Reprinted from

# Ninth International Symposium

# Machine Processing of

# Remotely Sensed Data

with special emphasis on

# Natural Resources Evaluation

**June 21-23, 1983**

# Proceedings

Purdue University
The Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907 USA

# AN ALGORITHM FOR INTERPOLATION OF DIGITAL IMAGERIES USING PIECE WISE HYPERSURFACE APPROXIMATION

A.D. KULKARNI

National Remote Sensing Agency
Hyderabad, India

K. SIVARAMAN

College of Engineering
Trivandrum, India

ABSTRACT

Here, an algorithm for interpolation by piece-wise polynomial approximation has been developed and implemented. An hypersurface approximation is carried out by quadratic surface defined over two dimensional space to the digital picture in the neighbourhood of point to be interpolated using orthogonal polynomials as basis functions. Kernel for interpolation has been evaluated with the help of polynomial basis functions, which enables efficient implementation of the algorithm.

## I. INTRODUCTION

Interpolation is a process of estimating intermediate values of a continuous event from discrete samples. Interpolation is used extensively in processing remotely sensed data for correcting various spatial distortions, magnification and reduction of imageries to match the required scale. The limitations of classical polynomial interpolation approaches like Lagrange interpolation are well discussed by Hou and Andrews.[1] They introduced the idea of interpolation by spline functions. They have developed an algorithm for interpolation using cubic spline functions. Recently, Keys has developed an algorithm for interpolation by cubic convolution method.[2] In cubic convolution method, he has defined a kernel for interpolation. There are some other methods for interpolation like nearest neighbour, bilinear interpolation. Cubic convolution interpolation is more accurate than nearest neighbour or linear interpolation. However, it is not as accurate as cubic spline interpolation method. Often, cubic convolution is preferred in practice over cubic spline interpolation method. This is because of its computational efficiency. Cubic spline interpolation requires greater computational exertion than cubic convolution. This is due to the fact that in the case of cubic spline, the interpolation coefficients are obtained by solving tri-diagonal matrix problem and they need to be evaluated at each data point.

Here, in this paper, an algorithm has been developed for interpolation by piecewise polynomial approximation. Here, the hypersurface approximation that has been used by Chittineni[3,4], Morgenthaler and Rosenfeld[5] has been utilized. Thus, approximation has been carried out by quadratic surface defined over two dimensional space or digital picture function in the neighbourhood of the point to be interpolated using orthogonal polynomials as basis functions. With these polynomial coefficients interpolation kernel has been evaluated, which in turn, enables the efficient implementation of the algorithm. Thus, the algorithm has an efficiency that of cubic convolution algorithm at the same time accuracy wise it is comparable with earlier spline interpolation methods.

## II. HYPERSURFACE FITTING USING ORTHOGONAL BASIS FUNCTIONS

Let $X = (x_0, x_1, \ldots\ldots, x_n)^T$ be a point in n dimensional space. Let $r_0$ be a rectangular region in n dimensional space. Without loss of generality, the coordinate system is chosen such that centres of the region $r_0$ is at the origin. Thus, we can write

$$\sum_{x_i \in r_0} x_i = 0 \quad \text{for all } i \tag{1}$$

Let $[Si(x),\ 0 \leq 1 \leq N]$ be a set of orthogonal basis functions defined over region $r_0$. Let f(x) be the digital picture function. Let g(x) be the estimate of f(x) as weighted sum of the basis functions, i.e.,

$$g(x) = \sum_{i=0}^{N} a_i Si(x) \tag{2}$$

where, $[a_i, 0 \le i \le N]$ are set of coeffi-cients. The total squared error $E^2$ can be written as,

$$E^2 = \sum_{x \varepsilon r_0} (f(x) - g(x))^2 \qquad (3)$$

Using the orthogonal properties of the basis functions, from Eq.(2) & (3), the coefficients $a_i$ that minimizes $E^2$ can be obtained as,

$$a_i = \sum_{x \varepsilon r_0} f(x) Si(x) / \sum_{x \varepsilon r_0} Si^2(x) \qquad (4)$$

## III. FITTING HYPERSURFACE USING DISCRETE ORTHOGONAL POLYNOMIALS

Let $\psi_i$ be the domain of $x_i$. Let $[P_{ij}(x_i), 0 \le j \le m]$ be the set of discrete orthogonal polynomials on $\psi_i$, $1 \le i \le n$. The set of n dimensional basis functions $Si(x)$, $0 \le l \le n$ can be constructed using one dimensional discrete orthogonal poly-nomials $P_{ij}(x_i)$ as follows:

$$\prod_{\ell=0}^{n} P_{10}(x_1), \ldots, \prod_{\ell=0}^{n} P_{1j}(x_1), \ldots, \prod_{\ell=0}^{n} P_{1i}(x_1) \qquad (5)$$

It can be shown that the above polynomials are orthogonal where $i \neq j$ for any l. Let $m_i = \Sigma x_i^k$ be the kth moment of $x_i$ over domain $x_i$. A few discrete orthogonal polynomials are given below:

$$\left.\begin{aligned} P_{i0}(x_i) &= 1 \\ P_{i1}(x_i) &= x_i \\ P_{i2}(x_i) &= x_i^2 - \frac{m_{i2}}{m_{i0}} \\ P_{i3}(x_i) &= x_i^3 - \frac{m_{i4}}{m_{i2}} \end{aligned}\right\} \qquad (6)$$

In terms of one dimensional discrete orthogonal polynomials, the n dimensional hyper quadratic surface can be written as

$$g(x) = a_0 \prod_{\ell=0}^{n} P_{10}(x_1) + \sum_{i=1}^{n} a_i P_{i1}(x_i) \prod_{\substack{\ell=1 \\ \neq i}}^{n} P_{10}(x_1) + \sum_{i=1}^{n} a_{1i} P_{i2}(x_i) \prod_{\substack{\ell=1, \\ \neq i}}^{n} P_{10}(x_1)$$

$$+ \sum_{\substack{i,j=1 \\ i<j}}^{n} a_{ij} P_{i1}(x_i) \prod_{\substack{\ell=1, \\ \neq i, \neq j}}^{n} P_{10}(x_1) \qquad (7)$$

In terms of coordinates $x_1, x_2, \ldots, x_n)$ the Eq.(7) can be written as,

$$g(x) = b_0 + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} b_{i1} x_i^2 + \sum_{\substack{i,j=1 \\ i<j}}^{n} b_{ij} x_i x_j \qquad (8)$$

where b's are given by,

$$\begin{aligned} b_0 &= a_0 - \sum_{i=1}^{n} \frac{m_{i2}}{m_{i0}} m_{i1} \\ b_i &= a_i \quad 1 \le i \le n \\ b_{i1} &= a_{i1} \quad 1 \le i \le n \\ b_{ij} &= a_{ij} \quad 1 \le i,j \le n \quad i<j \end{aligned} \qquad (9)$$

where a's are defined as,

$$\begin{aligned} a_0 &= \sum_{r_0} f(x) \\ a_i &= \frac{\sum_{r_0} x_i f(x)}{\sum_{r_0} x_i^2}, \quad 1 \le i \le n \\ a_{ii} &= \frac{\sum_{r_0} P_{i2}(x_i) f(x)}{\sum_{r_0} P_{i2}^2(x_i) f(x)}, \quad 1 \le i \le n \\ a_{ij} &= \frac{\sum_{r_0} x_i x_j f(x)}{\sum_{r_0} (x_i x_j)^2}, \quad 1 \le i,j \le n, \ i<j \end{aligned} \qquad (10)$$

The masks for coefficients $b_i$, that result by fitting hyper quadratic surface to image function can be obtained from Eq. (8), (9) and (10). For a 3 x 3 two dimen-sional neighbourhood, these masks are shown in Fig. 1.

## IV. INTERPOLATION KERNEL EVALUATION

For equispaced data, continuous inter-polation function, in the case of one dimensional data can be written as,

$$g(x) = \sum_k C_k U\left(\frac{x - x_k}{h}\right) \qquad (11)$$

In Eq. (11), $g(x)$ is interpolation function corresponding to sampled function $f(x_k)$ and $x_k$ are the interpolation notes. The $C_k$'s are parameters which depend upon sampled data. Eq.(11) converts the discrete data into a continuous function by an operation similar to that of convolution. In the case of two dimensional data for 3 x 3 neighbourhood, Eq.(11) can be written as,

$$g(x_1,x_2) = \sum_{\ell=-1}^{1} \sum_{m=-1}^{1} C_{j+1,k+m}\, U\left(\frac{(x_1-x_{1j})}{h_{x_1}}, \frac{(x_2-x_{2k})}{h_{x_2}}\right) \quad (12)$$

In Eq.(12), $C_{j+1,k+m}$ are parameters which depend upon sampled data. For two dimensional data, Eq.(8) representing hypersurface approximation can be rewritten as,

$$g(x_1,x_2) = b_0 + b_1x_1 + b_2x_2 + b_{11}x_1^2 + b_{22}x_2^2 + b_{12}x_1x_2 \quad (13)$$

Eq.(13), defines a continuous function at $x_1,x_2$ and coefficients b's correspond to sampled data as given by Eq.(9). Eq.(13) can be rewritten as,

$$g(x_1,x_2) = \sum_{\ell=-1}^{1} \sum_{m=-1}^{1} C_{j+1,k+m}\, (M_0+Ms_1+M_2s_2+M_{11}s_1^2+M_{22}s_2^2+M_{12}s_1s_2) \quad (14)$$

where $M_0$, $M_1$, $M_2$, $M_{11}$, $M_{22}$ and $M_{12}$ corresponds to masks shown in Fig. 1. $C_{jk} = f(x_1,x_2)$ where $f(x_{1_j}, x_{2_k})$ discrete image data $s_1$ and $s_2$ are given by,

$$s_1 = \frac{x_1 - x_{1j}}{h_{x_1}}$$

$$s_2 = \frac{x_2 - x_{2_k}}{h_{x_2}}$$

where $h_{x_1}$ and $h_{x_2}$ are sampling intervals.

It is obvious from Eq.(12) & (14), that interpolation kernel $U(s_1,s_2)$ can be written as,

$$U(s_1,s_2) = M_0 + M_1s_1 + M_2s_2 + M_{11}s_1^2 + M_{12}s_1s_2 \quad (15)$$

## V. ILLUSTRATIONS

The above developed algorithm has been implemented on VAX-11/780 computer system. As an illustration algorithm has been applied for Landsat data. Input data corresponds to an area from Landsat scene 161-043. The scene is resampled for magnification by a factor 2, by using cubic convolution, nearest neighbourhood, interpolation methods, by above developed algorithm. The results are shown in the plates. Plate 1 corresponds to input data and plates 2,3,4 correspond to interpolated output. For the example shown above the nearest neighbourhood took CPU time of about 3 mts. whereas, cubic convolution and hypersurface fitting methods took CPU time of 27 mts. and 28 mts., respectively.

## VI. SUMMARY

An algorithm for interpolation has been developed and implemented. In the developed algorithm interpolation has been carried out by fitting quadratic hypersurface, defined over two dimensional space, that approximates digital picture function in the neighbourhood of the point to be interpolated.

The hypersurface approximation is obtained by summation of orthogonal polynomials or basis functions. As an illustration, the method has been applied for Landsat data and results of the same are compared with other methods. The developed algorithm computer efficiency wise compares with cubic convolution interpolation algorithm.

## VII. ACKNOWLEDGEMENTS

The authors wish to thank Prof. B.L. Deekshatulu, Director, NRSA, for his guidance, encouragement and for providing all the required facilities for the work. The

Plate No.1
(INPUT)

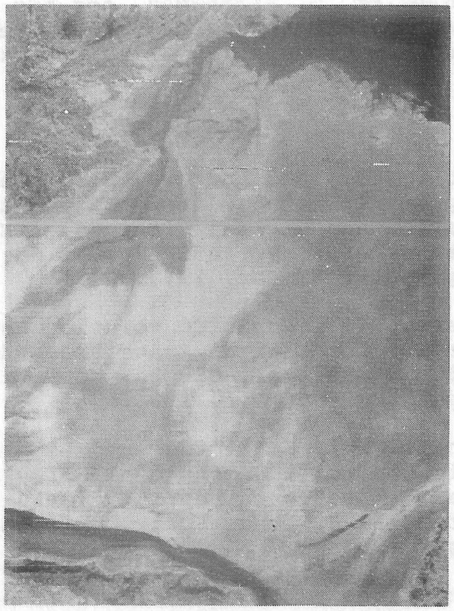

Plate No.2
(NEAREST NEIGHBOUR INTERPOLATION)

Plate No.3
(CUBIC CONVOLUTION INTERPOLATION)

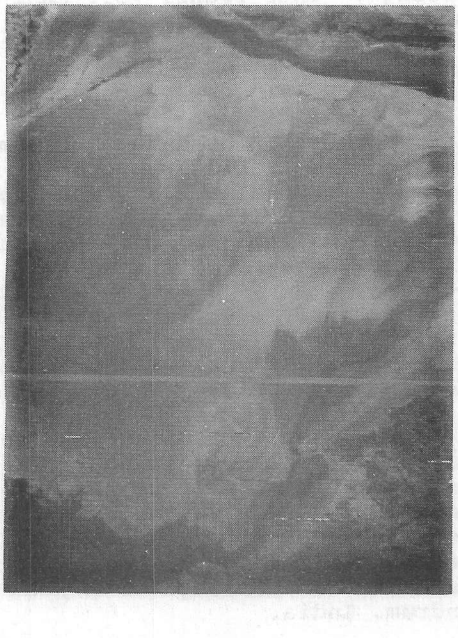

Plate No.4
(HYPER SURFACE APPROXIMATION INTERPOLATION)

authors are thankful to Mr. K. Vidyasagar, for software development support for cubic convolution algorithm. The authors wish to thank Mr. G. Rajaraman, for his secretarial assistance and other help rendered during the preparation of this work.

## VIII. REFERENCES

1. Hou, H.S., and Andrews, H.C. 'Cubic Spline for Image Inter-image Interpolation and Digital Fitting'. IEEE Trans. ASSP, Vol.ASSP 26, No.6, pp. 508-516, Dec. 1978.

2. Keys, R.G. 'Cubic Convolution Interpolation for Digital Image Processing' IEEE Trans. Vol.ASSP 2d, No.6, pp.1153-1160, Dec. 1981.

3. Chittineni, C.B. 'Edge and Line Detection in Multi dimensional Noisy Imagery Data'. Research report 5510-10-4-1-82, Conoco Inc. Ponaca City, Oklahoma, Feb. 1982.

4. Chittineni, C.B. 'Multidimensional Edge and Line Detection by Hypersurface Fitting Using Basis Functions' Proc. MPRSD Symposium at LARS, Purdue University, pp.245-254, July 1982.

5. Morgenthaler, D.S. and Rosenfeld, A. 'Multidimensional Edge by Hypersurface Fitting'. IEEE Trans. Vol. PAMI-3, No.4, pp.482-486, July '81.

## AUTHORS BIOGRAPHICAL DATA

A.D. Kulkarni. Graduated in Communications and Electronics Engineering from University of Poona in 1969, obtained Masters and Doctorate degrees in electrical engineering from Indian Institute of Technology, Bombay in 1971 and 1977, respectively. He is a member of Indian Society of Photo Interpretation and Remote Sensing. Since 1976, he is working as Software Engineer at the National Remote Sensing Agency, India.

K. Sivaraman. Graduated in Electronics and Communication Engineering from the Indian Institute of Science, Bangalore. Obtained his Masters degree from Indian Institute of Technology, Bombay, in 1971. At present, is working as Assistant Professor, at the College of Engineering, Trivandrum, India.

$$\frac{1}{9}\begin{matrix} -1 & 2 & -1 \\ 2 & 5 & 2 \\ -1 & 2 & -1 \end{matrix} \qquad \frac{1}{4}\begin{matrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{matrix}$$

$b_i$Mask($M_0$) $\qquad$ $b_{12}$Mask($M_{12}$)

$$\frac{1}{6}\begin{matrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{matrix} \qquad \frac{1}{6}\begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

$b_{11}$Mask($M_{11}$) $\qquad$ $b_1$Mask($M_1$)

$$\frac{1}{6}\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad \frac{1}{6}\begin{matrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{matrix}$$

$b_2$Mask($M_2$) $\qquad$ $b_{22}$Mask($M_{22}$)

Fig.1.