous zones and edge areas. Specified algorithms are assigned to interpolate each classified areas, respectively. In [26], a contrast -guided image interpolation method is proposed that includes contrast information into the image interpolation process. Given the image under interpolation, four binary contrast-guided decision maps (CDMs) are generated and used to guide the interpolation filtering through two sequential stages: 1) the 45° and 135° CDMs for interpolating the diagonal pixels and 2) the 0° and 90° CDMs for interpolating the row and column pixels.

Here we propose a new method of image interpolation using the Laplacian operator. The missing pixels are interpolated along the detected edge direction; this latter is estimated by the Laplacian operator in the 45° diagonal and 135°diagonal, horizontal and vertical directions. A cubic convolution interpolation is used after to interpolate the missing pixels along the strong edge. The paper is organized as follows: Section II describes the proposed interpolation method. Simulation results are shown in section III. Finally, the conclusion is drawn in section IV.

2 The proposed algorithm

A low resolution image LR can be considered to be a directly down sampled version of the HR image as shown in Figure 1, where the downsampling factor is 2. In this figure the circles are the missing pixels while the black dots are the pixels in the known LR image. That is, the HR image is restored by copying the LR image's pixels into an enlarged grid and then filling with the missing pixels. We can see from this figure that, for two times zoom, three quarters of the pixels of the HR image are missing. Also, three types of the missing pixels can be distinguished, pixels of type 1 with even coordinates; pixels of type 2 with even row index and odd column index, and finally pixels of type 3 with odd row index and even column index.

To estimate the missing pixels many interpolation methods have been proposed in the literature. Traditional interpolation methods such as the bilinear and bicubic interpolation methods may create blurring, produce ringing artifacts near the edges. These methods interpolate the missing pixels in the same direction, and lead to a jagged aliasing effect along edges. In contrast, edge-directed interpolation methods detect the edge orientations, and then interpolate along the detected edge direction. The main problem is that the detected edge are usually imprecise, mainly in the regions with complex textures.

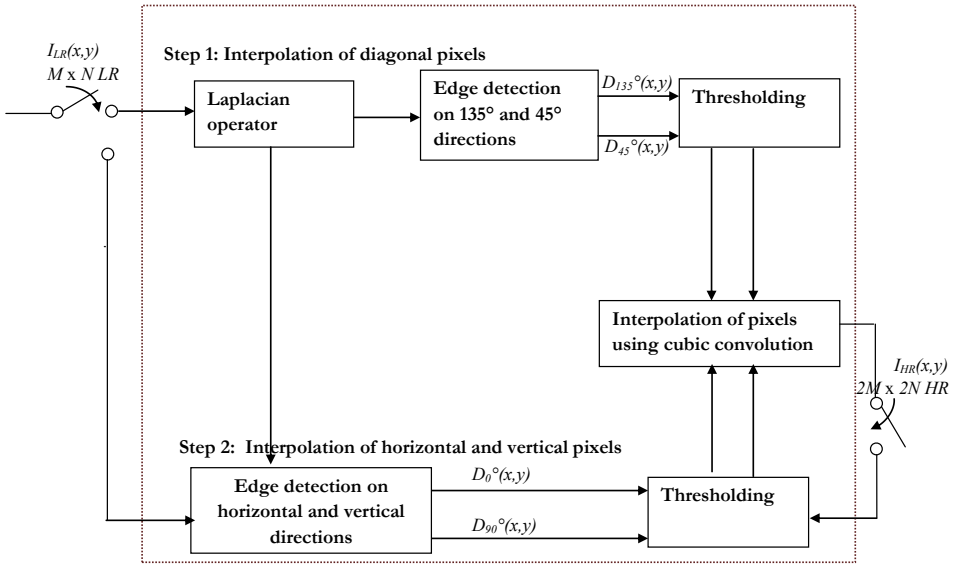


Figure 2: Framework of the proposed image interpolation based on Laplacian operator.

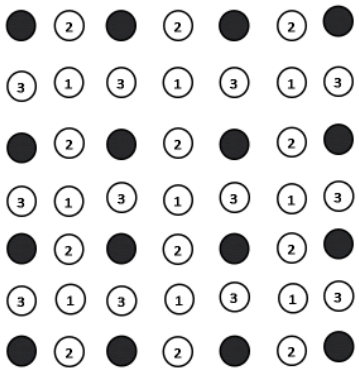


Figure 1: Formation of an LR image from an HR image by downsampling. The black dots represent the LR image pixels, while the other circles represent the missing HR pixels.

In order to avoid detecting weak edges, we propose a novel interpolation method based on the Laplacian operator. This method interpolates the missing pixel on the detected edge and then applying the cubic convolution interpolation along the estimated edge direction. An overview of the proposed framework is shown in Figure 2.

2.1 Edge direction detection

To interpolate the missing pixels in the high resolution image, the first step is to detect the edge directions using the Laplacian operator. The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. This transform highlights regions of rapid intensity change and is therefore often used for edge detection [27, 28]. The operator normally takes a single gray level image as input and produces another gray-level image as output.
The Laplacian $LP(x, y)$ of the LR image is given by:

$$LP(x,y) = \frac{\partial^2 LR}{\partial^2 x} + \frac{\partial^2 LR}{\partial^2 y} \tag{1}$$

This can be calculated using a convolution filter. Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. The commonly used small kernel can be written as below:

$$H_\alpha = \begin{bmatrix} \frac{\alpha}{\alpha+1} & \frac{1-\alpha}{\alpha+1} & \frac{\alpha}{\alpha+1} \\ \frac{1-\alpha}{\alpha+1} & \frac{-4}{\alpha+1} & \frac{1-\alpha}{\alpha+1} \\ \frac{\alpha}{\alpha+1} & \frac{1-\alpha}{\alpha+1} & \frac{\alpha}{\alpha+1} \end{bmatrix} \tag{2}$$

The parameter α controls the shape of the Laplacian and must be in the range 0.0 to 1.0. The choice of this parameter will be discussed later.
Two edge directions are examined for every missing pixel, horizontal and vertical directions for pixels of

type 2 and 3, 45° and 135° diagonal directions for pixels of type 1. The edge direction is detected using a window of size 5×5 in the Laplacian transform of the LR image. This window is chosen centered at the concerned missing pixel.

1. For every pixel of type 1 to be estimated whose coordinates $(x, y) = (2i, 2j)$, we estimate the strength of the edge in 45° and 135° diagonal directions as follows:

$$D_{45^\circ} = \sum_{m=-1}^{2}\sum_{n=-2}^{1} |LP(i+m, j+n) - LP(i+m-1, j+n+1)| \quad (3)$$

$$D_{135^\circ} = \sum_{m=-1}^{2}\sum_{n=-1}^{2} |LP(i+m, j+n) - LP(i+m-1, j+n-1)| \quad (4)$$

where LP is the Laplacian transform of the LR image.

2. For pixels of types 2 and 3 whose coordinates (x, y) can be written respectively as $(2i-1, 2j)$ and $(2i, 2j-1)$ (see Figure 4). We estimate the strength of the edges in the horizontal and vertical directions as follows:

$$D_{0^\circ} = \sum_{m=-2}^{2}\sum_{n=-2}^{1} |Lp(i+m, j+n) - Lp(i+m, j+n+1)| \quad (5)$$

$$D_{90^\circ} = \sum_{m=-2}^{2}\sum_{n=-2}^{1} |Lp(i+m, j+n) - Lp(i+m, j+n+1)| \quad (6)$$

For the rest of the paper, let D_1 denote, D_{0° for pixels of types 2 and 3, or D_{45° for pixels of type 1. Let D_2 denote D_{90° for pixels of types 2 and 3, or D_{135° for pixels of type 1.

$$\begin{cases} \text{if } (D1 - D2) < 0.5 \\ \text{The angle of orientation of the edge is } 45^\circ \text{ or } 0^\circ; \\ \text{elseif } (D2 - D1) < 0.5 \\ \text{The angle of orientation of the edge is } 135^\circ \text{ or } 90^\circ; \\ \text{else} \\ \text{the pixel is on homogeneous or textured region} \\ \text{end} \end{cases}$$

2.2 Missing pixel intensity estimation

The interpolation of the missing pixels is performed in two passes. In the first pass, we interpolate the pixels of type 1. In the second pass, we interpolate the pixels of types 2 and 3 using the known low resolution pixels and the already predicted pixels of type 1.

The estimated edge direction at a missing pixel position is used to estimate the pixel intensity value. This latter is calculated using the cubic convolution interpolation along the strongest edge.
For a missing pixel in a weak edge or textured region, the pixel value is estimated by combining the two orthogonal directional cubic interpolation results. If we denote by I_1 the estimation of the intensity value of the missing pixel at the 45° diagonal or horizontal direction and I_2 the estimation of the intensity at the 135° diagonal or vertical direction, the interpolation value of the intensity I of the missing pixel at location (x, y) in a homogenous or textured region is estimated as:

$$I = \frac{(w_1 I_1 + w_2 I_2)}{(w_1 + w_2)} \quad (7)$$

The weights w_1 and w_2 are computed as :

$$\begin{cases} w_1 = \dfrac{1}{1 + D_1^k} \\ w_2 = \dfrac{1}{1 + D_2^k} \end{cases} \quad (8)$$

where k is an exponent parameter adjusting the weighting effect, whose choice will be discussed in the next section.

2.3 Selection of the Laplacian filter

The results of the proposed method depend on the choice of the Laplacian shape α and the exponent k in Equation (8). These parameters cannot be directly determined for a given LR image because they depend on the types of the images. We derive α and k through training. Twenty-four 512×768 Kodak colour images [24] 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 values of α. The parameter α was separately set from 0.1 to 1 by step length 0.1. The average peak signal-to-noise ratio (PSNR) curves is shown for different cases in Figure 5. It can be seen that the higher PSNR values can be achieved when the greater α is used.

On the other hand, we can see from this figure that the PSNR achieves its maximum values when $k = 4$.

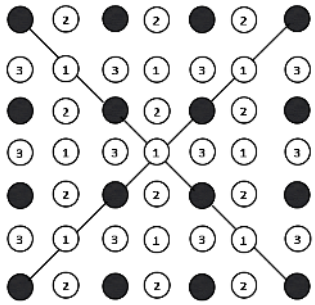


Figure 3: Interpolation of missing pixels in 135° and 45° diagonals .

(a) (b)

Figure 4: Interpolation of missing pixels (a) Type (1); (b) Type (2).

3 Simulation and results

To verify the performance, a set of experiments are performed to more comprehensively evaluate the effectiveness of the proposed method, this latter has been implemented in Matlab and tested on several images, and compared to the conventional interpolation algorithms, cubic convolution interpolation (CC) [12], directional filtering and data fusion (DFDF) [6], fast artifacts-free Image interpolation (ICBI) [8], new edge-directed interpolation (NEDI) [4], Accuracy improvements and improvement new edge-directed interpolation (iNEDI) [14], kernel Regression for Image Processing and Reconstruction (KR) [25] and Contrast-guided Image Interpolation (CGI) [26].

CC was implemented by Matlab "interp2" function, the other Matlab codes were available from the original authors. Refer to Figure 6, nine gray test images are selected and used for simulation. Nine gray test images as shown in Figure 6 were used for simulation.

We started from an original high-resolution gray image, this latter was down-sampled by a factor of two to obtain the low resolution image. The LR image was reconstructed using the previously cited methods (for the proposed method, α has been set to 1 and k to 4).

Various experimental factors are used to examine the quality of the interpolated images in terms of both subjective and objective measures, the first one is the subjective quality of the output images, and for this the results for each of the methods are shown in Figures 7-9. The second factor is the statistic parameters which are the quantitative measures namely the Peak signal to noise ratio (PSNR) was measured for each of the output images, to determine the difference between original image and the reconstructed image, The structural similarity (SSIM) measures the similarity between two images [21] and the Feature Similarity Index (FSIM) measures the feature similarity index between two images [22]. Tables 1, 2 and 3 show the resulting PSNR, SSIM and FSIM scores of the different methods.

Table 1: Comparison of PSNR (dB) results of the reconstructed images

Image	CC[12]	DFDF [6]	ICBI[8]	KR[25]	NEDI[4]	INEDI[14]	CGI[26]	Proposed
Bee	34,39	35,05	34,31	34,37	32,11	33,32	34,99	**35,37**
Face	40,53	40,84	39,79	40,26	38,41	39,19	41,20	**41,47**
Flinstones	26,93	27,03	27,00	26,72	25,28	27,03	27,58	**27,60**
Lena	33,81	33,77	33,87	33,96	32,79	33,20	34,32	**34,55**
Monarch	30,16	30,71	30,85	30,78	28,92	**31,26**	31,12	31,24
Parrot	33,31	33,32	33,10	30,76	32,20	33,49	34,19	**34,45**
Peppers	32.83	33.02	32.99	31.54	32.39	32.63	33.27	**33,45**
Rings	45,61	42,18	46,53	26,29	30,11	28,66	53,85	**54,44**
Watch	31,91	31,80	31,68	31,77	30,70	31,84	32,42	**32,55**
Average	34,38	34,19	34,45	31,82	31,43	32,29	35,88	**36,13**

Table 2: Comparison of SSIM results of the reconstructed images

Image	CC[12]	DFDF [6]	ICBI[8]	KR[25]	NEDI[4]	INEDI[14]	CGI[26]	Proposed
Bee	0,9934	0,9926	0,9940	0,9917	0,9883	0,9917	0,9941	**0,9943**
Face	0,9926	0,9915	0,9927	0,9906	0,9889	0,9916	0,9931	**0,9932**
Flinstones	0,9633	0,9632	0,9620	0,9642	0,9540	0,9637	0,9665	**0,9676**
Lena	0,9720	0,9721	0,9700	0,9716	0,9688	0,9692	0,9732	**0,9745**
Monarch	0,9858	0,9862	0,9862	0,9860	0,9820	0,9869	0,9879	**0,9885**
Parrot	0,9812	0,9806	0,9799	0,9806	0,9782	0,9807	0,9822	**0,9835**
Peppers	0,9597	0,9632	0,9562	**0,9642**	0,9641	0,9571	0,9606	0,9640
Rings	0,9999	0,9995	0,9996	0,9985	0,9892	0,9825	0,9999	**0,9999**
Watch	0,9809	0,9813	0,9790	0,9802	0,9728	0,9824	0,9835	**0,9839**
Average	0,9810	0,9811	0,9800	0,9808	0,9763	0,9784	0,9823	**0,9832**

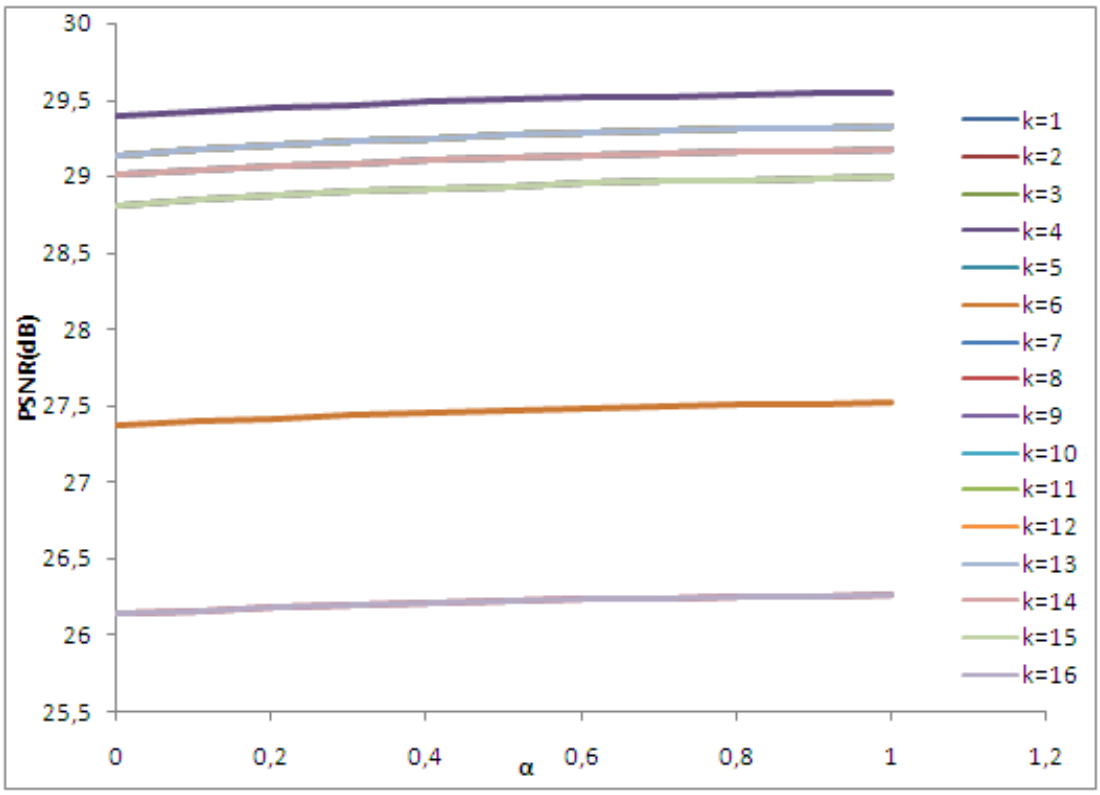


Figure 5: Selection of parameter k and the laplacian filter shape α.

The proposed method was compared with state-of-the-art of image interpolation methods, in the part of the subjective appearance of the resulted image, as can be seen in Figures 7-9, the filetering-based bicubic interpolation method [12] tends to blur image details and contains distinct artifacts and generates prominent jaggies along sharp edges, this method is in general inferior to the others in visual quality. The NEDI [4], KR [25] and INEDI [14] are very competitive in terms of visual quality, this is primarily because they preserve long edges well and yield sharper interpolation image, but these methods are still affected by the speckle noise and ringing effect which can influence the subjective quality of the image. DFDF [6] and ICBI [8] interpolation methods take a middle ground between the bicubic interpolation [12] and edge-directed interpolation NEDI [4], KR [25] and INEDI [14], they reproduce sharper large scale edges than the bicubic method, but the reconstruction of these methods is not as good as the method of [4] and [14] in subjective quality. It is clear that the proposed method and the algorithm of [26] produce a sharp high-resolution image and outperform the above-mentioned interpolation methods by producing a high visual image quality.

On the other hand, from tables 1, 2 and 3, the highest PSNR, SSIM and FSIM values of each row are shown in bold. The PSNR results demonstrate the outperforming of the proposed method over the above-mentioned methods for all testing images, and exceeds the average PSNR value of the second best method by 0.25 dB. The better PSNR improvement was 0.59 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 0.26 dB over the second best method. We can also see that our method achieves the highest SSIM and FSIM index.

4 Conclusion

In this paper, we have developed a novel method of image interpolation based on the Laplacian operator. This method consists in interpolating the missing pixel along the detected edge direction using the cubic convolution interpolation. From the experimental results, it's obvious that the proposed method gives the best enhancement in both quantitative metrics performance evaluation and subjective visual quality. Moreover, it generates a sharper high-resolution image with the highest PSNR, SSIM and FSIM index than the other methods. Therefore, the proposed method can keep a number of directional edges in the interpolation process, and suppress many of ringing, aliasing,

Figure 6: Set of images test (a) Bee; (b) Face; (c) Flinstones; (d) Lena; (e) Monarch; (f) Parrot; (g) Peppers; (h) Rings; (i) Watch

blurring and other visual artifact in zoomed image occurred by the habitual interpolation methods.

References:

[1] G. Ramponi, Warped distance for space-variant linear image interpolation, *IEEE Trans. Image Process* 62(5), 1999, pp. 629-639.

[2] J. W. Hwang and H. S. Lee, Adaptive image interpolation based on local gradient features, *IEEE Signal Process. Lett.* 11(3), 2004, pp. 359-362.

[3] K. Jensen and D. Anastassiou, Subpixel edge localization and the interpolation of still images, *Image Process. IEEE Trans.* 4(3), 1995, pp. 285-295.

[4] X. Li and M. T. Orchard, New edge-directed interpolation, *IEEE Trans. Image Process.* 10(10), 2001, pp. 1521-1527.

[5] D. Muresan and T. W. Parks, Prediction of image detail, *Process. Int. Conf. Image 2000.* 2, 2000.

[6] L. Zhang and X. Wu, An edge-guided image interpolation algorithm via directional filtering and data fusion, *Image Process. IEEE Trans.* 15(8), 2006, pp. 2226-2238.

[7] Y. Cha and S. Kim, The error-amended sharp edge (EASE) scheme for image zooming, *Image Process. IEEE Trans.* 16(6), 2007, pp. 1496-1505.

[8] A. Giachetti and N. Asuni, Fast Artifacts-Free Image Interpolation, *Br. Mach. Vis. Conf.*, Leeds UK, 2008, pp. 123-132.

[9] W. K. Carey and S. S. Hemami, Regularity-preserving image interpolation, *IEEE Trans. Image Process.* 8(9), 1999, pp. 1293-1297.

[10] Z. Dengwen, S. Xiaoliu, D. Weiming, Image zooming using directional cubic convolution in-

Table 3: Comparison of FSIM results of the reconstructed images

Image	CC[12]	DFDF [6]	ICBI[8]	KR[25]	NEDI[4]	INEDI[14]	CGI[26]	Proposed
Bee	0,9865	0,9869	0,9948	0,9906	0,9841	0,9941	0,9959	**0,9960**
Face	0,9871	0,9870	0,9954	0,9923	0,9854	0,9951	**0,9969**	0,9968
Flinstones	0,9668	0,9663	0,9751	0,9629	0,9658	0,9748	0,9785	**0,9787**
Lena	0,9787	0,9772	0,9868	0,9831	0,9750	0,9716	0,9881	**0,9886**
Monarch	0,9786	0,9801	0,9866	0,9794	0,9796	0,9860	**0,9885**	0,9885
Parrot	0,9823	0,9814	0,9860	0,9879	0,9807	0,9905	0,9915	**0,9918**
Peppers	0,9805	0,9815	0,9792	0,9824	0,9799	0,9783	0,9814	**0,9825**
Rings	0,9999	0,9989	0,9998	0,9988	0,9964	0,9840	0,9998	**0,9999**
Watch	0,9774	0,9782	0,9839	0,9849	0,9771	0,9774	0,9879	**0,9880**
Average	0,9819	0,9819	0,9875	0,9847	0,9804	0,9835	0,9898	**0,9901**

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

Figure 7: The interpolated images results of 'Lena' (a) Cubic convolution; (b) DFDF; (c) KR; (d) Contrast-edge ; (e) NEDI; (f) Proposed.

terpolation, *IET Image Process.* 6(6), 2012, pp. 627-634.

[11] M.-J. Chen and W.-L. Lee, A fast edge-oriented algorithm for image interpolation, *Image Vis. Comput.* 23(9), May 2005, pp. 791-798.

[12] R. Keys, Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoust. Speech Signal Process.* 29(6), May 1981, pp. 1153-1160.

[13] M. Unser, Splines: A perfect fit for signal and image processing, *IEEE Signal Process. Mag.* 16(6), Nov. 1999, pp. 22-38.

[14] N. Asuni and A. Giachetti, Accuracy improvements and artifacts removal in edge based image interpolation, *Proc 3rd Int Conf Comput Vis Theory Appl.*, VISAPP08, 2008.

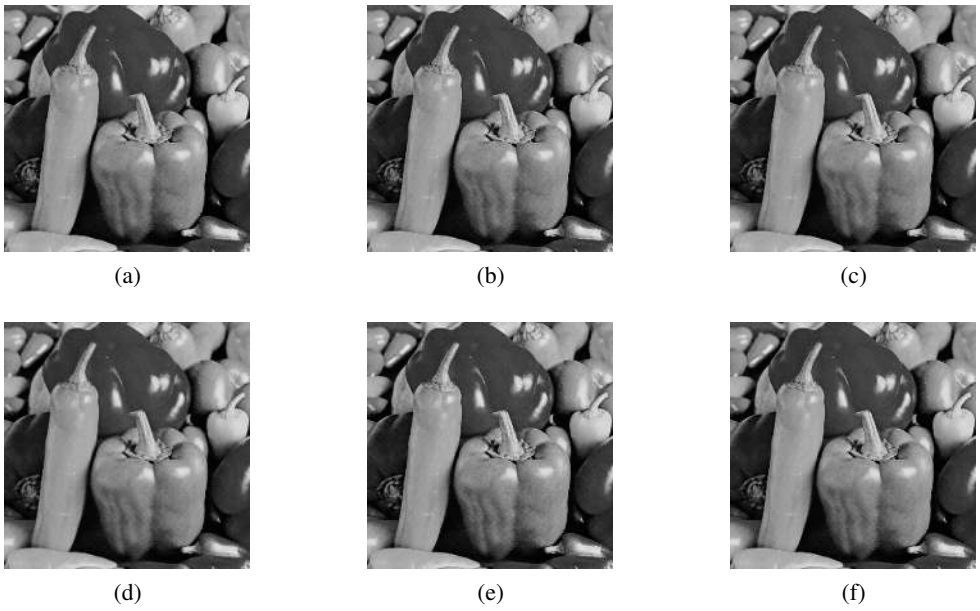
[15] S. Dube and L. Hong, An adaptive algorithm for image resolution enhancement, *Signals Syst. Comput. 2000 Conf. Rec. Thirty-Fourth Asilomar Conf.* 2, 2000.

[16] R. Kimmel, Demosaicing: image reconstruction from ccd samples, *IEEE Trans. Image Processing.* , 1999, pp. 1221-1228.

[17] J. Allebach and P. W. Wong, Edge-directed interpolation, *Proc. Int. Conf. Image Process. 1996.* 3, 1996, pp. 707-710.

[18] D. R. Cok, Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal, US Pat. No 4642678, 1987.

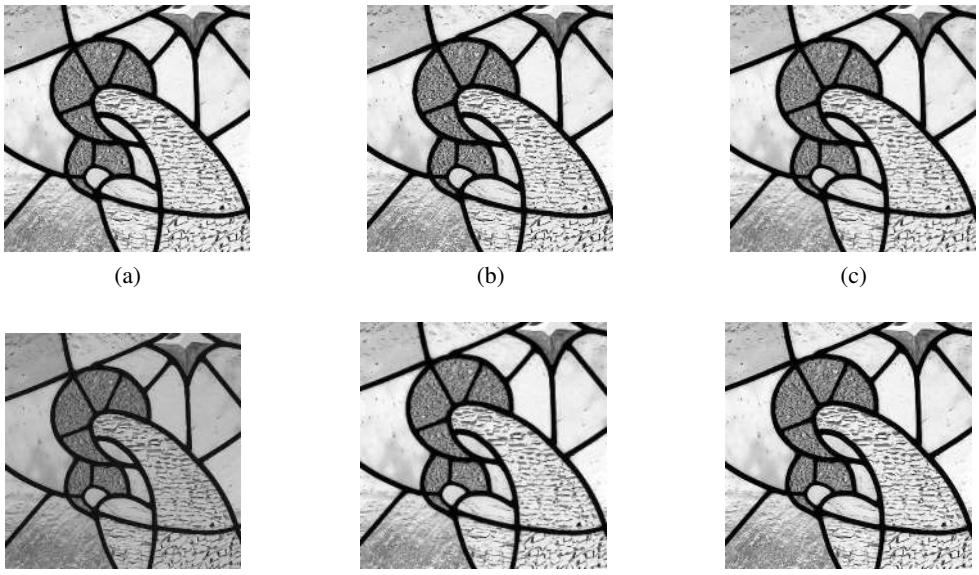
[19] M. Li and T. Nguyen, Markov Random Field Model-Based Edge-Directed Image interpola-



(a) (b) (c)

(d) (e) (f)

Figure 8: The interpolated images results of 'Peppers' (a) Cubic convolution; (b) ICBI; (c) KR; (d) NEDI ; (e) INEDI; (f) Proposed.



(a) (b) (c)

(d) (e) (f)

Figure 9: The interpolated images results of 'Mosaic' (a) Cubic convolution; (b) ICBI; (c) KR; (d) NEDI ; (e) INEDI; (f) Proposed.

tion, *Internation Conf. Image Proc.* , 2007, pp. 93–96.

[20] K. Adamczyk and A. Walczak, Application of 2D Anisotropic Wavelet Edge Extractors for Image Interpolation, *Human-Computer Systems Interaction Springer-Verlag Berlin Heidelberg.*, AISC 99, Part II:205–222, 2012.

[21] 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.* 13(4), Apr. 2004, pp. 600–612.

[22] Lin Zhang, Lei Zhang, X. Mou and D. Zhang, FSIM: A Feature Similarity Index for Image Quality Assessment, *IEEE Transactions on Image Processing.* 20(8), 2011, pp. 2378–2386.

[23] S. Ousguine, F. Essannouni, L. Essannouni,and D. Aboutajdine, High Resolution Image Reconstruction Using the Phase Correlation, *J. Inf. Organ.* 2(3), 2012, pp. 128-134.

[24] E. Dubois: Avalaible At. *http://www.site.uottawa.ca/ edubois/demosaicking*.

[25] H. Takeda, S. Farsiu and P. Milanfar, kernel Regression for Image Processing and Reconstruction, *IEEE Transactions on Image Processing* 16(2), February 2007, pp. 349–366.

[26] Z. Wei and M. Kai-Kuang, Contrast-guided Image Interpolation, *IEEE Transactions on Image Processing* 22(11), 2013, pp. 4271–4285.

[27] R.M. Haralick, Digital Step Edges from Zero Crossing of Second Directional Derivatives, *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6(1), Jan 1984, pp. 58-68.

[28] W. Xin, Laplacian Operator-Based Edge Detectors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(5), May 2007 , pp. 886-890.