Fast Computation of Image Scaling Algorithms Using Frequency Domain Approach

Prasantha H.S., Shashidhara H.L., and K.N.B. Murthy

PES Institute of Technology, 100 Ft Ring road, BSK 3[RD] stage, Bangalore-85
Prashanth_34@rediffmail.com,
shashihl@yahoo.com,
principal@pes.edu

Abstract. Image scaling algorithms play important role in many image scaling applications. Image scaling is the process of enlarging an image or reducing the size of an image to make it suitable to display on the given display device. The paper mainly focuses image zooming to fit the given image on a display device to view the details on a bigger display device. When the image is zoomed, artifacts like blurring, jagging and ghosting may arise. The main objective of the paper is to investigate and study the known algorithms for image scaling based on different comparative parameters in frequency domain. The different interpolation techniques such as nearest neighbor and bilinear are studied and compared in both spatial and frequency domain. The paper proposes a novel scheme for fast computation of the different image interpolation techniques.

1 Introduction

An image may be defined as a two-dimensional function f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. The field of digital image processing refers to processing digital images by means of a digital computer. Digital image processing involves the use of computer algorithms to perform image processing on digital images. The usefulness is apparent in many different disciplines covering medicine through remote sensing. The application of digital image processing includes medical applications, restorations and enhancements, digital cinema, image scaling, image compression, image transmission and coding, color processing, remote sensing, robot vision, facsimile, etc.

A fast algorithm for image interpolation is proposed for real-time enlargement of video images is discussed [8]. A novel method of nonlinear scaling is proposed to overcome the compatibility problem [7]. An adaptive resampling algorithm for zooming up images is based on analyzing the local structure of the image and applying a near optimal and least time-consuming resampling function that will preserve edge locations and their contrast. A method is proposed to take into account information about discontinuities or sharp luminance variations while doubling the input picture [5]. Popular methods such as bilinear Interpolation, cubic convolution and cubic convolution are investigated [4]. The different measures can be used to compare the different algorithms of interpolation [11] [6] [5] [10].

Aswatha Kumar M. et al. (Eds.): Proceedings of ICAdC, AISC 174, pp. 201–208.
springerlink.com

Section 1 discusses the introduction, Section 2 briefs about image scaling, Section 3 explains the different scaling techniques considered, Section 4 discusses the implementation details and Section 5 discusses the test results. Conclusions are discussed in the Section 6.

2 Image Scaling

In computer graphics, image scaling is the process of resizing a digital image. In the case of scaling up (image enlargement), the methods used involve first resizing the image by padding with zeros, followed by convolution with a two-dimensional, spatially invariant linear filter. Since we are dealing with matrices that represent discrete pixel values, we use the filter in the discrete domain. Hence each filter can be represented by the impulse function h, which is also called the mask of the filter. Scaling down (image reduction) increases the incidence of high frequencies and causes several pixels to collapse into one. Hence we need to apply a smoothing filter in order to minimize aliasing problems in the target image. Apart from fitting a smaller display area, image size is most commonly decreased (or sub-sampled or down-sampled) in order to produce thumbnails. There are several methods of increasing the number of pixels that an image contains, which evens out the appearance of the original pixels.

Non-adaptive algorithms include: nearest neighbor, bilinear, spline, sinc, lanczos and others. The more adjacent pixels they include, the more accurate they can become, but this comes at the expense of much longer processing time. Interpolation is the problem of approximating the value for a non-given point in some space, when given some values of points around that point. Nearest neighbor interpolation is the simplest method and basically selects the value of the nearest point, and assigns that value to the output point and does not consider the values of other neighboring points at all, yielding a piecewise-constant interpolate. Bilinear interpolation determines the grey level value from the weighted average of the four closest pixels to the specified input coordinates, and assigns that value to the output coordinates.

3 Image Scaling in the Frequency Domain

Image transforms are extensively used in image processing and image analysis. Transform is a basically mathematical tool, which allows us to move from one domain to another domain (time domain to frequency domain). Image transforms are useful for fast computation of convolution and correlation. The transforms do not change the information content present in the signal. Transforms play a significant role in various image processing applications such as image analysis, image enhancement, and image filtering and image compression as well. The frequency domain representation clusters the image data according to their frequency distribution. In frequency domain filtering, the image data is dissected into various spectral bands, where each band depicts a specific range of details within the image. The process of selective frequency inclusion or exclusion is termed “Frequency domain filtering”.

3.1 Fast Computation of Nearest Neighbor Interpolation

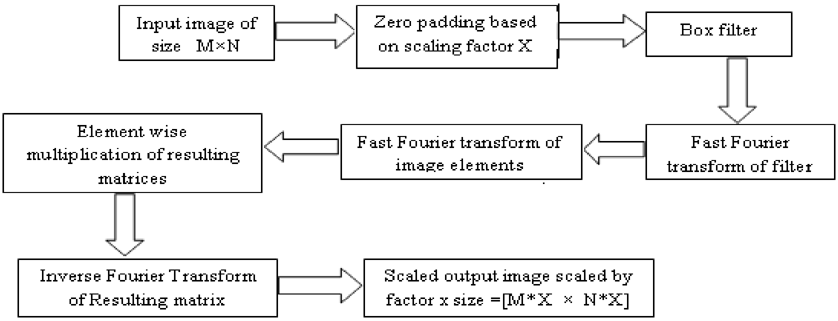

Fig. 1. Nearest Neighbor interpolation using frequency domain approach

In nearest neighbor method, replace each row by a number of identical rows, and likewise for columns. This is known as scaling by pixel replication. In formal terms, this is achieved by applying a box interpolation filter. Hence the rows of zeros have been replaced with a copy of the pixel values directly above them, and the columns of zeros took on the values of the pixels to their immediate left. In general, to scale an image x times in the horizontal direction and y times in the vertical direction, we need a mask of size y rows and x columns with all the elements taking a value of 1. If the original image had n rows and m columns, and we want to scale it up by a factor of x in the horizontal direction and y in the vertical direction, then the zero-padded image will have x*n+1 rows and y*n+1 columns.

3.2 Fast Computation of Bilinear Interpolation

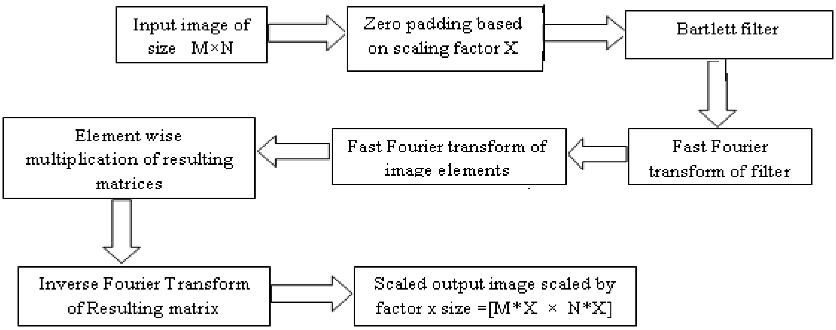

Fig. 2. Bilinear interpolation using frequency domain approach

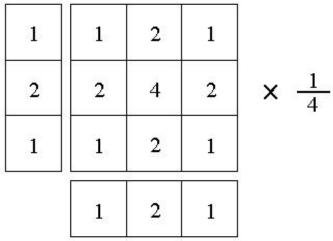

Fig. 3. Filter co-efficient

The figure 3 shows a matrix of a Bartlett filter used to scale an image up by a factor of 2 in both the horizontal and vertical directions. For example, in the case of scaling up 2 in both the horizontal and vertical directions, since each pixel is mapped into 4 pixels in the resulting image, we want the intensity to be 4 times greater than that of the original image in order to maintain the same average intensity level.

4 Implementation Details

The experiments are conducted at PESIT Multimedia Lab for different sets of input images of various file formats. The different interpolation algorithms considered are nearest neighbor, bilinear and spline interpolation for the comparison using different parameters such as MSE, PSNR, and quality index. The digital signal processor multimedia developer kit DM 642 operating at 600 MHZ is considered for the experimentation. All the source files are written in C programming language. The DM642 DSP is capable of executing eight instructions in parallel at the clock speed of 600 MHz If we are able to utilize these execution units then we can benefit from its processing power and obtain a high performance solution. Otherwise, the performance may not be satisfactory. The un-optimized codes for different interpolation techniques are considered for validation.

5 Test Results and Discussions

Experiments are conducted for different set of images of different resolution and file formats. A sample of the result is displayed for the further discussion. Different comparison parameters such as MSE, PSNR, Quality index (QI) and computation time are considered.

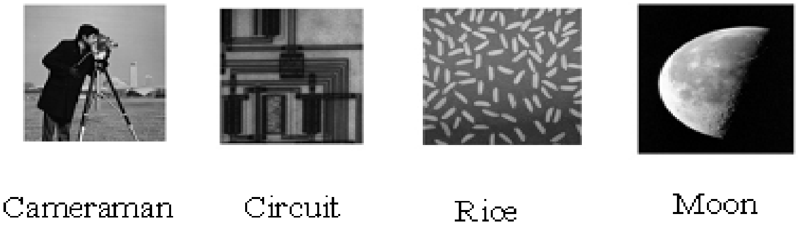

Fig. 4. Input images considered for the discussion

Fig. (a). Input image

Fig. (b). Box filter

Fig. (c). Bartlett filer

Fig. 5. Cameraman image (Scaled by factor 2 in both horizontal & vertical direction)

Fig.(a). Input image

Fig. (b). box filter

Fig. (c). Bartlett filer

Fig. 6. Cameraman image (Scaled by factor 4 in both horizontal and vertical direction)

The figure 4 indicates some of input images considered for the experimentation. The figure 5 and 6 indicates the original image and the subjective quality of the nearest neighbor and bilinear interpolation. The tables 1, 2, 3 and 4 shows the comparison of different parameters such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Quality index (QI) for different scaling factors 2, 4, and 8 for the cameraman, circuit, rice and moon images. The table 5 shows the result for spatial domain computation time and frequency domain computation time. Computation time is the total time involved to obtain the zoomed image.

Table 1. Table showing different comparison parameters for the Cameraman image

Scale factor	Nearest neighbor (Box filter)			Bilinear (Bartlett filter)		
	MSE	PSNR	QI	MSE	PSNR	QI
2	376.8179	22.3695	0.9515	261.2432	23.9604	0.9651
4	729.5694	19.5001	0.9066	497.8059	21.1602	0.9325
8	1193.9	17.3612	0.8454	807.7252	19.0582	0.8865

Table 2. Table showing different comparison parameters for the Circuit image

Scale factor	Nearest neighbor (Box filter)			Bilinear (Bartlett filter)		
	MSE	PSNR	QI	MSE	PSNR	QI
2	206.2694	24.9865	0.9830	133.5975	26.8728	0.9893
4	428.847	21.8078	0.9574	255.7305	24.053	0.9721
8	927.0181	18.4599	0.8910	590.7553	20.4167	0.9255

Table 3. Table showing different comparison parameters for the Rice image

Scale factor	Nearest neighbor (Box filter)			Bilinear (Bartlett filter)		
	MSE	PSNR	QI	MSE	PSNR	QI
2	372.82	22.42	0.943	201.24	25.093	0.961
4	734.564	19.47	0.88	407.805	22.026	0.921
8	98	18.21	0.7463	653.725	19.976	0.801

Table 4. Table showing different comparison parameters for the Moon image

Scale factor	Nearest neighbor (Box filter)			Bilinear (Bartlett filter)		
	MSE	PSNR	QI	MSE	PSNR	QI
2	86.2853	28.7714	0.9973	53.276	30.865	0.9985
4	215.948	24.7873	0.9924	136.1032	26.792	0.9965
8	563.3191	20.6233	0.9817	356.579	22.609	0.9903

Table 5. Comparison of computation time using spatial and frequency domain methods

Scale factor	Spatial domain (in milliseconds)		Frequency domain (in milliseconds)	
	Nearest neighbor	Bilinear	Nearest neighbor	Bilinear
2	19	42	14	40
4	34	80	28	71
8	55	125	49	112

From the experimental results, it is clear that bilinear interpolation algorithmic yields better results than nearest neighbor interpolation. Bilinear interpolation has higher value of PSNR and Quality index for scaling factors 2, 4 and 8 which is shown in the table 1, 2, 3 and 4 for all the input images considered. For any scaling factors, bilinear interpolation has higher value of PSNR and quality index. Also the subjective quality of the image for bilinear interpolation is better than nearest neighbor interpolation which is shown in figure 5 and 6. For the moon image shown in the table 4, PSNR and Quality index for any scaling factor is high which indicates that if the variation in the intensity levels for the given image is less, then it is possible to scale-up or zoom the image without sacrificing the quality. Similarly for cameraman image shown in the table 1, since the variations are more, if we scale the image, quality also suffers. The computation time required using frequency domain methods are comparatively less compared to spatial domain methods which is shown in table 5.

6 Conclusions

Bilinear interpolation can be used to zoom the image since it provides better results. But the main problem with bilinear interpolation is the computation time. Hence it is required to reduce the computation time which can be done by using frequency domain methods. Since filter based techniques are designed using frequency domain approach hence there is comparable reduction in computation time and complexity compared to spatial domain approach. Based on results, it is clear that even in frequency domain box filter based algorithm requires less time than Bartlett filter based algorithm but Bartlett filter usage yielding better quality image than box filter use. Also computation time can be further reduced by using fast computing transforms in place of FFT and IFFT.

References

1. Prasantha, H.S., Shashidhara, H.L., Balasubramanya Murthy, K.N.: Comparative analysis of different interpolation schemes in image processing. In: International Conference on Advanced Communication Systems (ICACS), India, pp. 17–24 (January 2007)
2. Prasantha, H.S., Shashidhara, H.L., Balasubramanya Murthy, K.N.: Image Scaling comparison using Universal Image Quality Index. In: International Conference on Advances in Computing, Control and Telecommunication Technologies (ICACCTT), India, pp. 859–863 (December 2009)

3. Gao, R., et al.: Image zooming algorithm based on partial differential equations technique. International Journal of Numerical Analysis and Modeling 6(2), 284–292
4. Matsuoka, R., et al.: Comparison of Image Interpolation Methods Applied to Least Squares Matching. In: CIMCA (2008)
5. Battiato, S., et al.: A locally adaptive zooming algorithm for digital images. Image and Vision Computing, 805–812 (2002)
6. Yuan, S., et al.: High accuracy WADI image interpolation with local gradient features. In: Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication System, Hongkong (2005)
7. Shi, Z., et al.: A novel nonlinear scaling method for video images. In: International Conference on Computer Science & Software Engineering, pp. 357–360.
8. Mihov, S.G., et al.: Interpolation algorithms for image scaling. Electronics (2005)
9. Lehmann, T.M., et al.: Survey: Interpolation Methods in Medical Image Processing. IEEE Transactions on Medical Imaging 18(11) (1999)
10. Acharya, T., et al.: Computational Foundations of image interpolation algorithms. ACM Ubiquity 8 (2007)
11. Wang, Z., et al.: A universal image quality Index. IEEE Signal Processing Letters XX(Y) (2002)