BICUBIC B-SPLINE APPROXIMATION BY LEAST SQUARES

Inkilä, Keijo
Helsinki University of Technology
Institute of Photogrammetry and Remote Sensing
Otakaari 1, 02150 Espoo 15
Finland
Comission III

Abstract

The generation of basis functions for two-dimensional cubic B-spline approximation by tensor product construction is first described. Then the least squares solution of the approximation problem is considered. Special attention is paid on computational efficiency, solution of rank deficient problems and application of constraints. Finally,some generalizations of the presented method are briefly discussed.

0. Introduction

Spline functions are widely used as approximating functions because of their favourable theoretical and computational properties. In this paper bicubic splines are used to approximate functions of two variables, that is, for surface fitting problems, which frequently arise in photogrammetry and remote sensing (e.g. approximation to height values or grey values). A bicubic spline is a piecewise polynomial of order 4 (of degree 3) along the lines parallel to the coordinate axes, a piecewise polynomial of order 7 along any other lines and it has continuous first and second partial derivatives. The consideration of bicubic splines has been chosen because of their practical importance. The generalization to the spline approximation of any order (and dimension as well) should be obvious after having examined the presented case.

The paper consists of 6 sections. In section 1 B-splines which provide a convenient basis for splines are introduced. In section 2 a basis for bicubic splines is constructed from the tensor product of two one-dimensional cubic B-spline sets. In section 3 surface fitting by least squares is considered. In sections 4 and 5 the solution of a rank deficient and a constrained problem is dealt with, respectively. Finally, in section 6 some generalizations are made to the presented method.

1. Cubic B-splines

Let $s_1, s_2, \ldots, s_{n+4}$ be an strictly increasing knot sequence,i.e.

$$s_1 < s_2 < \ldots < s_{n+4}$$

For this knot sequence the sequence of n cubic B-splines $B_1(x), B_2(x), \ldots, B_n(x)$ can be generated. This sequence is uniquely defined by the following conditions /3/:

(i) Each $B_i(x)$ is a cubic spline of order 4 with knots $s_1, \ldots, s_{n+4}$

(ii) $B_i(x) > 0$, if $s_i < x < s_{i+4}$
$= 0$, otherwise

(iii) $\sum_i B_i(x) = 1$, for $x \in [s_4, s_{n+1}]$ (scaling)

The importance of B-splines for approximation is mainly based on the following facts:

1) The sequence $B_1(x), B_2(x), \ldots, B_n(x)$ is a basis of the space of the cubic splines with end points s_4 and s_{n+1} and with (interior) knots $s_5, \ldots, s_n$ /3/.
2) The values of B-splines at given points can be computed efficiently and stably using a recurrence relation for B-splines /3,5/.
3) Due to small (local) support of B-splines (condition (ii)) the linear system arising in an approximation problem is sparse/3/.

2. B-spline Representation of a Bicubic Spline

Let us divide the xy-plane into rectangular panels by lines

$x = s_1$ $\qquad s_1 < s_2 < \ldots < s_{nx+4}$
$x = s_2$
$- \; - \; -$
$x = s_{nx+4}$

$y = t_1$ $\qquad t_1 < t_2 < \ldots < t_{ny+4}$
$y = t_2$
$- \; - \; -$
$y = t_{ny+4}$

Let $M_1(x), M_2(x), \ldots, M_{nx}(x)$ and $N_1(y), N_2(y), \ldots, N_{ny}(y)$ be cubic B-spline sequences generated for the knot sequences $s_1, \ldots, s_{nx+4}$ and $t_1, t_2, \ldots, t_{ny+4}$, respectively. Define

$$B_{ij}(x,y) = M_i(x) N_j(y), \quad i=1,\ldots,nx \quad \& \quad j=1,\ldots,ny$$

The functions $B_{ij}(x,y)$ are called tensor product B-splines /1/.A function $B_{ij}(x,y)$ is a hill function which is non-zero only over the rectangle $s_i < x < s_{i+4}$ and $t_j < y < t_{j+4}$, i,e. it has a small support only. It can be shown that the functions $B_{ij}(x,y)$ are a basis of the linear space of the bicubic splines defined on the rectangle R (Fig. 1)

$s_4 < x < s_{nx+1}$

$t_4 < y < t_{ny+1}$

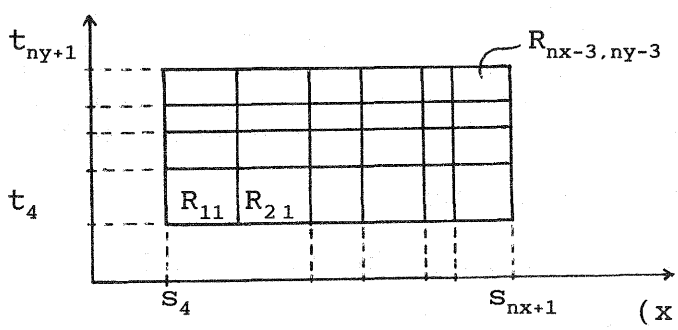

FIG. 1. The division of the rectangle R into rectangular panels by knots s_i and t_j.

with the knots $s_5,\dots,s_{nx}$ and $t_5,\dots,t_{ny}$. Consequently, a bicubic spline $s(x,y)$ has on the rectangle R a unique representation of the form /4/

$$s(x,y) = \sum_{i=1}^{nx} \sum_{j=1}^{ny} c_{ij} B_{ij}(x,y)$$

where c_{ij} are real coefficients. Their determination by least squares is the subject of the next section.

3. Least Squares Approximation

Let the data consist of the values

$$z_k = f(x_k, y_k) + e_k \;, \qquad k=1,\dots,m, \quad m>n=nx{*}ny$$

where $f(x,y)$ is a function to be approximated and e_ks are unknown errors. Define

$$v_k = \sum_{i=1}^{nx} \sum_{j=1}^{ny} c_{ij} B_{ij}(x_k, y_k) - z_k \;, \qquad k=1,\dots,m$$

or

$$\mathbf{v} = \mathbf{B}\mathbf{c} - \mathbf{z}$$

where

$$\mathbf{v}^t = (v_1 \;\; v_2 \;\; \dots \;\; v_m)$$
$$\mathbf{z}^t = (z_1 \;\; z_2 \;\; \dots \;\; z_m)$$
$$\mathbf{c}^t = (c_{11} \;\; c_{12} \;\; \dots c_{1ny} \;\; c_{21} \;\; \dots \;\; c_{nxny})$$
$$\mathbf{B} = (b_{ij}), \quad b_{ij} = B_{uv}(x,y), \quad u=\mathrm{int}[(j-1)/ny]+1, \; v=j-(u-1)ny$$

The particular problem of least squares approximation is to find c that minimizes $\mathbf{v}^t\mathbf{P}\mathbf{v}$, where $\mathbf{P}$ is a positive definite mxm-matrix (weight matrix).

It can be shown (see /3/ for the one-dimensional case) that the solution of the ls-problem is unique (rank($\mathbf{B}$)=n), if and only if, for some $1 \le k_1 < k_2 < \dots < k_n \le m$, the data point (x_{ki}, y_{ki}) lies inside the rectangle

$$s_u < x < s_{u+4} \qquad \text{where} \quad u=\mathrm{int}[(i-1)/ny]+1$$
$$t_v < y < t_{v+4} \qquad \qquad \quad v=i-(u-1)ny$$

A ls-problem is conventionally solved via the normal equations. Alternatively, it can be solved by means of orthogonal transformations /4,5,6/. Both approaches are familiar to photogrammetrists and are therefore not considered more here. Instead, we deal with some means to reduce the computational burden, which may easily grow considerable.

Due to small support of tensor product B-splines the matrix $\mathbf{B}$ is a sparse matrix. It is easy to find out that if the data points are numbered panelwise in the order $R_{11}, R_{12}, \dots, R_{1,ny-3}, R_{21}, \dots, R_{nx-3,ny-3}$, then the matrix $\mathbf{B}$ takes the form

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} & \mathbf{B}_{14} & & & \\ & \mathbf{B}_{22} & \mathbf{B}_{23} & \mathbf{B}_{24} & \mathbf{B}_{25} & \ddots & \\ & & \ddots & & & & \\ & & & \mathbf{B}_{nx-3,nx-3} & \dots & \mathbf{B}_{nx-3,nx} & \end{bmatrix}$$

i.e. **B** is a banded matrix with bandwidth 3ny+4. The band contains, however, many zeros, because each submatrix $\mathbf{B}_{rs}$ is a rectangular matrix of band form with bandwidth 4 and with ny columns and its number of rows equal to the number of data points in the panels $R_{r1}, R_{r2}, \ldots, R_{r,ny-3}$. Some extra computational savings may thus be attained by applying general sparse matrix methods for the solution.

A particularly favourable case from the computational point of view results, if the data is given at the nodes of a rectangular grid, that is,

$$z_{kl} = f(x_k, y_l) + e_{kl}, \quad k=1,\ldots,mx \text{ for } l=1,\ldots,my \quad (m=mx\,my)$$

Then the ls-problem $\mathbf{Bc} \cong \mathbf{z}$ can equivalently be written as /2,3,7/

$$(\mathbf{M} \otimes \mathbf{N})\mathbf{c} \cong \mathbf{z}$$

or

$$\mathbf{M}\mathbf{C}\mathbf{N}^t \cong \mathbf{Z}$$

or

$$\begin{cases} \mathbf{MY} \cong \mathbf{Z} \\ \mathbf{NC}^t \cong \mathbf{Y}^t \end{cases}$$

where

$\mathbf{M}=(m_{ij})$,	$m_{ij}=M_j(x_i)$,	$i=1,\ldots,mx$,	$j=1,\ldots,nx$
$\mathbf{N}=(n_{ij})$,	$n_{ij}=N_j(x_i)$,	$i=1,\ldots,my$,	$j=1,\ldots,ny$
$\mathbf{C}=(c_{ij})$,	$i=1,\ldots,nx$,	$j=1,\ldots,ny$	
$\mathbf{Z}=(z_{ij})$,	$i=1,\ldots,mx$,	$j=1,\ldots,my$	
$\mathbf{c}=\text{vec}\mathbf{C}$	("vec" means here the formation of a vector		
$\mathbf{z}=\text{vec}\mathbf{Z}$	from the rows of a matrix)		

The solution of a (large) ls-problem can thus be replaced by the successive solutions of two smaller (matrix) ls-problems. Computational superiority of the regular case is immediately obvious by examining the following table.

Matrix	Order	Bandwidth	Remarks
B	$m \times n$	3ny+4	$m=mx \times my$ & $n=nx \times ny$
M	$mx \times nx$	4	
N	$my \times ny$	4	

4. Rank Deficient Case

If the data is distributed arbitrarily, it may be difficult or even impossible to choose the knots s_i and t_j so that the condition for the full-rank problem is satisfied. If rank(**B**)=r<n, then after r orthogonal transformations we arrive at the situation (column interchanges are included)

$$\mathbf{QB} = \begin{array}{cc} \begin{array}{cc} r & n-r \end{array} & \\ \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & 0 \end{bmatrix} & \begin{array}{l} r \\ m-r \end{array} \end{array}$$

where $\mathbf{R}_{11}$ is a upper triangular matrix

The most straightforward way to have a ls-solution is, of course, to put the last n-r unknowns equal to zero. This may, however, lead to unwanted fluctuations caused by pairs of the c_i having large values of opposite sign /4/. A smoother surface results, if the minimum-norm condition $\|\mathbf{c}\|$=minimum is applied for a unique solution.

The minimum-norm solution can be computed in several ways but the following algorithm is particularly convenient in this context /6/:
1) Reduce $[\mathbf{R}_{11}\ \ \mathbf{R}_{12}]$ to the lower triangular form $[\tilde{\mathbf{R}}_{11}\ \ 0]$ by multiplying it from the right by the orthogonal matrix $\mathbf{K}^t$
2) Solve $\mathbf{y}_1$ from $\tilde{\mathbf{R}}_{11}\mathbf{y}_1 = \mathbf{g}_1$, where $\mathbf{g}_1$ contains the first k elements of $\mathbf{g} = \mathbf{Q}\mathbf{z}$
3) Compute $\mathbf{c} = \mathbf{K}\begin{bmatrix}\mathbf{y}_1\\ 0\end{bmatrix}$

5. Constraints

In surface fitting problems the need often arise that some components of the solution vector **c** should satisfy some specified constraints. For example, it may be required that the fitted spline or its derivatives take specified values at certain points or along the certain lines parallel to coordinate axes /4/. Such equality constraints may be expressed in the form

$$\mathbf{C}\mathbf{c} = \mathbf{d}$$

where **C** is a $m_c$xn-matrix ($m_c \leq n$) with rank(**C**)=k_c ($k_c \leq m_c$). The equality constrained ls-problem (lse-problem) can then be stated as:

$$\text{Minimize} \quad \|\mathbf{B}\mathbf{c} - \mathbf{z}\| \quad \text{subject to} \quad \mathbf{C}\mathbf{c} = \mathbf{d}$$

We choose to solve this problem by a method which makes use of an orthogonal basis for the null space of **C** /6/. If

$$\mathbf{C} = \mathbf{H}\mathbf{R}\mathbf{K}^t$$

is any orthogonal decomposition of **C** and **K** is partitioned as

$$\mathbf{K} = [\underset{k_c}{\mathbf{K}_1}\ \ \underset{n-k_c}{\mathbf{K}_2}]$$

then the columns of $\mathbf{K}_2$ form an orthogonal basis of the null space of **C** and all solutions of **Cc**=**d** are thus of the form

$$\mathbf{c} = \mathbf{C}^+\mathbf{d} + \mathbf{K}_2\mathbf{y}_2$$

where $\mathbf{C}^+$ is the pseudoinverse of **C** and $\mathbf{y}_2$ is an arbitrary vector with n-k_c elements. The vector $\mathbf{y}_2$ is now determined so that $\|\mathbf{B}\mathbf{c}=\mathbf{z}\|$ is minimized, that is, by solving the derived problem

$$\mathbf{B}\mathbf{K}_2\mathbf{y}_2 \cong \mathbf{z} - \mathbf{B}\mathbf{C}^+\mathbf{d}$$

It can be shown that this problem is of full-rank, if rank(**D**)=n, where $\mathbf{D}^t=[\mathbf{C}^t\ \ \mathbf{B}^t]$. Then the unique solution of the original lse-problem is

$$\hat{\mathbf{c}} = \mathbf{C}^+\mathbf{d} + \mathbf{K}_2\hat{\mathbf{y}}_2$$

If, on the other hand, rank(**D**)<n, then the original lse-problem has a unique minimum-norm solution given by

$$\tilde{c} = C^{+} d + K_2 \tilde{y}_2$$

where $\tilde{y}_2$ is the unique minimum-norm solution of the derived problem.

In practise usually rank(**C**)=m_c and we can choose **H**=**I**. Then the solution algorithm involves only the following steps:

1) Postmultiply **D** by an orthogonal matrix **K** such that

$$\begin{bmatrix} C \\ B \end{bmatrix} K = \begin{array}{cc} \begin{bmatrix} F_{11} & 0 \\ F_{21} & F_{22} \end{bmatrix} & \begin{array}{c} p \\ m \end{array} \\ \begin{array}{cc} p & n-p \end{array} & \end{array}$$

, where F_{11} is a lower triangular matrix

2) Solve y_1 from the lower triangular system $F_{11} y_1 = d$
3) Compute $\bar{z} = z - F_{21} y_1$
4) Solve the ls-problem $F_{22} y_2 \cong z$ for $\hat{y}_2$
5) Compute $\hat{c} = K \begin{bmatrix} y_1 \\ \hat{y}_2 \end{bmatrix}$

The possibility to include also linear inequality constraints in a surface fitting problem would be highly desirable. This would allow us, for example, to set upper/lower bounds for the unknowns or to require that the fitted surface is concave up or down. In many applications such additional information could be very valuable for guaranteeing a reasonable solution. The introduction of inequality constraints leads, however, to a rather complicated problem, whose consideration is beyond the scope of this paper.

6. Some Complementary Remarks

In this section we make some generalizations to the presented method.

Exploitation of other types of data

We have assumed above that the data consists of the values of the function to be approximated only. Values of other functionals can, however, be exploited also, if available. In particular, derivative information is often available and it can be easily and efficiently exploited in B-spline approximation.

Multiple knots

We made in Section 1 the assumption that the knot sequence is strictly increasing. No computational problems, however, result, if we allow at most 4 successive knots to coincide ($s_i < s_{i+4}$). This makes it possible to construct piecewise representations which are less smooth than cubic splines based on the relation /3/:

No. of continuity conditions + knot multiplicity = 4

It is, however, obvious that in tensor product approximation only the discontinuities parallel to the coordinate axes can be taken care efficiently.

Parametric surfaces

Many surfaces (e.g. closed surfaces) cannot be represented by explicit functions, but only in a parametric form:

$$x = f_x(u,v)$$
$$y = f_y(u,v)$$
$$z = f_z(u,v)$$

where u and v are the parameters. Basically, we then have to solve three two-dimensional approximation problems, one for each coordinate. Some features of the parametric case (e.g. the choice of parametrization) need, however, special attention to be paid for.

References

/1/ Bethel, James S.: Surface Approximation and On-line Quality Assessment in Digital Terrain Models. Dissertation, Purdue University, 1983.

/2/ Blaha, G.: A Few Basic Principles and Techniques of Array Algebra. Bulletin Geodesique, no. 3, 1977.

/3/ de Boor, C.: A Practical Guide to Splines. Springer-Verlag, New York, 1978.

/4/ Hayes, I.G., Halliday, I.: The Least-squares Fitting of Cubic Spline Surface to General Data Sets. I.Inst. Maths Applics 1974, pp. 89 - 103.

/5/ Inkilä, K.: Approximation of Curves and Surfaces by B-splines. The Photogrammetric Journal of Finland,no.1 ,1988.

/6/ Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems. Prentice-Hall, New Jersey, 1974.

/7/ Rauhala, U.: Introduction to Array Algebra. Photogrammetric Engineering and Remote Sensing, no. 2, 1980.