699

# A Method for Fair B-spline Interpolation and Approximation by Insertion of Additional Data Points

Tzvetomir Ivanov Vassilev*

Department of Computer Science, Queen Mary and Westfield College,

Mile End road, London E1 4NS

phone: 0171 975 5220, fax: 0181 980 6533

e-mail: ceco@dcs.qmw.ac.uk

**Abstract**

An efficient method for interpolation and approximation of both curve and surface points using B-splines is described. An automatic fairing is presented based on minimizing an energy functional. Additional data points, used as degrees of freedom for the fairing, are inserted only where the curve (the surface) needs them. This reduces the number of the unknowns to a minimum which makes the algorithm very fast and efficient especially when a huge amount of data is concerned. Results of applying the algorithm for about 15000 face data points, subject to measurement errors due to the digitization, are presented at the end of the paper.

**Keywords and Phrases:** Data interpolation and approximation, B-splines, fairing using curve and surface energy.

## INTRODUCTION

To construct visually acceptable interpolations for sets of data points has been a long standing research problem. The main difficulty arises from the fact that even when very efficient schemes based on splines are used the resulting curve (or surface) is often not "fair" enough and it has extraneous "bumps" and "wiggles" (for example as shown in

---

* **Address in Bulgaria** (valid **after 10 Oct 1995**)
Tzvetomir Vassilev, Dept. of Informatics, TU Rousse
8 Studentska street, 7017 Rousse, BULGARIA
phone: ++359 82 44507 276, fax: ++359 82 449791
e-mail: ceco@ait.tu-rousse.bg

Figures 4a, 5a, 6a, 7a, 8a). The objective of the work, described in this paper, is to develop a method for fair spline interpolation and approximation for 3D curves and surfaces that is efficient enough to be applied on large amounts of data.

Determining the parameterization of the curve is a fundamental problem to interpolation. Three basic methods have been widely used: uniform[1], chord length[2] and "centripetal" parameterization[3]. Uniform knot sequence does not usually give satisfactory results because it does not take into account the geometry of the points. There are also cases where chord length and centripetal methods fail to give good results. Several other algorithms for "optimal" parameterization have been described in the literature[4,5,6] but once the optimum has been found, the curve is determined simply by using the interpolation conditions or by a least squares fit. No other shape constraints are applied.

One approach for constructing a fair curve is to apply additional mathematical conditions to preserve its shape. For example Carl de Boor[2] proposed the "taut spline" that preserves convexity of the data and Fritsch and Carlson[7] constructed a $C^1$ cubic interpolant that preserves the monotonicity. Another possibility is to use an interactive fairing scheme[8,9], where the user decides which of the control points should be moved. Sapidis and Farin[10] suggested an automation of this process showing how to determine where a knot should be removed and a new one inserted using the curvature plots as a fairness criterion. Note that the resulting curve after the fairing is always only an approximation to the original one.

An alternative approach exploits minimization of the energy of the curve (surface). This dates back to 1966 when Schweikert suggested the idea of a spline in tension[11]. Here the standard cubic spline is enriched by adding new exponential terms and degrees of freedom, called tension parameters. The works of Salkauskas[12] and Foley[13] use a $C^1$ interpolant that minimizes weighted $L_2$-norms of the second, and the first and the second derivatives respectively. Celniker and Gossard[14] constructed a shape that naturally resists stretching and bending. This leads to a functional expressed as an integral of a weighted sum of the first and second parametric derivatives of the curve (or surface). In this method $C^1$ continuous cubic Hermite polynomials are used as a basis for curves and $C^1$ triangular elements for surfaces. Nowacki and Lü[15] went further and adopted a fairness criterion that contains $L_2$-norms of the second and the third order parametric derivatives. Their implementation uses $C^2$ Hermite quintic polynomial curves with area constraints resulting in a non-linear system of equations that is solved numerically.

An interpolation scheme, using the fairness norm[14] together with Catmull-Clark surfaces, is described by Halstead et al.[16] for data sets with arbitrary topology. In order to add degrees of freedom for the fairing, the original control mesh is subdivided twice to decouple the interpolation constraints from each other. This, however, dramatically increases the number of control points. As a result the method is too slow and practically inapplicable for large amounts of data.

This paper describes an algorithm for fair interpolation and approximation with $C^2$ continuous cubic B-splines and integral fairness criteria. Additional data points (ADP) are used as degrees of freedom for minimizing the energy of the curve (surface), resulting in a linear system of equations. The ADP are not distributed evenly but inserted only where necessary, where a lack of data is observed. The parameterization is based on a new method for incorporating the data geometry and the method is presented for both curves and surfaces. The scheme was developed to be applicable for large amounts of data.

The rest of the paper is organized as follows. In the second section the construction of a fair spline curve is described, including evaluation of the fairness norm, interpolation, approximation and choice of additional data points. In the third section the method is extended to tensor product surfaces. Finally the results and conclusions are presented in the fourth and fifth sections respectively.

## FAIR B-SPLINE CURVES

### Evaluating the fairness norm

Let $\mathbf{w}(u)=[x(u), y(u), z(u)]$ be a space curve parameterized by $u$ and let $\mathbf{V}$ be a column vector of its B-spline control points. Celniker and Gossard[14] suggested the following energy functional

$$E = \int_{\text{curve}} (\alpha \dot{\mathbf{w}}^2(u) + \beta \ddot{\mathbf{w}}^2(u))\mathrm{d}u \qquad (1)$$

where $\dot{\mathbf{w}}(u)$ and $\ddot{\mathbf{w}}(u)$ are the first and second derivatives in respect to the parameter $u$. The energy of the curve is represented as a weighted sum of its stretching and bending terms, where $\alpha$ and $\beta$ are freely selected coefficients.

A single cubic B-spline segment can be written as $\mathbf{w}_s(u)=\mathbf{V}_s^{\mathrm{T}}\mathbf{B}_s$, where $\mathbf{B}_s$ is the B-spline basis functions vector. Evaluating the integral (1) for the function $\mathbf{w}_s$ will result in the following quadratic form

$$E_s = \mathbf{V}_s^{\mathrm{T}}\mathbf{K}_s\mathrm{V}_s, \qquad (2)$$

where $\mathbf{K}_s$ is a 4×4 symmetric matrix, whose entries are calculated by solving the integral (1). $E_s$ is called the "local fairness norm" and $\mathbf{K}_s$ is the "local stiffness matrix" for the segment $\mathbf{w}_s(u)$. $\mathbf{K}_s$ can be represented as a weighted sum of its stretching and bending terms

$$\mathbf{K}_s = \alpha\mathbf{K}_{1s} + \beta\mathbf{K}_{2s}. \tag{3}$$

For a uniform knot distribution, using de Boor's formula[2] one can easily find that

$$\mathbf{K}_{1s} = \int_0^1 \dot{\mathbf{B}}_s \dot{\mathbf{B}}_s^{\mathrm{T}} \mathrm{d}u = \frac{1}{120}\begin{bmatrix} 6 & 7 & -12 & -1 \\ 7 & 34 & -29 & -12 \\ -12 & -29 & 34 & 7 \\ -1 & -12 & 7 & 6 \end{bmatrix}$$

$$\mathbf{K}_{2s} = \int_0^1 \ddot{\mathbf{B}}_s \ddot{\mathbf{B}}_s^{\mathrm{T}} \mathrm{d}u = \frac{1}{6}\begin{bmatrix} 2 & -3 & 0 & 1 \\ -3 & 6 & -3 & 0 \\ 0 & -3 & 6 & -3 \\ 1 & 0 & -3 & 2 \end{bmatrix}. \tag{4}$$

Now, after we have expressed the local fairness norm for a single segment, the global fairness norm $E$ for a B-spline curve can be found as a sum of the local fairness norms over each of the B-spline polynomial segments

$$E = \mathbf{V}^{\mathrm{T}}\mathbf{K}\mathbf{V},\ \mathbf{K}=\alpha\mathbf{K}_1+\beta\mathbf{K}_2, \tag{5}$$

where $\mathbf{K}$ is a quadratic symmetric band matrix obtained from the local stiffness matrices. Minimizing (5) for a particular choice of degrees of freedom will ensure the fairing of the curve.

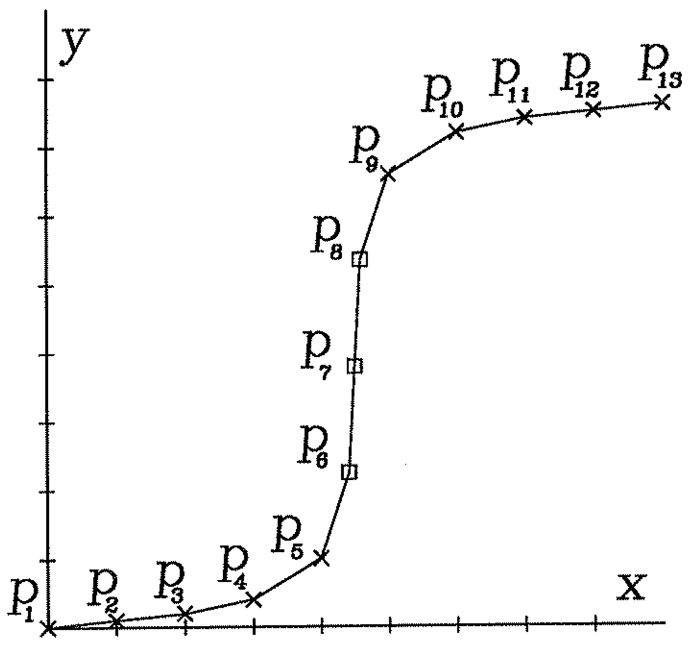



**Figure 1** Data set for fair interpolation with additional data points

**Fair B-spline interpolation with ADP**

We will modify slightly the original problem of interpolation for the purposes of fairing. The new task is: given a sequence of data points $\mathbf{p}_i$, $i$=1,...,$n$-1, some of them unknown and used as degrees of freedom, find the vector **V** of B-spline control points of a curve, which passes through the points $\mathbf{p}_i$ and minimizes the fairness norm (5). This is demonstrated in Fig. 1, where the original data set is $\{\mathbf{p}_1,..., \mathbf{p}_5, \mathbf{p}_9,..., \mathbf{p}_{13}\}$ and the unknown data points are $\{\mathbf{p}_6, \mathbf{p}_7, \mathbf{p}_8\}$. Using the interpolation constraints with end conditions set as degrees of freedom, the control points can be found by solving

$$\mathbf{AV} = \mathbf{P}, \qquad (6)$$

where **A** is a coefficient matrix and **P** is a column vector of data points, some of which unknown. Using the blossoming principle[17] for cubic splines with uniform knot sequence

$$\mathbf{A} = \frac{1}{6}\begin{bmatrix} 4 & 1 & 0 & 0 & .. & 0 \\ 1 & 4 & 1 & 0 & .. & 0 \\ : & : & : & : & :: & : \\ 0 & .. & 0 & 1 & 4 & 1 \\ 0 & .. & 0 & 0 & 1 & 4 \end{bmatrix}, \qquad (7)$$

where rows 0 and $n$ are the unknown end conditions.

In order to separate the known from unknown data points, we can rewrite the right hand side of the system (6) as follows

$$\mathbf{P} = \mathbf{DX}+\mathbf{D}_0, \qquad (8)$$

where **D** is a $(n+1)\times r$ sparse matrix, containing only "0"s and "1"s; **X** is a column vector of unknown data points with length $r$; $\mathbf{D}_0$ is a column vector of known data points with length $(n+1)$, containing a data point, where it is known, and "0"s elsewhere. The system matrix **A** is a symmetric band nonsingular matrix and its unique inverse $\mathbf{A}^{-1}$ can be found. The Cholesky factorization method[18] is very efficient in such cases. Having in mind (7) and (8) the system (6) becomes

$$\mathbf{V} = \mathbf{A}^{-1}(\mathbf{DX}+\mathbf{D}_0). \qquad (9)$$

Now the fairness norm (5) for the curve can be expressed as

$$E = (\mathbf{DX}+\mathbf{D}_0)^{\mathrm{T}}\mathbf{K}_{\mathrm{A}}(\mathbf{DX}+\mathbf{D}_0) \qquad (10)$$

where $\mathbf{K}_{\mathrm{A}} = \mathbf{A}^{-\mathrm{T}}\mathbf{K}\mathbf{A}^{-1}$. $\mathbf{K}_{\mathrm{A}}$ is symmetric and positive definite, so **X** can be found by setting the gradient of (10) to zero

$$\mathbf{D}^{\mathrm{T}}\mathbf{K}_{\mathrm{A}}\mathbf{DX} + \mathbf{D}^{\mathrm{T}}\mathbf{K}_{\mathrm{A}}\mathbf{D}_0 = 0. \qquad (11)$$

**Fair B-spline approximation with ADP**

Given a sequence of data points $\mathbf{p}_i$, $i$=1,...,$m$, some of them unknown, the problem is to find the vector **V** of B-spline control points of a curve, which passes close to the points $\mathbf{p}_i$ and minimizes the fairness norm (5). The original problem of least squares approximation with B-spline curves has been solved by Carl de Boor[2]. Let **S** be the linear space of all spline functions of order $k$ ($k$=3 for the cubic case), defined over a specific knot sequence. A discrete inner product for this space, which is a reasonable approximation to the continuous inner product $\int_a^b g(x)h(x)\mathrm{d}x$, is given by

$$<g, h> = \sum_{i=1}^{m} g(x_i)h(x_i)w_i \,, \tag{12}$$

where **x** is a sequence of data points in some interval [$a$, $b$] and **w** sequence of weights. The norm induced by the inner product (12) is expressed $\|g\|_2 = \sqrt{<g,g>}$. De Boor[2] proves that a function $f$ is a best approximation from the space **S** to an unknown function $p$ with respect to the norm $\|p-f\|_2$ if and only if

$$\sum_{j=0}^{n} < B_i, B_j > \mathbf{v}_j = < B_i, p >, \; i = 0,...,n. \tag{13}$$

Since some of the data points are unknown, we rewrite the inner products on the right hand side of (13) as

$$\mathbf{AV}=\mathbf{DX}+\mathbf{D}_0, \tag{14}$$

where **A** [($n$+1)×($n$+1)] is the system matrix; **D** [($n$+1)×$r$] is a sparse matrix; **X** is a column vector of unknown data points with length $r$; and $\mathbf{D}_0$ is a column vector with length ($n$+1), containing the known terms of the inner products. **A** is a symmetric band nonsingular matrix and its unique inverse $\mathbf{A}^{-1}$ can be found. From now on the problem is the same as for the case of interpolation and can be solved in the same way.

**Controlling the approximation error**

The error of approximation can be calculated using the formula

$$E_A = \frac{1}{m-1}\sum_{i=1}^{m} \|\mathbf{p}_i - \mathbf{w}(u_i)\|^2 \,. \tag{15}$$

One can control the value of $E_A$ by varying the number of polynomial segments $L$ of the B-spline curve. Finding the optimal $L$ for reconstructing a curve from a given "noisy" data set is problematic. Using too many segments will make the curve follow the noise and thus have many unwanted undulations, while too few segments may result in failure of the curve to represent the characteristics of the shape. Hence, the least $L$ should be found that satisfies a tolerance constraint. This can be done in the following way:

1. Find an initial value $L_0$ for which the tolerance requirement is violated. This is done using a binary search starting with $L_0 = m/2$.
2. Increase $L_0$ by one and find the approximation with ADP as derived before.
3. Check the tolerance requirement.
4. Repeat step 2 and 3 until the tolerance requirement is satisfied.

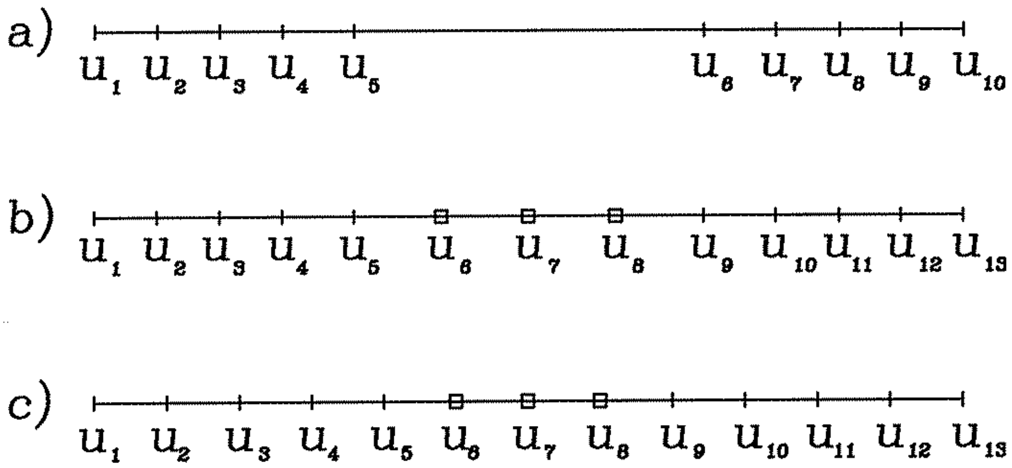



**Figure 2** Algorithm for new knots (data points) insertion

**Parameterization and choice of the additional data points**

When solving the problem for fair interpolation (approximation) we assumed that the new data points were to be inserted in known places. In this section we describe how these places may be determined. In many practical cases, when the points are obtained by measurement, we have to deal with the problem of a lack of data, because for some intervals measurements cannot be performed for physical reasons. As shown by the various experiments the author conducted (see Fig. 4a, 5a, 6a, 7a), the curve tends to have extraneous inflection points and undulations in and around these intervals. Therefore the additional data points should be inserted precisely where a lack of data is observed. Fig. 2 demonstrates the insertion algorithm for the data set in Fig. 1. It is as follows:

1. Calculate the parameter values $u_i$, $i=1,\ldots,n$, using the chord length or centripetal formula (Fig. 2a).
2. Compute the average length between two parameter values $\Delta u = u_n/(n-1)$.
3. For each interval $[u_i, u_{i+1}]$ find the number $n_i = \text{round}(\Delta u_i/\Delta u)$ of the new parameter values (data points) and insert them (Fig. 2b).

4. Recalculate the sequence $u_i$ this time using the uniform formula (Fig. 2c).

Note that the actual interpolation (approximation) is performed with the parameterization from step 4 (Fig. 2c). Steps 1-3 are needed only to determine where the additional data points are to be inserted. The last step is necessary because for the uniform case the stiffness matrix **K** can be computed exactly from integral (1). Otherwise, if we stop at step 3, numerical methods must be used to solve (1) which will decrease the efficiency of the algorithm. Although it might seem as if the final parameterization is uniform, this is not the case, because obviously the sequence $\{u_1,...,u_5, u_9,...,u_{13}\}$ in Fig. 2c is not uniform in respect to the original data set $\{\mathbf{p}_1,...,\mathbf{p}_5, \mathbf{p}_9,...,\mathbf{p}_{13}\}$ in Fig. 1. In fact the knot vector (Fig. 2c) is only an approximation to this in Fig. 2b but it is good enough, because the smoothness of the curve is ensured by the minimizing of the fairness norm (5).

## FAIR B-SPLINE SURFACES

### Fairness criterion

Let $\mathbf{w}(u,v)$ be a surface parameterized by $u$ and $v$, and let $\mathbf{V}$ be a $n_u \times n_v$ matrix of its control points. The following energy functional is considered in this paper

$$\iint (c_1 \mathbf{w}_{uv}{}^2 + c_2 \mathbf{w}_{u^2v}{}^2 + c_3 \mathbf{w}_{uv^2}{}^2 + c_4 \mathbf{w}_{u^2v^2}{}^2)\,\mathrm{d}u\mathrm{d}v\,, \qquad (16)$$

where the suffixes mean partial derivatives in respect to the parameters $u$ and $v$. A tensor product B-spline surface can be expressed as

$$\mathbf{w}(u,v) = \mathbf{B}_u{}^T \mathbf{V} \mathbf{B}_v, \qquad (17)$$

where $\mathbf{B}_u$ and $\mathbf{B}_v$ are vectors of the B-spline basis functions.

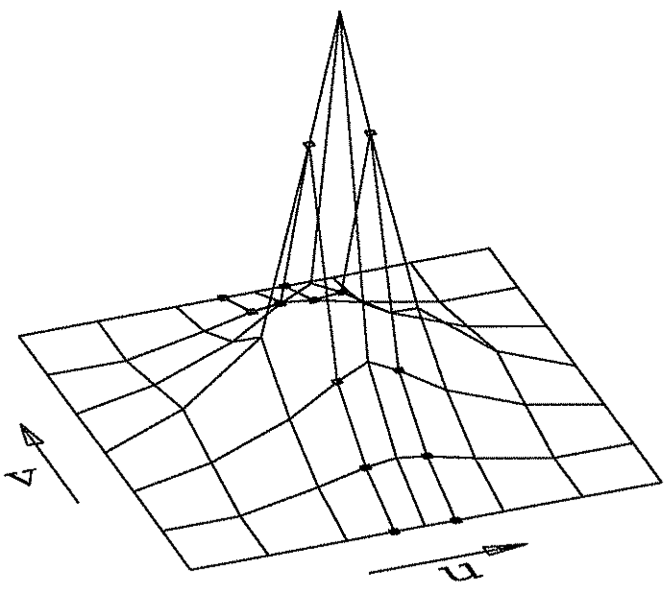



**Figure 3** Surface data set with new data points inserted in direction of parameter $u$

**Interpolation and approximation with ADP**

The interpolation (approximation) constraints result in the following system:

$$\mathbf{A}_u\mathbf{V}\mathbf{A}_v = \mathbf{D} \text{ or } \mathbf{V} = \mathbf{A}_u^{-1}\mathbf{D}\mathbf{A}_v^{-1}. \quad (18)$$

Suppose we have inserted additional data points only in the direction of the parameter $u$ (see Fig. 3), then (18) becomes:

$$\mathbf{V} = \mathbf{A}_u^{-1}(\mathbf{D}_u\mathbf{X}_u+\mathbf{D}_0)\mathbf{A}_v^{-1}, \quad (19)$$

where $\mathbf{X}_u$ is a $r_u \times n_v$ matrix of the unknowns and $\mathbf{D}_0$ is a $n_u \times n_v$ matrix of the known data points.

Substituting (17) and (19) in (16) and finding its minimum leads to a system

$$\mathbf{D}_u^T\mathbf{K}_{Au}\mathbf{D}_u\mathbf{X}_u + \mathbf{D}_u^T\mathbf{K}_{Au}\mathbf{D}_0 = 0, \quad (20)$$

where $\mathbf{K}_{Au}$ and $\mathbf{D}_u$ are the same as in (11) but computed for the knot sequence $u_i$. Finding the solution of (20) requires solving $n_v$ linear systems with $r_u$ unknowns. Note that the system matrix $\mathbf{D}_u^T\mathbf{K}_{Au}\mathbf{D}_u$ is common and it can be computed and decomposed in advance which speeds up the method. A detailed proof of (20) is given in the Appendix.

If we insert additional data points in direction of $v$ we will obtain nearly the same system as (20) but with matrices computed for the sequence $v_i$. Once all the data points have been found the control points are calculated from (18).

**Parameterization and choice of degrees of freedom**

The parameterization and the choice of degrees of freedom need to be slightly modified for tensor product surfaces. New parameter values (data points) for the sequence $u_i$ will appear in each one of the $n_v$ isoparametric curves (see Fig. 3), which may be computed as follows:

1. For each interval $[u_i, u_{i+1}]$, using steps 1-3 from the previous chapter for each curve, find the number $n_{imax} = \max(n_{ij}, j=1,\dots,n_v)$.
2. For each interval $[u_i, u_{i+1}]$ insert $n_{imax}$ new parameter values.
3. Recalculate the sequence $u_i$, using the uniform formula $u_i = i$, thus making the parameterization common for all the isoparametric curves, which is necessary for tensor product surfaces.

Controlling the error of approximation can be done in the same way as for curves but this time the values $L_u$ and $L_v$ for both parameter directions should be varied.

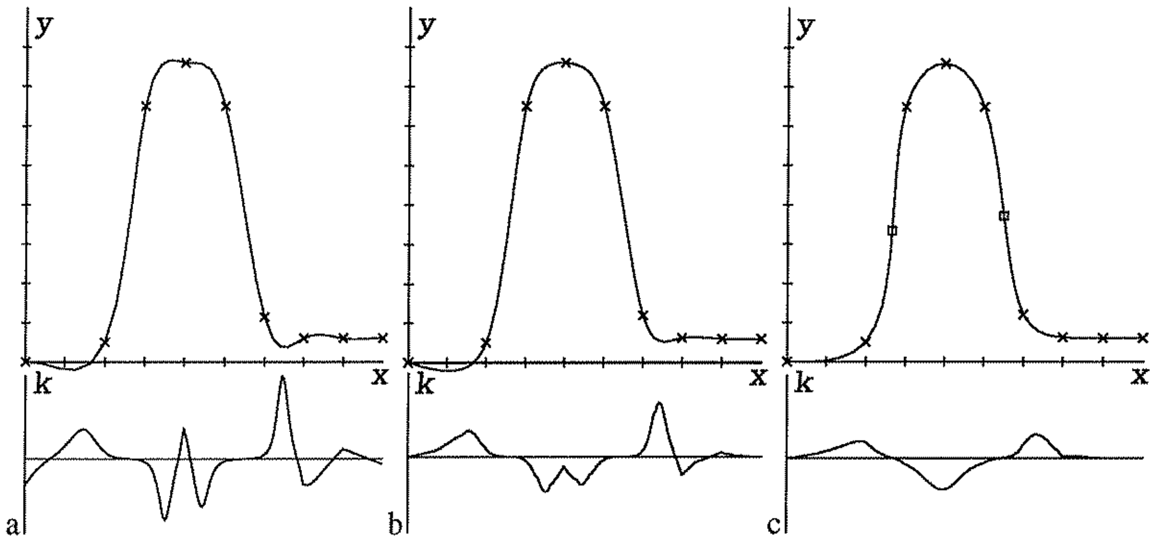



**Figure 4** Data set 1: a. Cubic B-spline interpolation; b. Fair interpolation with ACP; c. Fair interpolation with ADP.

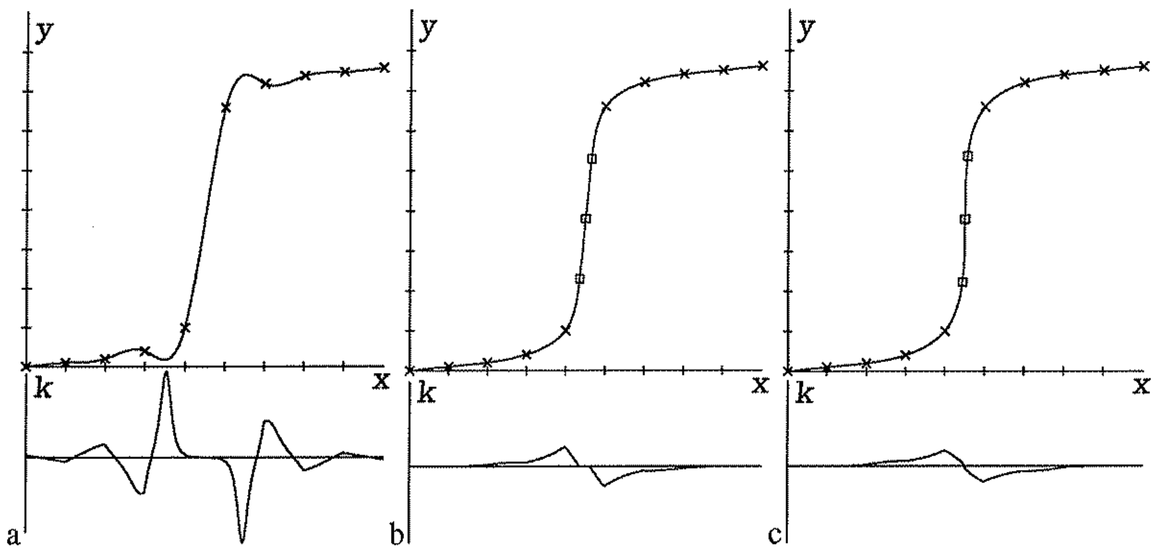



**Figure 5** Data set 2: a. Cubic B-spline interpolation; b. Fair interpolation with ADP and $\alpha$=1.0, $\beta$=0.2; c. Fair interpolation with ADP and $\alpha$=0.2, $\beta$=1.0.

## RESULTS

The algorithms were implemented on a Silicon Graphics Indy workstation using the OpenGL library to render the surfaces. The results are shown in Fig. 4-8. The original data points are marked with crosses and the new ones (inserted by the algorithm) with squares. The curvature distribution is given below every curve as an indicator for the fairness. All the experiments (except Fig. 5) were conducted with coefficient values in (1) $\alpha$=$\beta$=1.

An alternative method for fair interpolation has been implemented that follows from the idea for inserting enough additional control vertices so that the interpolation constraints are decoupled from each other[16]. This method uses the same energy functional but additional control points (ACP) as degrees of freedom. Fig. 4 compares the quality of the schemes with the data set given in Table 1. Type 1 end conditions (given tangent vectors) have been imposed for the cubic B-spline interpolation. The method with ACP obviously improves the smoothness of the curve but it neither cures the violated monotonicity in interval one nor removes the extraneous inflection points in intervals 6, 7.

Fig. 5 demonstrates the results with another data set (Table 2). It also shows the influence of the parameters $\alpha$ and $\beta$, which, as seen in Fig. 5b and c, can be used as shape controls. Fair B-spline least squares approximation (Fig. 6) was applied on 119 points of a face profile obtained through digitization and subject to substantial noise. The effect of fairing is obvious in the areas of the forehead and the mouth. The performance of the different methods for this data set is given in Table 3. The efficiency of the scheme with ADP improves by a factor of 14 compared with that of ACP.

Fig. 7 shows B-spline surface interpolation of the data points in Fig. 3. The unwanted bumps and wiggles (Fig. 7a) were removed by the fairing (Fig. 7b). Least squares B-spline fair approximation has been applied on a "noisy" data set, of about 15000 points. The extraneous undulations (Fig. 8a) were smoothed by the algorithm (Fig. 8b), automatically inserting new data points where necessary. The performance of the different methods for this data set is given in Table 4. It is obvious that the fairing is not too expensive compared to the pure approximation, the quality however is much better.

**Table 1** Data set 1, Figure 4

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| X | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 | 7.0 | 8.0 | 9.0 |
| Y | 0.0 | 0.5 | 6.5 | 7.6 | 6.5 | 1.2 | 0.62 | 0.6 | 0.6 |

**Table 2** Data set 2, Figures 1, 2, 5

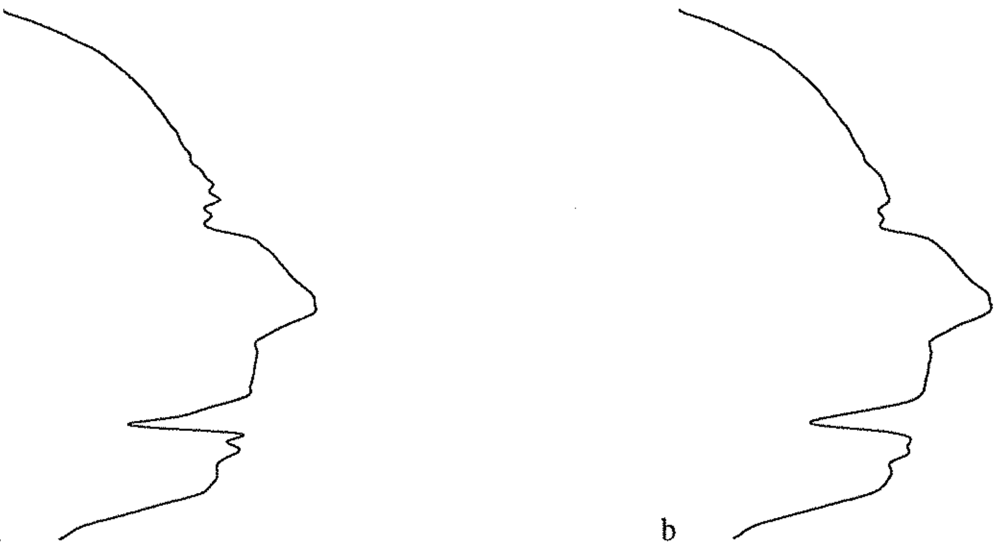
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| X | 0.0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 | 7.0 | 8.0 | 9.0 |
| Y | 0.0 | 0.1 | 0.2 | 0.4 | 1.0 | 6.6 | 7.2 | 7.4 | 7.5 | 7.6 |

**Table 3** Performance of the different methods for approximating curves

| Method | Figure | Number of polynomial segments $L$ | Time (s) |
|---|---|---|---|
| Fair interpolation with ACP | _ | 354 | 71.3 |
| Fair interpolation with ADP | _ | 125 | 4.9 |
| Approximation | Fig. 6 a | 70 | 0.05 |
| Fair approximation with ADP | Fig. 6 b | 70 | 1.2 |

**Table 4** Performance of the different methods for approximating surfaces

| Method | Figure | Number of segments | | Time (s) |
|---|---|---|---|---|
| | | $L_u$ | $L_v$ | |
| Fair interpolation | _ | 130 | 130 | 21.1 |
| Approximation | Fig. 8 a | 56 | 70 | 3.2 |
| Fair approximation | Fig. 8 b | 56 | 70 | 6.8 |



a b

**Figure 6** Face data set: a. Least squares approximation; b. Fair approximation with ADP.

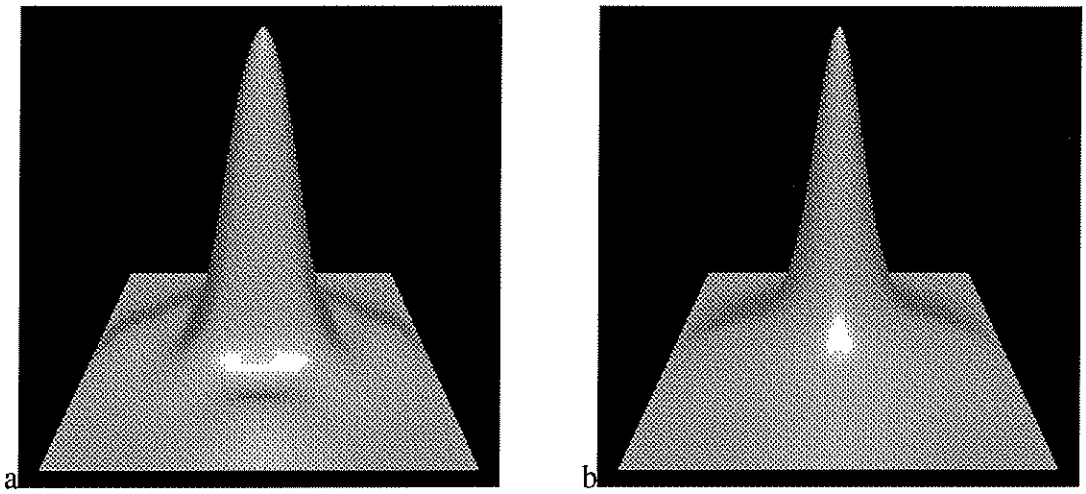


**Figure 7** Surface interpolation: a. Spline interpolation;
b. Fair spline interpolation with ADP.

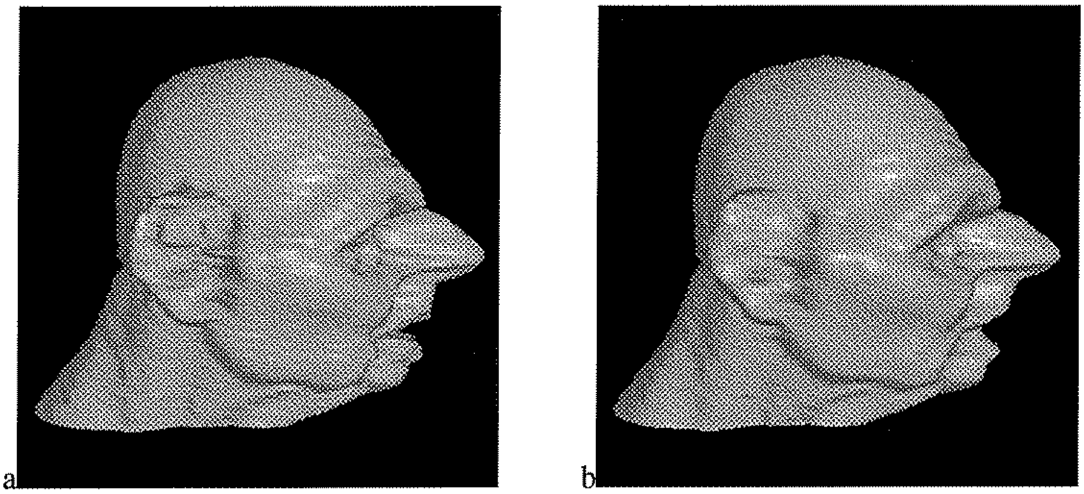


**Figure 8** Face data set: a. Least squares spline approximation;
b. Fair spline approximation with ADP.

## CONCLUSIONS

A method for fair interpolation and approximation of both curve and surface points using B-splines has been described. The suggested energy functional for surfaces (16) preserves the two-dimensionality of the arrays of control and data points. Solving this kind of system, as Farin[1] shows, leads to $O(n^4)$ computations, which are normally $O(n^6)$ if the points are arranged in a vector. Although the algorithms were implemented for the cubic case, they could be easily extended for splines of any order $k$. Additional data points (ADP) are inserted as degrees of freedom for the fairing only where necessary. This approach proved to be very efficient especially for unevenly distributed points, often the case with measured or digitized data. An important feature is that the algorithms for ADP insertion and for the actual fairing are independent. So, if for some practical applications another insertion scheme proves to be more appropriate, it can be used without any limitations with the described fairing procedure. The idea of ADP can be extended to other methods for global fair interpolation (approximation). It will improve the performance, especially when a larger amount of data is concerned.

## ACKNOWLEDGEMENTS

The author would like to thank the British Council and Microsoft who sponsored his work. Figures 6 and 8 are produced from data obtained from a digitization of a Spitting Image puppet of Margaret Thatcher. Thanks to Spitting Image production for providing the puppet and to the UK Advisory Committee on Computer Graphics for funding the digitization. The author is also grateful to his supervisor Mel Slater for his cooperation.

## REFERENCES

1. Farin, G. Curves and Surfaces for Computer Aided Geometric Design. Academic Press, Inc. 1993.
2. de Boor, C. A Practical Guide to Splines. Springer-Verlag, New York, 1978.
3. Lee, E. Choosing nodes in parametric curve interpolation. Computer-Aided Design, Vol. 21, No 6, 1989, pp 363-370.
4. Hartley, P. and Judd, C. Parameterization and shape of B-spline curves. Computer-Aided Design, Vol. 12, No 5, 1980, pp 235-238.
5. Nielson, G. and Foley, T. A survey of applications of an affine invariant norm. In T. Lyche and L. Shumaker, editors, Mathematical Methods in CAGD, pp 445-468, Academic Press, 1989.
6. Marin, S. P. An approach to data parameterization in parametric cubic spline interpolation problems. Approximation Theory, Vol. 41, 1984, pp 64-86.
7. Fritsch, F.N. and Carlson, R.E. Monotone piecewise cubic interpolation. SIAM Journal on Numerical Analysis. Vol. 17, No. 2, April 1980, pp. 238-246.
8. Kjellander, J.A.P. Smoothing of cubic parametric splines. Computer-Aided Design, Vol. 15, No 3, May, 1983, pp 175-179.
9. Rogers, D. F., Satterfield, S. G. and Rodriguez, F. A. Shiphull, B-spline surfaces and CAD/CAM. Computer Graphics - Theory and Applications, Springer-Verlag, 1983.

10. Sapidis, N. and Farin, G. Automatic fairing algorithm for B-spline curves. Computer-Aided Design, Vol. 22, No 2, March, 1990, pp 121-129.

11. Schweikert, D. G. An interpolation curve using a spline in tension. J. Math. Physics, Vol. 45, 1966, pp 312-317.

12. Salkauskas, K. $C^1$ splines for interpolation of rapidly varying data. The Rocky Mountain Journal on Mathematics, Vol. 14, 1984, pp. 239-250.

13. Foley, T. A. A shape preserving interpolant with tension controls. Computer Aided Geometric design, Vol. 5, 1988, pp. 105-118.

14. Celniker, G. and Gossard D. Deformable curve and surface finite elements for free-form shape design. Proceedings of SIGGRAPH, Vol. 25, No 4, July, 1991, pp 257-266.

15. Nowacki, H. and Lü, X. Fairing composite polynomial curves with constraints. Computer Aided Geometric Design, Vol. 11, 1994, pp. 1-15.

16. Halstead, M., Kass, M. and DeRose, T. Efficient, fair interpolation using Catmull-Clark Surfaces. Proceedings of SIGGRAPH, Annual series, 1993, pp 35-44.

17. Ramshaw, L. Blossoms are polar forms. Computer Aided Geometric Design, 6, 1989, pp 232-358.

18. Forsythe, G. and Moler, C. Computer solution of linear algebraic systems. Prentice-Hall Inc., 1967, pp 114-119.

**APPENDIX**

Equation (16) is a sum of four similar terms and here the solution to the first one is given. From (17)

$$\mathbf{w}_{uv}{}^{2} = \dot{\mathbf{B}}_{v}{}^{T}\mathbf{V}^{T}\dot{\mathbf{B}}_{u}\dot{\mathbf{B}}_{u}{}^{T}\mathbf{V}\dot{\mathbf{B}}_{v}, \quad (21)$$

where the dots mean first derivative. We have to compute

$$\frac{\partial}{\partial \mathbf{X}_{u}}\iint c_{1}\mathbf{w}_{uv}{}^{2}\mathrm{d}u\mathrm{d}v = c_{1}\frac{\partial}{\partial \mathbf{X}_{u}}\iint \dot{\mathbf{B}}_{v}{}^{T}\mathbf{V}^{T}\dot{\mathbf{B}}_{u}\dot{\mathbf{B}}_{u}{}^{T}\mathbf{V}\dot{\mathbf{B}}_{v}\mathrm{d}u\mathrm{d}v =$$

$$c_{1}\frac{\partial}{\partial \mathbf{X}_{u}}\iint \dot{\mathbf{B}}_{v}{}^{T}\mathbf{A}_{v}{}^{-T}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})^{T}\mathbf{A}_{u}{}^{-T}\dot{\mathbf{B}}_{u}\dot{\mathbf{B}}_{u}{}^{T}\mathbf{A}_{u}{}^{-1}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})\mathbf{A}_{v}{}^{-1}\dot{\mathbf{B}}_{v}\mathrm{d}u\mathrm{d}v =$$

$$2c_{1}\iint \mathbf{D}_{u}{}^{T}\mathbf{A}_{u}{}^{-T}\dot{\mathbf{B}}_{u}\dot{\mathbf{B}}_{u}{}^{T}\mathbf{A}_{u}{}^{-1}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})\mathbf{A}_{v}{}^{-1}\dot{\mathbf{B}}_{v}\dot{\mathbf{B}}_{v}{}^{T}\mathbf{A}_{v}{}^{-T}\mathrm{d}u\mathrm{d}v =$$

$$2c_{1}\mathbf{D}_{u}{}^{T}\mathbf{A}_{u}{}^{-T}(\int \dot{\mathbf{B}}_{u}\dot{\mathbf{B}}_{u}{}^{T}\mathrm{d}u)\mathbf{A}_{u}{}^{-1}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})\mathbf{A}_{v}{}^{-1}(\int \dot{\mathbf{B}}_{v}\dot{\mathbf{B}}_{v}{}^{T}\mathrm{d}v)\mathbf{A}_{v}{}^{-T} =$$

$$2c_{1}\mathbf{D}_{u}{}^{T}\mathbf{A}_{u}{}^{-T}\mathbf{K}_{1u}\mathbf{A}_{u}{}^{-1}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})\mathbf{A}_{v}{}^{-1}\mathbf{K}_{1v}\mathbf{A}_{v}{}^{-T}. \quad (22)$$

Results similar to (22) will be obtained solving the remaining three terms of (16). Then the system of equations reads (⇔ means if and only if)

$$\mathbf{D}_{u}{}^{T}\mathbf{A}_{u}{}^{-T}(\alpha_{u}\mathbf{K}_{1u}+\beta_{u}\mathbf{K}_{2u})\mathbf{A}_{u}{}^{-1}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})\mathbf{A}_{v}{}^{-1}(\alpha_{v}\mathbf{K}_{1v}+\beta_{v}\mathbf{K}_{2v})\mathbf{A}_{v}{}^{-T} = 0 \Leftrightarrow$$

$$\mathbf{D}_{u}{}^{T}\mathbf{A}_{u}{}^{-T}\mathbf{K}_{u}\mathbf{A}_{u}{}^{-1}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})\mathbf{A}_{v}{}^{-1}\mathbf{K}_{v}\mathbf{A}_{v}{}^{-T} = 0 \Leftrightarrow$$

$$\mathbf{D}_{u}{}^{T}\mathbf{K}_{Au}(\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{0})\mathbf{K}_{Av} = 0. \quad (23)$$

Here $c_1 = \alpha_u\alpha_v$, $c_2 = \beta_u\alpha_v$, $c_3 = \alpha_u\beta_v$, $c_4 = \beta_u\beta_v$. The matrix $\mathbf{K}_{Av}$ is symmetric, band and positive definite and its inverse $\mathbf{K}_{Av}^{-1}$ exists. Multiplying both sides of (23) with $\mathbf{K}_{Av}^{-1}$ results in

$$\mathbf{D}_{u}{}^{T}\mathbf{K}_{Au}\mathbf{D}_{u}\mathbf{X}_{u}+\mathbf{D}_{u}{}^{T}\mathbf{K}_{Au}\mathbf{D}_{0} = 0. \quad (24)$$