International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 6 521 - 526

Performance Analysis of Image Scaling Algorithms

P. S. Parsania,
Research Scholar, Department of Computer Science,
Sardar Patel University,
Vallabh Vidyanagar, India
pankaj_parsania@yahoo.com

P. V. Virparia,
Director, Department of Computer Science,
Sardar Patel University,
Vallabh Vidyanagar, India
pvvirparia@yahoo.com

***Abstract*—** Image scaling is the process of resizing a digital image, it is one of the main operations in image processing task such as computer graphics, gaming, medical image processing, virtualization, camera surveillance and quality control. Different image interpolation algorithms are used in resizing digital images. To resize an image, every pixel in the new image must be mapped back to a location in the old image in order to calculate a value of new pixel. There are many algorithms available for determining new value of the pixel, most of which involve some form of interpolation among the nearest pixels in the old image. In this paper, we used Nearest-neighbor, Bilinear, Bicubic, Bicubic B-spline, Catmull-Rom, Lanzcos of order two and Lanzcos of order three algorithms for image interpolation. Each algorithms generates varies artifact such as aliasing, blurring and moiré which results in to different look to the scaled image. This paper presents analysis of quality and computational time consideration of images while using these interpolation algorithms.

Keywords- *Image Scaling, Image resize, Interpolation, Nearest-neighbor, Bilinear, Bicubic, cubic B-spline, Catmull-Rom, Lanzcos*

*****

I. INTRODUCTION

Image scaling is a procedure used in expanding and contracting digital images, but is often used with different terminologies in literature like image interpolation, image resampling and image resize [1]. There are many algorithms currently in use for the resizing of digital images. Most of them attempt to reproduce a visually attractive replica of the original. With the recent integration of multimedia into almost every aspect of daily lives, consumers are viewing images, video, and other visual data on a wide variety of products, ranging from televisions to computer screens to a large selection of hand-held devices. [2]. While enlarging an image it is not possible to discover any more information in the image than already exists, and image quality inevitably suffers. However, there are several methods of increasing the number of pixels that an image contains, which can be created out of original pixels. These methods are often termed as image interpolation algorithms.

When the image is interpolated from a higher resolution to a lower resolution, it is called as image down-scaling or down sampling. On the other hand, when the image is interpolated from a lower resolution to a higher resolution, it is referred as image up-scaling or up-sampling [3]. Image interpolation has a variety of applications in the areas of computer graphics, digital image editing, medical image reconstruction; for instance scaling up is used to enlarge images for HDTV or medical image displays and scaling down is applied to shrink images to fit mini-size LCD panel in portable instruments. It is also a part of many commercial image processing tools or freeware graphic viewers such as Adobe Photoshop CS2 software, IrfanView, Fast Stone Photo Resizer, Photo PosPro, XnConvert etc.

There are various algorithms currently in use for scaling of digital images. Some of the common image scaling algorithms are the nearest neighbor, bilinear, and bicubic [4]. Other image scaling algorithms such as Cubid B-Spline and Catmull-Rom generates better image quality. Lanczos algorithm utilizes 2-lobed, 3-lobed or 4-lobes sinc window function to implement interpolation [5]. These algorithms uses higher order interpolators which take more surrounding pixels into consideration, and thus also require more computation time, and retain the most of image details after an interpolation. They are extremely useful when the image requires multiple translation, rotations or distortions in separate steps. However, for single-step enlargements or image operations, these higher order algorithms provide diminishing visual improvement and processing time increases significantly.

Interpolation using Shape-Preserving Approach [6], classification and stitching [7], grid based image interpolation [8], auto-regression based method [9], seam carving [10] and interpolation used with Artificial Neural Networks [11] are some of the novel techniques used for interpolating images. Although these algorithms perform well, they require a lengthy processing time due to their complexity. This is intolerable for real-time image scaling. Hence, these algorithms have not been considered for the comparative analysis in this paper.

In this paper, various image scaling algorithms are implemented using ImageJ, an open source java software developed by Wayne Rasband at National Institute for Health (NIH) used for image processing. These algorithms are used for scaling digital images with scaling parameters in x and y directions. The result obtained is presented and summarized with quality and time complexity parameters.

II. IMAGE SCALING ALGORITHMS

The basic principle of image scaling is to increase the image pixel numbers, so a low resolution image is converted to a high resolution image. There are many different types of scaling algorithms, each resulting in a different look to final image. Thus, it is best if the quality, or visible distinction for each pixel, is retained throughout the interpolation function. Interpolation function is performed by convolution operation which involves a large number of addition and multiplication operations. Hence, a trade-off is required between computation complexity and quality of the interpolated image [12].

When a small image is enlarged, for example if an image is scaled as shown in Fig. 1., the color values of original 4 adjacent pixels marked A, B, C, and D in (a) were filled in the new A, B, C, and D locations in (b) accordance with the scaling

factor. But there are a large number of pixels which values are unknown between A, B, C, and D, such as P. so the values of these pixels should be calculated through interpolating estimation [13]. In this paper, we used Nearest-neighbor, Bilinear, Bicubic, Bicubic Cubic B-spline, Catmull-Rom, Lanzcos of order two and Lanzcos of order three algorithms for image scaling.

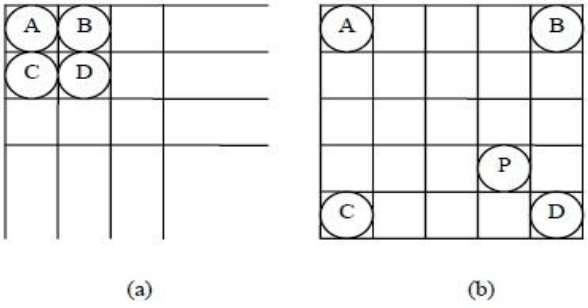

Figure 1. Schematic diagram of image enlargement

A. *Nearest Neighbour*

The Nearest Neighbor interpolation is the fastest and simplest option. It simply takes the color of a pixel and assigns it to the new pixels that are created from that pixel. Due to this simplistic approach, it does not create an anti-aliasing effect. Using this method one finds the closest corresponding pixel in the source (original) image for each pixel in the destination image. New pixels are made the same as others close-by. The pixels or dots of color are duplicated to create new pixels as the image grows. It creates pixilation or edges that break up curves into steps or jagged edges. This form of interpolation suffers from normally unacceptable effects for both enlarging and reduction of images. Nearest Neighbor interpolation is considered to be incapable of producing photographic quality work. [14].

B. *Bilinear Interpolation*

Bilinear interpolation takes a weighted average of the four neighborhood pixels to calculate its final interpolated value. The result is much smoother image than the original image. When all known pixel distances are equal, then the interpolated value is simply their sum divided by four. This technique performs interpolation in both directions, horizontal and vertical. This technique is give better result than nearest neighbor interpolation and take less computation time compare to bicubic interpolation [15].

C. *Bicubic Interpolation*

Bicubic goes one step beyond bilinear by considering the closest 4x4 neighborhood of known pixels for a total of 16 pixels. Since these are at various distances from the unknown pixel, closer pixels are given a higher weighting in the calculation. Bicubic produces noticeably sharper images than the previous two methods, and is perhaps the ideal combination of processing time and output quality. For this reason it is a standard in many image editing programs including Adobe Photoshop, printer drivers and in-camera interpolation [16].

D. *Cubic B-Spline*

As bicubic interpolation, the cubic B-spline interpolation algorithm also interpolates from the nearest sixteen source pixels. However, this algorithm uses B-spline interpolating functions instead of cubic splines, which in general yield quite smooth results. It performs a convolution with a two-dimensional non separable filter, so its complexity is increased. In contrast, bicubic interpolation uses a convolution with a separable filter, and hence its complexity is less. Despite this performance difference, cubic B-spline has interesting characteristics of smoothness that make it a good option in some cases [5].

E. *Catmull-Rom Interpolation*

Catmull-Rom is a local interpolating spline developed for computer graphics purposes. Its initial use was in design of curves and surfaces, and has recently been used several applications. Catmull-Rom splines are a family of cubic interpolating splines formulated such that the tangent at each point is calculated using the previous and next point on the spline. The results are similar to ones produced by bicubic interpolation with regards to sharpness, But the Catmull-Rom reconstruction is clearly superior in smooth signal region [5].

F. *Lanczos interpolation*

Lanczos interpolation function is a mathematical formula used to smoothly interpolate the value of a digital image between its samples. It maps each sample of the given image to a translated and scaled copy of the Lanczos kernel, which is a sinc function windowed by the central hump of a dilated sinc function. The sum of these translated and scaled kernels is then evaluated at the desired pixel [17]. Lanczos interpolation has the best properties in terms of detail preservation and minimal generation of aliasing artifacts for geometric transformations not involving strong down sampling. The number of neighboring pixels considered varies as the order of the kernel. If the order is chosen to be 2, 16 pixels are considered while if the order is 3, 36 neighboring pixels are utilized for interpolation. However the higher order Lanczos interpolation require high computational time, which make them not suitable for the many commercial software [18].

III. RESULT ANALYSIS

All seven algorithms are implemented using ImageJ. ImageJ is an open source java software developed by Wayne Rasband at National Institute for Health (NIH) used for image processing. The ImageJ distribution for Windows includes a Java compiler which allows to compile plugins inside ImageJ. Plugins are small Java modules for extending functionality of ImageJ by using simple standardized interface. Plugin can be created, edited, compiled and organized through the *plugin* menu in ImageJ's main window.

For this paper we used ImageJ 1.50i software with java 1.8.0_77 (64bit). The software installed on a standard Window 10 Professional machine having Intel® Core i5 CPU @ 3.30 Ghz and installed DDR3 RAM of 4GB, is used for compilation and execution of java programs. Two input test image of size 512x512 (*Girl and House*) were downloaded from the USC-SIPI Image Database provided by the University of Southern California. Two images were scaled down to 128x128 pixel size for this research. The corresponding interpolated output images of size 256x256 produced by seven image scaling algorithms are shown in Fig. 2. Also, comparison of time complexity of all seven image scaling algorithms is was performed for *Girl* image.

 ISSN: 2321-8169
521 - 526

(a)

(a)

(b)

(b)

(c)

(c)

ISSN: 2321-8169
521 - 526

(d)

(d)

(e)

(e)

(f)

(f)

(g)

(g)

(h)

(h)

Figure 2. (a) Input image (Girl and House) (b) Nearest-neighbour (c) Bilinear (d) Bicubic, (e) Cubic B-spline (f) Catmull-Rom

(g) Lanczos order 2 (h) Lanczos order 3

For estimating the quality of each scaling algorithm, two test images (*Girl and House*) were used. While dealing with different algorithms it always face a trade-off between the three image scaling artifacts – aliasing, blurring and moiré. Nearest neighbour interpolation produces a high aliasing effect resulting in jagged edges. Bilinear interpolation reduces the aliasing effect but causes a moderate blurring of the edges. Bicubic interpolation produces a moderate aliasing, blurring and moiré effect. Cubic B-spline interpolation make the curve more smoothly and the image edge more perfect which exhibit no discontinuity, corners, or abrupt changes to the scaled image. Catmull-Rom interpolation is much superior in smooth signal region and provides the sharpness as Bicubic algorithms with better quality of the scaled image. Lanczos interpolation of order two and three delivers an image quality very similar to that of Bicubic, but these algorithms are computationally very intensive as they require 16 and 36 pixels respectively for interpolating an output pixel value. Lanczos interpolation also utilizes the sine function repeatedly which itself requires a large number of addition and multiplication operations.

Comparison of time complexity of all seven image scaling algorithms is shown in Table 1 for *Girl* image. Time complexity was measured for scaling 256x256 image to 512x512 (by scaling factor of 2 or 200 percentage) and 1024x1024 (by scaling factor of 4 or 400 percentage). Computationally, nearest neighbour interpolation is the least intensive as it considers only one pixel for interpolation Bilinear uses 4 diagonal pixels for interpolation and therefore, requires more computation than the nearest neighbour method. Bicubic, cubic B-Spline and Catmull-Rom uses 16 pixels to interpolate image with different functional parameters. Lanczos interpolation of order 2 and 3 are computationally very intensive as they require 16 and 36 pixels respectively for interpolating an output pixel value hence Lanczos order 3 uses the maximum time of all the seven algorithms due to more number of pixel consideration.

TABLE I. TIME COMPLEXITY OF SCALING ALGORITHMS

Scaling Algorithms	Processing Time in Second(s)	
	200 % Scaling Factor	*400 % Scaling Factor*
Nearest neighbour	0.016	0.031
Bilinear	0.046	0.067
Bicubic	0.078	0.342
Cubic B-Spline	0.110	0.381
Cutmull-Rom	0.114	0.475
Lanczos order 2	0.891	4.596
Lanczos order 3	2.141	9.914

IV. CONCLUSION

Image scaling is a basic image operation used in image processing and plays an important role in many computer graphics applications. Although there are myriad of interpolation algorithms available in literature, not all of them are suitable for implementing real-time image applications. In this paper, we discussed the main non-adaptive image scaling algorithms suitable for real-time applications. A comparative analysis of seven image interpolation algorithms is presented based on the results of their software implementation.

Of the seven image interpolation algorithms evaluated, nearest neighbour scaling algorithm is the most simple and fast algorithm. It has the advantages of fast speed, but it can bring significant distortion and it will appear mosaic and produces a high aliasing effect. Bilinear interpolation method is more complex than the nearest neighbour method, and so it has larger calculation. It has no gray discontinuity defects and has satisfactory results. Bicubic and Cubic B-spline generates interesting result and can get relatively clear picture quality. Bicubic uses less time complexity compared to Catmull-Rom, Lanczos order 2 and Lanczos order 3. Catmull-Rom algorithm offer the best result between image quality and computational complexity for real-time image interpolation applications. Lanczos order 2 and Lanczos order 3 using maximum time computational complexity as they require 16 and 36 pixels respectively for interpolating an output pixel value, hence not suitable for real-time applications.

REFERENCES

[1] H. S. Prasantha, H. L. Shashidhara, and K. N. Balasubramanya Murthy, "Image Scaling Comparison Using Universal Image Quality Index," International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 859-863, Dec. 2009.
[2] E. E. Danahy, S. S. Agaian and K. A. Panetta, "Algorithms for the resizing of binary and grayscale images using a logical transform," Electronic Imaging, International Society for Optics and Photonics, pp. 64970Z-64970Z, Feb. 2007.
[3] S. Safinaz, "An Efficient Algorithm for Image Scaling with High Boost Filtering," International Journal of Scientific and Research Publications, vol, 4, no. 5, May 2014.
[4] R. C. Gonzalez, Digital image processing. Pearson Education India; 2009.
[5] W. Burger, and M. J. Burge, Digital image processing: an algorithmic introduction using Java, 1st ed., Springer, 2009.
[6] G. X. Zhang, M.M. Cheng, S.M. Hu, and R.R. Martin, "A Shape-Preserving Approach to Image Resize," Pacific Graphics 2009, vol. 28, no. 7, Sept. 2009.
[7] N. Muelle and T. Q. Nguyen, "Image interpolation using classification and stitching," Proc. of the IEEE International Conference on Image Processing, pp. 901-904, Oct. 2008.
[8] H.B. Kekre, T Sarod and S Thepade, "Grid based image scaling technique," International Journal Of Computer Science And Applications., vol. 1, no. 2, Aug. 2008.
[9] W. Zhe, Z. Jiefu and Z. Mengchu, "A Fast Autoregression Based Image Interpolation Method," Proc. of the IEEE International Conference on Networking, Sensing and Control, pp.1400-1404, April. 2008.
[10] D. Domingues, A. Alahi and P. Vandergheynst, "Stream carving: an adaptive seam carving algorithm," IEEE International Conference on Image Processing, pp. 901-904, Sept. 2010.
[11] F. Liang and K. Xie, "An Image Interpolation Scheme combined with Artificial Neural Network", In Proc. of the Third International Conference on Natural Computation, pp. 99-102, 2007.
[12] C. Singh, S. Singh, R. Saini and A. K Saini, "A Comparative Analysis of Image Scaling Algorithms". International Journal of Image, Graphics and Signal Processing, pp.55-62, April. 2013
[13] D. Han, "Comparison of commonly used image interpolation methods," ICCSEE, Atlantic Press, Hangzhou, China. Jan. 2013.
[14] K. Shah, J. Pandya and S. Vahora, "A Survey On Super Resolution Image Reconstruction Techniques", International Journal of Engineering Research & Technology, vol. 2 no. 4, pp. 1897-1901, April.2013.
[15] S. Fadnavis, "Image Interpolation Techniques in Digital Image Processing: An Overview," International Journal of Engineering Research and Applications, vol,. 4, no. 10, pp.70-73, Oct. 2015
[16] A. Prajapati, S. Naik and S. Mehta, "Evaluation of Different Image Interpolation Algorithms", International Journal of Computer Applications, Vol.58, Issue.1, pp. 6-12, Jan. 2012.
[17] P. S. Parsania and P. V. Virparia, "A Comparative Analysis of Image Interpolation Algorithms," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no.1, Jan. 2016, pp. 29-34, DOI 10.17148/IJARCCE.2016.5107
[18] Wikipedia website. [Online]. Available: http://en.wikipedia.org/wiki/Lanczos_resampling, 2016