Image Filtering and Scaling Algorithms

Slavy G. Mihov[1] and Georgy S. Zapryanov[2]

Abstract **– Image filtering and resizing are essential for the purpose of correct image visualization. In this paper, parallel between different algorithms is made, and as basis for this comparison are used three methods: Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Subjective Evaluation (SE). Here are given the results from the execution of the above algorithms.**

Keywords **– filtering, scaling, resizing, MSE, PSNR.**

I. INTRODUCTION

Image resizing or scaling (magnification or reduction) is one of the main operations in image processing. It is done whenever resolution change has to be done. For example, for viewing images and video are used devices with fixed resolution, such as plasma panels and LCD displays. Digital resizing is also applied for scaling of video data of devices based on Digital Micromirror Device™ (DMD™) technology, developed by Texas Instruments. DMD is in the core of Digital Light Processing (DLP) which is used in a large scale in the contemporary fully digital viewing systems. In the present moment such systems are produced in various forms: video projector, high definition displays, home theatre, standard TV etc. More and more popular become High Definition TV (HDTV) that also has fixed resolution, higher than that of DVD and films broadcasted at the TV channels. Also, when printing photographs taken by digital cameras, an adapting of the input resolution to that of the printer should be done. The quality of the image of the devices mentioned above tends to depend more and more on the resizing techniques used.

Image filtering is another main operation in image processing. It is used whenever is a necessity for removing artifacts from the image or adjusting some details in it. Image processing and especially image scaling comes at no cost. By using an algorithm for such a purpose, usually this introduces various artifacts in the image, some of which desirable and some – not. When in need of eliminating completely or lowering as much as possible these negative effects, the image is been processed by some filtering algorithms. Various types of them are widely used in practice. This paper concentrates primarily on the family of convolution filters.

The aim of the current paper is to create a methodology for comparison of some of the most common algorithms for image filtering and interpolation. The interpolation methods are compared via executing those algorithms upon three types of images: synthetic (vector) and two types of real photographs. For noting the changes introduced in the processed image by each of the algorithms, we use two error metrics: the MSE and the PSNR in the RGB color space.

II. EVALUATION METHODS

A. General Issues

In the present paper is made a comparison between the following scaling algorithms: Box Algorithm (Nearest Neighbor - NN) [3, 5]; Simple (Pixel) Interpolation (SI), Bilinear Interpolation (BI) [3, 5]; Bicubic-Spline Interpolation (BSI) (a = 0, -1/2, -1) [1, 3]; Directional Interpolation (DI) [5]; Data Dependant Triangulation (DDT) [2]; "Marbella" [4] and Functional Interpolation (FI) [5] – sin and exp.

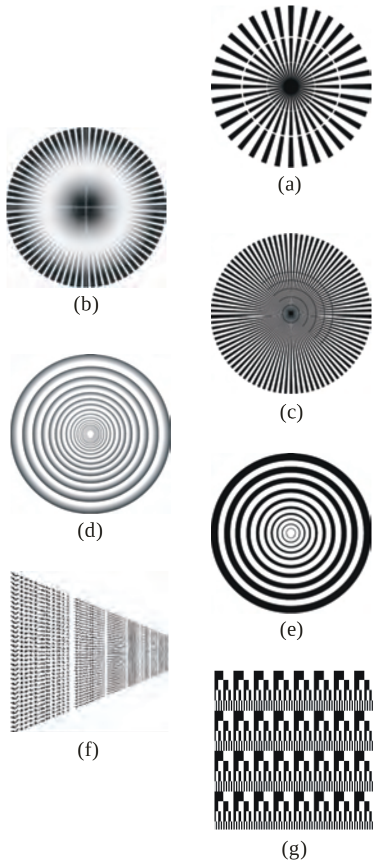


Fig. 1. Synthetic vector images ("miras")

For the purpose of comparing the scaling algorithms were created 7 synthetic vector images (miras) (Fig. 1). These

[1]Slavy G. Mihov is with the Technical University - Sofia, Bul. "Kl. Ohridsky" No. 8, 1100 Sofia, Bulgaria, E-mail: smihov@abv.bg
[2]Georgy S. Zapryanov is with the Technical University - Sofia, Bul. "Kl. Ohridsky" No. 8, 1100 Sofia, Bulgaria,
E-mail: gszap@tu-sofia.bg

vector images are with four different resolutions: 256x256, 512x512, 800x800 and 1024x1024 pixels. For their creation is used the principle described in [3].

Images (a), (c), (e) and (g) are used for estimating the quality when transforming sharp edges; Images (a), (b) and (c) – for estimating the systematic sub pixel shift in different parts of the image; Images (b) and (d) – for estimating the transformation of low contrast details; Image (f) – for estimating the conversion of tiny details; The last image is the label "VirtLab1311", repeated several times and displayed in perspective.

Except these 7 vector images, 14 photographs are tested too: 7 images from the Kodak test suite with resolution 768x512 pixels and 7 similar high quality photographs.

B. Image Scaling

The corresponding vector images are scaled up and down to suitable resolutions. The quality of the applied algorithm is estimated by comparing the result images with the original ones or with images got by third party algorithms (in the current case, for these purposes is used Photoshop 7.0 Bicubic algorithm – PB). Estimating the quality of the scaling process of the photographs is accomplished by following a strict succession of actions. These three steps are: scaling the image up or down using some of the tested algorithms, recovering the original resolution using the same methods and measuring MSE and PSNR between the original and the final image. There is no limitation for the value of the scaling coefficient, such as being integer. It can be even a floating point number. Furthermore, scaling down can be done with one algorithm, and scaling up – with another (for estimating the quality of combinations of algorithms).

When processing the selected image, for obtaining the final resolution is formed a so called "Frame", comprising of four images: the original one, the intermediate one – obtained at the first step of the processing (reduction or magnification); the final one – obtained at the second step of the processing (for one-step image processing the intermediate and the final images are identical) and another one – represents the visual (RGB) difference between the original and the final images. In the software is realized the possibility for creating more than one frames, when processing an image (vector or raster), in order to study the variation of the algorithm's quality, for different scaling coefficients.

C. Parallel Processing

The modern tendency in computer technique development is increasing the computing power by usage of more than one Central Processing Units (CPU) working cooperatively. Multiprocessor systems become more and more popular. Dual core processors and the preceding Hyper Threading Technology follow the same tendency, performing several physical or logical processors in one chip. The main principle of work is dispersing the computing load between the processors and thus each one processes a part of the common job, increasing the total performance.

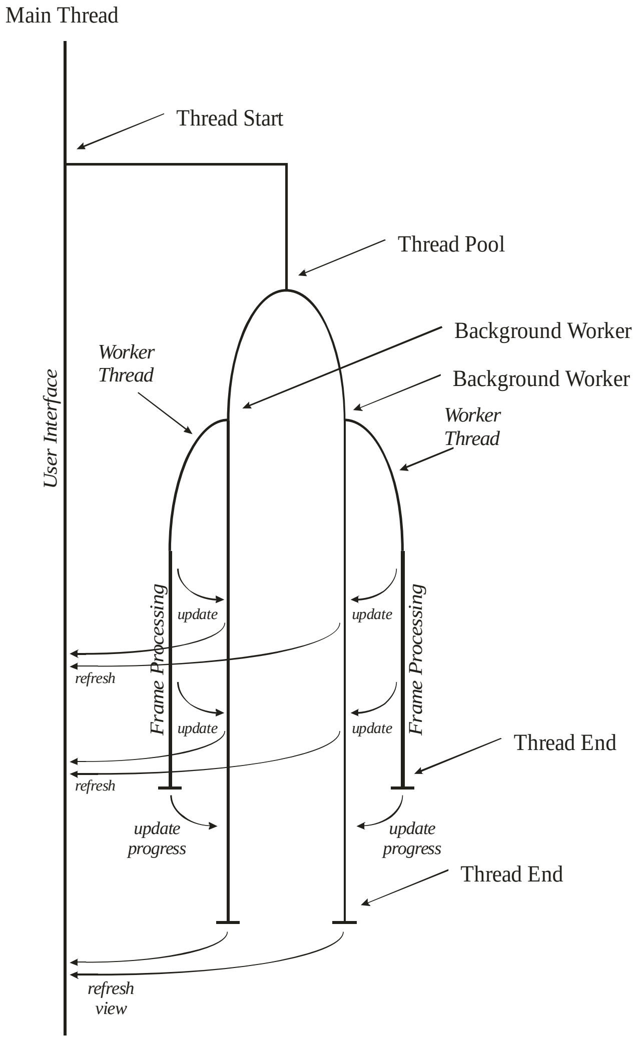


Fig. 2. Parallelism in image processing

For a program or just a mere computation procedure, to use the benefit of a multiprocessor system, is required to have a suitable structure. It should possess some sort of parallelism. Its algorithm should be organized in several autonomous linear sections. At runtime, these sections are meant for processing by different CPUs simultaneously and by this mean the whole computation load is dispersed among the available processing units. A proper dispersion of the computation load aims to balance the usage of all the CPUs equally and thus gives them equal portions of the entire task to complete.

In the case of image processing, image scaling and filtering tend to be the heaviest and most time consuming operations to be taken. The program flow is organized in such a way that effectively uses the capabilities of a computer system with two processors (this case includes systems with a single dual-core processor or a processor using Hyper Threading Technology). The independent work of both of the CPUs is organized in several threads running simultaneously. Fig. 2 illustrates the sequential flow of all the threads involved, the moments of their creation and synchronization.

As seen from Fig. 2, here are used three techniques for splitting the sequential program flow and creating new threads. The first one (Thread Start) is the simplest of all and is known being thread-unsafe. It is used just to create a new thread and call it for execution. The newly started thread performs all the computation in image processing, while the main thread (the one it was created by) serves the user interface.

As mentioned above, the computation procedure is designed for processing by two CPUs (either physical or logical). The second technique for manipulating multiple threads (Thread Pool) implements this idea. The flow of the current thread is split into two identical threads, which differ only by their IDs (identification numbers). The operating system is responsible for the assignment of each of them being executed by one of the two processors. This is the stage where the computation load is dispersed and balanced among the CPUs. This technique tends to be thread-safe; in general case can involve more than two threads and thus is widely used in practice.

The next thing that follows is one and the same for all the working processors (the threads they execute are identical). The third technique used for manipulating threads is called Background Worker. It forms the current thread into two parallel ones. The first of them is the so called Worker Thread, that performs the heavy computation of image processing and the second one is its master thread responsible its control. Background Worker embeds numerous features for synchronization and communication between the threads it comprises. It is thread-safe and is extremely suitable for performing heavy computations in background regime.

Fig. 2 also shows the most important communications between the threads, by which they synchronize their work and exchange results. The parallel threads exist only until they complete their part of the common job, after which they terminate. Only the main thread remains and continues program's sequential flow.

D. Image Filtering

In the present project are used the following filtering algorithms: Mean Filter, Median Filter, Gaussian Smoothing, Conservative Smoothing, Laplacian Filter, Variance Filter, Laplacian of Gaussian (LoG Filter), Sharpening Filter and Custom (user-defined convolution) Filter. They are used to eliminate or partially decrease the presence of noise and various artifacts in images. Their usage can be combined with that of some scaling algorithms, as means to recover an image from the introduced artifacts by the preceding manipulation (usually a preceding resizing that had distorted the image in some way). Most of the filters mentioned above belong to the family of convolution filters. The mere filtering is done by convolving (performing the operation convolution) the image with a so called kernel. The kernel is a square two-dimensional integer matrix with resolution of 3x3, 5x5, 7x7 or 9x9. The size of the kernel should be an odd number. For practical purposes kernels larger than 9x9 aren't used. Even kernels with dimensions 9x9 are not very common.

The image and the kernel used in convolution are shown schematically on Fig. 3. The kernel shown is a non-square matrix with dimensions 2x3, which does not conflict with math theory in general. The image is represented as a grayscale image by an integer matrix 6x9. The values I_{ij} of the matrix (values form 0 to 255) resemble the intensities of the individual pixels the image consists of. In case of a color image, (a 24-bit RGB image, for example) it is represented by several matrixes of the same kind, one for each channel.

$\mathbf{I_{11}}$	$\mathbf{I_{12}}$	$\mathbf{I_{13}}$	$\mathbf{I_{14}}$	$\mathbf{I_{15}}$	$\mathbf{I_{16}}$	$\mathbf{I_{17}}$	$\mathbf{I_{18}}$	$\mathbf{I_{19}}$
$\mathbf{I_{21}}$	$\mathbf{I_{22}}$	$\mathbf{I_{23}}$	$\mathbf{I_{24}}$	$\mathbf{I_{25}}$	$\mathbf{I_{26}}$	$\mathbf{I_{27}}$	$\mathbf{I_{28}}$	$\mathbf{I_{29}}$
$\mathbf{I_{31}}$	$\mathbf{I_{32}}$	$\mathbf{I_{33}}$	$\mathbf{I_{34}}$	$\mathbf{I_{35}}$	$\mathbf{I_{36}}$	$\mathbf{I_{37}}$	$\mathbf{I_{38}}$	$\mathbf{I_{39}}$
$\mathbf{I_{41}}$	$\mathbf{I_{42}}$	$\mathbf{I_{43}}$	$\mathbf{I_{44}}$	$\mathbf{I_{45}}$	$\mathbf{I_{46}}$	$\mathbf{I_{47}}$	$\mathbf{I_{48}}$	$\mathbf{I_{49}}$
$\mathbf{I_{51}}$	$\mathbf{I_{52}}$	$\mathbf{I_{53}}$	$\mathbf{I_{54}}$	$\mathbf{I_{55}}$	$\mathbf{I_{56}}$	$\mathbf{I_{57}}$	$\mathbf{I_{58}}$	$\mathbf{I_{59}}$
$\mathbf{I_{61}}$	$\mathbf{I_{62}}$	$\mathbf{I_{63}}$	$\mathbf{I_{64}}$	$\mathbf{I_{65}}$	$\mathbf{I_{66}}$	$\mathbf{I_{67}}$	$\mathbf{I_{68}}$	$\mathbf{I_{69}}$

$\mathbf{K_{11}}$	$\mathbf{K_{12}}$	$\mathbf{K_{13}}$
$\mathbf{K_{21}}$	$\mathbf{K_{22}}$	$\mathbf{K_{23}}$

Fig. 3. Image and kernel used in convolution

The values K_{ij} of the second matrix are the coefficients of the kernel. Eq. 1 shows the mere mathematical expression of operation convolution between the image and the kernel.

$$O(i,j)=\sum_{k=1}^{m}\sum_{l=1}^{n} I(i+k-1,\, j+l-1)K(k,l) \tag{1}$$

Here: I(i, j) stands for the intensity of pixel (I_{ij}) with coordinates (i, j) from the image; K(i, j) stands for the coefficient of the kernel K_{ij}; O(i, j) stands for the final value of pixel (i, j), after completing the convolution.

E. Formal Metrics

For objective estimate of an algorithm's quality is used the main approach in digital image processing, based on computing the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). MSE (Eq. 2) represents the average square error (difference) in the intensities [$I(i;j)$ *and* $I'(i;j)$] of a given color (channel) of the corresponding pixels from both of the images and it is dimensionless quantity. Because of its simplicity, it is the most common method for objective measurement of image quality, based on a reference image.

$$MSE=\frac{1}{m.n}\sum_{i=1,j=1}^{m,n}\left(I(i;j)-I'(i;j)\right)^2 \tag{2}$$

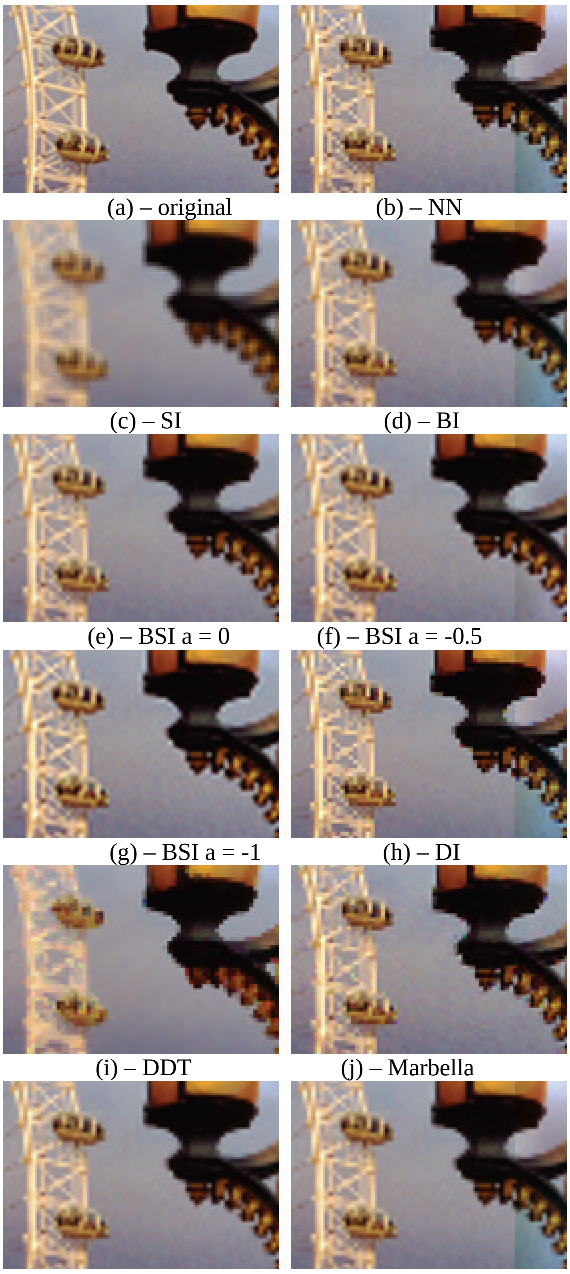
PSNR (Eq. 3) represents the square of the ratio between the maximum of the signal (the maximal possible value of color's intensity (I_{max})) and the mean square error (difference) in the intensities [$I(i;j)$ *and* $I'(i;j)$] of a given color (channel) of the corresponding pixels from both of the images and is measured in decibels (*dB*).

$$PSNR=10\lg\frac{I_{\max}^2.m.n}{\sum_{i=1,j=1}^{m,n}\left(I(i;j)-I'(i;j)\right)^2}\quad[dB] \tag{3}$$

The estimate can be done by computing MSE and PSNR for every single color (Red, Green, Blue), as well as totally for the three of them.

III. RESULTS

For estimating the quality of each scaling algorithm, several experiments with different test images are competed. It is compared the results from execution of different algorithms upon one and the same image, as well as execution of one and the same algorithm upon different images. Combinations of algorithms executed upon one and the same image are tested too. The results got are compared in between and with pre-created images. Some of the images were resized, using Photoshop 7.0 Bicubic algorithm, and then compared with those got from the execution of this program.



(a) – original (b) – NN (c) – SI (d) – BI (e) – BSI a = 0 (f) – BSI a = -0.5 (g) – BSI a = -1 (h) – DI (i) – DDT (j) – Marbella (k) – FI (sin) (l) – FI (exp)

Fig. 4. ROI – algorithms' effects

The main artifacts in the images, that can be noted and determined as results of the scaling process, are:

1. Ringing – appearance of a single wave near a sharp edge in the image.
2. Overshooting – appearance of several such waves.
3. Sub-pixel shift – translation of the image to some direction. It is relevant to special features of the algorithm's realization. Sometimes it has no influence over the visual quality of the image, but has significant influence over the formal metrics.
4. Aliasing – so called "step effect" – irregularity of the sharp diagonal edges of the image. It emerges in the vector, as well as in the photographic images. It is most noticeable in Nearest Neighbor, DI, DDT and somewhat in Marbella.
5. Blurring – insignificant sharpness of the image after scaling. In fact it appears for all the interpolation algorithms, but is most distinctive for Simple Interpolation (SI). Boosting the sharpness increases the rest of the artifacts and vice versa – reducing the artifacts looses image's sharpness.

Some of these artifacts, regarded as negative effects, can be significantly reduced by applying some of the filtering procedures to that image. Various high-pass and low-pass filters (and combinations of them) are used to sharpen or smoothen different details in the image. This affects image quality and its total perception can be improved.

IV. CONCLUSION

The method for estimating the scaling algorithms offered in the paper is a convenient tool that allows implementing simple and efficient experiments with similar methods to be performed in order to improve their characteristics. This method is applicable for testing filtering algorithms too and even combinations of both types. The results achieved in this research, outline the directions for further tests and algorithm development.

For some methods, such as SI, DI and DDT it is not quite reasonable, attempts for improvements to be done, while for others (BSI-s, BI and FI) is recommended to be done efforts for improving the sharpness and the precision of the interpolation. Marbella also has potential for improving the quality. NN is suitable for image preview and for some specific applications.

REFERENCES

[1] D. Lancaster, "Using Cubic Spline Basis Functions for Image Pixel Interpolation", www.tinaja.com/glib/basis.pdf.

[2] D. Su, P. Willis, "Image Interpolation by Pixel Level Data-Dependant Triangulation", Computer Graphics Forum, No. 1, 2004.

[3] D. Vatolin, S. Putilin, "Image scaling methods evaluation and comparative results between different methods", Proceedings of Graphicon-2003, pp 202-207, Moscow, September 2003 (in Russian).

[4] S. Battiato, "A new edge-adaptive zooming algorithm for digital images", Proceedings of Signal Processing and Communications SPC, pp 144-149, Marbella, Spain, 2000.

[5] http://.efg2.com/Lab/Library/ImageProcessing/Algorithms.htm