International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 9 (2017) pp. 114–123

Speeding up edge guided image interpolation using the integral image technique

Said OUSGUINE[1], Fedwa ESSANNOUNI[1], Leila ESSANNOUNI[2], Mohammed ABBAD[1] and Driss ABOUTAJDINE[1]

[1]LRIT Research Laboratory (associated unit to CNRST, URAC n° 29),
Mohammed V University, Faculty of Sciences Rabat, Morocco
lamak07@hotmail.fr

[2]LASTIMI Research Laboratory,
Mohammed V University, High School of Technology Sale, Morocco

Abstract: **Image interpolation and super-resolution are subjects of great interest in image processing. The main task behind any image interpolation method is to obtain superior high-resolution images with improved visual quality, and recuperate rich texture and sharp edges using minimal calculation. In this sense, the aim of this paper is to present a new method which is effective in view of high resolution quality, processing speed, and economic performance by reducing the amount of calculations required to perform image edge directed image interpolation. This can be attained by operating in an integrated manner image interpolation using 4-directional image gradient information which defines differences in intensities between contiguous pixels. The proposed method is implemented and tested over several gray images, and compared to many interpolation methods in the state of the art. It has been demonstrated to be able to obtain faithfully better results than the habitual interpolation methods in terms of image quality metrics (PSNR, SSIM, FSIM) as well as the the perceptual visual appearance of the reconstructed images.**

Keywords: Image interpolation, Cubic convolution, Image integration, Edge detection.

I. Introduction

Image interpolation aims to generate a high-resolution image from low-resolution counterparts, its process consists in estimating a set of unknown pixel values from a set of knowing pixels, to improve the image quality and have a better visual appearance in the output. Image interpolation is referred in literature by many terminologies, such as resizing, upscaling, zooming, enhancement and enlargement; in addition, it was widely required in various applications for many areas, from digital photography to video communication, satellite remote sensing, objet recognition, medical imaging, surveillance and high-resolution television.
Varieties of image interpolation methods exist. The conventional polynomial-based interpolation methods such as bicubic interpolation, cubic spline, cubic convolution and nearest neighbor treat the whole image in a uniform way. They use polynomial approximation to compute each missing pixel values from its adjacent known pixels. They are easy for implementation and performing, moreover, they work well in smooth areas. However, these techniques have less complexity calculation cost during the interpolation process, there common drawback is stilling the annoying artifacts such as jaggies, blurring, aliasing and ringing near edges, due to their incapacity to adapt to varying pixel structures in the image. This leads to a blurred image to some extent. To overcome this drawback, a number of adaptive interpolation methods [[1]-[20]] have been developed to reduce these visual artifacts and improve the quality of the reconstructed high-resolution image. The major step practiced in these methods is that they employ models that explicitly or implicitly extract the characteristics of local features such as dominant image structure and edge information of the LR image in order to spatially adapt the interpolation coefficients to better match the local structures nearby the edges.
The important groups among the adaptive algorithms are edge-directed interpolation methods which follow the principle that no interpolation across the edges in the image is allowed. This can be provided into two main steps: the first one is to detect the edge characteristics, including localization, orientation and intensity variation of the extracted edges in the neighboring pixels. Then, the second one is to interpolate the missed pixels along the estimated edge-direction as close to the original as possible, to restore more accurate and visually pleasing results.

II. Related Work

To solve this problem, several image interpolation methods that involve more evolved models have been recently proposed in the literature either in adaptive class or non-adaptive class or by combining the two classes to improve the subjective quality of the interpolated images. Ramponi [1] suggested the warped distance to the interpolated pixel instead of a uniform one, thus modifying the one dimensional kernel of a separable interpolation filter. The computational complexity of the method remains low, but the upscaling quality of oriented edges neither horizontally nor vertically is poor. An improvement of this technique was made by Huang and Lee [2] which consists in modifying the two-dimensional in-

terpolation kernel by the characteristics of the local gradient features.

Other methods of interpolation based on edge detection were introduced in [3, 18, 4, 13]. In [3], Jensen proposes to estimate the orientation of each edge in the image by using projection onto orthonormal basis and modify the interpolation process to avoid interpolating across the edges. An iterative edge-directed scheme was developed in [18]. The authors use an estimate of the edge mapping of the high-resolution image to guide a bilinear interpolation such that interpolation across edges is avoided. Another approach, NEDI, was proposed by Li and Orchard in [4]. They exploit the geometric duality between the covariance of a low-resolution image and the covariance of a high-resolution image, with a help of linear prediction, to guide the interpolation. The missed pixels can be predicted using estimated covariance through 2D Kalman filter approach.

An improved method of NEDI was proposed by Asuni and Giachetti [13] INEDI (Improved New Edge-Directed Interpolation) to modify the algorithm of Li [4] by varying the size of the training window according to the edge size. Li and Zhang [6] suggest to interpolate the missing pixels in predetermined multiple directions, and then fuse the directional interpolation results by linear minimum mean-square-error estimation (LMMSE). However, the fusion of two orthogonal directions is insufficient in some situations. 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 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 performing the 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 recommended by Giachetti and Asuni [8]. This method relies on the combination of two steps. First, an adaptive algorithm is applied by interpolating locally pixel values along the direction where second order image derivative is lower. Then the interpolated values are modified using an iterative refinement minimizing differences in second order image derivatives information, thus maximizing second order derivative values and smoothing isolevel curves.

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. 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 area. In [14], a contrast-guided image interpolation method is proposed. It 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 $^\circ$ and 135° CDMs for interpolating the diagonal pixels and 2) the 0° and 90° CDMs for interpolating the row and column pixels.

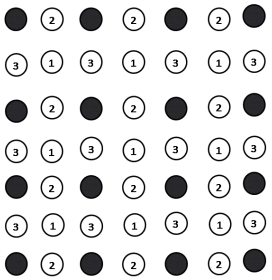


Figure. 1: Formation of an LR image from an HR image by downsampling. The black dots represent the LR image pixels, while the other circles represent the missing HR pixels.

The aim of this paper 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. Thus a novel method of image reconstruction is proposed, based on image interpolation using the integral image, which consists in estimating the edge-directions using 4 directional images, then interpolating the missing pixels along the detected edge direction using a cubic convolution. The missing pixels are interpolated along the detected edge direction; the latter is estimated in the 45° and 135°diagonal, horizontal and vertical directions. After that, a cubic convolution interpolation is used to interpolate the missing pixels along the strong edge.

The organization of the rest of this paper is shown as follows: Section II describes the proposed interpolation method. Simulation results are shown in section III. Finally, the conclusion is drawn in section IV.

III. The proposed algorithm

Let us consider a LR image, which is a downsampled version of the HR image by a factor of two as shown in Figure 1. In this figure, the circles are the pixels to be interpolated while the black dots are the available pixels in the known LR image. That is, the HR image is reconstructed by copying the LR images pixels to high-resolution 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 missed. 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.

Various interpolation methods has been proposed to estimate the unknowing pixels however the main problem found in traditional methods is the presence of blurring and ringing artifacts near the edges, which can lead to a jagged aliasing effect. In contrast, edge-directed interpolation methods detect the edge orientations, and then interpolate along the detect-

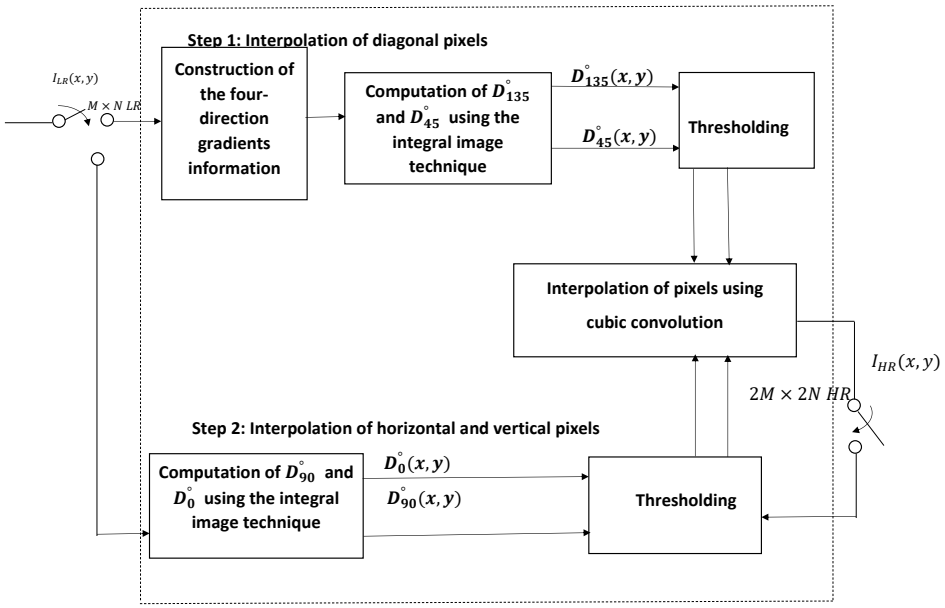


Figure. 3: Framework of the proposed image interpolation using the integral image technique.

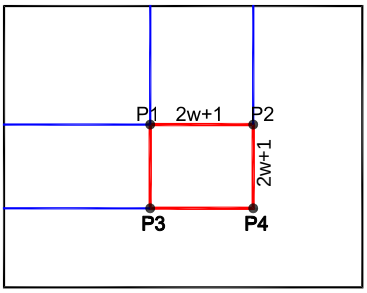


Figure. 2: A description of computing the summed windows in a directional gradient image.

ed edge direction. The main drawback of these approaches is the enormous number of calculations especially if a large window is used to detect edge orientation. Here we propose a fast technique for speeding up the edge directed interpolation methods. An overview of the proposed algorithm is shown in Figure 3.

A. Edge direction detection

To interpolate the missing pixels in the high resolution image, the first step is to detect the edge directions using 4-directional image gradient information representing differences in color intensities between adjacent pixels :

$$\begin{aligned} G_{0^\circ}(x,y) &= |I_{LR}(x,y) - I_{LR}(x,y+1)|, \\ G_{90^\circ}(x,y) &= |I_{LR}(x,y) - I_{LR}(x+1,y)|, \\ G_{45^\circ}(x,y) &= |I_{LR}(x,y) - I_{LR}(x+1,y-1)|, \\ G_{135^\circ}(x,y) &= |I_{LR}(x,y) - I_{LR}(x+1,y+1)|. \end{aligned} \quad (1)$$

For each pixel (x,y), we compute the four directional gradients sums in a window of size $(2w+1, 2w+1)$ centred at (x,y). Let $D_\theta(x,y)$ denote the sum of the pixels in gradient image for the direction θ. Thus $D_\theta(x,y)$ can be defined as follows:

$$\underset{\theta\in\{0^\circ,45^\circ,90^\circ,135^\circ\}}{D_\theta(x,y)} = \sum_{\substack{x-w\le x'\le x+w \\ y-w\le y'\le y+w}} G_\theta(x',y'), \quad (2)$$

To improve processing speed, D_θ can be computed using the integral image technique. The summed area table is a data structure and algorithm for quickly and efficiently generating the sum of values in a rectangular subset of a grid. In the image processing domain, it is also known as an integral image. It was first introduced to computer graphics in 1984 by Frank Crow for use with mipmaps [27]. In computer vision it was first used within the Viola–Jones object detection framework in [25]. However, historically, this principle is very well known in the study of multi-dimensional probability distribution functions, namely in computing 2D (or ND) probabilities (area under the probability distribution) from the respective cumulative distribution functions [26].
As the name suggests, the value at any point (x,y) in the summed area table is just the sum of all the pixels above and to the left of (x,y), inclusive :

$$\underset{\theta\in\{0^\circ,45^\circ,90^\circ,135^\circ\}}{IG_\theta(x,y)} = \sum_{\substack{x'\le x \\ y'\le y}} G_\theta(x',y'). \quad (3)$$

Moreover, the summed area table can be computed efficiently in a single pass over the gradient image, using the fact that the value in the summed area table at (x,y) is just:

$$\begin{aligned} IG_\theta(x,y) &= G_\theta(x,y) + IG_\theta(x-1,y) \\ &\quad + IG_\theta(x,y-1) - IG_\theta(x-1,y-1). \end{aligned} \quad (4)$$

Once the summed area table has been computed, the task of

evaluating the intensities over any window requires only four array references. This allows for a constant calculation time that is independent of the size of the rectangular window. That is, using the notation in the Figure 2, having $P_1 = (x - w, y - w), P_2 = (x + w, y - w), P_3 = (x - w, y + w)$ and $P_4 = (x + w, y + w)$, the sum of $G_\theta(x, y)$ over the rectangle spanned by P_1, P_2, P_3 and P_4 is:

$$\begin{aligned} D^w_\theta(x,y) &= \sum_{\substack{x-w\le x'\le x+w \\ y-w\le y'\le y+w}} G_\theta(x',y') \\ &= IG_\theta(P_4) + IG_\theta(P_1) - IG_\theta(P_2) - IG_\theta(P_3). \end{aligned} \quad (5)$$

This calculation is a constant time that is independent of the choice of w.

Two edge directions are examined for every missing pixel, horizontal and vertical directions for pixels of type 2 and 3, while 45° and 135° diagonal directions are examined for pixels of type 1.

For the rest of the paper, let D_1 denote $D^w_{0^\circ}$ for pixels of types 2 and 3 and $D^w_{45^\circ}$ for pixels of type 1. Also, D_2 denotes $D^w_{90^\circ}$ for pixels of types 2 and 3 and $D^w_{135^\circ}$ for pixels of type 1.

if $(D1 - D2) <$ TH
The edge orientation is on 45° or 0°;
elseif $(D2 - D1) <$ TH
The edge orientation is on 135° or 90°;
else
the pixel is on homegenous or textured region
end

Where TH is a threshold parameter determined experimentally, the choice of TH will be discussed in the next section.

Algorithm 1 The integral image technique

Input: low-resolution image LR
Output: high-resolution image HR

1. Extraction of the edge directions
   (a) Compute the directional image gradient G in the 45° and 135° diagonal, horizontal and vertical directions using eq. (1).
   (b) Use the integral image technique for calculating the summation of the four directional gradient $D^w_\theta(x, y)$ as in eq. (5).
   (c) Estimate the edge directions using the difference between the two orthogonal directional integral.
2. if the missing pixel is on the strong edge, we estimate the pixel intensity value using the 1-D cubic convolution, otherwise if the missing pixel is on weak edge or textured region, we Combine the two orthogonal directional cubic interpolation results using the weighted average of directional estimation (WADE) according to eq. (6).

B. Missing pixel intensity estimation

The interpolation of the missing pixels using the proposed method consists of three stages. In the first stage, we interpolate the pixels of type 1 using the four nearest known pixels from the LR image along the estimated edge. In the second stage, the pixels of types 2 and 3 are generated, we will not only using the known pixels from the LR image but also the already predicted pixels of type 1 obtained from the first stage. The estimated edge direction at a missing pixel position is used to estimate the pixel intensity value. The latter is calculated using the 1D cubic convolution interpolation along the strongest edge directly.

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 by using the weighted average of directional estimation (WADE)[28]. 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 weights w_{p1} and w_{p2} combining I_1 and I_2 are computed as follows:

$$\begin{cases} w_{p1} = \exp(-\frac{D_1^2}{2\sigma^2}) \\ w_{p2} = \exp(-\frac{D_2^2}{2\sigma^2}) \end{cases} \quad (6)$$

the parameter σ is used as a scale of the weights w_{p1} and w_{p2}.

Finally, the interpolation value I of the missing pixel at location (x, y) in a homogenous or textured region is estimated as: :

$$I = \frac{(w_{p1}I_{_1} + w_{p2}I_{_2})}{(w_{p1} + w_{p2})} \quad (7)$$

The results of the proposed algorithm depend on the choice of the threshold TH and the scale σ of the directional weights in equation 6. These parameters cannot be directly determined for a given LR image because they depend on the types of the images. We derive σ and TH through number of training. We use Twenty-four 512×768 Kodak color images [29] as training sample to obtain the optimal factors of threshold TH and the scale σ. These images were first converted into grey images and then downsampled to obtain their LR counterparts. The HR images were reconstructed from the LR ones using the proposed method with the different values of TH and the scale σ. The parameter threshold TH was separately set from 0.1 to 1 using step length of 0.1, and the scale σ vary from 0 to 1 using step length of 0.1. The Figure 5 shows the average peak signal-to-noise ratio (PSNR) curves using different values of σ and TH.

From this figure, the PSNR achieves its maximum value when $\sigma = 0.7$ and $TH = 0.7$.

The proposed algorithm as defined up to now only gives a 2 x 2 image interpolation. For a larger factor (power-of-two), it requires multiple executions of such 2 x 2 interpolation sequentially. For MxM image interpolation where M is not a power of two, one can first use the aforementioned N x N image interpolation (where the value of N is chosen as a power of two and as close to M as possible, but being smaller), followed by performing a 2-D bicubic image interpolation with a rational number (bounded by 2) of image enlargement factor.

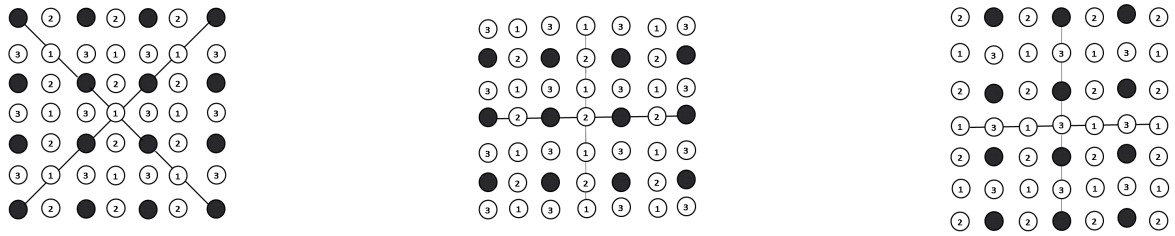


Figure. 4: Interpolation of the missing pixels a) Type 1 (b) Type (2) (c) Type (3).

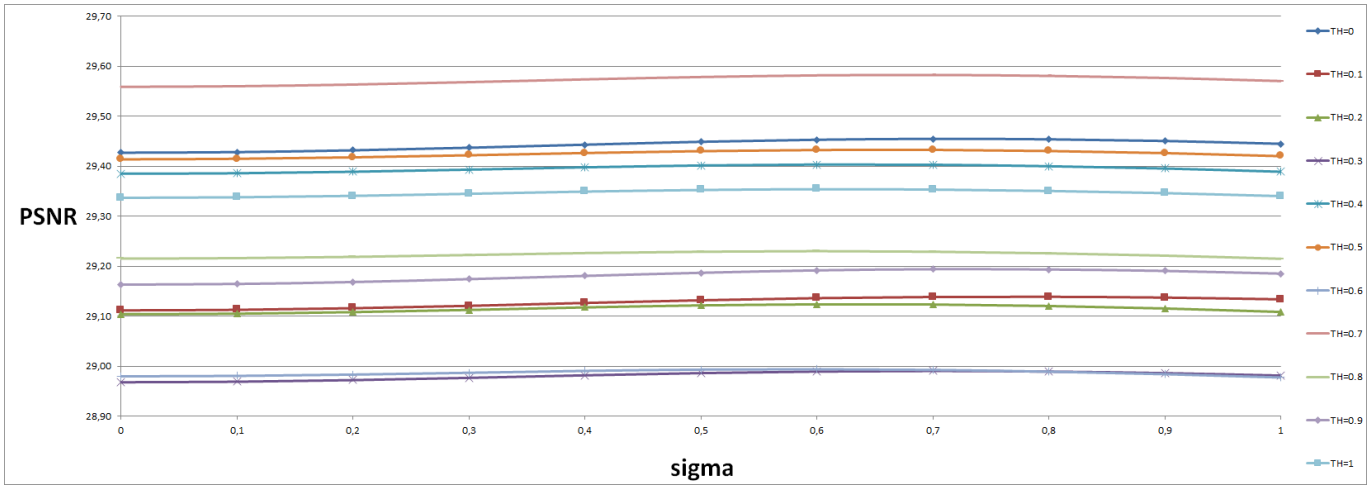


Figure. 5: Selection of the scale parameter σ and the threshold TH.

IV. Simulation results

For demonstration of the accuracy of the proposed method, extension experiments are conducted to evaluate the effectiveness of the proposed method in subjective visual effect and objective quality. This method has been implemented in Matlab, tested on several images and compared to the conventional interpolation algorithms: Cubic Convolution interpolation (CC) [12], Directional Filtering and Data Fusion (DFDF) [6], Iterative Curve Based Interpolation (ICBI) [8], New Edge-Directed Interpolation (NEDI) [4], Improved New Edge-Directed Interpolation (INEDI) [13], Kernel Regression (KR) [15].

The bicubic interpolation is performed with the matlab built in functions, whereas the codes of the other compared methods are kindly provided by their authors.
For thoroughness and fairness of our comparison study, over eight gray test images including 'lena', which contain various edges and texture contents, have been widely selected and used for the evaluation to analyze and verify the performance of the proposed method, as shown in Figure 6.
Following the conventional setting, the original image was first directly downsampled by a factor of 2 to get the input LR image, then, different interpolation methods are applied to the input LR image, for generating the reconstructed HR image. To further conduct the performance evaluation of the proposed method in comparison with the other interpolation methods, several tests are set in both qualitative subjective and quantitative objective evaluation. In the part of subjective visual quality of the output images, the images results for each of the interpolation methods are displayed in Figures 7-10.
For the objective measurement of the different interpolation methods the Peak Signal to Noise Ratio (PSNR), and the two perceptual metrics the Structural SImilarity Measure (SSIM) [21] and the Feature SImilarity Measure (FSIM) [22], which is the quality assessment criterion between the original image and corresponding interpolating image, are used to assess the quality of the reconstructed images.

Tables 1, 2 and 3 list the PSNR, SSIM and FSIM scores of the different methods on the test images.
From Table 1, the highest PSNR value of each row is shown in bold. It is apparent that the proposed method outperforms the above-mentioned interpolation methods around three evaluating criterion, and achieves the highest PSNR, SSIM and FSIM scores for all testing images, which fully demonstrates that the interpolation results by the proposed method are the best both objectively and subjectively. More specifically, the proposed method exceeds the average PSNR value of the second best method (ICBI) by 1.7 dB. Note that on the image 'Parrot' which has rich texture, the proposed method gains 0.98 dB respectively over the second best method INEDI [13] algorithm. We can see also that our method achieves almost the highest SSIM and FSIM values.
The proposed method was compared with state-of-the-art of image interpolation methods in the part of the visual appearance of the resulted image. As can be shown in Figures 7-10, the filtering-based bicubic interpolation method [12]

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

Image	CC[12]	DFDF [6]	ICBI[8]	KR[15]	NEDI[4]	INEDI[13]	Proposed (w=1)	Proposed (w=2)	Proposed (w=3)
Bee	34.39	35.05	34.31	34.37	32.11	33.32	34.98	**35.15**	35
Face	40.53	40.84	39.79	40.26	38.41	39.19	40.69	41.27	**41.31**
Flinstones	26.93	27.03	27	26.72	25.28	27.03	**27.52**	27,49	27,24
Lena	33.81	33.77	33.87	33.96	32.79	33.2	34.26	**34.45**	34.35
Monarch	30.16	30.71	30.85	30.78	28.92	**31.26**	31.03	31.09	30.8
Parrot	33.31	33.32	33.1	30.76	32.2	33.49	34.07	34.43	**34.48**
Rings	45.61	42.18	46.53	26.29	30.11	28.66	54.08	54.39	**54.5**
Watch	31.91	31.8	31.68	31.77	30.7	31.84	32.25	**32.46**	32.36
Average	34.58	34.34	34.64	31.86	31.32	32.25	36.11	**36.34**	36.25

Table 2: Comparison of SSIM results of the reconstructed images

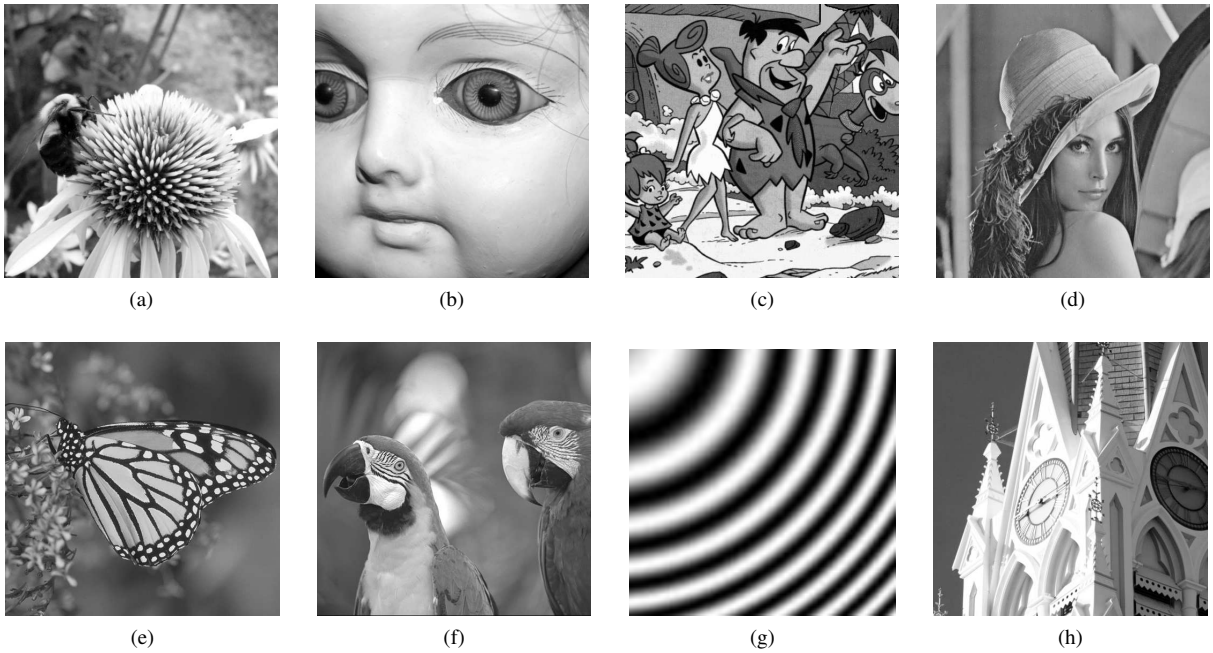
Image	CC[12]	DFDF [6]	ICBI[8]	KR[15]	NEDI[4]	INEDI[13]	Proposed (w=1)	Proposed (w=2)	Proposed (w=3)
Bee	0.9934	0.9926	**0.994**	0.9917	0.9883	0.9917	0.9933	0.9939	0.9939
Face	0.9926	0.9915	0.9927	0.9906	0.9889	0.9916	0.992	0.9928	**0.993**
Flinstones	0.9633	0.9632	0.962	0.9642	0.954	0.9637	0.9672	**0,9673**	0,9661
Lena	0.972	0.9721	0.97	0.9716	0.9688	0.9692	0.9732	**0.9743**	0.9739
Monarch	0.9858	0.9862	0.9862	0.986	0.982	0.9869	0.9875	**0.9881**	0.9878
Parrot	0.9812	0.9806	0.9799	0.9806	0.9782	0.9807	0.9822	0.9832	**0.9834**
Rings	0.9999	0.9995	0.9996	0.9985	0.9892	0.9825	0.9999	**0.9999**	0.9999
Watch	0.9809	0.9813	0.979	0.9802	0.9728	0.9824	0.9812	0.9834	**0.9835**
Average	0.9836	0.9834	0.9829	0.9829	0.9778	0.9811	0.9846	**0.9854**	0.9852

Table 3: Comparison of FSIM results of the reconstructed images

Image	CC[12]	DFDF [6]	ICBI[8]	KR[15]	NEDI[4]	INEDI[13]	Proposed (w=1)	Proposed (w=2)	Proposed (w=3)
Bee	0.9865	0.9869	0.9948	0.9906	0.9841	0.9941	0.9953	**0.9957**	0.9956
Face	0.9871	0.987	0.9954	0.9923	0.9854	0.9951	0.9961	0.9967	**0.9968**
Flinstones	0.9668	0.9663	0.9751	0.9629	0.9658	0.9748	0.9779	**0.9782**	0.9774
Lena	0.9787	0.9772	0.9868	0.9831	0.975	0.9716	0.988	**0.9884**	0.9883
Monarch	0.9786	0.9801	0.9866	0.9794	0.9796	0.986	0.9871	**0.9881**	0.9874
Parrot	0.9823	0.9814	0.986	0.9879	0.9807	0.9905	0.9912	0.9917	**0.9918**
Rings	0.9999	0.9989	0.9998	0.9988	0.9964	0.984	0.9999	**0.9999**	0.9999
Watch	0.9774	0.9782	0.9839	0.9849	0.9771	0.9774	0.986	**0.9875**	0.9875
Average	0.9822	0.982	0.9886	0.985	0.9805	0.9842	0.9902	**0.9908**	0.9906

Table 4: Average computational time (second) of different interpolation methods

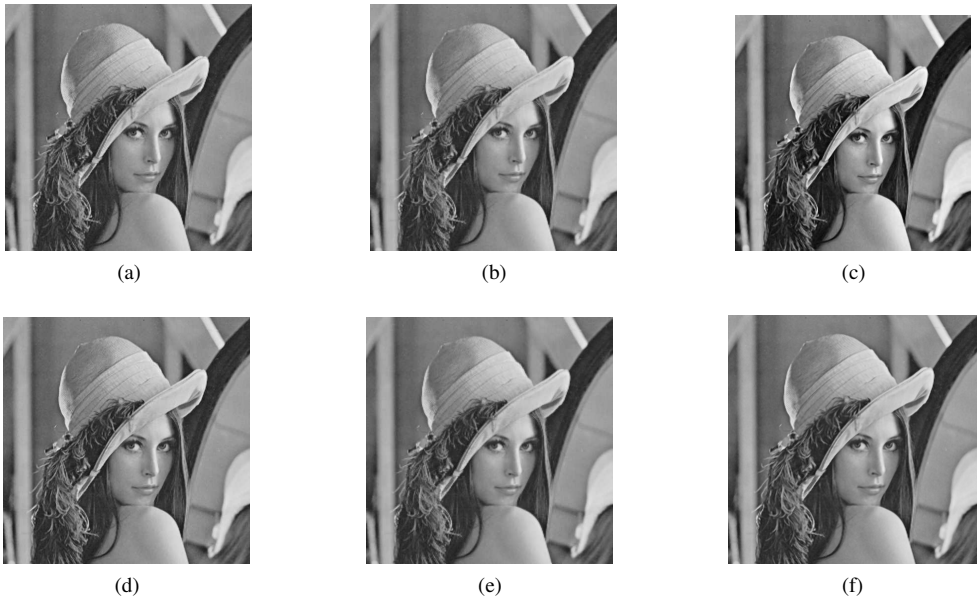
	CC[12]	DFDF [6]	ICBI[8]	KR[15]	NEDI[4]	INEDI[13]	Proposed (w=2)	Proposed (w=3)
Computational Time (second)	0.06	13.44	95.12	16.46	14.41	428.29	2.40	2.36



(a) (b) (c) (d)

(e) (f) (g) (h)

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



(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 w=2.

produces some blur and jaggy artefacts along edges due to non-adaptive filtering, so it has worst quality than the other interpolation methods. The NEDI [4] and INEDI [13] 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 because they break the geometric duality between the LR covariance and HR covariance, 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] and INEDI [13], they reproduce sharper large scale edges than the bicubic method, but DFDF produce blocky edges, ICBI can retain sharper edges than DFDF but it also tends to result in strong discontinuities on a long edge. It is clear that the proposed method produces a sharp high-resolution image and outperforms the above-mentioned interpolation methods by producing a high visual image quality and recover continues edge and fine details.

Another advantage of our proposed method is that it yields an acceptable computational complexity. We test the run time of an un-optimized Matlab implementation of the proposed method and other interpolation methods on an Intel Core i5

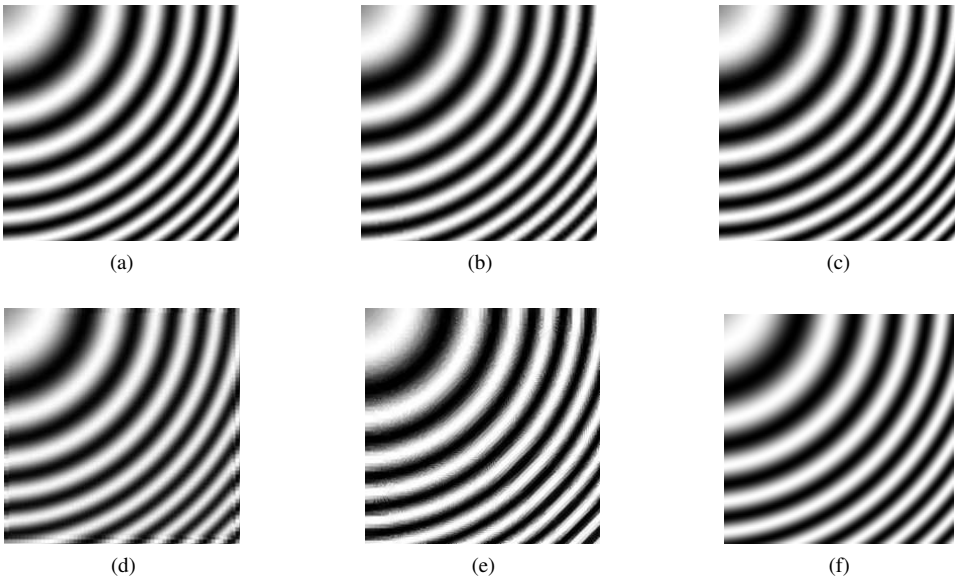


Figure. 8: The interpolated images results of 'Rings' (a) Cubic convolution; (b) ICBI; (c) KR; (d) NEDI ; (e) INEDI; (f) Proposed w=2.

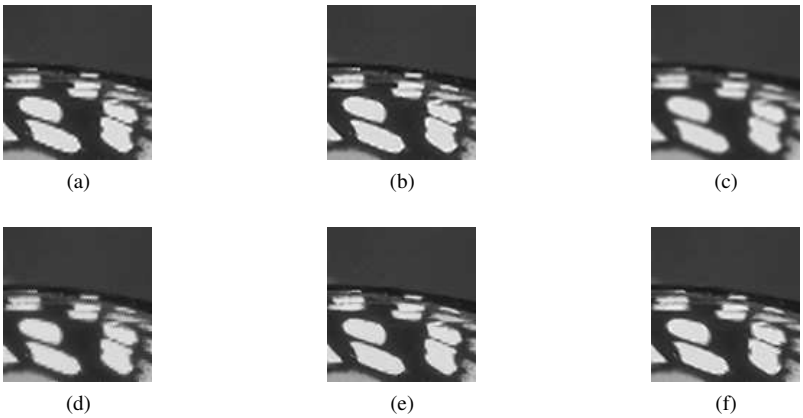


Figure. 9: The interpolated images results of the part of 'Monarch' (a) Cubic convolution; (b) ICBI; (c) KR; (d) NEDI ; (e) INEDI; (f) Proposed w=2.

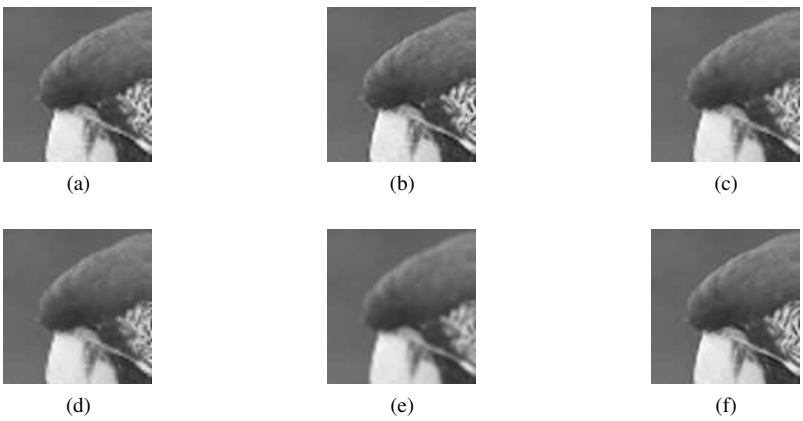


Figure. 10: The interpolated images results of the part of 'Parrot' (a) Cubic convolution; (b) ICBI; (c) DFDF; (d) NEDI ; (e) KR; (f) Proposed w=2.

3.10 GHz processor, for the enlargement of the 256×256 LR image to a 512×512 HR image by a factor of 2×2. We made use the realization codes provided by the authors on their websites for other methods on comparison. As shown

in Table 4, our proposed method yields the smallest running time among the adaptive methods. Specifically, the processing time of NEDI [4] and DFDF [6] are 6.10 times and 5.69 times slower than that of the proposed algorithm. Also the execution time of the proposed method is relatively independent of the choice of the training window size (w).

V. Conclusion

This paper aims to develop an efficient and rapid image interpolation method using the integral image technique. In the proposed method, we make consideration of edge detection, we detected the edge strength based on the LR image on four directions, we adjusted the estimated edge parameters, and then, based on these parameters, we interpolated the missing pixels using the cubic convolution along the detected edge orientation. Experimental results indicate the viability and efficiency of the proposed over the existing interpolation methods in terms of both objective assessment and subjective visual quality, which translates by the PSNR, SSIM and FSIM values, while reducing annoying artifacts such as jagging around edges suffered by habitual interpolation methods which lead to missing much image details. Furthermore, the proposed method is also fast due to the use of the integral image.

References

[1] G. Ramponi, *Warped distance for space-variant linear image interpolation*, IEEE Trans. Image Process. vol. 62, no. 5, pp. 629-639, 1999.

[2] J. W. Hwang and H. S. Lee, *Adaptive image interpolation based on local gradient features*, IEEE Signal Process. Lett. vol. 11, no. 3, pp. 359-362, 2004.

[3] K. Jensen and D. Anastassiou, *Subpixel edge localization and the interpolation of still images*, Image Process. IEEE Trans. vol. 4, no. 3, pp. 285-295, 1995.

[4] X. Li and M. T. Orchard, *New edge-directed interpolation*, IEEE Trans. Image Process. vol. 10, no. 10, pp. 1521-1527, 2001.

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

[6] L. Zhang and X. Wu, *An edge-guided image interpolation algorithm via directional filtering and data fusion*, Image Process. IEEE Trans. vol. 15, no. 8, pp. 2226-2238, 2006.

[7] Y. Cha and S. Kim, *The error-amended sharp edge (EASE) scheme for image zooming*, Image Process. IEEE Trans. vol. 16, no. 6, pp. 1496-1505, 2007.

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

[9] W. K. Carey and S. S. Hemami, *Regularity-preserving image interpolation*, IEEE Trans. Image Process. vol. 8, no. 9, pp. 1293-1297, 1999.

[10] Z. Dengwen, S. Xiaoliu, and D. Weiming, *Image zooming using directional cubic convolution interpolation*, IET Image Process. vol. 6, no. 6, pp. 627-634, 2012.

[11] M.-J. Chen and W.-L. Lee, *A fast edge-oriented algorithm for image interpolation*, Image Vis. Comput. vol. 23, no. 9, pp. 791 - 798, May 2005.

[12] R. Keys, *Cubic convolution interpolation for digital image processing*, IEEE Trans. Acoust. Speech Signal Process. vol. 29, no. 6, pp. 1153-1160, May 1981.

[13] 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.

[14] Z. Wei and M. Kai-Kuang, *Contrast-guided Image Interpolation*, IEEE Transactions on Image Processing. vol. 22, no. 11, pp. 4271-4285, 2013.

[15] H. Takeda, S. Farsiu and P. Milanfar, *kernel Regression for Image Processing and Reconstruction*, IEEE Transactions on Image Processing. vol. 16, no. 2, pp. 349-366, February 2007.

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

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

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

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

[20] M. Li and T. Nguyen, *Markov Random Field Model-Based Edge-Directed Image interpolation*, pp. 93-96, 2007.

[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. vol. 13, no. 4, pp. 600-612, Apr. 2004.

[22] Lin Zhang, Lei Zhang, X. Mou and D. Zhang, *FSIM: A Feature Similarity Index for Image Quality Assessment*, IEEE Transactions on Image Processing. vol. 20, no. 8, pp. 2378-2386, 2011.

[23] S. Ousguine, F. Essannouni, L. Essannouni, M. Abbad and D. Aboutajdine, *A New Image Interpolation Using the Bilateral Filter*, International Journal of Imaging and Robotics, vol. 16, no. 2, pp. 91-104, 2016.

[24] S. Ousguine, F. Essannouni, L. Essannouni, M. Abbad and D. Aboutajdine, *A New Directional Image Interpolation Based on Laplacian Operator*, International Journal of Signal Processing, vol. 1, pp. 1-11, 2016.

[25] P. Viola and M. Jones, *Robust Real-time Object Detection*, International Journal of Computer Vision, vol. 18, no. 11, pp. 10491075, 2002.

[26] A. Finkelstein and neeratsharma, *Double Integrals By Summing Values Of Cumulative Distribution Function*, Wolfram Demonstration Project, 2010.

[27] F. Crow, *Summed-area tables for texture mapping*, SIGGRAPH'84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, pp. 207-212, 1984.

[28] H. Zhang and M. Rong, *Deinterlacing algorithm using gradient-guided interpolation and weighted average of directional estimation*, IET Image Processing, vol. 9, no. 6, pp. 450460, Jun. 2015.

[29] E.Dubois, *Avalaible At*, http://www.site.uottawa.ca/ e-dubois/demosaicking

[30] B. Amin and W. Xiaolin, *Image interpolation with hidden Markov model*, IEEE International Conference in Acoustics Speech and Signal Processing (ICASSP), pp. 874-877, 2010.

[31] C. Hsin-Hui and J. J. Ding, *Structural similarity-based nonlocal edge-directed image Interpolation*, Picture Coding Symposium (PCS), vol. 9, no. 6, pp. 289-292, 2013.

[32] H. C. Stanley, T. D. Vo and T. Q. Nguyen, *Subpixel motion estimation without interpolation*, IEEE International Conference in Acoustics Speech and Signal Processing (ICASSP), pp. 722-725, 2010.

[33] E. Hosogai and T. Yuichi, *Iterative edge-directed image upscaling with high directionality*, Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA), pp. 1-4, 2014.