

Optimization of PSNR of Zooming Image using Cubic Spline Interpolation Method

Ankita Agrawal, Arpan Shah, Ramesh Bharti

Abstract— In this paper, we have presented the cubic spline interpolation method for zooming in an image. The cubic spline algorithm is programmed in MATLAB as m-file. Firstly, according to the zooming intensity, sufficient zeroes are inserted between each pixel and then those zeroes are replaced by the respective values generated by the m-file. Finally a zoomed in image is constructed.

Index Terms— Digital image processing, zoom in, cubic spline interpolation method, MATLAB.

$$S_i(x) = \begin{cases} S_1(x) & \text{if } x_1 < x < x_2 \\ S_2(x) & \text{if } x_2 < x < x_3 \\ \vdots & \vdots \\ S_{n-1}(x) & \text{if } x_{n-1} < x < x_n \end{cases}$$

I. INTRODUCTION

The field of digital image processing refers to processing digital images using digital computers. A digital image is comprised of finite number of values each of which has a location and a value. Generally speaking these values are referred to as pixels. Using MATLAB, this process becomes a bit easier as the principles involved in MATLAB for matrix manipulations can easily be used in image processing. One of the major applications of image processing is object or face recognition, in addition to which zooming in and zooming out an image is also a very important field of interest. In this paper, we are zooming in an image using cubic spline interpolation method. We programmed the method in MATLAB as m-file and tested it on 'cameraman.tif'.

Cubic Spline Interpolation Method: This method relies on constructing a polynomial of low degree between each pair of known data points. If a first degree polynomial is used, it is called linear interpolation. For second and third degree polynomials, it is called quadratic and cubic splines respectively. The higher the degree of the spline, the smoother we get the curve. To obtain a smoother curve (or finer zooming), cubic splines are frequently recommended. The fundamental idea behind cubic spline interpolation is based drawing smoother curves through a number of points. This spline consists of weights attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights, resulting in a pleasingly smooth curve. A cubic polynomial is the minimum order polynomial that guarantees the generation of C_0, C_1, C_2 , curves. Therefore, it is the best known methods resort to piecewise cubic curve constructed by the individual third degree polynomials assigned to each subinterval.

Mathematical Analysis of Cubic Spline Interpolation Process: The essential idea is to fit a piecewise function of the form

where $S_i(x)$ is a third degree polynomial defined by $S_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i$ (1)
 x_i are the data points.

and $i = 1, 2, 3,$

The first and second derivatives of eq. (1) are given by

$$S_i'(x) = 3a_i(x-x_i)^2 + 2b_i(x-x_i) + c_i \quad (2)$$

$$S_i''(x) = 6a_i(x-x_i) + 2b_i \quad (3)$$

where i being same as $i = 1, 2, \dots, n-1$

Our spline will need to conform to the following stipulations.

1. The piecewise function $S_i(x)$ will interpolate all data points.
2. $S_i(x)$ will be continuous on the interval $[x_1, x_n]$
3. $S_i'(x)$ will be continuous on the interval $[x_1, x_n]$
4. $S_i''(x)$ will be continuous on the interval $[x_1, x_n]$

Following the stipulations, we deduce the formulas for the coefficients as

$$a_i = \frac{M_{i+1} - M_i}{6h} \quad (4)$$

$$b_i = \frac{M_i}{2} \quad (5)$$

$$c_i = \frac{y_{i+1} - y_i}{h} - \left(\frac{M_{i+1} + 2M_i}{6}\right)h \quad (6)$$

$$d_i = y_i \quad (7)$$

y_i is the pixel color intensity at data point x_i

After calculating the values of M_i using eq.(8), we calculate the values of a_i, b_i, c_i and d_i for each pair of adjacent pixel and calculate the respective values for the zeroes filled in between them. The formula for filling the values in their respective positions was generated by hit and trial method. Now after constructing the image matrix and using the 'imshow' command we get the zoomed image.

For cubic spline, we calculate the values of M_i in terms of y_i using eq.(8) and after calculating the values M_2, M_3, \dots, M_{n-1} , we find the values of M_1 and M_n using eq.(9) and eq. (10)

Manuscript received June 20, 2014.

Ankita Agrawal, B-tech-M-tech Integrated scholar, Jagan Nath University

Arpan Shah, Asst. Professor, Jagan Nath University

Ramesh Bharti, Asst. Professor, Jagan Nath University

$$\begin{pmatrix} 6 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & 4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 4 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} M_1 \\ M_3 \\ M_4 \\ \vdots \\ M_{n-3} \\ M_{n-2} \\ M_{n-1} \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} y_1 - 2y_2 + y_3 \\ y_2 - 2y_3 + y_4 \\ y_3 - 2y_4 + y_5 \\ \vdots \\ y_{n-4} - 2y_{n-3} + y_{n-2} \\ y_{n-3} - 2y_{n-2} + y_{n-1} \\ y_{n-2} - 2y_{n-1} + y_n \end{pmatrix} \quad (8)$$

[6] M.-J. Chen, C.H. Huang, W.L. Lee, "A fast edge-oriented algorithm for image interpolation", Image and Vision Computing, Vol. 23, Issue 9, 1 September 2005, pp. 791-798.
 [7] WANG Jiechao et al., "Two New Proposed Image Zooming Methods", International School Beijing University of Posts and Telecommunications Beijing, China.
 [8] Hakran Kim et al., "Curvature Interpolation Method for Image Zooming", IEEE transactions on image processing, Vol. 20, No. 7, July 2011.

Results: Taking 'cameraman.tif' as sample grey scale image and the zooming intensity as 2, we can compare the results as the psnr value was found to be 148.48 and the execution time was 384.1070 seconds.



Fig. 1 A part of cameraman.tif of size 50x50



Fig. 2. 4x Zoomed in image of size 200x200

So, we found that the zoomed image is much clear as compared to normally available zooming applications.

REFERENCES:

[1] Digital Image Processing using MATLAB by R.C. Gonzalez, R.E. Woods and S.L. Eddins.
 [2] Cubic Spline Interpolation by Sky McKinley and Megan Levine
 [3] J. M. Almira and A. E. Romero, "Image zooming based on sampling theorems", vol. 2011, no. 1, (2011), pp. 22, ISSN: 1887-1097.
 [4] D. M. Monro, P. D. Wakefield, "Zooming with Implicit Fractals", International conference on image processing, 26-29, Oct, CA, USA, pp. 913-916.
 [5] S. S. Rifman, "Digital rectification of ERTS multispectral imagery", in Proc. Symp. significant Results Obtained from ERTS-I (NASA SP-327), Vol. 1, sec. B, 1973, pp. 1131-1142.