

Iterative Image Interpolation vs. Traditional Interpolation Methods

Samia Sh. Lazar

College of Archaeology
University of Mosul, Mosul, Iraq

Received on: 13 / 6 / 2010

Accepted on: 21 / 12 / 2010

ABSTRACT

Image interpolation is an important image processing operation applied in diverse areas ranging from computer graphics, rendering, editing, medical image reconstruction, and online image viewing. In this paper, image is interpolated from a lower resolution (LR) to a higher resolution (HR) based on the combination of two different procedures. First, an adaptive algorithm interpolating locally image pixel values along the direction is applied, where second order image derivative is low. Then interpolated values are modified using an iterative refinement to minimize the differences in second order image derivatives, maximize the second order derivative values and smooth the curves. The first algorithm itself provides edge-preserving images that are measurable better than those obtained with conventional methods presented in this literature. Objective and subjective tests on a series of natural images show the advantages of the used technique over existing approaches.

Keywords: Iterative Image Interpolation, Traditional Interpolation Methods.

استكمال الصورة التكراري مقابل طرائق الاستكمال التقليدية

سامية لازار

كلية الآثار، جامعة الموصل

تاريخ قبول البحث: 2010/12/21

تاريخ استلام البحث: 2010/06/13

المخلص

إن استكمال قيم الصورة عملية مهمة لمعالجة الصورة حيث طبقت في مجالات متنوعة منها رسومات الحاسوب وإعادة والتحرير، وإعادة بناء صورة طبية، وعرض صورة على الإنترنت. في هذا البحث، استكملت الصورة قيمها من الدقة الواطنة إلى الدقة العالية استناداً على تطبيق إجراءات مختلفين. أولاً، تطبيق الخوارزمية التكرارية لاستكمال قيم نقاط الصورة محلياً على طول الاتجاه حيث المشتقة الثانية للصورة تكون واطئة. ثم القيم المولدة تعدل بتطبيق تقنية تكرارية لتقليل الاختلافات في المشتقة الثانية للصورة، إذ يزيد قيم المشتقة الثانية ويصقل المنحنيات. فالخوارزمية الأولى ذاتها توفر صوراً مع الحفاظ على حافاتها على نحو قابل للقياس بشكل أفضل من تلك المكتسبة بالطرائق التقليدية المقدمة في هذا البحث. وتشير الاختبارات الموضوعية والذاتية على مجموعة من الصور الطبيعية إلى فوائد التقنية المقترحة مقارنة بالطرائق الموجودة حالياً.

الكلمات المفتاحية: استكمال الصورة التكراري، طرائق الاستكمال التقليدية.

1. Introduction

Image interpolation is a technology for image resolution conversion that generates a high-resolution image from its low-resolution image counterpart [5]. In the digital age, interpolation has become increasingly important and, with the introduction of digital imaging, the subject of image interpolation has become increasingly popular [1]. It has been widely applied to various image/video capture/display devices such as digital still cameras, digital video cameras, printers, and multiple-function peripherals [5]. In literature, image interpolation techniques are referred to many terminologies such as image resizing, image resampling, digital zooming, and image magnification or enhancement [6].

Image interpolation algorithms can be grouped in two categories: adaptive and non-adaptive [6]. The adaptive methods exploit geometric structure or regularity of image features, such as step-edge, ridge and valleys. The orientation of such features is first extracted from the image, and then the interpolation is locally adapted to perform interpolation along the features, rather than across them [1]. In non-adaptive methods, certain computations are performed indiscriminately to the whole image for interpolation regardless of its contents. The computational logic of an adaptive image interpolation technique is mostly dependent upon the intrinsic image features and contents of the input image whereas computational logic on non-adaptive image interpolation technique is fixed irrespective of the input image features [6].

Besides the traditional interpolation techniques such as nearest-neighbor, bilinear, and bicubic interpolation, several algorithms have been used to enhance the performance of interpolating. In this paper, the applied Iterative Curve Based Interpolation (ICBI) method uses the local second order information to adapt the interpolation and an iterative refinement used to remove artifacts while preserving relevant image features and natural texture. The objective and subjective tests that are performed to evaluate the performance of the applied method against other traditional interpolation methods show that the ICBI method is clearly an improvement. The tests show that the ICBI method provides a reasonable reconstruction of the missing information and requires a considerably reasonable computational power.

The rest of this paper is organized as follows: Section 2 presents the concept of interpolation and reviews the conventional methods for image interpolation. In section 3, introduce the Iterative Curve Based Interpolation (ICBI), which considers the effects of the curvature continuity and enhancement. Section 4 presents some experimental results in terms of the image quality using the ICBI and some of traditional interpolation methods. Section 5 draws some conclusions.

2. Interpolation

Interpolation is a process used to estimate an image value at a location in between image pixels. Interpolation methods determine the value for an interpolated pixel by finding the point in the input image that corresponds to a pixel in the output image and then calculate the value of the output pixel by computing a weighted average of some set of pixels in the vicinity of the point. The weightings are based on the distance for each pixel from the point [7].

The fastest scheme for calculating interpolated pixel is nearest-neighbor assignment (sometimes called zero-order interpolation) [4]. In this scheme, the output pixel is assigned the value of the pixel that the point falls within. No other pixels are considered [7]. The only advantage of nearest neighbor interpolation is its simplicity (no calculations are required to derive the output pixel value) and low computation [6]. However, it provides the worst results in terms of smoothness; and it does not generate any new data points [3].

A smoother interpolated image is obtained with bilinear (first order) interpolation. This algorithm uses the four input pixels surrounding the point (x,y) to estimate the output pixel (Fig. 1). Bilinear interpolating is usually implemented by first convolving the input image along its rows, creating new interpolated columns of pixels, and then along the new columns to create new interpolated rows of pixels, with a triangle weighting function in both directions [4]. The intensity value $I(x,y)$ at the interpolated point P at (x,y) in the image can be considered as

$$I(x, y) = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} I(x_1, y_1) + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} I(x_2, y_1) + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} I(x_1, y_2) + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} I(x_2, y_2) \quad (1)$$

where $I(x_1, y_1)$, $I(x_1, y_2)$, $I(x_2, y_1)$, and $I(x_2, y_2)$ are the intensity values of the four neighboring pixels [6].

Bilinear interpolation uses more memory than the nearest neighbor method, and requires slightly more execution time [3]. Although the bilinear interpolation is smooth, there remains a jagged pattern along the curved road at the top. This is aliasing that originated in the digitization of the photograph [4].

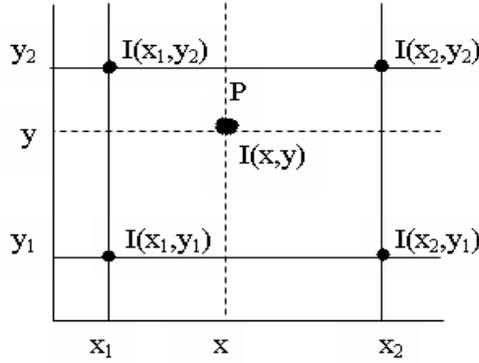


Figure (1): Bilinear interpolation using 4 neighboring points

The smoothing incurred with bilinear interpolation may be avoided with bicubic (second-order) interpolation, at the expense of more computation [4]. The output pixel value is a weighted average of pixels in the nearest 4-by-4 neighborhood [7]. The general form for a bicubic interpolation is given by [6] as

$$I_{xy} = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{21}x^2y + a_{12}xy^2 + a_{22}x^2y^2 + a_{30}x^3 + a_{03}y^3 + a_{31}x^3y + a_{13}xy^3 + a_{32}x^3y^2 + a_{23}x^2y^3 + a_{33}x^3y^3 \quad (2)$$

where $a_{i,j}$ is 4x4 array containing the values of the 16 surrounding points.

x : horizontal distance between the point and the four points left of it, between (0-1)

y : vertical distance between the point and the four points below it, between (0-1)

The number of pixels considered affects the complexity of computation, therefore, the bicubic method takes longer time than bilinear interpolation. However, the greater the number considered, the more accurate effect is, so there is a tradeoff between processing time and quality [7].

3. Iterative Curve Based Interpolation

Iterative curve based interpolation (ICBI) method is based on the composition of two different procedures. The first is a simple rule-based hole-filling method (filling gabs in the enlarge grid at locations $(2x+1, 2y+1)$) which computes new samples by interpolating along the direction where the second order image derivative is lower. The second refines the values of the interpolated pixels through an iterative algorithm trying

to force second order derivative continuity with some additional constraints for edge preservation [2]. First, the energy term is introduced as

$$\begin{aligned}
 U_c(2x+1,2y+1) = & \\
 & w_1 \left(\left| I_{11}(2x,2y) - I_{11}(2x+1,2y+1) \right| + \left| I_{22}(2x,2y) - I_{22}(2x+1,2y+1) \right| \right) + \\
 & w_2 \left(\left| I_{11}(2x,2y) - I_{11}(2x+1,2y-1) \right| + \left| I_{22}(2x,2y) - I_{22}(2x+1,2y-1) \right| \right) + \\
 & w_3 \left(\left| I_{11}(2x,2y) - I_{11}(2x-1,2y+1) \right| + \left| I_{22}(2x,2y) - I_{22}(2x-1,2y+1) \right| \right) + \\
 & w_4 \left(\left| I_{11}(2x,2y) - I_{11}(2x-1,2y-1) \right| + \left| I_{22}(2x,2y) - I_{22}(2x-1,2y-1) \right| \right)
 \end{aligned} \tag{3}$$

where I_{11} , I_{22} are local approximations of second order derivatives, computed as

$$I_{11}(2x+1,2y+1) = I(2x-2,2y-2) + I(2x+2,2y+2) - 2I(2x+1,2y+1) \tag{4}$$

$$I_{22}(2x+1,2y+1) = I(2x-2,2y+2) + I(2x+2,2y-2) - 2I(2x+1,2y+1) \tag{5}$$

The energy term in equation (3) sums local directional changes of second order derivatives. Weights w_i are set to 1 when the first order derivative in the corresponding direction is not larger than a fixed threshold and to 0 otherwise. In this way smoothing is avoided because there is a strong discontinuity. The minimization of this term roughly corresponds to force continuity in the local curvatures of the continuous surface ideally describing the image. The only problem of this method can be an excessive smoothing, which can be reduced by adding an energy term to enhance the absolute value of the second order derivative [2].

$$U_e(2x+1,2y+1) = -|I_{11}(2x+1,2y+1) + I_{22}(2x+1,2y+1)| \tag{6}$$

This energy alone is not able to produce very good and natural interpolated images, therefore it is necessary to add a derived potential to the curvature continuity and curvature enhancement term to improve the quality of interpolation results in edge regions. The potential is defined as

$$U_i(2x+1,2y+1) = f(I) \Big|_{2x+1,2y+1} I(2x+1,2y+1) \tag{7}$$

with

$$f(I) = -\frac{I_1(x,y)^2 I_{22}(x,y) - 2I_1(x,y)I_2(x,y)I_{12}(x,y) + I_{22}(x,y)^2 I_1(x,y)}{I_1(x,y)^2 + I_2(x,y)^2} \tag{8}$$

The ICBI method considers the effects of the curvature continuity, and curvature enhancement. By proper weighting between these effects, the ICBI method produces perceptually pleasant image. However, the performance depends on the chosen parameters [8].

4. Experimental Results

The experiment presented on four images “img1”, “img2”, “img3” & “img4” which are used as benchmark images for testing to demonstrate the generalization ability of the used approaches. The ICBI method has been compared with two traditional interpolation methods namely bilinear and bicubic methods. The algorithms used in this paper implemented in matlab running on a PC with Intel(R) Pentium(R), 1.70GHz CPU and 512MB Ram. The test images as shown in Figures (2a), (3a), (4a) & (5a) are resized by a factor of two, four and eight with the decimation approach

respectively. Then the images in Figures (2 b-d), (3 b-d) (4 b-d) and (5 b-d) are interpolated to 256×256 by a factor of two, four and eight, respectively using the bilinear, bicubic, and ICBI interpolation methods. The interpolated images are compared with the original high-resolution images for subjective evaluation.

For subjective performance evaluation, the Peak Signal-to-Noise Ratio (PSNR) is used and defined as in equation (7) below:

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right), \quad (9)$$

$$\text{with } MSE = \frac{1}{H \times W} \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} |L_{x,y} - Y_{x,y}|^2 \quad (10)$$

where $L_{x,y}$ and $Y_{x,y}$ are the pixels in the original and interpolated images at location (x,y) , respectively. H & W are the height and width of the image, respectively [8]. The PSNR numbers indicate the overall image quality with errors between the original and interpolated results pixel-by-pixel [5]. The results of PSNR on the test images (Img, Img2, Img3 and Img4) are summarized in Table (1). The ICBI method produced clearly the bested results, but the most interesting fact is that the iterative method is relevantly slower than bilinear and bicubic methods.

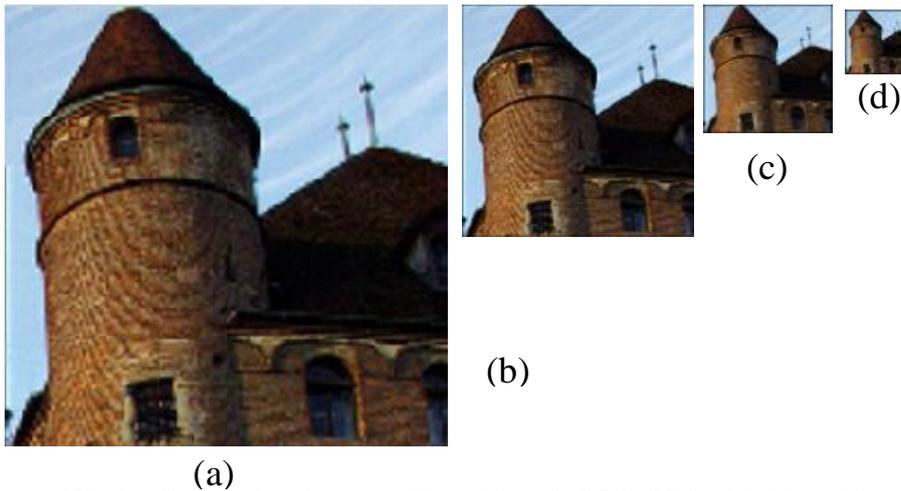


Figure (2): (a) Original HR image “*Img1*” with 256×256 . (b) “*Img1*” resized to 128×128 . (c) “*Img1*” resized to 64×64 (d) “*Img1*” resized to 32×32 .

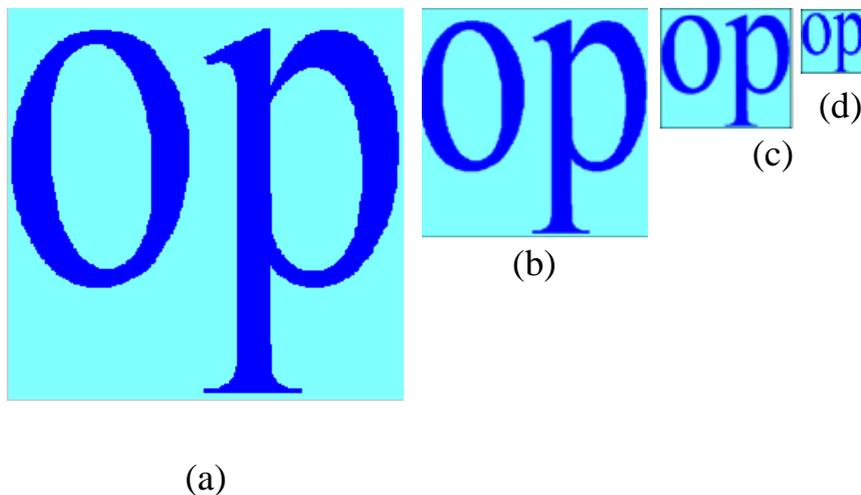


Figure (3): (a) Original HR image “*img2*” with 256×256 (b) “*Img2*” resized to 128×128 (c) “*Img2*” resized to 64×64 (d) “*Img2*” resized to 32×32 .

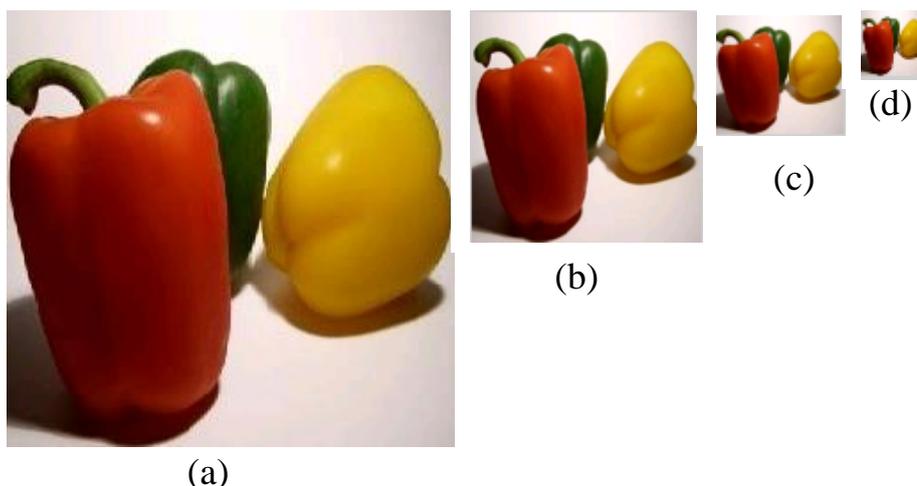


Figure (4): (a) Original HR image “*Img3*” with 256×256 (b) “*Img3*” resized to 128×128 (c) “*Img3*” resized to 64×64 (d) “*Img3*” resized to 32×32.

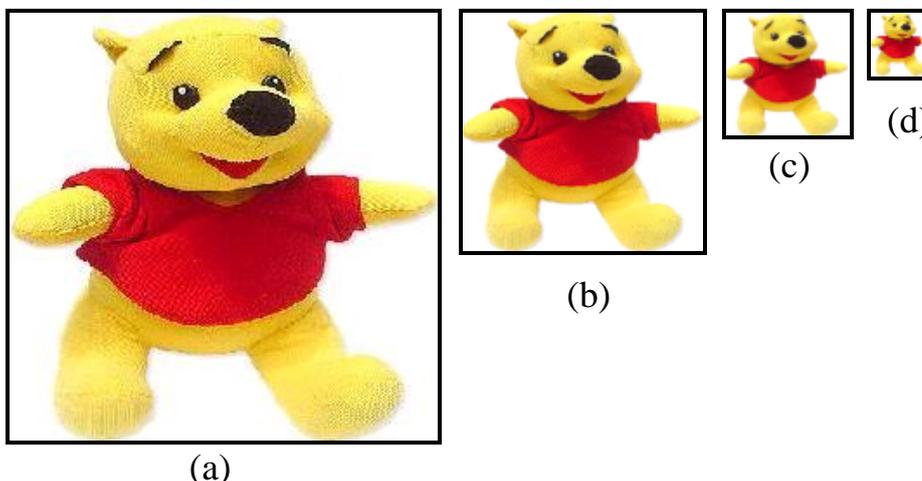


Figure (5): (a) Original HR image “*Img4*” with 256×256 (b) “*Img4*” resized to 128×128 (c) “*Img4*” resized to 64×64 (d) “*Img4*” resized to 32×32.

Table 1: PSNR values in dBs of interpolating “*Img1*”, “*Img2*”, “*Img3*”, & “*Img4*” images with bilinear, bicubic, and ICBI interpolation methods.

<i>Images</i>	<i>Image Size</i>	<i>Bilinear Interpolation</i>	<i>Bicubic Interpolation</i>	<i>ICBI interpolation</i>
<i>Img1</i>	128×128	31.556	32.709	32.740
	64×64	25.563	26.276	26.499
	32×32	22.297	22.757	23.135
<i>Img2</i>	128×128	25.500	25.988	26.483
	64×64	21.926	22.490	23.023
	32×32	19.041	19.587	20.159
<i>Img3</i>	128×128	33.253	34.252	34.984
	64×64	27.135	28.122	28.802
	32×32	23.087	23.853	24.064
<i>Img4</i>	128×128	27.126	27.406	27.547

	64×64	24.670	25.089	25.236
	32×32	22.443	22.894	23.004

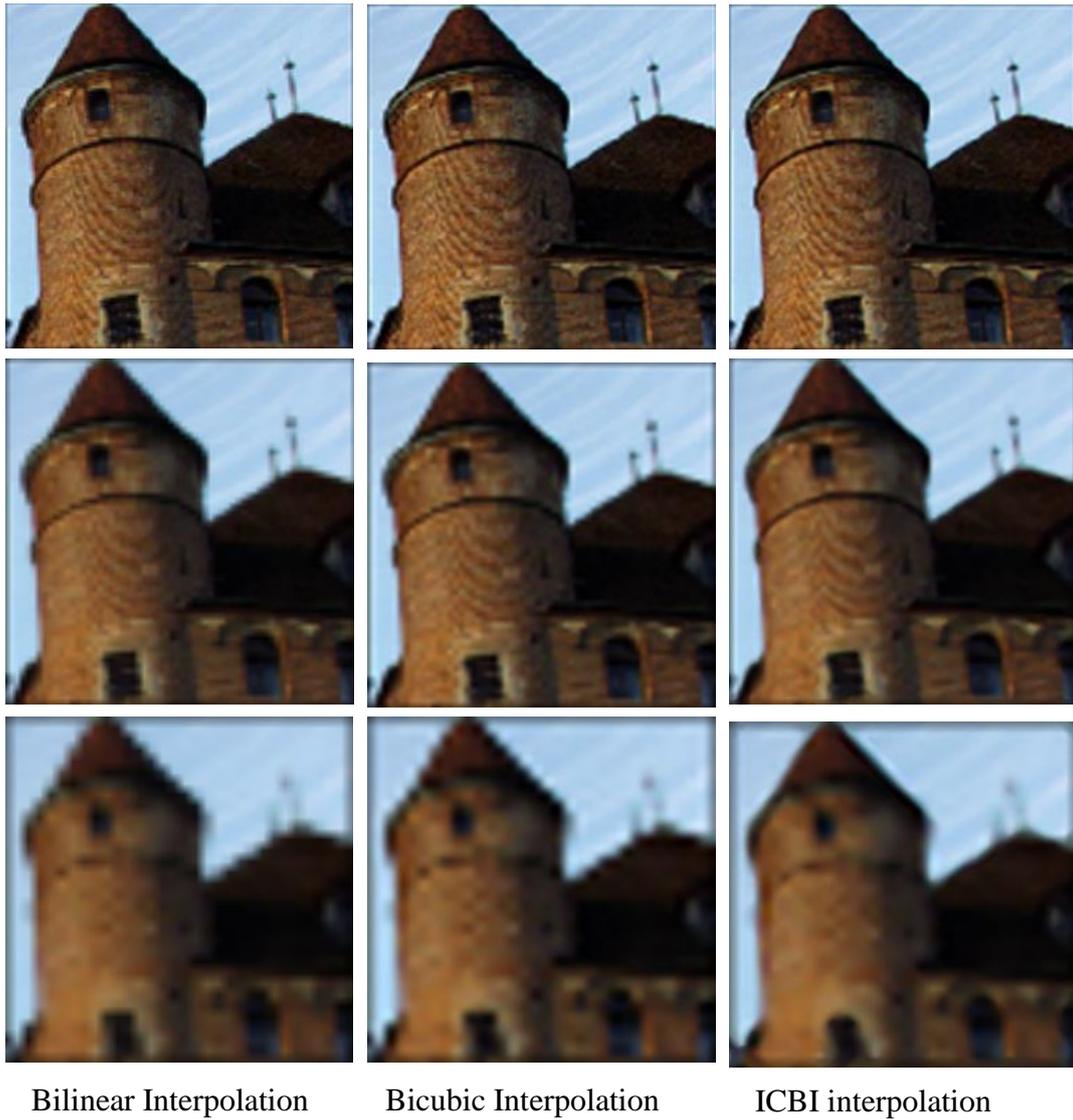
In order to compare perceived image quality, the tested RGB images are taken and resized them by a factor of two, four and eight with three different algorithms (bilinear, bicubic, and ICBI). Then a group of 15 people are asked to compare them in order to select the method providing the best average “perceived quality”. All the different possible couples of corresponding images are presented (in random order) to the subjects involved in the test, who are asked to choose the preferred image for each of them. The sum of the successful comparisons for each interpolation method is then taken as the quality score of the method itself.

For visual quality comparisons, Figures (6), (7), (8) & (9) show the resolution enhanced portions of “Img1”, “img2”, “Img3” and “img4” images by using bilinear, bicubic, and ICBI methods. It can be noticed that the bilinear interpolation made some jaggedness along the edges and blurred the image. The bicubic method made less blurring than the bilinear method, but it made more observable jagged edges. The used ICBI method creates images that have natural aspect and present less artifacts than the images interpolated with bilinear and bicubic technique. So, images interpolated with ICBI are always rated as the best. The program steps of the performance the ICBI method is presented as followings:

```

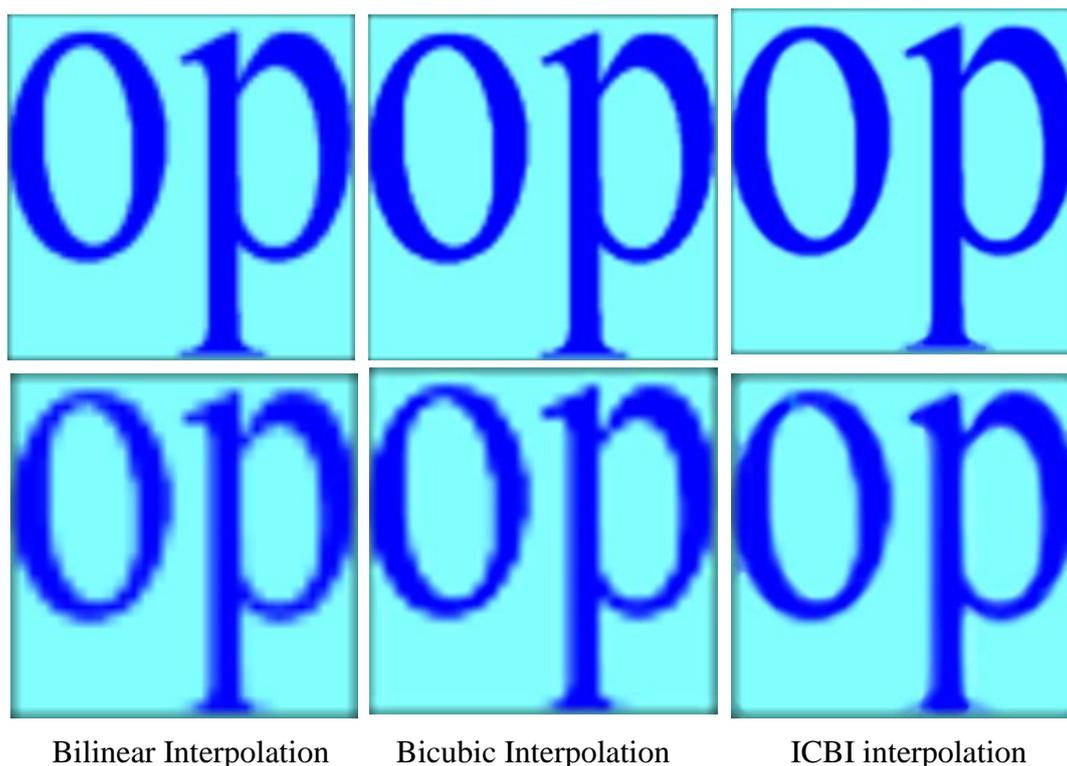
ImgExp(x,1)=(ImgExp(x-1,1)+ImgExp(x+1,1))/2;    % left column
ImgExp(x,W)=(ImgExp(x-1,W)+ImgExp(x+1,W))/2;    % right column
ImgExp(1,y)=(ImgExp(1,y-1)+ImgExp(1,y+1))/2;    % top row
ImgExp(H,y)=(ImgExp(H,y-1)+ImgExp(H,y+1))/2;    % bottom row
% Compute second order derivatives
A1=abs(ImgExp(x-1,y-1+s)-ImgExp(x+1,y+1-s));
A2=abs(ImgExp(x+1-s,y-1)-ImgExp(x-1+s,y+1));
B1=(ImgExp(x-1,y-1+s)+ImgExp(x+1,y+1-s))/2;
B2=(ImgExp(x+1-s,y-1)+ImgExp(x-1+s,y+1))/2;
% iterative refinement
for St=1:Mnum          % Mnum: Maximum number of iterations
if St < Mnum/4
step = 1;
elseif St < Mnum/2
step=2;
elseif St < 3*Mnum/4
step=2;
end
% computation of derivatives
C1(x,y)=(ImgExp(x-1+s,y-1)-ImgExp(x+1-s,y+1))/2;
C2(x,y)=(ImgExp(x+1-(2*s),y-1+s)-ImgExp(x-1+(2*s),y+1-s))/2;
D1(x,y)=ImgExp(x-1+s,y-1)+ImgExp(x+1-s,y+1)-2*ImgExp(x,y);
D2(x,y)=ImgExp(x+1,y-1+s)+ImgExp(x-1,y+1-s)-2*ImgExp(x,y);
D3(x,y)=(ImgExp(x-s,y-2+s)-ImgExp(x-2+s,y+s)+ImgExp(x+s,y+2-s)-
s)-ImgExp(x+2-s,y-s))/2;

```

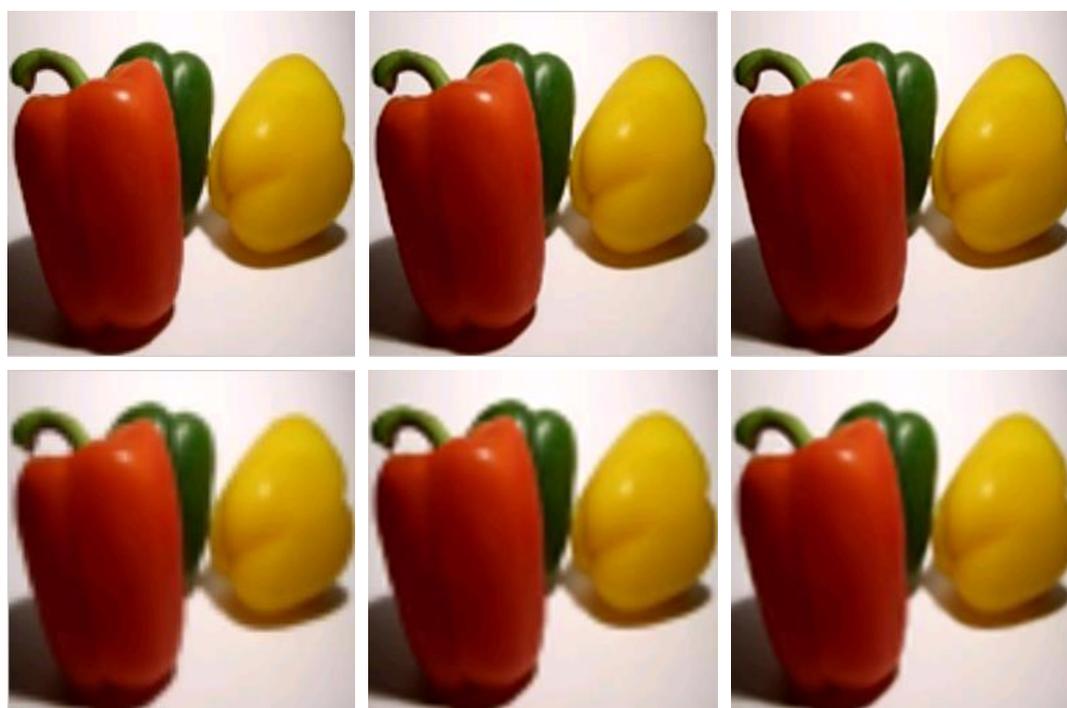


Figure(6): Interpolating of “*Img1*” image with bilinear, bicubic, and ICBI interpolation methods, respectively. Upper line interpolating “*Img1 with size 128×128*”, middle line interpolating “*Img1 with size 64×64*”, and lower line interpolating “*Img1 with size 32×32*”





Figure(7): Interpolating of “*Img2*” image with bilinear, bicubic, and ICBI interpolation methods, respectively. Upper line interpolating “*Img2* with size 128×128 ”, middle line interpolating “*Img2* with size 64×64 ”, and lower line interpolating “*Img2* with size 32×32 ”



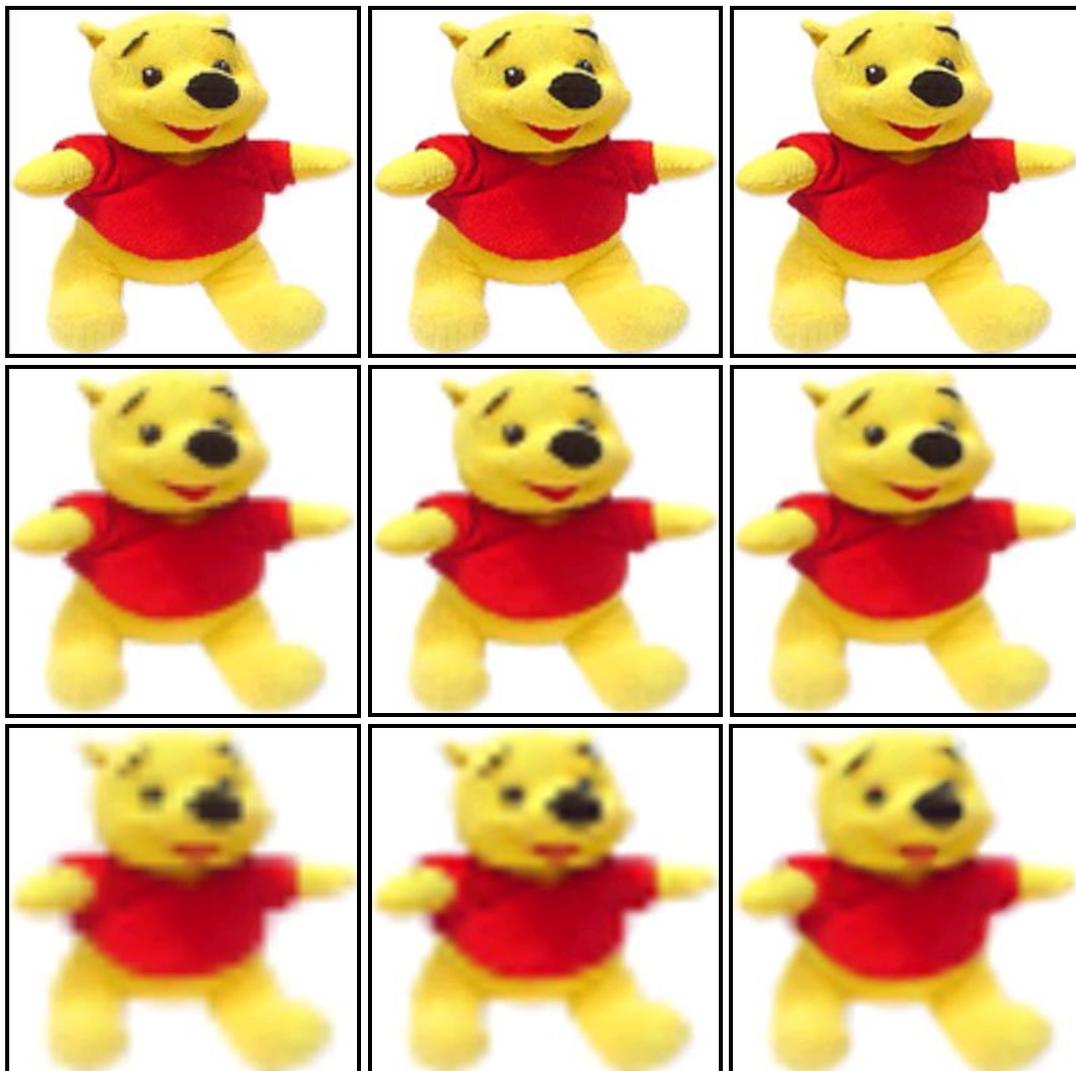


Bilinear Interpolation

Bicubic Interpolation

ICBI interpolation

Figure(8): Interpolating of “*Img3*” image with bilinear, bicubic, and ICBI interpolation methods, respectively. Upper line interpolating “*Img3* with size 128×128 ”, middle line interpolating “*Img3* with size 64×64 ”, and lower line interpolating “*Img3* with size 32×32 ”



Bilinear Interpolation

Bicubic Interpolation

ICBI interpolation

Figure(9): Interpolating of “*Img4*” image with bilinear, bicubic, and ICBI interpolation methods, respectively. Upper line interpolating “*Img4* with size 128×128 ”, middle line interpolating “*Img4* with size 64×64 ”, and lower line interpolating “*Img4* with size 32×32 ”

5. Conclusions

In this paper, the implementation of the Iterative Curvature Based Interpolation (ICBI) method and the traditional (bilinear, bicubic) methods on “img1” and “img2” images are presented. The results show that the traditional methods put the original pixels in an enlarged grid and fill holes averaging neighboring pixels in two steps whereas, the ICBI method is interpolated directionally in order to preserve locally second order derivatives. Thus the (ICBI) adds to the previous method an iterative refinement that try to reduce second order derivatives changes keeping original pixel values fixed and also preserving large gray level variations and smoothing.

The performance of the ICBI method has been verified with extensive simulation and comparison with bilinear and bicubic interpolation methods. Simulation results showed that the presented method has achieved outstanding perceptual performance with consistent subjective and objective performance. Finally, because of iteration in ICBI method and manual programming in matlab, the execution time will increase when using this technique.

REFERENCES

- [1] Andrew G., 2009, “Least-Squares Optimal Interpolation for Direct Image Super-Resolution”, Ph.D. thesis, Massey University, Palmerston North, New Zealand, p13-14.
- [2] Giachetti A., and Asuni N., 2008, “Accuracy improvements and artifacts removal in edge based image interpolation”, In Proc. 3rd Int. Conf. Computer Vision Theory and Applications, Funchal, Madeira, Portugal, January 22-25, Volume 1, p3.
- [3] Muserref T., 2008, “Digital Image Processing of Remotely Sensed Oceanographic Data”, MSc. thesis, Department of Scientific Computing, The Graduate School of Applied Mathematics, The Middle East Technical University, p41.
- [4] Robert A. S., 2007, “Remote Sensing: Models and Methods for Image Processing”, By Elsevier Inc. All rights reserved, 3rd Edition, p300-304.
- [5] Sheng-Fu. L., Hung-Ming. C., and Yi-Che L., 2009, “Image Enlargement by Applying Coordinate Rotation and Kernel Stretching to Interpolation Kernels”, Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, p2-14.
<http://www.hindawi.com/journals/asp/2010/576831.html>
- [6] Tinku A., and Ping-Sing T., 2007, “Computational Foundations of Image Interpolation Algorithms”, By ACM Ubiquity, Vol. 8, p1-4.
- [7] User’s Guide, 2008, “Image Processing Toolbox™”, By The MathWorks, Inc., 6th Edition, p(6-3).
- [8] Wing-Shan T., Chi-Wah K., and Wan-Chi S., 2009, “A Modified Edge Directed Interpolation for Images”, 17th European Signal Processing Conference (EUSIPCO), Glasgow, Scotland, August 24-28-2009, By EURASIP, p283.