Spatial and Frequency Domain Comparison of Interpolation Techniques in Digital Image Processing

Miklós Póth, Tibor Szakáll

Subotica Tech, Serbia, Department of Informatics
pmiki@vts.su.ac.rs, szakall.tibor@gmail.com

Abstract:This article presents a comparison of different interpolation techniques both in the spatial and frequency domains. Various methods are presented, from a very simple linear interpolation to more complex cubic convolution. The Fourier analysis is used to compare the methods in the frequency domain. Methods are compared by doubling the images in both spatial directions and calculating the signal-to-noise ratio between the original and interpolated images. The paper will also include some notes on subjective image quality.

Keywords: interpolation, spatial domain, frequency domain, signal-to-noise ratio.

1 Introduction

Interpolation is closely related to reconstruction of continuous functions based on known discrete samples. In today's digital world where a need of storing, processing and transmitting data is constantly growing, it is not difficult to find fields of application of interpolation. In most cases the easiest and most widely used solution to these problems is interpolation, where the approximating function is constructed to match the values of the discrete function.

There can be many reasons for interpolation. In digital image processing one of the reasons may be increasing or zooming the picture. For example, when an image seen on the screen is changed into full screen mode. On the other hand, it is a well-known fact that interpolation can neither improve the quality of the picture, nor can it introduce new details into it. In most cases it degrades the quality of the image. Another common reason for interpolation is when during transmission the image gets corrupted, and its quality needs to be improved. The next reason for interpolation can be image rotation (different from *n*90°*) when original pixels will not match the grid points. In that case interpolation helps to find the pixel values in grid points. Interpolation is fundamental in digital image processing: it connects the discrete world with the continuous world [2, 3, 4, 10, 11].

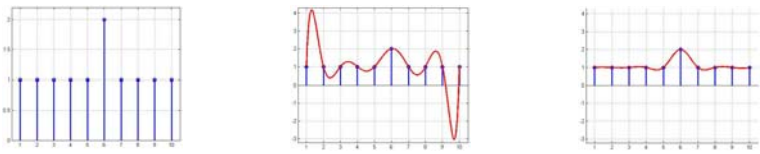
There is quite a large number of interpolation methods. In order to gain better insight into them, these algorithms are classified as described below. The basic classification divides interpolation algorithms into two main categories: the first group uses the same algorithm on the whole image surface (convolution methods), while the second group adapts according to the neighbouring pixels (adaptive methods) [2, 6]. Convolution-based methods can be further classified based on the used function set (polynomial, exponential, trigonometric, Gaussian, etc). This paper is primarily concerned with polynomial functions.

This paper is organized as follows: Section 2 shows why polynomial interpolation is used very rarely, Section 3 describes interpolation artifacts, Section 4 presents a detailed analysis of the interpolation kernels, and Section 5 discusses the experimental results, followed by the conclusions drawn in Section 6.

2 Problems with Polynomial Interpolation

When dealing with polynomial interpolation functions, it is known that when the aim is to construct a continuous function that passes through x points, a polynomial must be used whose degree is $x-1$. This polynomial is unique, and can be constructed using Lagrange basis functions. These polynomials are very rarely used because they have an oscillatory property. That is, if there is a big difference between consecutive pixel values, the polynomial will have great oscillations throughout the whole interval. This property degrades the interpolated image quality because in edge areas there is a big difference in pixel luminances. The classical Lagrange method cannot solve this problem efficiently, so alternative solutions must be used. One possibility of overcoming this problem is to use piecewise polynomials which in the joining points have C2 continuity. A short example is given to illustrate oscillatory.

Let's suppose there is a group of 10 pixels, shown in Figure 1(a). The pixels form a constant line, except one which has value 2. To interpolate this signal a polynomial of degree 9 is needed. After constructing this interpopla ting polynomial it has great oscillations throughout the whole interval as seen in Figure 1(b). This would lead to serious mistakes if the polynomial were sampled with a smaller period (for example 0.5).



(a) (b) (c)

Figure 1. (a) Original signal, (b) Interpolated signal with polynomial of degree 9, (c) Interpolated signal with piecewise polynomial.

Piecewise polynomials solve this problem quite efficiently. Certain pixels have influence only on their neighbours, so there are no oscillations on the whole interval (Figure 1(c)). Figure 2 shows a comparison of these two methods.

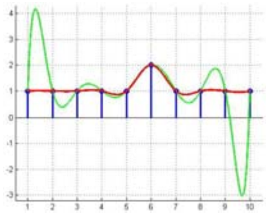


Figure 2. Comparison of polynomial and piecewise polynomial interpolations.

3. Interpolation Artifacts

3.1 Image Expansion

When increasing the image size, interpolation aims to find the best approximation of colors and luminance values based on neighboring pixels. Figure 3. shows an example of changing the image size. Pixels of the original image are separated, and the values of the unknown pixels are calculated based on their neighboghood.

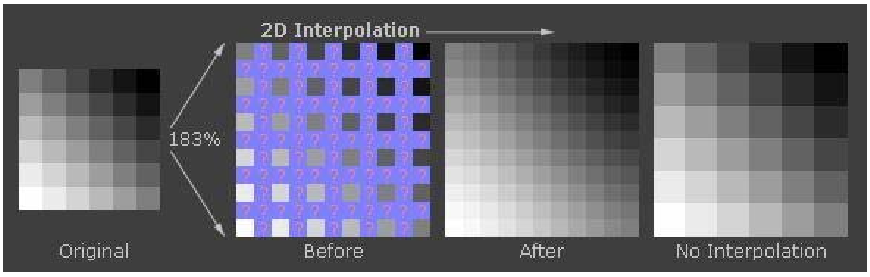


Figure 3

Increasing a digital image by interpolation

Large scaling factors considerably degrade the image quality. If the image size is to be doubled in both directions, for example from 100-by-100 to 200-by200 pixels, we have to calculate 30000 new pixels from only 10000 original values.

3.2 Blurring, Aliasing and Ringing

All non-adaptive methods are trying to find the best compromise between three unwanted artifacts: ringing (edge halos), blurring and aliasing (Figure 4).

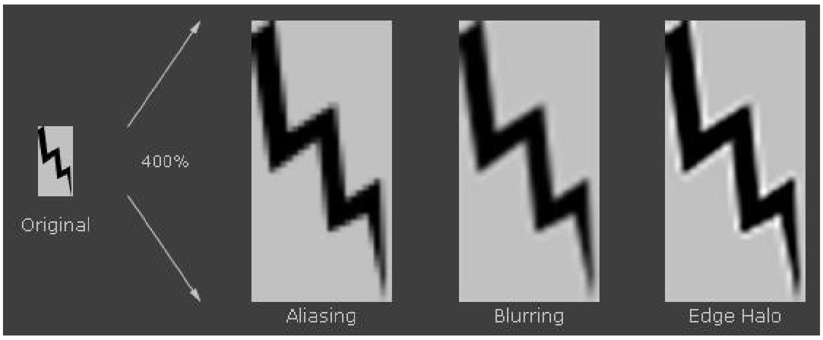


Figure 4
Interpolation artifacts: Aliasing, blurring and ringing

All methods will introduce some of those effects. Even the most advanced non-adaptive methods will trade one of these effects for the other two.

3.3 Decreasing Image Size

The main reason for decreasing image size is to reduce the needed memory to store the image, or when we want to transmit the image and quality is not the crucial factor. When increasing image size, the problem is aliasing, but when decreasing image size, the problem is moire pattern which depends on the interpolation method. Moire pattern is mostly visible on fine textures.

Figure 5 shows that decreasing the image size can cause loss of details in the image. The original (artificial) image consists of 1 pixel wide black and white vertical lines. If the even columns of the image are omitted, the resulting image will be white. If the odd lines are omitted, the resulting image will become black. By averaging, (let's say 2x2 area) the whole image will become grey. So there are situations where downsizing artifacts cannot be eliminated.

Figure 5
Loss of details caused by image downsizing

4 Spatial and Spectral Analysis of Interpolation Kernels

4.1 Linear Interpolation

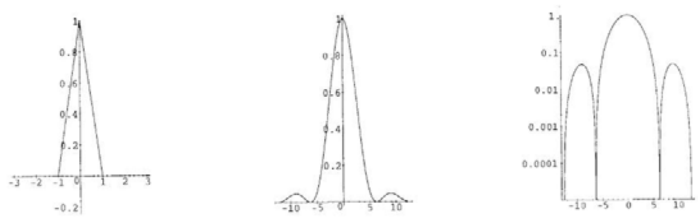
Linear interpolation is a first degree method that places a straight line between two input samples. For the $\left(x_0, x_1\right)$ interval and for function values f_0 and f_1, the interpolation function has the form:

$$f\left(x\right) = a_1 x + a_0 \,. \tag{1}$$

In spatial domain, linear interpolation is equivalent to convolution of the input image with the following kernel:

$$h(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{else} \end{cases} \tag{2}$$

This kernel is shown in Figure 6.



(a) (b) (c)

Figure 6
Linear interpolation. (a) Kernel. (b) Magnitude of Fourier transform.
(c) Logrithmic plot of magnitude

Another interpretation of linear interpolation is shown in Figure 7. Here the aim is to find a pixel with coordinates *(u,v)* based on four known pixels (green, orange, red and blue). In order to obtain the desired pixel, multiply the intensity of the red dot is multiplied with the surface of the red parallelogram, the intensity of the green dot with the green parallelogram, and so on. In the end up these products are summed up.

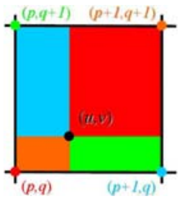


Figure 7
Interpretation of bilinear interpolation

4.2 Cubic Convolution

Cubic convolution is an efficient third degree interpolation method that approximates the ideal *sinc* function quite well [6, 7, 8]. The function is defined on four intervals: (-2,-1), (-1,0), (0,1) and (1,2). Outside the (-2,2) interval, the interpolation function is zero. The analytic form of the function is:

$$h(x)=\begin{cases} a_{30}|x|^3+a_{20}|x|^2+a_{10}|x|+a_{00} & |x|<1 \\ a_{31}|x|^3+a_{21}|x|^2+a_{11}|x|+a_{01} & 1\le|x|<2 \\ 0 & \text{else} \end{cases} \quad (3)$$

The unknown parameters a_{ij} can be determined from the following constraints:

1. $h(0)=1$ and $h(x)=0$ for $|x|=1,2$.
2. h is continuous for $|x|=0,1,2$.
3. h has a continuous first derivative for $|x|=0,1,2$.

Based on these constraints, a system of 8 equations with 7 unknowns can be established, so the system has one degree of freedom (we are free to choose one of the parameters). It is convenient to choose *a31=a*. In this case, the result is:

$$h(x)=\begin{cases} (a+2)|x|^3-(a+3)|x|^2+1 & |x|<1 \\ a|x|^3-5a|x|^2+8a|x|-4a & 1\le|x|<2 \\ 0 & \text{else} \end{cases} \quad (4)$$

Let *h(x)* be convex for *x=0*, and concave for $|x|=1$:

$$h''(0)=-2(a+3)<0 \quad \rightarrow \quad a>-3$$

$$h''(1)=-4a>0 \quad \rightarrow \quad a<0$$

So the value of a can vary between -3 and 0. Mathematically the most accurate result is obtained by choosing $a = -0.5$. Based on image content, other values for the parameter a are possible. For instance, if the goal is to enhance edges (visually desirable property), suitable values for a would be $a = -0.75$ or $a = -1$.

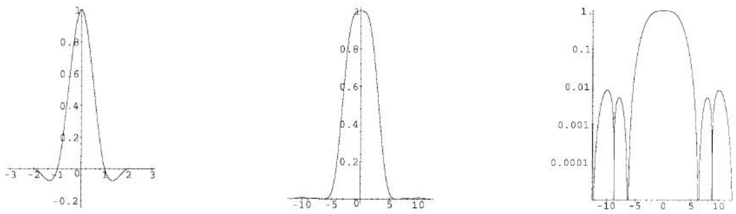


Figure 8
Cubic convolution interpolation. (a) Kernel. (b) Magnitude of Fourier transform.
(c) Logrithmic plot of magnitude

Figure 9 shows the comparison between linear and cubic convolution interpolators. The ideal *sinc* function for reference can also be seen.

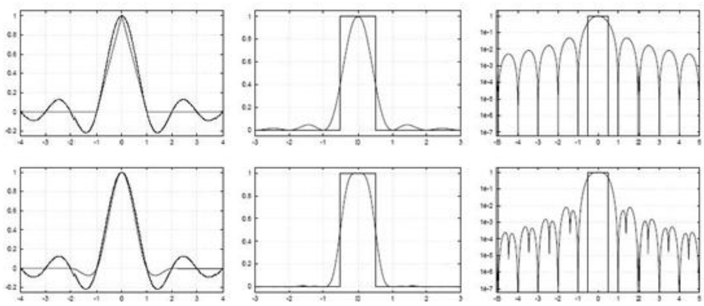


Figure 9
Comparison of linear and cubic convolution interpolators.

Another measure that can be used to compare the quality of interpolation kernels is the first derivative of its Fourier transform at *f=-1/2*. Table 1 shows these values along with the value for the ideal *sinc* interpolation.

Table 1
Comparison of first derivative of the Fourier transforms of linear and cubic kernels

Kernel	$\widetilde{H}^{(1)}(-1/2)$
$h_{Linear}(x)$	1.621
$h_{Cubic}(x)$	2.321
$h_{Sinc}(x)$	∞

The improvement of cubic convolution over linear is considerable, but the advantage of higher convolution kernels over cubic is only marginal. That is the reason why most of commercial image processing software use cubic convolution: it is relatively fast and computationally cheap.

5 Experimental Results

This section will present an analysis of the quality of interpolation methods based on the quality of the reconstructed image. The testing was done by shrinking the 256x256 pixel image to size 128x128 pixels. In the next step the image was expanded to its origial size of 256x256 pixels. Image expansion was done using the nearest neighbour, linear and cubic convolution methods. However, the fastest, nearest neighbour gives the worst visual results, thus it will not be discussed further. After expansion, the signal-to-noise ratio was compared between the original and the reconstructed images. Moreover, the spectra of the reconstructed images were compared. In this paper the test image used is the Cameraman. Figure 10 shows the test image Cameraman before and after interpolation with the appropriate spectra.

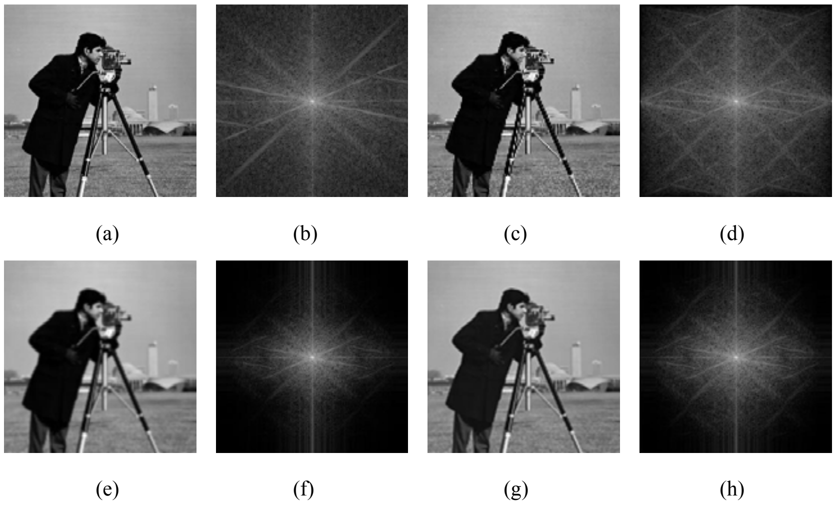


Figure 10
(a) Test image Cameraman. (b) Spectrum of Cameraman. (c) Cameraman after nearest neighbor interpolation. (d) Spectrum of (c). (e) Cameraman after bilinear interpolation. (f) Spectrum of (e). (g) Cameraman after cubic convolution interpolation. (h) Spectrum of (g).

Based on spectrum images, the following conclusions can be drawn: not every interpolation method shown degrades the low frequency components of the test image (the middle part of the interpolated spectra differ very little from the original). Differences become visible on higher frequencies. High frequency components suffer the least damage at the nearest neighbor interpolation and the most damage at linear interpolation. But the nearest neighbor considerably degrades the image, so it is not enough to watch the spectra only. The signal-to-noise ratio between the original and the reconstructed image also has to be taken into consideration. This ratio is defined as follows

$$PSNR = 20\log_{10}\frac{255}{\sqrt{MSE}} \tag{5}$$

where MSE is

$$MSE = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}\left[I(x,y) - I'(x,y)\right]^2 \tag{6}$$

In formula (6) M and N are are the size of the image I, while I' is the reconstructed image.

Table 2
Signal-to-noise ratio in decibels of various test images

	Nearest neighbor	Linear	Cubic
Lena	24.14	26.77	28.44
Cameraman	22.37	24.70	26.23
Clock	24.98	27.35	29.36
Moon	27.44	31.07	34.39
Noise	8.99	11.46	11.80

As can be seen, the nearest neighbor gives the smallest PSNR, and the cubic method gives the highest. However, the nearest neighbor preserves a fair amount of high frequency content of the image, because of the low signal-to noise ratio the quality of the resulting image is very modest. Based on empirical results it can be stated that in order to gain an acceptable visual quality, the PSNR has to be at least 25-30dB (higher is better). If there are no pressing quick calculation times, it is well-worth performing cubic convolution.

Conclusion

In this paper three interpolation techniques are compared both in spatial and in the freqency domain. The formula for the cubic convolution case was derived and suggestions were made concerning how to decide the value of the changable parameter a. In the experimental part the spectra of the images were calculated and the PSNR of the different methods compared after shrinking and expanding the images.

References

[1] Randy Crane: *A Simplified Approach to Image Processing-classical and modern techniques in C*, Prentice Hall PTR, Upper Saddle River, New Jersey 07458

[2] T. Lehmann, C. Gonner, K. Spitzer: *Survey: Interpolation Methods in Medical Image Processing*, IEEE Trans. Medical Imaging, vol. 18, pp. 1049-1075, 1999

[3] Erik H. Meijering, Karel J. Zuiderveld, Max A. Viergever: *Image Reconstruction by Convolution with Symmetrical Piecewise* n-*th Order Polynomial Kernels,* IEEE Transactions on Image Processing, Vol. 8, No. 2, February 1999

[4] Michael Unser, Akram Aldroubi, and Murray Eden: *Fast B-spline Transforms for Continuous Image Representation and Intrpolation,* IEEE Trans. on Patt. An. and Mach. Intell., vol. 13, no. 3, March 1991, pp. 277-285

[5] Miodrag Popović: *Digital signal processing (Digitalna obrada signala)*, Nauka, Beograd, 1994

[6] Robert G. Keys: *Cubic Convolution Interpolation for Digital Image Processing*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-29, No. 6, December 1981, pp.1153-1160

[7] George Wolberg: *Digital Image Warping*, IEEE Computer, Society Press, Los Alamitos, California, 1990

[8] Rafael C. Gonzalez, Richard E. Woods: *Digital Image Processing*, Prentice Hall, 2002

[9] Erik Meijering: *A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing*, Proceedings of the IEEE, Volume 90, Issue 3, Mar 2002/

[10] D. P. Mitchell, A. N. Netravali: *Reconstruction Filters in Computer Graphics*, Comp, Graph., vol. 22., no. 4, pp. 221-228, 1988

[11] Einar Maeland: *On the Comparison of Interpolation Methods*, IEEE Transactions on Medical Imaging, Vol. 7., No. 6., September 1988, pp. 213-217