High resolution Image Interpolation Using 2D Spline Model

[1] *JAYATI DOSHI*, [2] *DR. Y. P. KOSTA*, [3]*ASSISTANT PROFESSOR SUNERA KARGATHARA*

[1, 2, 3] **Department Of Electronics and communication Engineering, Marwadi Education Foundation Group of Institutes, Gujarat Technological University, Gujarat, India.**

ec.jayati@gmail.com, ypkosta@gmail.com, kargatharasunera@gmail.com

ABSTRACT: Image interpolation to higher resolution is an important feature in many digital imaging applications. Practical applications require scaling of images at any fractional values. This paper describes the use of B-spline interpolation for resizing of an image, may it be zooming or compression at any fractional values. First the problem of determining the B-spline coefficients for the exact signal interpolation is described. The classical approach for 1D spline is extended to 2D spline. The equation for 2D spline interpolation and the algorithm for applying it to the image is described in the paper. The simulation results for cubic spline interpolation are shown in the paper.

KEYWORDS: ***Interpolation method; image zooming; compression*.**

1. INTRODUCTION

Digital image processing has shown a huge growth in past few decades. Its applications in various fields have increased to a great extent in the field such as digital telecommunication, internet broadcasting, medical imaging etc. Among all the processing techniques that are applied on a digital image, image resizing is one of the important techniques. It is indispensable because it affects the bandwidth and the storage requirements tremendously. It is a processes that introduces (or eliminates) the pixels to (or from) the original image.

The simplest and most commonly used technique is the replication method. Although this method is fast and simple it produces very poor result if the expansion factor is greater than three [3]. In any resizing technique distortion in the original image is unavoidable. But we can minimize it using different techniques.

One idea to resize (especially to compression) an image or a signal is to convert it from the digital domain to continuous domain and then apply the necessary scaling.

The rest of the manuscript is divided as follows: first the basic algorithm for image resizing is discussed in section. It is followed by the spline mathematics and physical interpretation. Then the proposed 2D spline algorithm is described and at last the simulation results are shown.

2 BASIC ALGORITHM FOR IMAGE RESIZING

Standard process of resizing is to fit the data with continuous image model and then resample this function on the gird appropriate to the desired scaling. Applying filter prior to sampling can give more accurate results. But it is difficult to implement the optimal projection pre-filter when sinc or higher order splines are used.

The performance of an algorithm is decided by the model’s ability to reproduce up to specified degree n [4,5]. Higher order interpolation methods work well for image magnification and rotation, but it causes aliasing in image reduction.

For projection based image resizing, the flow of the algorithm can be given as: -

1. Fit the discrete data with the continuous function using interpolation.
2. Scale the continuous function.
3. Finally, resample the function at integers.

One of the fundamental issues in signal processing is switching between the continuous and discrete signal domain. It is the question that arises during the acquisition process where the analog signal is converted to a series of numbers and similarly during the representation in continuous domain. One approach to these problems is provided by Shannon's sampling theory. But there are four major limitations to this approach:

1. It relies on the use of ideal filters.
2. The band limited hypothesis which is in contradiction to the idea of finite duration signal.
3. The band limiting operation tends to generate Gibbs oscillations.
4. The basis function, sinc(x), has a very slow decay, which makes computation in signal domain very inefficient.

An alternative approach to problem of switching between the continuous and discrete domain is by using Splines. The detailed description about spline interpolation is given in the next section.

3. SPLINE MATHEMATICS AND PHYSICAL INTERPRETATION

Splines are piecewise polynomials that are connected smoothly to each other. The joining points are known as *knots*. For n^{th} order spline, n+1 coefficients are required for each segment of the curve. But we need to impose additional constraints for smoothness purpose, i.e. continuity of spline and its derivatives up to the order (n-1). Thus there is only one degree of freedom per segment.

B-spline expansion for any signal s(x) can be given as follows:

$$S(x) = \sum_{k \in Z} C(k)\beta^n(x-k) \quad (1)$$

which involves the integer shift of the central B-spline of degree *n* denoted by $\beta^n(x-k)$ (shown in figure 1); *c(k)* are the parameters of the model. B-spline is a bell shaped function. It is constructed by (n+1) fold convolution of rectangular pulse β^0:

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & otherwise \end{cases} \quad (2)$$

$$\beta^n = \beta^0 * \beta^0 * \dots * \beta^0(x) \quad (3)$$

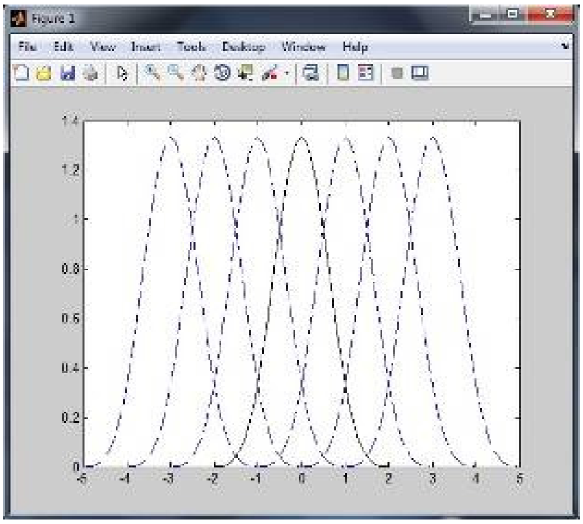

Figure 1 : Integer shift of central cubic spline

B-spline of degree 0 to 5 are shown in the figure 2. As each B-spline is characterized by sequence of coefficients *C(k)* which is convenient structure for discrete signal even though the spline model is continuous.

It is very easy to use B-spline for mathematical calculations because derivative and integral of spline can be calculated very easily,

$$\frac{d\beta^n}{dx} = \beta^{n-1}\left(x+\frac{1}{2}\right) - \beta^{n-1}\left(x-\frac{1}{2}\right) \quad (4)$$

$$\int_{-\infty}^{x} \beta^n(x)dx = \sum_{k=0}^{+\infty} \beta^{n+1}\left(x-\frac{1}{2}-k\right) \quad (5)$$

Once these basic operations are known, manipulation of mathematical equations involving splines is very easy.

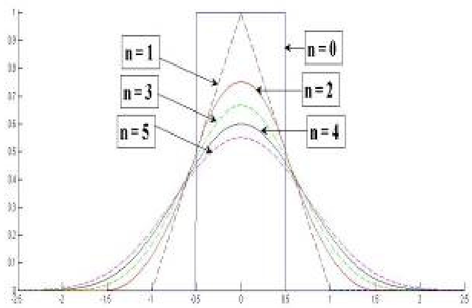

Figure 2 Spline basis function

Now we want the spline model such that it should be the perfect fit for the integers. So for the given samples of the signal *S(k)*, the spline coefficients *C(k)* can be given as

$$\sum_{k\in Z} C(k)\beta^n(x-k)|_{x=k} = S(k) \tag{6}$$

This equation can be re-written in the form of convolution as

$$S(k) = (b_1^n * C)(k) \tag{7}$$

Where b_m^n is the discrete B-spline kernel defined as

$$b_m^n(k) = \beta^n(^x/_m)|_{x=k} \tag{8}$$

Now, defining the inverse convolution operator

$$(b_m^n)^{-1}(k) \overset{z}{\leftrightarrow} {}^1/_{B_1^n(z)} \tag{9}$$

the solution is found by inverse filtering [6]

$$C(k) = (b_1^n)^{-1} * S(k) \tag{10}$$

Here the B-spline filter $(b_1^n)^{-1}$ is an all pole system. It can be implemented by using cascade of first order causal and anti-causal recursive filters [7].

4. PROPOSED ALGORITHM FOR IMAGE INTERPOLATION

Generally, whenever an *n* dimensional signal is to be interpolated, 1D spline is used. 1D spline is discussed in the section III. This equation is applied to each dimension one after another. But this is very time consuming.

Instead of this approach, we apply 2D spline directly to the image. The equation for this is given as

$$S(x,y) = \sum_{m\in Z}\sum_{n\in Z} c(m,n)\beta^n(x-m)\beta^n(y-n) \tag{11}$$

where S(x,y) represents the pixel value, C(m,n) is the parameters of the B-spline model while $\beta^n(x-m)$ and $\beta^n(y-n)$ are the shifted central B-spline of degree *n*.

Using this equation for the calculation and development of the algorithm, results are same as those obtained using 1D spline, but the processing time is reduced.

Algorithm for interpolating an image using this equation can be given as follows: -

1. Make the necessary assumptions for the boundary conditions based on the spline model intended to use.
2. Calculate the spline coefficients C(m,n) for all m,n ϵ Z using the known values of S(x,y).
3. Decide the scaling factor.
4. Based on the scaling factor, calculate the values S(x',y').

Here, x and y belongs to the set of real numbers. So it can take any real value. Thus scaling can be done for any fractional value as well. Also image compression can be achieved along with the zooming with good results.

This interpolation can also be viewed as interpolation via filtering technique. Here

Results of image resizing (both zooming and compression) are shown in the next section.

5. SIMULATION RESULTS

Simulation result for curve interpolation using cubic spline is shown in figure 3. For the sake of comparison, we have shown the nearest neighbor interpolation along with the cubic spline interpolation. The curve between the two points represents the cubic spline interpolation whereas the straight line joining the two points represents the nearest neighbor interpolation.

Results of zooming of images using cubic spline interpolation are shown below. Figure 4 shows the zooming for gray scale images awhile the figure 5 shows the zooming for colored images. Here the zooming factor is 5x.

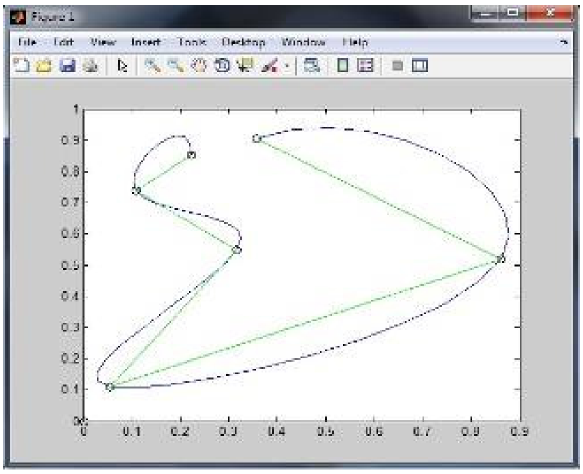

Figure 3 Cubic Spline v/s Nearest neighbor interpolation

Figure 4: Gray scale image zooming

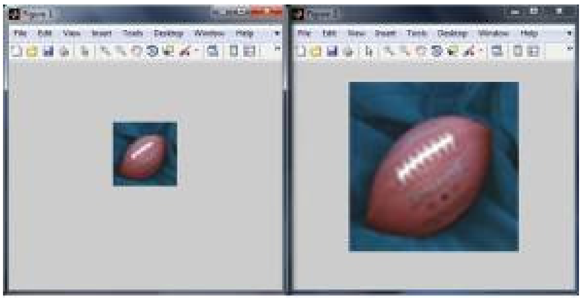

Figure 5: Colored Image zooming

Results for the compression of the image are show in the figures below. Figure 6 shows the compression of the gray scale image by 1.5 scaling factor. And figure 7 represents the compression of colored image by 1.5 scaling factor. It can be observed that the results of image compression are also very faithful. Thus spline can be used for high quality image resizing.

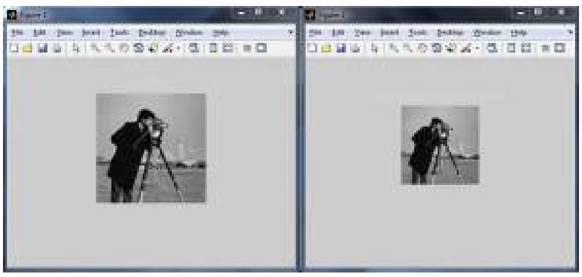

Figure 6 Gray sale image compression

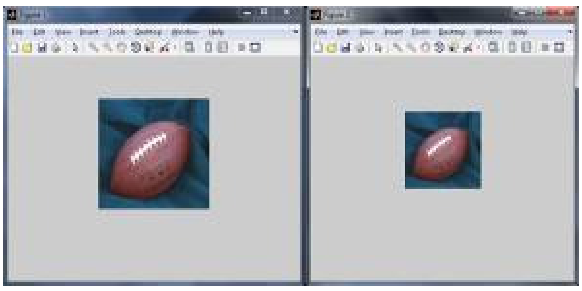

Figure 7 Colored image compression

6. CONCLUSION

Spline interpolation can be used for image resizing for practical applications as it is possible to have fractional scaling factor. The proposed algorithm for 2D spline provides same result as 1D spline but the processing time is reduced. It can also be seen from the spline model that it provides an efficient bridge between continuous and discrete domain.

5. REFERENCES

[1] G. Strang, *Linear algebra and its applications*, Cengage learning, 2009

[2] R. C. Gonzalez and R. E. Woods, Digital image processing, Pearson, 2009

[3] .M. Unser, A. Aldroubi, M. Eden, “Enlargement or Reduction of Digital Images With Minimum Loss of Information”, IEEE Trans. on Image Processing, Vol. 4, No. 3, March 1995, pp. 247-258.

[4] G. Strang and G. Fix, “A Fourier analysis of the finite element variational method,” in *Constructive Aspect of Functional Analysis* ,Rome, Italy; Edizioni Cremonese, pp 796-830, 1971

[5] C. de Boor, “Quasiinterpolation and approximation power of multivariate splines,” in *Computation of Curves and Surfaces,* W. Dahmen, Ed. Boston, MA: Kluwer, pp 313-345, 1990

[6] Unser, M.; Aldroubi, A.; Eden, M., "B-spline signal processing. I. Theory," *Signal Processing, IEEE Transactions on* , vol.41, no.2, pp.821,833 , Feb 1993.

[7] M. Unser, A. Aldroubi, and M. Eden, “Fast B-spline transforms for continuous image representation md interpolation,”lEEE Tran. Pattern Anal. Machine Intell., vol. 13, no. 3, pp. 277-285, 1991

[8] Unser, M.; Aldroubi, A.; Eden, M., "B-spline signal processing. II. Efficiency design and applications," *Signal Processing, IEEE Transactions on* , vol.41, no.2, pp.834,848, Feb 1993

[9] Matiu-Iovan, L., "Some aspects of implementing a cubic spline interpolation algorithm on a DSP," *Electronics and Telecommunications (ISETC), 2012 10th International Symposium on* , vol., no., pp.291,294, 15-16 Nov. 2012

[10] Matiu-Iovan, L.; Frigura-Iliasa, F.M., "Algorithms of cubic B-spline interpolation extended for m>2," *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*, vol., no., pp.1,4, 18-20 July 2012