International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Computational Time Complexity of Image Interpolation Algorithms

P.S. Parsania[1*], P. V. Virparia[2]

[1] College of Food Processing Technology & Bio Energy, Anand Agricultural University, Anand, India
[2] Department of Computer Science Sardar Patel University, Vallabh Vidyanagar, India

**Corresponding Author: panakaj_parsania@yahoo.com, Tel.: +91-99090-08655*

Available online at: www.ijcseonline.org

Accepted: 14/Jul/2018, Published: 31/July/2018

Abstract— Image Interpolation is an important operation in many image processing software and applications. It is a process of enlarging or reducing the image size. To resize an image, every pixel in new image is calculated using the values of the pixels in old image. 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. After interpolating new values for pixel, it is important to preserve the image quality. As a result of digital image operations, various methods suffer from different edge-related visual artifacts such as aliasing, edge blurring, and jaggies effect. For our study we have used Nearest-neighbor, Bilinear, Bicubic, Cubic B-spline, Catmull-Rom, Lanczos of order two and Lanczos of order three image interpolation algorithms. In this paper, an attempt is made to evaluate different image interpolation algorithms to compare time performance on Intel Core i3, i5 and i7 processors supported with different hardware configuration. The result shows that more time is required to compute the larger image. However, the time can be minimized using higher end hardware configuration.

Keywords— Interpolation, Computational Complexity, adaptive, non-adaptive, image quality, resize, scaling.

I. INTRODUCTION

Image plays an important role in our everyday activity. Throughout the world, people use an image for social communication over Facebook, WhatsApp and other applications of similar type. These images are communicated over the internet using varied devices such as desktop computer, laptop or mobile phones. With the development of such a sophisticated device, there is a requirement to change the image size to fit the requirement of particular device. This can be possible using image interpolation. To resize an image new pixels needs to be calculated based on the original image. 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 and attempt to reproduce a visually attractive replica of the original.

Image interpolation is a process of resizing or scaling a digital image to a large size that results in generating a high-resolution image from the available low-resolution image. Interpolation is a method through which new pixel is constructed from the available pixel in an image. Image interpolation is a necessity when you want to enlarge or reduce image size. To enlarge an image new pixel needs to be added while reducing image size pixels from the image needs to be removed. As the image is enlarged or reduced, quality of image degrades and results into many artifacts such as aliasing, blurring, ringing and blocky or jaggies effect.

The image interpolation algorithms can be categories into two different methods. One is Non-Adaptive interpolation algorithms and other is known as Adaptive interpolation algorithms [1]. For our research, we used seven Non-Adaptive interpolation algorithms. These algorithms consider all the pixels in an image with similar priority depending upon the type of algorithm used. The logic used to find unknown pixel in a resized image remains constant, without considering the image features. Non-Adaptive algorithms include Nearest-neighbour, Bilinear, Bicubic, Cubic B-spline, Catmull-Rom, Lanczos of order two and Lanczos of order three image interpolation algorithms. Depending on the complexity of the algorithm they use somewhere between 0 to 256 adjacent pixels for interpolating [2]. Time and processing speed required while interpolating an image increase with the increase in a number of pixels considered to enlarge an image. There is a strong relationship between the quality of an image generated and the number of pixels considered to interpolate an image. High processing time is required by the Central Processing Unit to generate visually good looking enlarged image [3]. So, the quality of an image depends upon, how the interpolation algorithm is implemented.

The rest of the paper is organized as follows, Seven Non-Adaptive image interpolation algorithms are explained in section II. Results and related discussion explains about the software, hardware, and image used for research and its output generated is done in section III. Findings and concluding remarks are given in section IV.

II. IMAGE INTERPOLATION ALGORITHMS

Image interpolation algorithms convert or resize a digital image from one resolution (dimension) to another resolution without losing the visual content in the picture. There are numbers of image interpolation algorithms, each algorithm derives different result of output image depending upon the logic implemented. Thus, the algorithm is considered best if the resulted image maintains the quality without losing the feature of an image. In this paper, we used Nearest-neighbour, Bilinear, Bicubic, Bicubic Cubic B-spline, Catmull-Rom, Lanczos of order two and Lanczos of order three algorithms for generating images and measure time performance on different hardware platform [4].

II.1.*Nearest Neighbor*

The Nearest neighbour interpolation is most simple to implement as it only considers one pixel, the closest one to interpolate the point and does not consider the values of neighbouring points at all. It requires the least computation and takes least processing time. Using the Nearest Neighbour algorithm, the empty spaces will be filled in with the closest neighbouring pixel value. It simply makes each pixel larger by replicating the new pixel. The pixels or dots of colour are replicated to create new pixels as the image grows. This interpolation method is very efficient and does not create an anti-aliasing effect. The quality of an image generated using nearest neighbour is very poor as it creates pixelation or edges that break up curves into steps or jagged edges. This is not the suitable interpolation method for enlarging images because it results in blocky images so it is not used in high-quality imaging applications [5].

II.2.*Bilinear*

Bilinear interpolation identifies the four nearby pixels in an image and takes the distance-weighted average of these four pixels to determine new value. Since new a pixel is estimated according to the relative position of neighboring four pixels, results in smoother images than the Nearest Neighbor interpolation. The new image generated using the bilinear interpolation method will have smooth edges compared to the original image. If all the four pixels are equal distance from the computed pixel then the intensity of the new pixel will be simply average of four neighbor pixels [6]. This method will generate the blurring effect in an image. Since it takes more number of pixels for computation than Nearest Neighbor interpolation, it requires more processing time and produces better quality realistic image output [7].

II.3.*Bicubic*

Bicubic interpolation is an advanced image interpolation algorithm as it considers four by four adjacent pixels for a sum of 16 pixels. These sixteen pixels are at a different position from the computed pixel, more weight is given to nearby pixels according to its distance. The color intensity of the new pixel will be calculated by these 16 pixels according to their weighted average [8]. Bicubic interpolation generates sharper images than the nearest neighbor and bilinear interpolation methods. This method gives jaggies effect around sharp boundaries lines, which are more visible with the contrast color interpolated image. Computational time for this method is more as it takes more number of pixels for calculation. Bicubic interpolation algorithm produces an eye-pleasing image and it is a standard in the majority of image editing software [9].

II.4.*Catmull-Rom*

Catmull-Rom interpolation is a spline function which helps in preserving edges, results in a sharper image compared to Bicubic. Catmull-Rom interpolation uses tangent of each control points in calculation by using previous and next control points on the curve. The curve will pass through all the control points. If one of the control points is changed, the curve will affect between the two control points only. The curve will be generated on each segment by using two control points and tangent to a curve at each control point [10]. Image generated using this method is as sharp as Bicubic interpolation but generates more smoothness at edges. So, this method is clearly superior in the smooth signal region [11].

II.5.*Cubic B-Spline*

Splines are piecewise polynomials to generate continues signal by connecting different knots. Cubic B-Spline interpolation also requires the adjoining 16 source pixels to interpolate an image using B-Spline interpolating function. It generates smooth results at the edges of the image. The computational complexity of Cubic B-Spline interpolation is higher compared to Bicubic interpolation as it uses a two-dimensional non-separable filter, wherein the Bicubic interpolation uses a convolution function with a separable filter. In general, if you want a smoother result, Cubic B-Spline will perform better over Bilinear and Bicubic interpolation [12].

II.6.*Lanczos*

This method is based on the 2-lobed or 3-lobed Lanczos window function as the interpolation function, hence termed as Lanczos order two and Lanczos order three interpolation. It is to be used as a low-pass filter to smoothly interpolate the value of a digital signal between its samples. The effect of each input sample on the interpolated values is defined by the filter's reconstruction kernel, known as the Lanczos kernel [13]. It is the normalized sinc function, windowed by a central lobe of a horizontally stretched sinc function [14]. The Lanczos order two and order three interpolation algorithms use source image intensities at 16 pixels and 36 pixels respectively. The computational complexity of both these algorithms is high compared to all the above algorithms but provides high sub-pixel accuracy, better preservation of small-scale structures in an image and fewer generations of aliasing artifacts. The major disadvantage is there may be ringing artifacts just before and after abrupt changes in the sample values, which may lead to clipping artifacts [15].

III. RESULTS AND DISCUSSION

For time computation complexity, we used three different types of hardware configurations and the image interpolation was performed using Imagej. ImageJ is an image processing open source software with pre-installed Java. The software is developed by Wayne Rasband at the National Institute of Health (NIH). ImageJ is used to design user specific plugins for image operation to be performed on selected images. This software provides user extensibility using plugins or recorded macros. Plugins and macros can be added to the Plugin menu to create, edit and compile Java programming codes. ImageJ support verities of image file format including JPEG, TIFF, PNG, GIF, BMP, LUT, FIT, PGM and row data formats. It also supports multi-threaded image operations to performed parallel, utilizing multi-core facilities of CPU. For this research we have used three different types of hardware configuration having Intel® Core i3 CPU @ 2.40 GHz and DDR3 RAM of 2GB, Intel® Core i5 CPU @ 2.50 GHz and DDR3 RAM of 4 GB and Intel® Core i7 CPU @ 2.70 GHz and DDR3 RAM of 8GB, installed with the latest copy of ImageJ 1.50i software with java 1.8.0_77 to support 64 bits computer architecture for compilation and execution of interpolation algorithms.

For our research, we downloaded Lena image of size 512 X 512 pixels from the USC-SIPI Image Database. The image was scaled down to 128 X 128 pixels size as our test image and generated enlarged images of size 256X256 (2X), 512X512 (4X), 1024X1024 (8X), 1280xX1028 (10X), 2560X2560 (20X) and 6400X6400 (50X). The computational complexity of these images is measured using three different hardware configuration. Fig. 1 show the input test image (Lena) of size 128X128 Pixels. Fig. 2 to Fig. 8 shows the output images of size 256x256 (2X) generated using Nearest-neighbor, Bilinear, Bicubic, Cubic B-spline, Catmull-Rom, Lanczos of order two and Lanczos of order three image interpolation algorithms. However, valid coloured photographs can also be published.

Figure 1. Input test image Lena (128 X 128) Pixels.

Figure 2. Nearest Neighbor Intrpolation.

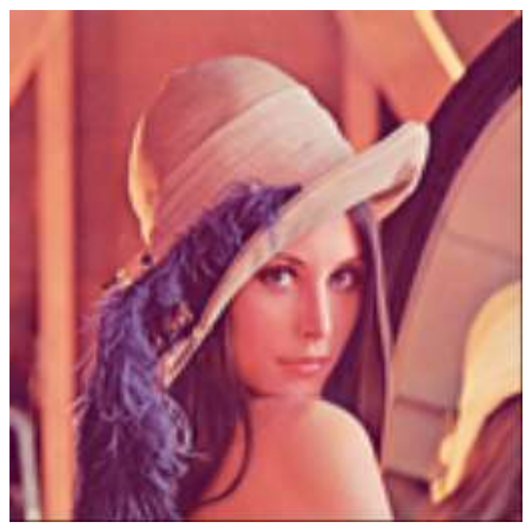

Figure 3. Bilinear Interpolation.

Figure 4. Bicubic Interpolation.

Figure 5. Catmull–Rom Interpolation.

Figure 6. Cibic B-Spline Interpolation.

Figure 7. Lanczos Order 2 Interpolation.

Figure 8. Lanczos Order 3 Interpolation.

Table 1. Computational Time in seconds using Intel® Core i3 CPU @ 2.40 GHz and DDR3 RAM of 2 GB

Algorithms	***2X***	***4X***	***8X***	***10X***	***20X***	***50X***
Nearest Neighbour	0.015	0.028	0.062	0.094	0.312	1.469
Bilinear	0.031	0.047	0.125	0.218	0.797	3.329
Bicubic	0.047	0.172	0.64	0.969	3.922	22.735
Catmull-Rom	0.063	0.218	0.906	1.422	5.516	33.829
Cubic B-Spline	0.068	0.235	0.922	1.438	5.766	34.658
Lanczos 2	0.359	1.813	8.125	13.188	51.112	324.031
Lanczos 3	0.891	4.109	18.048	28.861	112.114	706.799

Table 2. Computational Time in seconds using Intel® Core i5 CPU @ 2.50 GHz and DDR3 RAM of 4 GB

Algorithms	2X	4X	8X	10X	20X	50X
Nearest Neighbour	0.011	0.023	0.045	0.068	0.218	1.282
Bilinear	0.023	0.036	0.094	0.156	0.531	2.814
Bicubic	0.039	0.156	0.442	0.687	2.712	19.023
Catmull-Rom	0.049	0.234	0.536	0.826	3.323	19.612
Cubic B-Spline	0.051	0.256	0.547	0.849	3.368	20.819
Lanczos 2	0.314	1.134	5.853	9.432	42.817	245.275
Lanczos 3	0.608	2.902	12.492	19.926	81.373	515.078

Table 3. Computational Time in seconds using Intel® Core i7 CPU @ 2.70 GHz and DDR3 RAM of 8 GB

Algorithms	2X	4X	8X	10X	20X	50X
Nearest Neighbour	0.008	0.013	0.031	0.046	0.187	0.875
Bilinear	0.015	0.019	0.077	0.093	0.375	2.391
Bicubic	0.028	0.078	0.359	0.562	2.203	13.853
Catmull-Rom	0.033	0.109	0.427	0.667	2.766	17.406
Cubic B-Spline	0.039	0.152	0.462	0.688	2.797	17.456
Lanczos 2	0.203	1.063	4.86	7.886	32.835	203.204
Lanczos 3	0.484	2.375	10.44	16.637	69.514	439.577

Comparison of time complexity of all seven image scaling algorithms was performed for Lena image as shown in Table 1, 2 and 3 using Intel® Core i3, Intel® Core i5, and Intel® Core i7 respectively.

Data represented in Tables 1, 2 and 3 are shown using the line chart for the corresponding interpolated output images of size 256X256 (2X), 512X512 (4X) and 1024X1024 (8X) using all seven algorithms in Fig 9, 10 and 11 respectively. Data presented in above three tables show that if output image size increased the algorithm requires more computational time. Even if we compare the results. The trends clearly suggest that if more pixels are to be computed, the corresponding algorithm require more computational time.

From the above mentioned three tables and figures it is very clear that higher the processing speed and available RAM, the computational time decrease. The computational complexity of Nearest Neighbour interpolation is least while Lanczos order three demands the most processing time. As data shows in above Table 3, higher-end hardware configuration of Intel® Core i7 CPU @ 2.70 GHz and DDR3 RAM of 8GB, Nearest Neighbour uses 0.008 while Lanczos order three uses

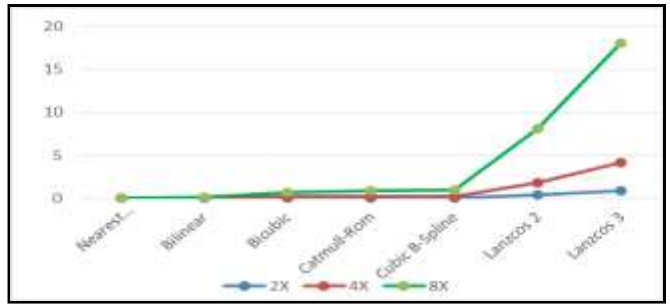

Figure 9. Line Chart for Computational Time in seconds using Intel® Core i3 CPU for interpolated output images of size 2X, 4X and 8X.

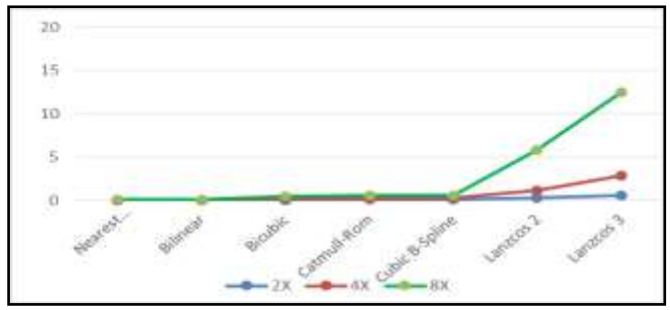

Figure 10. Line Chart for Computational Time in seconds using Intel® Core i5 CPU for interpolated output images of size 2X, 4X and 8X.

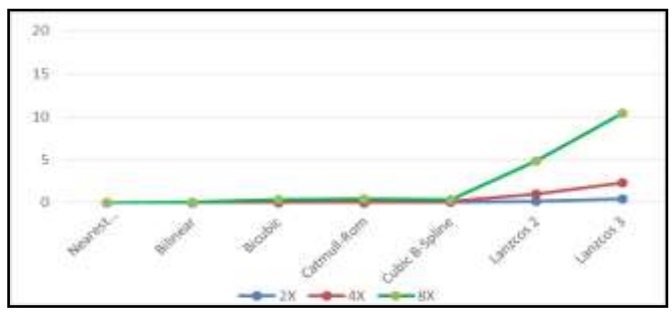

Figure 11. Line Chart for Computational Time in seconds using Intel® Core i7 CPU for interpolated output images of size 2X, 4X and 8X.

0.484 second to generate 256X256 (2X) image. Increase in time for processing is due to the nature of the algorithms implemented, Nearest Neighbour just copies the value of nearest one pixel while Bilinear interpolation uses four pixels to compute the value of a new pixel. Bicubic, Catmull-Rom and cubic B-Spline uses 16 pixels to interpolate an image with different functional parameters, hence the time is almost similar for these three algorithms to generate a 256X256 (2X) output image. Bicubic requires 0.028, Catmull-Rom requires 0.033 and Cubic B-Spline requires 0.039 second. The difference in time computation is due to the different type of mathematical calculation required for computing new pixel. Lanczos order two and order three requires 16 and 32 pixels, and the computational time of 0.203 and 0.484 second respectively. Higher processing time for Lanczos order two and order three is due to the complex nature of algorithms to interpolate new pixel.

IV. CONCLUSION

Image interpolation is the basic requirement for image graphic software. In this paper, we used seven different interpolation algorithms using ImageJ open source platform with three different hardware configuration to measure computational complexity. As the source image size 128X128 (1X) increase to 2X, 4X, 8X, 10X, 20X and 50X, a higher number of pixels are required for calculating. This results into additional time required to generate the corresponding resized image.

Estimating computational complexity of different image interpolation algorithms is difficult because it depends upon several factors: image size, response time, available RAM, the processing speed of a computer and the complexity of implemented algorithms. Computationally efficient algorithms such as Nearest Neighbour and Bilinear do not provide visually pleasing images. Combination of computational complex and image quality makes the interpolation algorithms viable for different applications in image processing. As shown in Table 1, 2 and 3 and Fig.2 to Fig. 8 the Bicubic, Catmull-Rom and cubic B-Spline uses moderate computation complexity and generates visually better quality images than Nearest Neighbour and Bilinear algorithms. Lanczos order two and three generates visually good images but the computational complexity increase and hence these algorithms are not suitable for real-life applications.

REFERENCES

[1] P. Bhatt, S. Patel, A. Shah, S. Patel, "*Image Enhancement Using Various Interpolation Methods*" IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), Vol. 2, No.4, pp.799-803, 2012.

[2] D. Doma, *"Comparison of Different Image Interpolation Algorithms*", West Virginia University Libraries, 2008.

[3] J. Titus, S. Geroge, "*A Comparison Study On Different Interpolation Methods Based On Satellite Images"*, International Journal of Engineering Research & Technology, Vol. 2, Issue 6, pp. 82-85 2013.

[4] P. S. Parsania, P. V. Virparia, "*Image Quality Comparison using PSNR and UIQI for Image Interpolation Algorithms*", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 12, pp. 21679-21687, 2016.

[5] K. Shah, J. Pandya, S. Vahora, *"A Survey On Super Resolution Image Reconstruction Techniques"*, International Journal of Engineering Research & Technology, Vol. 2, No. 4, pp. 1897-1901, 2013.

[6] G. Kaur, J. Kaur, *"A Comparative Study of Image Demosaicing"*, International Journal of Computer Sciences and Engineering, Vol. 3, Issue. 7, pp. 98-102, 2015.

[7] 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, 2015.

[8] A. Sinha, M. kumar, A. Jaiswal, R. Saxena, *"Performance Analysis of High Resolution Images Using Interpolation Techniques in Multimedia Communication System"*, Signal & Image Processing: An International Journal, Vol. 5, No. 2, pp. 39-49, 2014.

[9] A. Prajapati, S. Naik, S. Mehta, *"Evaluation of Different Image Interpolation Algorithms"*, International Journal of Computer Applications, Vol. 58, Issue. 1, pp. 6-12, 2012.

[10] T. Wu, B. Bai, P. Wang, *"Parallel Catmull-Rom Spline Interpolation Algorithm for Image Zooming Based on CUDA"*, International Journal of Applied Mathematics & Information Science, Vol. 7, No. 2, pp. 533-537, 2013.

[11] W. Burger, M. J. Burge, *"Digital image processing: an algorithmic introduction using Java"*, 1st ed., Springer, India, pp. 400-401, 2009.

[12] D. Han, "*Comparison of Commonly Used Image Interpolation Methods*", In the Proceedings of 2nd International Conference on Computer Science and Electronics Engineering, France, pp.1556-1559, 2013.

[13] S. Singh, T. Gulati, *"Upscaling Capsule Endoscopic Low Resolution Images"*, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue. 5, pp. 40-46, 2014.

[14] C. Suresh, S. Singh, R. Saini, A. K. Saini, "*A Comparative Analysis of Image Scaling Algorithms"*, International Journal of Image, Graphics and Signal Processing, Vol. 5, Issue. 4, pp - 55-62, 2013.

[15] P. S. Parsania, P. V. Virparia, "*Performance Analysis of Image Scaling Algorithms*", International Journal on Recent and Innovation Trends in Computing and Communication, vol. 4, Issue 6, pp. 521-526, 2016.

Authors Profile

Mr. P. S. Parsania pursed MScIT in the year 2004 and in pursuit for Ph.D. from Department of Computer Science, Sardar Patel University, Vallabh Vidyanagar, Gujarat, India. Currently, he is working as an Assistant Professor at College of FPTBE, AAU, Anand, Gujarat, India. His area of interests are Digital Image Processing, Data Structure, Java, and Database Management System.

Dr. P. V. Virparia joined the Department of Computer Science of Sardar Patel University, Vallabh Vidyanagar in 1989 and currently working as a Professor. He completed his MCA in 1989 from Sardar Patel University and Ph. D. in 2002 from Sardar Patel University. TEN research scholars have completed their Ph.D. (Computer Science) under his guidance. Currently, FOUR students are doing their Ph. D. under the guidance of him. Also, three students have completed their M.Phil. (Comp. Sc.) under his supervision. His publications include one book, two chapters in book and 40 papers in International Journal, 16 papers in National Journals and 47 papers in national conferences/seminars. His research interests include the areas of Computer Simulation & Modeling, Data Mining, Networking and IT enabled services. He is an editor and editorial review board member in several journals/magazines.

