

A RESCURSIVE CUBIC SPLINE INTERPOLATION METHOD FOR THE NOISE REMOVAL IN IMAGE PROCESSING

N. GAJALAKSHMI ¹, S. KARUNANITHI, AND M. MALARVIZHI

ABSTRACT. In this paper, we proposed a recursive cubic spline interpolation method for image processing using Extended Kalman Filter techniques. This recursive cubic spline interpolation is widely used in image (signal) information compression to zoom in or out to correct spatial distortions. The recursive cubic spline interpolation method is applied to interpolate the numerical information between existing known environments and extend them in the centralized system. We proceed in the same way towards the unknown environment and extend it into the distributed system. The cubic spline recursive interpolation method (algorithm) detects impulse noise in the image. Noise and noise free images are detected by changing the value of the pixel element relative to the maximum and minimum gray value of an image. If the processing pixel is different from 0 or 255, the pixel is free from noise. Otherwise, the processing pixel is noisy, which is processed by the recursive cubic spline interpolation method. We use edge cutting techniques, histogram analyzer and matrix sizing techniques that act as image processors. Based on the edge cutting techniques, the noise terms are identified and controlled (minimized) through the recursive cubic spline interpolation method. The extended kalman filter techniques proposed are further applied to extend the size of the image matrix to be processed. The recursive cubic spline interpolation method eliminates the noise of salt and pepper and preserves smoothness of the original image. The proposed recursive cubic spline interpolation method is based on the neighboring noise-free pixels and the noise-free output pixels as indicated above.

¹*Corresponding author*

2010 *Mathematics Subject Classification.* 65D18, 68U10.

Key words and phrases. Edge cut, Histogram analyzer, kalman filter, array dimensions, recursive cubic spline interpolation method.

1. INTRODUCTION

In mathematics, the term spline is received from the name of an adaptable segment, generally utilized by drafters to help with drawing bended lines. Here, a spline is numerical function that is piecewise characterized by polynomial functions, and which has a high degree of smoothness at the points where the polynomial parts are connected (which are known as nodes) [3].

Interpolation is one of the most significant functions that can be utilized in the process of estimating the intermediate values of a set of known discrete sampling points [9, 14]. Interpolation is widely used in image processing and getting informations from the image and compressing the image to amplify or diminish and to address the images [6, 15]. Interpolations can be classified into two main categories, global and partial interpolations [8, 10]. Global interpolations are depending on building a single equation that fits all information points [12]. This equation is generally a high-grade polynomial equation. In spite of the fact that these interpolations result in smooth curves, they are generally not appropriate for engineering applications, since they are subject to strong oscillations and overcoming in intermediate points. Partial interpolations are based on the construction of a low-degree polynomial between each pair of known information points [2, 7]. The greater and the smooth curve obtained with the increase in the degree of stretch marks. To obtain a smooth curve, cubic splines are as often as possible recommended [5, 11]. This methodology consists in adapting the cubic polynomials and pair of adjacent points [1, 4].

This paper is composed as follows, in section 2, the basic ideas related to cubic spline interpolation method are discussed. We implemented recursive cubic spline interpolation method in section 3. We extended this method to denoise the image through some image filtering techniques like Extended Kalman filter and numerical illustration are shown in this section. Finally, section 4 concludes the paper.

2. SPLINE INTERPOLATION

Interpolation is the process of defining a function that accepts specific values at specific points. These points are used to generate lower power polynomials that approach a complicated function during the interval $[a, b]$. The number of point is connected through smooth curves and is interpolated by the weights

which are the coefficients in the cubic polynomials. The necessary concept behind this is to adapt any function by parts as indicated in the following equation,

$$(2.1) \quad F(x) = \begin{cases} f_1(x) & \text{if } x_1 \leq x \leq x_2 \\ f_2(x) & \text{if } x_2 \leq x \leq x_3 \\ \dots & \\ f_{n-1}(x) & \text{if } x_{n-1} \leq x \leq x_n \end{cases},$$

where $f_i(x)$ is a cubic polynomial function which is characterized in the following equation

$$(2.2) \quad f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i,$$

where a_i, b_i, c_i and d_i are unknown constants. The number ' n ' of known information points that have intervals ' $n - 1$ ', is shown in the Figure 1. The unknown

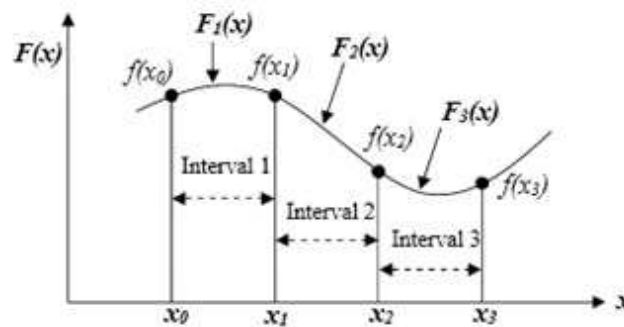


FIGURE 1. Piecewise cubic spline interpolation

constants are proceeding with the following steps:

- (1) The values of the function should be the same in the internal nodes ($n - 1$).
- (2) The first and last functions must pass through the two end points x_0 and x_n .
- (3) The first derivatives must be the same in internal points $n - 1$.
- (4) The second derivatives equal to the internal nodes $n - 1$.

Natural spline assumes that the second derivatives at the end nodes (points) are zero (i.e., $f''_i(x_0) = f''_i(x_n) = 0$). The cubic spline in the interval (x_{i-1}, x_i) is

given by the equation (2.3),

$$(2.3) \quad F_i(x) = \frac{f''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f''(x_i)}{6(x_i - x_{i-1})}(x - x_i)^3 + \left[\frac{f(x_{i-1})}{(x_i - x_{i-1})} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6} \right] (x_i - x) + \left[\frac{f(x_i)}{(x_i - x_{i-1})} - \frac{f''(x_i)(x_i - x_{i-1})}{6} \right] (x - x_{i-1}).$$

Equation (2.3) contains only two unknowns $f''(x_i)$ and $f''(x_{i-1})$, the second derivatives at the end points of the interval $(x_i - x_{i-1})$ which are determined in equation (2.4).

$$(2.4) \quad \begin{aligned} & (x_i - x_{i-1})f(x_{i-1}) + 2(x_{i+1} - x_{i-1})f(x_i) + (x_{i+1} - x_i)f(x_{i+1}) \\ &= \frac{6}{(x_{i+1} - x)}[f(x_{i+1}) - f(x_i)] \\ &+ \frac{6}{(x_i - x_{i-1})}[f(x_{i-1}) - f(x_i)]. \end{aligned}$$

Equation (2.4) is written for $(n - 1)$ internal points which generate $(n - 1)$ simultaneous equations for second derivatives.

3. RECURSIVE CUBIC SPLINE INTERPOLATION METHOD

Spline interpolation is a useful technique for interpolation between known informations and dynamical coefficients. These two functions are used to eliminate the noise of salt and pepper present in the noisy image. Salty and pepper noise can take on gray levels 0s and 255s, but the random noise impulse can take on any value between 0 and 255. This paper describes the removal of salty and pepper noise in the image using the filter recursive spline interpolation.

A spline-based methodology is proposed to eliminate the noise of very high density of salt and pepper in color and grayscale images. The algorithm consists of two phases, the first phase detects if the pixel is noisy or noiseless. The second stage removes the noisy pixel using the recursive cubic spline interpolation method. The proposed recursive cubic spline interpolation method is based on neighboring noiseless pixels and previous noiseless output pixels.

3.1. Problem Formulation. The initial step of the proposed algorithm recognizes impulse noise in the image. If the processing pixel is different from "0" or "255", and it is free from noise. Otherwise, the processing pixel is noisy, which is processed by the method recursive cubic spline interpolation method (RCSIM). The steps of RSIF are explained as follows:

Step 1: Consider a 3×3 matrix window, the processing pixel considered is P_{ij} .

Step 2: If $0 < P_{ij} < 255$ then P_{ij} is noise-free pixel and its value remains unaltered.

Step 3: If $P_{ij} = 0$ or 255 then P_{ij} is a noisy pixel, which is processed as follows:

Step 3a: If they chose window contains all the elements such as 0's and 255's, the processing pixel (P_{ij}) is replaced by the average value of the 3×3 window.

Step 3b: If they chose window does not contain all elements such as 0's and 255's, and apply the RCSIM algorithm to the remaining pixels. The average value of the RCSIM replaces the noisy pixel.

Step 4: Repeat steps 1 to 3 until all the pixels in the entire image.

Recursive cubic spline interpolation method equation is given in (3.1)

$$(3.1) \quad y = \begin{cases} x & \text{if } 0 < x < 255 \\ F(x) & \text{otherwise,} \end{cases}$$

where $F(x)$ is the output of the spline interpolation filter and x is the gray value of the input image.

3.2. Filtering Techniques. The filter plays a vital role in many common applications, such as applications power supplies, radio communications and so on. Digital filters are often much higher order and are finite impulse response filters that allow a linear phase response. The filter helps to minimize glare and reflections, improves colors, and reduces light entering the lens and more. Each lens filter has a specific purpose, and is designed to offer a specific effect that can help to improve the image.

Extended Kalman filter (EKF):

EKF works by changing nonlinear models with each step into a linear system of equations. In a single variable value model and its derivative; the generalization for multiple variables and equations is the Jacobin matrix. Linearized equations are used similarly to the standard kalman filter.

As in many cases approximating a nonlinear system with the linear model, EKF will not work well. On the contrary to the KF standard for linear systems, the EKF is not tested as an option in any way; it is simply an extension of the linear system technique in a wider class of problems.

Histogram Analyzers:

The histogram is utilized to graphically summarize and move the distribution and to process the information set. Histogram analysis tool calculates the histogram information from the original and modified image.

Illustration of Recursive Cubic Spline Interpolation Method

From the following image, we get the pixel values,



FIGURE 2. Sample Image to process the noise

Every single pixel in the image is checked for the noise.

$$\begin{bmatrix} 100 & 125 & 112 \\ 120 & 130 & 116 \\ 103 & 118 & 143 \end{bmatrix} \rightarrow \begin{bmatrix} 100 & 125 & 112 \\ 120 & 130 & 116 \\ 103 & 118 & 143 \end{bmatrix}$$

If the rendering pixel in the selected window contains a noise-free pixel, no further process is required. For example, "130" is the noise-free processing pixel.

Therefore, it is unaltered. Then,

$$\begin{bmatrix} 149 & 47 & 88 \\ 61 & 0 & 255 \\ 3 & 0 & 255 \end{bmatrix}.$$

This pixel values are arranged into one dimensional format

$$\begin{array}{cccccc} X & 0 & 1 & 2 & 3 & 4 \\ F(x) & 149 & 47 & 88 & 61 & 3 \end{array}$$

Here $h = 4, h + 1 = 5, h_0 = h_1 = h_2 = h_3 = h_4 = 1$, to find $x = 2.5$. Recursive Cubic Spline interpolation is

$$\begin{aligned} F_i(x) &= \frac{f''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f''(x_i)}{6(x_i - x_{i-1})}(x - x_{i-1})^3 \\ &+ \left[\frac{f(x_{i-1})}{(x_i - x_{i-1})} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6} \right] (x_i - x) \\ &+ \left[\frac{f(x_i)}{(x_i - x_{i-1})} - \frac{f''(x_i)(x_i - x_{i-1})}{6} \right] (x - x_{i-1}). \end{aligned}$$

Interpolate at 2.5, here $x_i = 3, x_{i-1} = 2, x_{i+1} = 4$,

$$(3.2) \quad F(2.5) = 74.5 - f''(2)(0.0625) - f''(3)(0.0625).$$

To find $f''(2)$ and $f''(3)$, we get $f''(2) + 4f''(3) + f''(4) = -186$. The calculated value of $f''(4) = 0$,

$$(3.3) \quad f''(2) + 4f''(3) = -186,$$

$$(3.4) \quad f''(1) + 4f''(2) + f''(3) = -408,$$

and hence $f''(0) = 0$,

$$(3.5) \quad 4f''(1) + f''(2) = 858.$$

Solving the simultaneous equations (3.3) to (3.5) and we obtain the $f''(2)$ and $f''(3)$ as

$$\begin{bmatrix} 0 & 1 & 4 \\ 1 & 4 & 1 \\ 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} f''(1) \\ f''(2) \\ f''(3) \end{bmatrix} = \begin{bmatrix} -186 \\ -408 \\ 858 \end{bmatrix}.$$

By row reduction method, $f''(1) = 246.78, f''(2) = -64.5, f''(3) = -5.36$, and from (3.2), we have $F(2.5) = 84.45$.

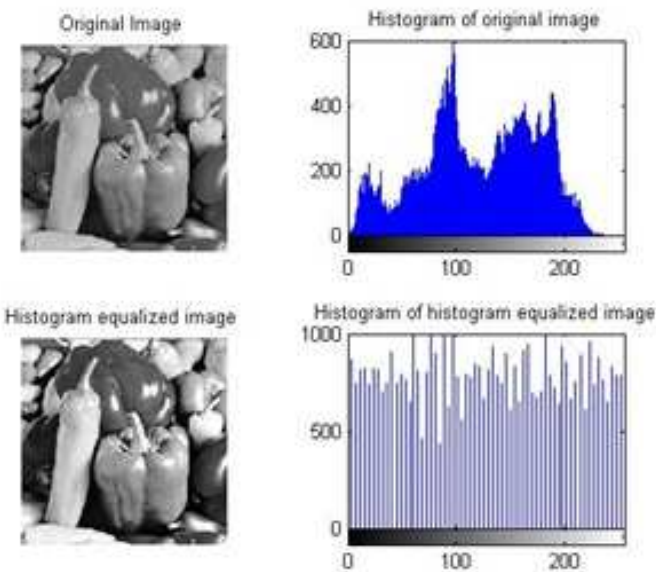


FIGURE 3. Histogram equalized image, Histogram of histogram equalized image of figure 2

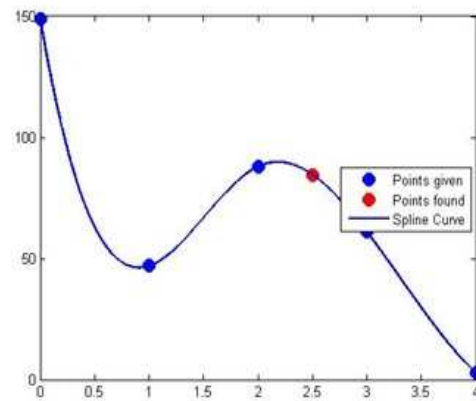


FIGURE 4. Interpolation value at 2.5

From the plot we interpolate at 2.5, we arrived $F(2.5)$ as 84.5625, this shows that similar values are arrived during the interpolation process and processing the corresponding image pixel as an input.

Illustration 2

Consider the following sample image.



FIGURE 5. Sample image to process the noise

From the image we get the pixel values

$$\begin{bmatrix} 162 & 67 & 111 \\ 75 & 0 & 155 \\ 5 & 0 & 255 \end{bmatrix}$$

This pixel values are arranged into one dimensional format

$$\begin{array}{c|cccccc} X & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline F(x) & 162 & 67 & 111 & 75 & 155 & 5 \end{array}$$

Here $h = 5, n + 1 = 6, h_0 = h_1 = h_2 = h_3 = h_4 = 1$, to find $x = 3.5$. Recursive Cubic Spline interpolation is:

$$\begin{aligned} F_i(x) &= \frac{f''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f''(x_i)}{6(x_i - x_{i-1})}(x - x_{i-1})^3 \\ &+ \left[\frac{f(x_{i-1})}{(x_i - x_{i-1})} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6} \right] (x_i - x) \\ &+ \left[\frac{f(x_i)}{(x_i - x_{i-1})} - \frac{f''(x_i)(x_i - x_{i-1})}{6} \right] (x - x_{i-1}). \end{aligned}$$

Interpolate at 3.5, here $x_i = 4, x_{i-1} = 3, x_{i+1} = 5$:

$$(3.6) \quad F(3.5) = 115 - 0.0625f''(3) - f''(4)(0.0625).$$

To find $f''(3)$ & $f''(4)$, we get,

$$(3.7) \quad f''(3) + 4f''(4) + f''(5) = -1380.$$

In a similar way calculate, $f''(5) = 0$,

$$(3.8) \quad f''(2) + 4f''(3) + f''(4) = -696,$$

$$(3.9) \quad f''(1) + 4f''(2) + f''(3) = -480.$$

Proceeding further we get, $f''(0) = 0$, therefore,

$$(3.10) \quad 4f''(1) + f''(2) = 834.$$

Solving the simultaneous equations (3.7) to (3.10) and obtain $f''(3)$ and $f''(4)$:

$$\begin{bmatrix} 0 & 0 & 1 & 4 \\ 0 & 1 & 4 & 1 \\ 1 & 4 & 1 & 0 \\ 4 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f''(1) \\ f''(2) \\ f''(3) \\ f''(4) \end{bmatrix} = \begin{bmatrix} -1380 \\ 696 \\ -480 \\ 834 \end{bmatrix}.$$

Solving for $f''(1)$, $f''(2)$, $f''(3)$ and $f''(4)$, we get,

$$f''(1) = 277.84, f''(2) = -277.35, f''(3) = 351.56, f''(4) = -432.89.$$

Equation (3.6) becomes, $F(3.5) = 111.3625$.

From the original image, we use the histogram analyzer to get the histogram of original image, then the histogram image is denoised into the histogram equalized image. Again we denoise to the histogram of the histogram equalized image. Simulation results are arrived through MATLAB.

4. CONCLUSION

The simulated results shows in this paper prove that recursive cubic spline interpolation method gives better denoised the image quality at high density images, and we ensure that the smoothness of the original image is retained. Also we compared the original and noisy image with histogram analyzer method. Further we interpolated the corresponding pixel values through recursive cubic spline interpolation method. The stable and smooth characteristics of interpolation function is to retain the smoothen area of the original image.

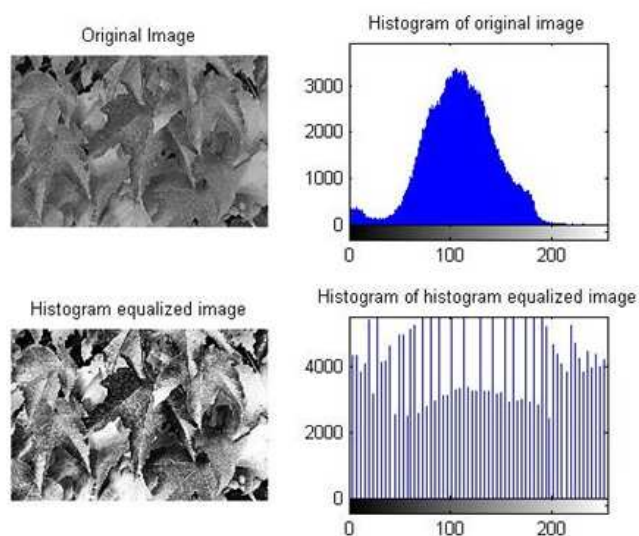


FIGURE 6. Histogram equalized image, Histogram of histogram equalized image of figure 5

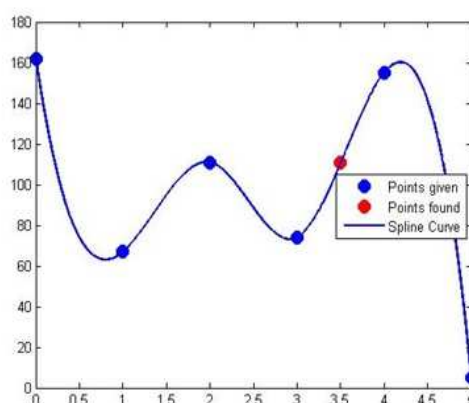


FIGURE 7. Interpolation value at 3.5

REFERENCES

- [1] C.D.A. BOOR: *Practical guide to splines*, Berlin, Germany, Springer-Verlag, 1978.
- [2] B. CHANDA B, D.D. MAJUMDER: *Digital Image processing and analysis*, Prentice Hall of India, 1st edition, 2002.
- [3] S.A. DYER, J.S. DYER: *Cubic spline Interpolation, Part 1*, IEEE Instrumentation and measurement magazine, (1) (2001), 44-46.

- [4] M. UNSER: *Splines – A Perfect fit for Signal and Image Processing*, IEEE Signal Processing Magazine, **16**(6) (1999), 22-38.
- [5] T. VEERAKUMAR, ESAKIKIRAJAN, I. VENNILA: *Recursive Cubic Spline Interpolation Filter Approach for the Removal of High density salt and pepper noise*, Signal, Image And Video Processing (Springer), **8** (2014), 159-168.
- [6] D. LANCASTER: *A Review of Some Image Pixel Interpolation Algorithms*, Synergetics, 11 pages.
- [7] R.H. CHAN, C.W. HO, M. NIKOLOVA: *Salt and pepper noise removal by median - type noise detectors and detail – Preserving regularization*, IEEE Trans.Image Process, **4**(10) (2005), 1471-1485.
- [8] I. PITAS, A.N. VENETSANOPOULOS: *Order statistics in digital image processing*, Proceedings of IEEE, **80**(12) (1992), 1893-1921.
- [9] K.S. SRINIVASAN, D. EBENEZER: *A new fast and efficient decision based algorithm for removal of high - density impulse noises*, IEEE Signal Processs. Lett., **14**(3) (2007), 189-192.
- [10] W.K. PRATT: *Digital image Processing*, John Wiley & Sons, New York, 2nd Edition, 1991.
- [11] S. JAYARAMAN, S. ESAKKIRAJAN, T. VEERAKUMAR: *Digital Image Processing*, Chap.9, Tata McGraw Hill, New Delhi (2009).
- [12] H. SPATH: *One Dimensional Spline Interpolation Algorithms*, CRC Press, 1995.
- [13] J.S. LIM: *Two Dimensional Signal and Image Processing*, Englewood Cliffs, NJ, Prentice-Hall, 1990.
- [14] I.S. REED: *Notes on Image Data Compression Using Liner Spline Interpolation*, Los Angeles, CA: Dept. Elect. Eng. Suthern Calif, 1981.
- [15] M. UNSER, A. ALDROUBI, M. EDEN: *B-Spline Signal Processing: Part II-Efficient Design and Applications*, IEEE Trans. Signal Processing, **41** (1993), 834-848.

PG & RESEARCH DEPARTMENT OF MATHEMATICS,
GOVERNMENT THIRUMAGAL MILLS COLLEGE, TAMIL NADU , INDIA.
Email address: n.gajalakshmi@rocketmail.com

PG & RESEARCH DEPARTMENT OF MATHEMATICS,
GOVERNMENT THIRUMAGAL MILLS COLLEGE, TAMIL NADU , INDIA.
Email address: kap232008@gmail.com

PG & RESEARCH DEPARTMENT OF MATHEMATICS,
GOVERNMENT THIRUMAGAL MILLS COLLEGE, TAMIL NADU , INDIA.
Email address: malar2308@gmail.com