A survey on evaluation methods for image interpolation

2009 Meas. Sci. Technol. 20 104015

(http://iopscience.iop.org/0957-0233/20/10/104015)

Meas. Sci. Technol. **20** (2009) 104015 (9pp) doi:10.1088/0957-0233/20/10/104015

A survey on evaluation methods for image interpolation

Angelos Amanatiadis and Ioannis Andreadis

Laboratory of Electronics, Department of Electrical and Computer Engineering, Democritus University of Thrace, 12 Vas. Sofias Str, GR-67100, Xanthi, Greece

E-mail: aamanat@ee.duth.gr and iandread@ee.duth.gr

Received 2 January 2009, in final form 6 April 2009
Published 4 September 2009
Online at stacks.iop.org/MST/20/104015

Abstract
Image interpolation is applied to Euclidean, affine and projective transformations in numerous imaging applications. However, due to the unique characteristics and wide applications of image interpolation, a separate study of their evaluation methods is crucial. The paper studies different existing methods for the evaluation of image interpolation techniques. Furthermore, an evaluation method utilizing ground truth images for the comparisons is proposed. Two main classes of analysis are proposed as the basis for the assessments: performance evaluation and cost evaluation. The presented methods are briefly described, followed by comparative discussions. This survey provides information for the appropriate use of the existing evaluation methods and their improvement, assisting also in the designing of new evaluation methods and techniques.

Keywords: interpolation, evaluation methods, error measurement, performance analysis

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Interpolation of image-sampled data is required in many consumer, medical and industrial imaging applications. Image interpolation represents an arbitrary continuously defined function as a discrete sum of weighted and shifted basis functions. The ideal image interpolation algorithm should preserve the qualitative characteristics of the output image since interpolated images suffer from artefacts, such as blurring, discontinuities in edges and checkerboard effects. Furthermore, the applied methods should meet some quantitative attributes especially when they are oriented for real-time imaging applications such as mobile phones or digital cameras. The algorithms should introduce low computational cost and low memory requirements in order to meet the hard real-time requirements of such implementations.

Widely used algorithms such as the nearest-neighbour and bilinear interpolation [1] exhibit computational simplicity, but severe blurring problems particularly in edge regions. Linear approaches are those most frequently used due to the fact that although nonlinear methods, such as bicubic interpolation [2] and spline interpolation [3, 4], produce better results they have a larger computational burden and involve blurring. Edge-directed interpolations have been attempted to overcome such shortcomings by applying a variety of operators according to the edge directions [5, 6]. The main disadvantages of this category are the need of an extra processing stage for the edge extraction and their ability to identify only certain angles of edges. A Markov random field model-based edge-directed interpolation method is proposed in [7] which relates interpolated images to the minimal energy state of a 2D random field. Neural networks have also been used for image interpolation [8]. These on the other hand require a large number of cells in the networks, rendering them computational-wise too intensive. Fuzzy interpolation approaches have been proposed [9, 10] for two-dimensional signal resampling, with optimal visual results but lack in simplicity and might require additional processing for edge identification. Area-based interpolation [11–13] computes each interpolated pixel by proportional area coverage of a filtering window which is applied to the input image. The Mitchell and Lanczos [14, 15] interpolation methods use a window function of limited spatial support in order to reduce spectral leakage and loss of spatial resolution, requiring however a great amount of hardware resources. Adaptively, quadratic image interpolation [16] has recently been reported with very good visual results but the computational cost for its coding remains its main disadvantage.

The development of image interpolation techniques has attracted significant attention, but relatively fewer attempts have been spent on their evaluation techniques. Several and different evaluation techniques can be found in literature related to image interpolation [17–20]. However, due to the unique characteristics and wide applications of image interpolation, a separate study of their evaluation methods is crucial. Furthermore, the differences between Euclidean, affine and projective transformations require different evaluation approaches in contrast to the conventional metric methods. Such kinds of surveys which present different measurement evaluation methods for several digital image processes have been proved to be a very helpful tool for researches in the past years.

In this paper, we present a novel classification scheme consisting of two different categories. The first category includes all the methods that can evaluate the fidelity criteria of an algorithm. The presented methods can evaluate the performance features of the output-interpolated image in terms of artefacts, blurring, discontinuities in edges or checkerboard effects. A new evaluation method utilizing ground truth optically captured images for comparisons. More specifically, the proposed method measures the efficiency of the algorithm through sets of interpolated images. The first image is digitally interpolated, using the algorithm under evaluation and the second image is optically acquired corresponding to the ground truth image. The second evaluation category includes methods which measure the cost features of the algorithm itself. The computational burden of the algorithm and its memory requirements can be evaluated using these methods. Finally, we present the applicabilities and trade-offs between the evaluation methods. The appropriate use of both analysis categories is crucial in defining which image interpolation algorithm performs better in specific conditions.

The rest of the paper is organized as follows. In section 2, we provide an introduction to image interpolation. In section 3, we classify the evaluation methods in two main categories. In section 4, we describe the existing evaluation methods which measure the performance characteristics of the algorithms. Furthermore, a new evaluation method is also proposed. In section 5, we present the methods for cost analysis of the interpolation algorithms. In section 6, we provide a further discussion presenting the applicabilities and trade-offs between all presented evaluation methods. Finally, we give concluding remarks in section 7.

2. Image interpolation

Image interpolation can be described as the process of using known data to estimate values at unknown locations. The interpolated value $f(\mathbf{x})$ at coordinate $\mathbf{x}$ in a space of dimension q can be expressed as a linear combination of samples $f_{\mathbf{k}}$ evaluated at integer coordinates $\mathbf{k} = (k_1, k_2, \ldots, k_k) \in Z^q$

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in z^q} f_{\mathbf{k}} \varphi_{\text{int}}(\mathbf{x} - k) \quad \forall\, \mathbf{x} = (x_1, x_2, \ldots, x_q) \in R^q. \tag{1}$$

The sample weights are given by the values of the basis function $\varphi_{\text{int}}(\mathbf{x} - \mathbf{k})$ [18]. The main idea behind interpolation is to specify a basis function which approximates the original input image in order to acquire the output-interpolated image. Since the basis interpolating function is defined on a continuous-valued parameter, it may be evaluated at arbitrary points. The resulting values comprise the interpolated image. An important requirement is that the interpolating function must pass through the known image data values. The order of interpolation refers to the maximum number of continuous derivatives that the interpolating function is required to possess. For this reason, higher-ordered B-spline image interpolation results tend to look smoother and more continuous. However, this is at the cost of increased computation.

The simplest of the B-spline interpolators are of the zero and first orders. These are known as pixel replication and bilinear interpolation [1], respectively. In pixel replication, each output pixel simply obtains the value of the closest input pixel. In bilinear interpolation, the basis function is piecewise linear which means that each output pixel may be computed as a linear combination of up to four input pixels. Both pixel replication and bilinear interpolation are very common, and they are known to perform satisfactorily for interpolating smooth textures. On the other hand, they offer little in terms of edge and detail rendition. Third-order, B-spline interpolation presents better interpolation quality than pixel replication and bilinear interpolation [21]. However, one drawback with cubic B-spline interpolation is that a considerable amount of computation is required to specify the interpolating function. Another drawback of cubic B-spline interpolation is that once the interpolating function has been obtained, still more computations are required to recover the desired output samples.

Each of the algorithms described above amounts to the application of a single linear filter. Since the same filter is applied to obtain each output pixel, these methods tend to average image data across high-contrast boundaries such as edges. The result is that interpolations from these methods are blocky or blurry. This is especially true in the cases of pixel replication and bilinear interpolation. In response to this problem, a wide variety of nonlinear approaches have been proposed. In edge-directed algorithms, the idea is to find edges in the source data, and to render them continuous and sharp in the output image. However, this approach introduces an extra step for edge identification. In local regions where edges are detected, the approach is to find the description of an edge which passes through the local region. This edge description may be used to interpolate without blurring the values of pixels which lie on different sides of the edge. In regions where edges are not detected, many of the edge-directed algorithms use bilinear interpolation, since it usually provides very satisfactory results for smooth textures, at minimal computational cost. Other nonlinear approaches to image interpolation employ stochastic models for image data. The objective is to obtain a high spatial resolution rendering which is optimal in some well-defined sense.

As we can see, interpolation algorithms vary in the used interpolation function making them susceptible or not in certain conditions as well as changing their computational cost. A decision for which interpolation algorithm performs best in terms of performance and computational cost is a crucial but complicated process; thus evaluation methods should be clearly defined, categorized and carefully used.

3. Evaluation methods

The evaluation methods were classified into two categories. The first category studies the performance of the algorithms based on fidelity criteria. Both interpolation algorithms and output images are subjected to evaluation analysis which can clarify the efficiency of each algorithm. However, evaluating output interpolated images requires the application of the algorithm in certain image transformations. In literature, the most widely used transformation for evaluation is the scaling transformation [3, 16, 22]. In our analysis, we expand the evaluation also in Euclidean transformations such as rotation and translation. Furthermore, in order to have more thorough analysis we take into consideration not only the scaling-up process, which is mainly used in literature, but the scaling-down process as well.

Until now, all methods characterize different interpolation algorithms by simply computing the fidelity criteria based on comparisons between interpolated images without the use of any prior knowledge of the assumed perfect interpolated image. In this category, we propose an evaluation method which uses optically transformed images as the ground truth for the comparisons. The method relies on the fact that an optically transformed image corresponds to the ground truth image of a digitally interpolated one.

However, not all properties of the interpolation algorithms can be evaluated by the performance analysis. The second category directly treats the algorithms themselves by considering their complexity and memory utilization. These properties could be helpful for selecting suitable algorithms in specific applications. For example, camera mobile phones have limited hardware resources, such as memory and processing power; thus the implementation of sophisticated and complex interpolation algorithms is inefficient. Low-power design analysis is also necessary, especially for mobile multimedia devices.

The interpolation algorithms that we use for applying the evaluation methods on them are the three most widely used. The nearest neighbour, the bilinear and the bicubic interpolation, which are briefly described in section 2. The purpose of this paper is not to evaluate the previously mentioned algorithms but to demonstrate the applicability of the evaluation methods using these widely known interpolation algorithms.

4. Performance analysis

Transformation comparison tests in this category should include both synthetic and natural images. For scaling transformation, the scaling factors should be chosen from both integers and non-integers in order to realize differences and weaknesses between methods. Furthermore, a complete evaluation procedure should include analysis from several different images and present the average results. In certain cases, the minimum or maximum errors can be presented instead. In the following methods, all used images are greyscale using the equivalent equations. However, all equations can be adjusted to any colour space. For example, the peak signal-to-noise ratio (PSNR) of an interpolated colour image denoted by PSNR_p is [23]:

$$\mathrm{PSNR}_p = (4 \times \mathrm{PSNR}_Y + \mathrm{PSNR}_U + \mathrm{PSNR}_V)/6, \qquad (2)$$

where PSNR_Y, PSNR_U, PSNR_V are the corresponding PSNR values of the (4 : 1 : 1) Y, U and V components of the interpolated image, respectively.

4.1. Fourier analysis

The Fourier analysis compares the interpolation kernel against the sin*c* function, which is a perfect reconstructor for band-limited signals. Figure 1 shows the magnitude of the Fourier transform of each interpolation kernel. The magnitude of the Fourier transform for each kernel is plotted within the interval $0 \leqslant f \leqslant 2$ or $0 \leqslant \omega \leqslant 4\pi$, where $\omega = 2\pi f$. The interval $0 \leqslant f \leqslant 1/2$ is called pass band and the $f = 1/2$ or $\omega = \pi$ is the cutoff point. For quality assessment, two characteristics are of interest in the pass band which are closely related to the deviation from the ideal reconstructor: how fast the function starts to fall and the fall step. In the stop band, the best performance is indicated by the amplitude which is closest to zero presenting also the least slopes. Deviation within the pass band causes blurring, and large amplitudes of ripples and side lobes in the stop band translate into aliasing and fringes. In Fourier analysis, the sampling of an interpolated (continuous) image is equivalent to interpolating the (discrete) image with a sampled interpolation function [24].

4.2. Interpolation quality

This evaluation method requires the repetitive application of the image interpolation algorithm. Using a starting input image $I(x, y)$, we employ repetitively the interpolation algorithm under evaluation in order to acquire the final interpolated image $\hat{I}(x, y)$. The final image must have the same size and spatial resolution as the input image. Then, the interpolation quality is assessed by the pixel differences before and after the successive interpolations. Since all image transformations can be applied in this method, these criteria are excellent for visualizing the interpolation error, and thus is the most widely used evaluation method.

Next, we will apply the evaluation method in scaling and rotation transformations, using the root mean square error (RMSE) and the peak signal-to-noise ratio (PSNR) for measuring the interpolation error. Given an input

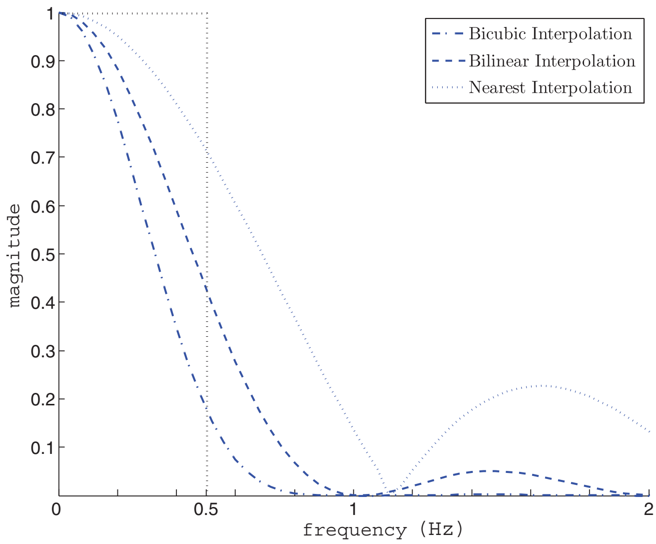

Figure 1. Magnitude of the Fourier transform in a linear scale for evaluation of pass band performance. The vertical dashed line determines the cutoff point for $f = 1/2$.

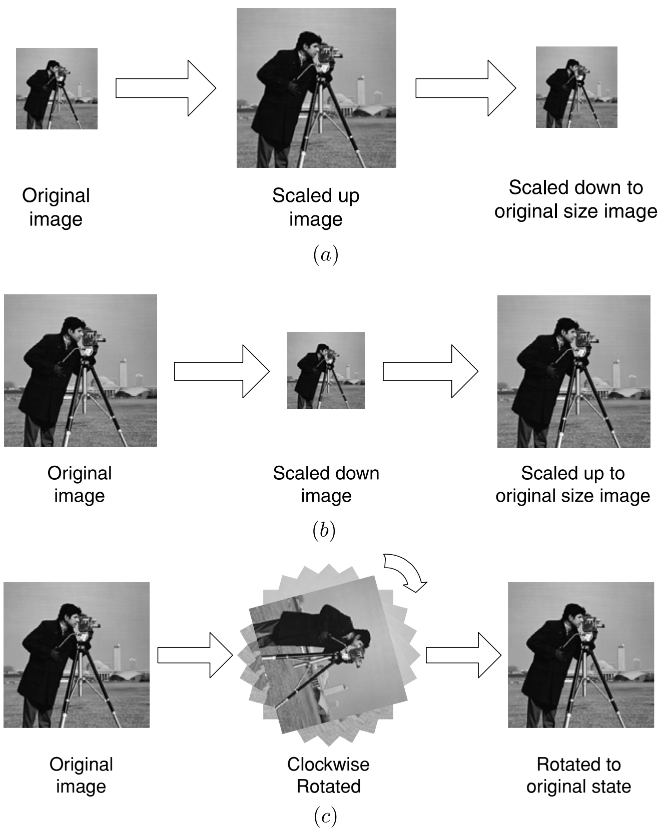

Figure 2. Repetitive image transformations required for the performance evaluation method: (*a*) scale down-after-up; (*b*) scale up-after-down; (*c*) successive rotations.

image $I(x, y)$, the testing procedure in order to evaluate an interpolation algorithm in scaling transformation is as follows. The method requires two scaling transformations performed by the same interpolation algorithm as shown in figure 2(*a*) for the *cameraman* image. The first transformation is to scale up an image by a magnification factor f. Then, the scaled image is scaled down by the same magnification factor f. The final image $\hat{I}(x, y)$ has the same spatial resolution as the original one but it has been twice interpolated. For rotation, the input image can be successively rotated 16 times by 22.5^0 in order to reach its initial state, as shown in figure 2(*c*). However, for even better evaluation a user can apply several magnification factors for scaling and several rotation degrees to obtain the average results. This can be useful for evaluating interpolation algorithms which present optimal results only in certain conditions [25]. The RMSE between the original image $I(x, y)$ and the interpolated image $\hat{I}(x, y)$ is given by [26]

Table 1. Comparisons between the reference *cameraman* image and the scaled down-after-up interpolated image.

	Interpolation method		
Metric	Nearest	Bilinear	Bicubic
PSNR (dB)	70.50	73.22	74.43
RMSE	0.0761	0.0556	0.0484

$$\mathrm{RMSE} = \left(\frac{1}{m \times n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (\hat{I}(x, y) - I(x, y))^2 \right)^{1/2}, \quad (3)$$

where the images are of size $m \times n$.

The PSNR metric can be also used which is closely related to RMSE. The PSNR in decibels (dB) between the original image $I(x, y)$ and the interpolated image $\hat{I}(x, y)$ is given by [27]

$$\mathrm{PSNR} = 20 \times \log_{10} \left(\frac{\mathrm{MAX}_I}{\mathrm{RMSE}} \right), \quad (4)$$

where MAX_I is the maximum pixel value of the images. When the pixels are represented using eight bits per sample, this is 255. To avoid border effects, it is recommended to extract centred subimages before computation. However, the actual value is not meaningful, and only the comparison between different interpolation results provides a measure of quality. The results presented in table 1 demonstrate the RMSE and PSNR metrics for the *cameraman* image after the scale down-after-up image transformation.

The reverse operation, of first scaling down and then scaling up, is shown in figure 2(*b*). While it may be expected that this reverse operation may present the same error as the previous procedure, the errors produced are different from the previous ones since many interpolation algorithms have different behaviour between scaling-up and scaling-down. Especially, in the case of scale up-after-down, the RMSE values of the scaled images by nearest neighbour are zero because the point sampling returns back the original image.

The main feature of this method is that many different interpolation errors can be evaluated in the results. For example, an algorithm which has outstanding performance in scaling process but inadequate performance in rotation, would present average overall performance results.

4.3. Step edge response

The following evaluation method is related to picture quality and especially to interpolation edge behaviour. Similar measurements can be extracted also from Fourier analysis as defined in section 4.1. However, this method can be applied also to interpolation algorithms which are not expressed by a basis function. These algorithms can be area-based interpolations, interpolations using neural networks and interpolations based on local features. While there are various types of edges in an image, step edges are visually more distinct than other types of edges. Therefore, preservation of the sharpness and continuity of the step edges is a key feature. The objective of this method is to evaluate whether the algorithm preserves the original step edges sharp and continuous. In order to evaluate this, a synthetic input image containing only two different area intensities of 225 and 25 is geometrically transformed. In the case of scaling transformation, the image is scaled up and depending on the algorithm used, an unwanted area might arise with a progressive transition from 225 to 25. The same effect can appear after translation of the image or after rotation. Generally, faster transition provides better performance. Figure 3, illustrates the step edge response from 225 to 25. More precisely, figures 3(*a*) and (*b*) present the spatial performance of each algorithm for scaling and rotation transformation, respectively. The results are more distinctive in the figure 3(*c*) performance diagram. For bilinear interpolation, a single new spatial area arises, caused by the modest low-pass filter that bilinear interpolation uses. New spatial areas are even more considerable in bicubic interpolation due to the over smoothing in the pass band. In this task, nearest neighbour interpolation presents the best results since it reproduces exactly the interpolated pixels. However, this feature also results in the strong aliasing effects that are associated with the nearest neighbour interpolation.

This evaluation method is also widely used in text-oriented interpolation methods [28, 29], where high-frequency input images are studied.

For calculating the kernel error we can compare $f(\mathbf{x})$ with $f_h(\mathbf{x})$ using the following equation:

$$\epsilon^2(h) = \int_{-\infty}^{\infty} (f(\mathbf{x}) - f_h(\mathbf{x}))^2 \, dx_1 \, dx_2 \cdots dx_q \quad \forall\, \mathbf{x} = (x_1, x_2, \ldots, x_q) \in R^q, \tag{5}$$

where $h > 0$ is the sampling step and $f_h(\mathbf{x})$ is the interpolated function in certain intervals denoted by h. This difference between $f(\mathbf{x})$ and $f_h(\mathbf{x})$ describes how fast the interpolated function f_h converges to the true function f, when the sample steps are becoming smaller and smaller.

4.4. Evaluation with ground truth images

Here we propose a method which facilitates the attributes of raw captured images, since modern imaging devices can perform the desired transformations without the use of digital image processing. More specifically, we can capture zoomed, rotated and translated images. Using this *a priori* knowledge, we can finally have sets of optically and digitally transformed images and apply the evaluation methods on these sets of images. Depending on the transformation certain procedures must be followed. For rotation and translation we only need the physical position change of the imaging device. Since we need high precision, a user can find some external equipment such as high-precision tripods very helpful. For scaling transformation, the optical zooming feature of the imaging device will be used.

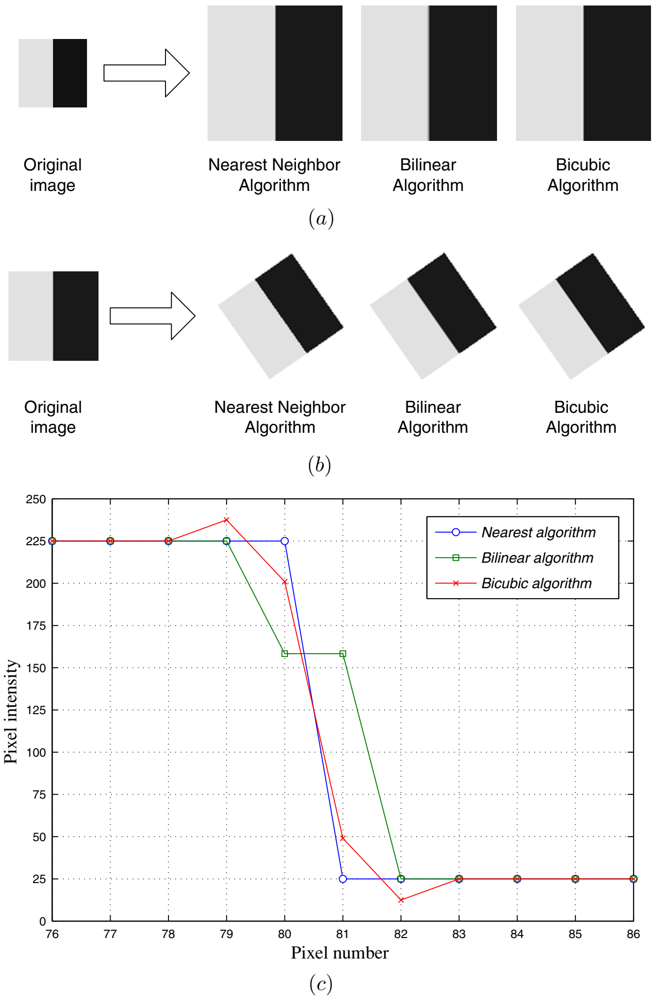

Figure 3. Step edge performance comparison of a synthetic original image containing only two different area intensities of 225 and 25: (*a*) scaled images; (*b*) rotated images; (*c*) performance diagram.

The focal length of a lens is defined as the distance in mm from the optical centre of the lens to the focal point, which is located on the solid-state sensor. A change in focal length allows the user to come closer to the subject or to move away from it. The optical zooming factor is calculated by

$$\text{Optical zooming factor} = \frac{f_{\max}}{f_{\min}}, \tag{6}$$

where $f_{\max}$ represents the maximum focal length and $f_{\min}$ the minimum focal length. Modern imaging devices provide an accurate estimation of (6) to the user's camera screen. Utilizing the previous feature, we can acquire pictures with different optical zooming settings as illustrated in figure 4. More specifically, with this procedure we can construct a database consisting of the original images,

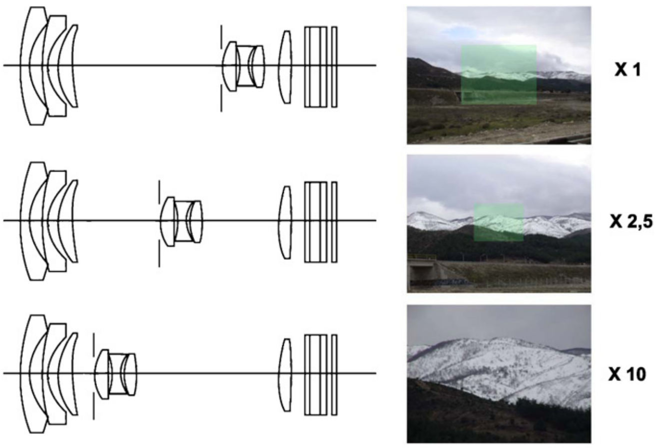

Figure 4. Various settings of an optical zooming system: ×1 magnification factor; ×2.5 magnification factor; ×10 magnification factor.

figure 5(*a*) and the optically scaled images, figure 5(*b*), with a known magnification factor. Then, a digitally scaled image, magnified by the known magnification factor, will be produced by the algorithm under study, figure 5(*c*).

The proposed method creates a ground truth image for comparisons with the under-evaluation algorithms. All further test images will digitally be interpolated from the original image, for example figure 5(*a*). An online database with similar sets of pictures can be created for comparison purposes and tests. The qualitative measurements can be done with metrics such as RMSE or PSNR, as mentioned in section 4.2. Comparison results for the scaling transformation are presented in table 2. Based on the results, bicubic interpolation demonstrates an overall superiority against the interpolations of bilinear and nearest neighbour.

4.5. Subjective observation

Judgements based on subjective observations are accepted and widely used in bibliography [5]. Experienced observers can report useful measurements especially when a number of images are evaluated and results are processed with the help of analytical methods. In figure 6, we demonstrate the visual results, scaling the *cameraman* image by a scaling factor of 1.5, in order to provide a subjective comparison between nearest neighbour, bilinear interpolation and bicubic interpolation.

Table 2. Comparisons between optically scaled image figure 5(*b*) (ground truth) and digitally scaled images figure 5(*c*) by different interpolation algorithms.

	Interpolation method		
Metric	Nearest	Bilinear	Bicubic
PSNR (dB)	67.28	67.39	67.56
RMSE	0.1102	0.1089	0.1067

Table 3. Number of operations per pixel.

	Interpolation method		
Operation	Nearest	Bilinear	Bicubic
Addition	2	16	22
Multiplication	0	18	29

5. Cost analysis

The second evaluation category includes methods for the cost analysis of the interpolation algorithm itself. The computational burden and memory requirements of the algorithm are evaluated in this category. Such methods provide overall criteria and especially indicate the applicability of each interpolation algorithm in certain applications such as real-time implementations.

5.1. Computational Burden

Defining the complexity of an algorithm is a crucial but difficult task in literature [30, 31]. However, some methods have been used in works related to image interpolation. The number of operations required for a pixel interpolation is a widely accepted method. This number includes the operations required for the convolution as well as the number required for the calculation of the basis function. The convolution of an $N \times N$ mask needs N^2 multiplications and $N^2 - 1$ additions. Table 3 reports the number of operations

(*a*) (*b*) (*c*)

Figure 5. The original image and the two under evaluation images: (*a*) original image; (*b*) optically scaled image (ground truth); (*c*) digitally scaled image.

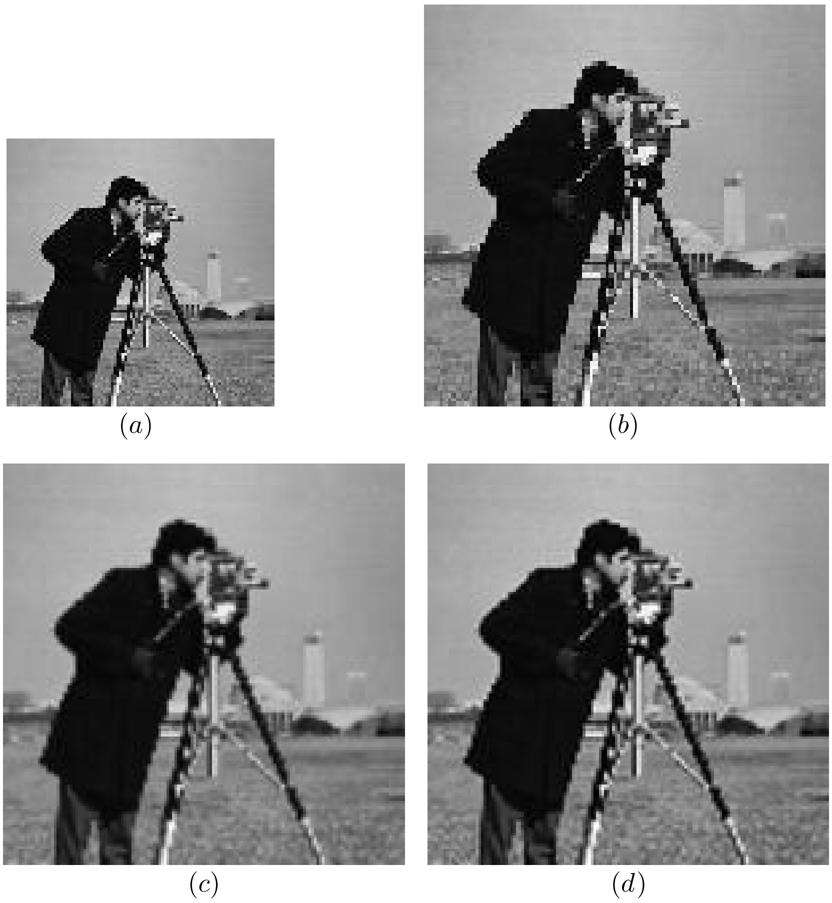

Figure 6. Visual interpolation results for *cameraman*: (*a*) original; (*b*) nearest interpolation; (*c*) bilinear interpolation; (*d*) bicubic interpolation.

per pixel for the nearest neighbour, bilinear and bicubic interpolations. However, this method is appropriate only for simple and straightforward algorithm implementations and its applicability is constrained in sophisticated algorithms. Additional operations need implementations which use prefiltering steps and basis functions that include mathematical functions such as sine or cosine.

5.2. *Runtime comparisons*

Execution times have also been used in [32, 33] for evaluating the computational cost of algorithms. The comparison tests are performed for several transformations on computers with known characteristics. The compared execution times provide a measure of the lowest complexity technique. For even better comparisons, detailed statistics such as internal core cycles, the number of sequential or non-sequential cycles and the number of program instructions executed can be provided from dedicated processor emulators.

Summary charts which include both performance and cost metrics are also very useful as they provide an overall analysis for each algorithm compared to others. Figure 7 shows a chart, where the x-axis corresponds to the execution time for the rotation of an image and the y-axis corresponds to the equivalent interpolation quality using PSNR measure. Based on the chart, bicubic interpolation demonstrates a PSNR superiority against the other interpolations, presenting however longer execution times.

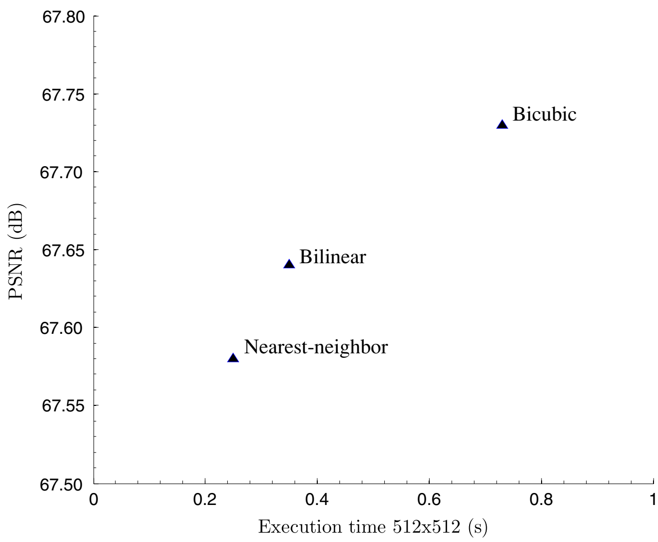

Figure 7. Performance chart for rotation transformation.

However, we should mention that in the last few years there has been an increase in research in the area of using graphics processing units (GPUs) for image processing algorithms instead of general purpose computing. These GPUs are designed to perform a limited number of operations on very large amounts of data. They typically have more than one processing pipeline working in parallel with each other. For this reason, detailed comparisons should be clearly discriminated based on the processor type used. Run-time comparisons for image interpolation algorithms using GPUs have been reported in [34, 35].

5.3. Data memory requirements

While the computational burden reflects mostly on the processor utilization, the demands on memory resources cannot be estimated with the same metrics. Imaging devices such as mobile phones, digital cameras and PDAs share a restricted amount of embedded memory for image processing. Thus, interpolation algorithms must be designed and optimized for these applications in order to meet the real-time performance restrictions. While most techniques use convolution operations, the data memory requirements are strictly connected to the contributing input pixels. The convolution of an $N \times N$ mask needs $N^2 + 1$ memory accesses. Furthermore, the amount of embedded memory needed is closely related to the basis function. For example, bilinear interpolation uses four neighbouring pixels for each calculation. This means that regardless of the image transformation, the operation can be done with only two line buffers. Thus, an 8 bit colour 640×480 pixel resolution input image would require an amount of $8 \times 3 \times 640 \times 2 =$ 3.75 KB of internal memory. However, bicubic interpolation which uses 16 neighbour input pixels, needs four line buffers for each operation which is equal to $8 \times 3 \times 640 \times 4 =$ 7.5 KB of internal memory.

Interpolation algorithm memory tests can be implemented also with the aid of a processor developer suite. The suite enables the user to specify specific layers of the memory that have specific characteristics. This functionality can help users to simulate certain memory models even for memory-limited applications. The algorithms, written in C language, are cross-compiled into the processor codes and mapped to the memory. The accurately simulated processor reads and executes the machine codes for memory testing. The process can easily be modified for various purposes and several processors. Simulation measurements give all the memory map details of the algorithm under evaluation such as the number of accesses on off-chip memory, the number of accesses on on-chip memory and wait times. The algorithm with lower accesses and wait times presents the best performance.

5.4. Power requirements

The power evaluation method can provide a metric of power consumption needed by the interpolation algorithm. In systems that involve multidimensional streams of signals such as images or video sequences, it has been shown that the majority of power cost is due to the off-chip memory interactions [36]. These interactions include data and instruction memory use. It is obvious that the data memory power consumption is closely related to the memory requirements of each interpolation method, and that the instruction memory power consumption is equivalent to the algorithm computational burden. The relatively low data memory requirements of image interpolation algorithms also present a low impact on the total power consumption. The dominant consumption is derived from the instruction memory use.

The starting point of this evaluation method is the description of the interpolation in a high-level language. Next, using processor developer suites we can acquire the number of instructions and core cycles needed from the algorithm. Next, with the use of estimators of energy and performance models we can have power consumption predictions for the under-evaluation algorithms [37, 38]. The models that describe the energy and performance characteristics of the memory layers must be real memory models, which are based on propriety memory models of industrial vendors. Summarizing, in power critical applications, image interpolation algorithms should follow design strategies for a low number of memory accesses since they reduce the power consumption due to memory traffic.

6. Further discussion

Although image interpolation is widely used in image processing, each application has its specific requirements. Thus, depending on the application, an appropriate choice of evaluation methods should be made. Furthermore, whether an evaluation method is optimal or not is associated with the interpolation technique, the content of the image and the geometric transform. The proposed evaluation categorization makes a discrete separation between performance and cost requirements.

For applications such as medical imaging, where the most crucial feature is the fidelity of the output image, the most appropriate methods are included in the performance category. More specifically, since the contents are low-frequency images the interpolation quality criteria should be used, since it can give an overall measure of quality. For applications which include high-frequency images, such as text or document processing, Fourier analysis or the step-edge response metrics should be preferred. However, sophisticated interpolation algorithms which can not be expressed by a single basis function such as neural network or fuzzy approaches can be evaluated only using the step-edge response method.

In contrast to quality performance demanding applications, the cost analysis is appropriate to applications with limited resources. Depending on the critical resource of the application, the equivalent evaluation method should be applied. If there are real-time constraints, the design and evaluation strategy should be based upon minimizing the runtime factor. Before establishing an interpolation algorithm in a mobile embedded device, power requirement analysis must be performed since in this application the most critical factor is power consumption which is equal to increased memory standby time. Data memory and computational burden analysis is vital in applications that interpolate large spatial resolution images such as GIS applications.

7. Conclusion

In this paper, we studied the methods so far proposed for image interpolation evaluation and comparison. We also proposed a new evaluation method utilizing ground truth optically captured images for comparisons. A method classification scheme has also been proposed. Each method we studied in this paper has advantages and limitations. From an application point of view, those that belong to different groups are more complementary than competitive. Besides, the performance of interpolation algorithms is influenced by many factors; thus only one evaluation method would not be adequate to evaluate all properties of an algorithm and different methods should be combined. This survey provides information for the appropriate use of the existing evaluation methods and their improvement, assisting also in the designing of new evaluation methods and techniques.

References

[1] Jain A K 1989 *Fundamentals of Digital Image Processing* (Upper Saddle River, NJ: Prentice-Hall)
[2] Keys R G 1981 Cubic convolution interpolation for digital image processing *IEEE Trans. Acoust., Speech, Signal Process.* **29** 1153–60
[3] Hwang J W and Lee H S 2004 Adaptive image interpolation based on local gradient features *IEEE Signal Process. Lett.* **11** 359–62
[4] Unser M, Aldroubi A and Eden M 1991 Fast B-spline transforms for continuous image representation and interpolation *IEEE Trans. Pattern Anal. Mach. Intell.* **13** 277–85
[5] Battiato S, Gallo G and Stanco F 2002 A locally adaptive zooming algorithm for digital images *Image Vis. Comput.* **20** 805–12
[6] Cha Y and Kim S 2007 The error-amended sharp edge (EASE) scheme for image zooming *IEEE Trans. Image Process.* **16** 1496–505
[7] Li M and Nguyen T Q 2008 Markov random field model-based edge-directed image interpolation *IEEE Trans. Image Process.* **17** 1121–8
[8] Plaziac N, Syst E M and Woodbury N 1999 Image interpolation using neural networks *IEEE Trans. Image Process.* **8** 1647–51
[9] Chen J L, Chang J Y and Shieh K L 2000 2D discrete signal interpolation and its image resampling application using fuzzy rule-based inference *Fuzzy Sets Syst.* **114** 225–38
[10] Ting H and Hang H 1997 Edge preserving interpolation of digital images using fuzzy inference *J. Vis. Commun. Image Represent.* **8** 338–55
[11] Amanatiadis A, Andreadis I and Konstantinidis K 2008 Design and implementation of a fuzzy area-based image-scaling technique *IEEE Trans. Instrum. Meas.* **57** 1504–13
[12] Amanatiadis A, Andreadis I and Gasteratos A 2007 A Log-Polar interpolation applied to image scaling *Proc. IEEE Int. Workshop on Imaging Systems and Techniques* pp 1–5
[13] Kim C H, Seong S M, Lee J A and Kim L S 2003 Winscale: An image-scaling algorithm using an area pixel model *IEEE Trans. Circuits Syst. Video Technol.* **13** 549–53
[14] Mitchell D P and Netravali A N 1988 Reconstruction filters in computer-graphics *Proc. 15th Ann. Conf. on Computer Graphics and Interactive Techniques* pp 221–8
[15] Grevera G J and Udupa J K 1996 Shape-based interpolation of multidimensional grey-level images *IEEE Trans. Med. Imag.* **15** 881–92
[16] Muresan D and Parks T 2004 Adaptively quadratic (AQua) image interpolation *IEEE Trans. Image Process.* **13** 690–8
[17] Lehmann T M, Gonner C and Spitzer K 1999 Survey: Interpolation methods in medical image processing *IEEE Trans. Med. Imag.* **18** 1049–75
[18] Thevenaz P, Blu T and Unser M 2000 Interpolation revisited *IEEE Trans. Med. Imag.* **19** 739–58
[19] Meijering E H W, Niessen W J and Viergever M A 2001 Quantitative evaluation of convolution-based methods for medical image interpolation *Med. Image Anal.* **5** 111–26
[20] Amanatiadis A and Andreadis I 2008 Performance evaluation techniques for image scaling algorithms *Proc. IEEE Int. Workshop on Imaging Systems and Techniques* p 114–8
[21] Unser M, Thevenaz P and Yaroslavsky L 1995 Convolution-based interpolation for fast, high-quality rotation of images *IEEE Trans. Image Process.* **4** 1371–81
[22] Ramponi G 1999 Warped distance for space-variant linear image interpolation *IEEE Trans. Image Process.* **8** 629–39
[23] Chuah C S and Leou J J 2001 An adaptive image interpolation algorithm for image/video processing *Pattern Recognit.* **34** 2383–93
[24] Parker J A, Kenyon R V and Troxel D E 1983 Comparison of interpolating methods for image resampling *IEEE Trans. Med. Imag.* **2** 31–9
[25] Chang C C, Chou Y C, Yu Y H and Shih K J 2005 An image zooming technique based on vector quantization approximation *Image Vis. Comput.* **23** 1214–25
[26] Loia V and Sessa S 2005 Fuzzy relation equations for coding/decoding processes of images and videos *Inf. Sci.* **171** 145–72
[27] Sung M M, Kim H J, Kim E K, Kwak J Y, Yoo J K and Yoo H S 2002 Clinical evaluation of JPEG2000 compression for digital mammography *IEEE Trans. Nucl. Sci.* **49** 827–32
[28] Ye J, Fu G and Poudel U P 2005 High-accuracy edge detection with blurred edge model *Image Vis. Comput.* **23** 453–67
[29] Leu J G 2000 Image enlargement based on a step edge model *Pattern Recognit.* **33** 2055–73
[30] Papadimitriou C H and Steiglitz K 1998 *Combinatorial Optimization: Algorithms and Complexity* (Mineola, NY: Courier Dover Publications)
[31] Ausiello G 1999 *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties* (Berlin: Springer)
[32] Chung K H and Chan Y H 2007 A low-complexity joint color demosaicking and zooming algorithm for digital camera *IEEE Trans. Image Process.* **16** 1705–15
[33] Lukac R and Plataniotis K N 2005 Digital zooming for color filter array-based image sensors *Real-Time Imaging* **11** 129–38
[34] Kelly F and Kokaram A 2004 Fast image interpolation for motion estimation using graphics hardware *Proc. SPIE* vol 5297 pp 184–94
[35] Ruijters D, Haar-Romeny B M and Suetens P 2008 Accuracy of GPU-based B-Spline evaluation *Proc. 10th IASTED Int. Conf. Computer Graphics and Imaging* p 117–22
[36] Catthoor F 1998 *Custom Memory Management Methodology: Exploration of Memory Organisation for Embedded Multimedia System Design* (Berlin: Springer)
[37] Wilton S and Jouppi N 1996 CACTI: an enhanced cache access and cycle time model *IEEE J. Solid-State Circuits* **31** 677–88
[38] Landman P and Rabaey J 1995 Architectural power analysis: The dual bit type method *IEEE Trans. Very Large Scale Integration (VLSI) Syst.* **3** 173–87