process is also studied with different energy forms and parametrization techniques. The test results of the surface interproximation process also show the same conclusion as the curve interproximation process.

The principal contributions of this paper are:

- It shows that the energy form has a much bigger impact on the generated curve/surface than the parametrization technique in the curve/surface interproximation process.
- It shows that one should avoid using approximated curve/surface energy forms in a curve/surface interproximation process since the generated curve/surface would be far from being a good approximation to the optimal curve/surface.
- It presents a surface interproximation scheme which guarantees C^2-continuity between adjacent patches.

Details of the paper are shown in the subsequent sections.

2. ANOTHER LOOK AT CURVE INTERPROXIMATION

Given a set of n 3D data

$$D_i = A_i \times B_i \times C_i$$

where

$$A_i = [a_i, b_i]; B_i = [c_i, d_i]; C_i = [e_i, f_i] \tag{1}$$

with $a_i \leqslant b_i, c_i \leqslant d_i$ and $e_i \leqslant f_i$, the objective is to construct a curve passing through all D_i with relatively smooth shape. This may be achieved by solving a *strain energy minimization problem* using cubic B-spline curves[5]. Note that D_i could be a point, a line segment, a rectangle, or a parallelepiped.

The strain energy of a curve segment $C(t)$ is defined as the integral of the squared curvature over the parameter space I[8]

$$E(C) = \int \kappa^2(s)\mathrm{d}s = \int_I \left(\frac{|C_t \times C_{tt}|}{|C_t|^3}\right)^2 |C_t|\mathrm{d}t \tag{2}$$

where $\mathrm{d}s$ is the curve length measure, and C_t and C_{tt} are the first and second derivatives of C with respect to t, respectively. Since it was commonly agreed that computing the exact strain energy is too difficult to be used as a design tool, several approximation methods had been developed. When the first derivative is small and perpendicular to the second derivative, the curve energy is typically approximated by the following form[4,5,8,15,26]:

$$E(C) = \int_I |C_{tt}|^2\mathrm{d}t \tag{3}$$

If $|C_t|$ is close to a constant, the curve energy has also been approximated by[12,17]

$$E(C) = \frac{1}{m(|C_t|)^3}\int_I |C_{tt}|^2\mathrm{d}t \tag{4}$$

where $m(|C_t|)$ denotes some kind of average of $|C_t|$ over the parameter space.

Another option is to use Kallay and Ravani's surface energy approximation for a curve[14]. In that case, the energy of a curve is approximated by the integral of a general quadratic form in the second derivatives of the curve over the parameter space:

$$E(C) = \int_I \Delta \cdot Q \cdot \Delta^{\mathrm{T}}\mathrm{d}t \tag{5}$$

where Q is a 3×3 symmetric coefficient matrix and $\Delta = (x_{tt}, y_{tt}, z_{tt})$ with $x(t), y(t)$ and $z(t)$ being the x, y and z components of $C(t)$, respectively. When Q is an identity matrix, one gets (3) as a special case.

On the other hand, if $|C_t|$ is close to a constant, since the computation of the curve energy defined in (2) is essentially a process of computing the integral of $|C_t \times C_{tt}|^2$, one has

$$\int_I |C_t \times C_{tt}|^2\mathrm{d}t = \int_I \Lambda \cdot R \cdot \Lambda^{\mathrm{T}}\mathrm{d}t \tag{6}$$

where $\Lambda = (y_t z_{tt}, y_{tt} z_t, x_{tt} z_t, x_t z_{tt}, x_t y_{tt}, x_{tt} y_t)$ and

$$R = \begin{pmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{pmatrix} \quad \text{with} \quad B = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \tag{7}$$

Hence, by allowing R to be an arbitrarily defined 6×6 symmetric coefficient matrix, one gets another general form for curve energy if the first derivative is close to a constant.

Minimizing strain energy has been used as a common means to optimize the shape of a curve or surface in shape design applications. It is important to know which of all the energy approximation methods provides the best approximation. In the following, we will investigate the performance of the above energy forms, including (2), in the curve interproximation process. Namely, in finding the optimal control points $P_i = (x_i, y_i, z_i)$ for a cubic B-spline curve $C(t)$ that interproximates D_i, we will use different energy functions in the optimization process to determine which one gives the best result. Furthermore, since the outcome of the interproximation process depends on knot parametrization too, the investigation will consider the impact of different parametrization techniques as well.

Let $(t_i), i = -2, -1, \ldots, n+3$, be a knot vector constructed using one of the following three methods: the uniform model, the centripetal model[18], or the relative chord length model[1,6]. Specifically, $t_{-2} = t_{-1} = t_0 = t_1 = 0, t_n = t_{n+1} = t_{n+2} = t_{n+3} = 1$, and

$$t_i - t_{i-1} = \frac{1}{(n-1)}, \qquad i = 2, \ldots, n-1 \tag{8}$$

for the uniform model,

$$t_i - t_{i-1} = \frac{|Q_i - Q_{i-1}|^{1/2}}{\sum_{j=2}^{n} |Q_j - Q_{j-1}|^{1/2}}, \qquad i = 2, \ldots, n-1 \tag{9}$$

for centripetal model, or

$$t_i - t_{i-1} = \frac{|Q_i - Q_{i-1}|}{\sum_{j=2}^{n} |Q_j - Q_{j-1}|}, \qquad i = 2, \ldots, n-1 \tag{10}$$

for the relative chord length model, where Q_i is the center

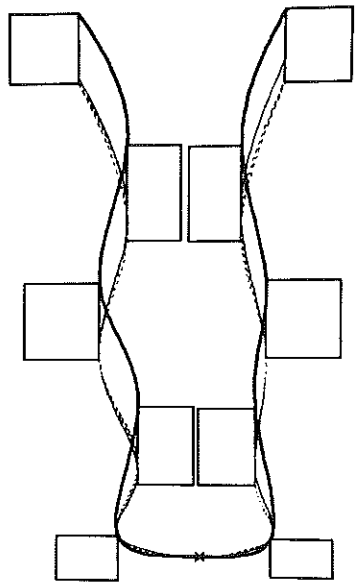


Figure 1 Case 1(a): cubic B-spline curve interproximation

Q in (14) is set to an identity matrix; for (5) the matrix Q in (14) is defined as follows:

$$Q = \begin{pmatrix} 0.90 & 0.10 \\ 0.10 & 0.90 \end{pmatrix}$$

Both matrices are 2×2 because $C(t)$ is a 2D curve. The objective function for (6) is constructed using (16) with R defined in (7). The objective function for (2) is an approximation of (2) constructed using the trapezoid method with 10 points for each span.

The minimization of the objective functions for (3) and (5) are carried out using the *E04NFF* subroutine *NAG*. This is a global optimization process for quadratic programming problems (the objective functions for (3) and (5) are quadratic functions). The solution returned by this subroutine is a global minimum. The initial configuration of the control points for (3) and (5) are the same. First, a set of interpolation points $Q_i = (a_i, b_i, c_i)$ is constructed as follows: for $i = 1, \ldots, n$, if D_i is a point, then Q_i is set to D_i, otherwise, Q_i is set to the center of D_i.

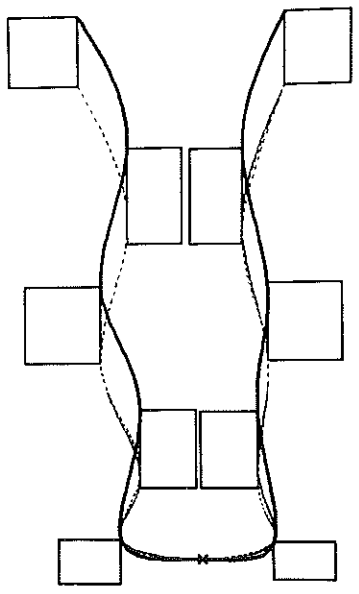


Figure 2 Case 1(b): cubic B-spline curve interproximation

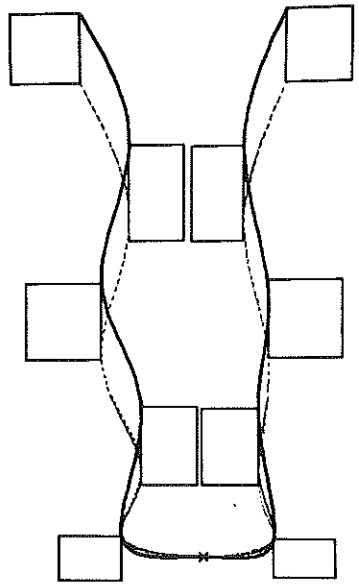


Figure 3 Case 1(c): cubic B-spline curve interproximation

Then the following equation is solved for the initial control points $P_i = (x_i, y_i, z_i)$ (see (15) for the definition of G):

$$(X, Y, Z)\begin{bmatrix} G & 0 & 0 \\ 0 & G & 0 \\ 0 & 0 & G \end{bmatrix} = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_n)$$

The objective functions for (2) and (6) are minimized using the *E04UCF* subroutine of *NAG*. This is a local optimization procedure for non-linear programming problems (recall that the objective functions for (2) and (6) are neither linear nor quadratic). To facilitate the minimum finding process, the solution returned by the subroutine *E04NFF* for (3) is used as the initial configuration of the control points for both (2) and (6). It is not guaranteed that in this case the solution returned by *E04UCF* is a global minimum. However, the resulting curve seems to be reasonably close to a global minimum. This can also be justified by comparing the (exact) strain energy of the resulting curve with the energies of the curves obtained using (3) and (5). The (exact) strain energies and the computation times of the interproximating curves are shown in *Tables 1* and *2*, respectively.

It can be seen from the test cases that the energy form

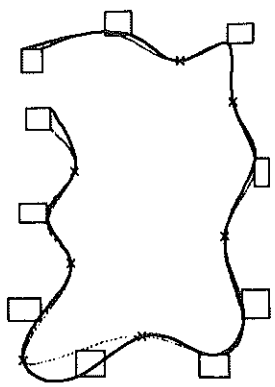


Figure 4 Case 2(a): cubic B-spline curve interproximation

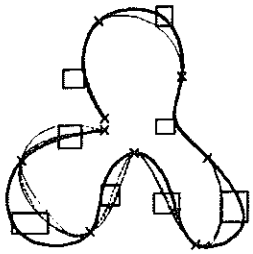


Figure 9 Case 3(c): cubic B-spline curve interproximation

parameter knots in u and $(n+6)$ parameter knots in v. Three different methods may be used to construct the uv parameter knots for the surface.

(1) *Uniform model*: the u and v parameter knots are evenly distributed (see (8)). The u parameter knots are: 0, 0, 0, 0, $1/(m-1), 2/(m-1), \ldots, (m-2)/(m-1)$, 1, 1, 1, 1. The v parameter knots are: 0, 0, 0, 0, $1/(n-1)$, $2/(n-1), \ldots, (n-2)/(n-1)$, 1, 1, 1, 1.

(2) *Average centripetal model*: for each row of the interproximating data, one u knot vector can be computed by the centripetal model, such as the one calculated in (9). After all the u knot vectors have been calculated, an average method is used to obtain the final u knot vector for the surface[22]. A similar approach can be used to obtain the v knot vector.

(3) *Average relative chord length model*: for each row of the interproximating data, one u knot vector can be computed by the relative chord length model, such as the one calculated in (10). After all the u knot vectors have been calculated, an average method is used to obtain the final u knot vector for the surface. A similar approach can be used to obtain the v knot vector.

Let $U = u_{-2}, u_{-1}, \ldots, u_{m+3}$ and $V = v_{-2}, v_{-1}, \ldots, v_{n+3}$ be the u and v knot vectors computed using one of the above approaches. The bicubic B-spline surface we are looking for may be expressed as:

$$S(u,v) = \sum_{i=1}^{m}\sum_{j=1}^{n} P_{i,j}, w_{i,3}(u)w_{j,3}(v), \quad u,v \in [0,1] \quad (18)$$

where $w_{i,3}(u)$ and $w_{j,3}(v)$ are defined in (12) with corresponding u and v knots and $P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ are 3D control points. The $P_{i,j}$ are to be chosen so that the strain energy of $S(u,v)$ over the parameter space is

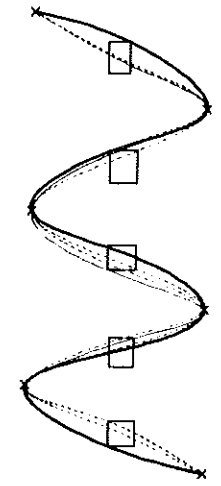


Figure 10 Case 4(a): cubic B-spline curve interproximation

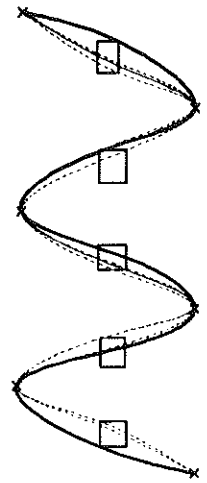


Figure 11 Case 4(b): cubic B-spline curve interproximation

minimum with the constraint that $S(u_i, v_j) \in D_{i,j}$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

The strain energy of a surface patch $S(u,v)$ is defined as the integral of the squared principal curvatures over the parameter space[10,14]:

$$E(S) = \int\int_D (\kappa_1^2 + \kappa_2^2) \mathrm{d}\sigma \quad (19)$$

where $\mathrm{d}\sigma$ is the surface area measure. Using the following equation:

$$\kappa_1^2 + \kappa_2^2 = (2H)^2 - 2\kappa$$

where H is the mean curvature and κ is the Gaussian curvature[5], one can get the energy form in terms of the first and second surface derivatives as follows:

$$E(S) = \int\int_D \left[\left(\frac{LG - 2MF + NE}{EG - F^2} \right)^2 - 2\frac{LN - M^2}{EG - F^2} \right] (EG - F^2)^{1/2} \mathrm{d}u\mathrm{d}v \quad (20)$$

where

$$E = S_u \cdot S_u \qquad F = S_u \cdot S_v \qquad G = S_v \cdot S_v$$
$$L = \mathbf{n} \cdot S_{uu} \qquad M = \mathbf{n} \cdot S_{uv} \qquad N = \mathbf{n} \cdot S_{vv}$$

and $\mathbf{n}$ is the unit normal vector.

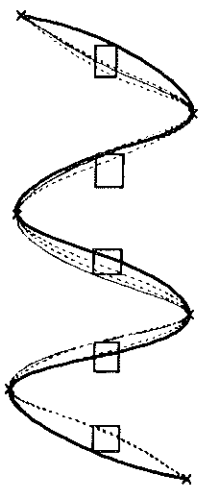


Figure 12 Case 4(c): cubic B-spline curve interproximation

Table 3 Strain energies and computation times (in second) of the interproximation surfaces

Energy Form	Uniform		Centripetal		Chord length	
	Energy	Time	Energy	Time	Energy	Time
(19)	43.401923	289.590000	43.546302	289.950000	52.730120	288.890000
(29)	56.154762	0.170000	55.125100	0.160000	66.741664	0.230000

considered too difficult to be used as a design tool, several approximation methods have been used. These include the *thin plate model*[3,24,25]

$$E(S) = \int\int_D [|S_{uu}|^2 + 2|S_{uv}|^2 + |S_{vv}|^2]\mathrm{d}u\mathrm{d}v, \tag{25}$$

which is a small deflection approximation of the surface curvature, the *membrane model*[27]

$$E(S) = \int\int_D [|S_u|^2 + |S_v|^2]\mathrm{d}u\mathrm{d}v \tag{26}$$

which is a small deflection approximation of the surface area, and a generalized form of the thin plate model[14]

$$E(S) = \int\int_D (\delta \cdot Q\dot{\delta}^{\mathrm{T}})\mathrm{d}u\mathrm{d}v \tag{27}$$

where Q is a 9×9 symmetric coefficient matrix and $\delta = (x_{uu}, x_{uv}, x_{vv}, y_{uu}, y_{uv}, y_{vv}, z_{uu}, z_{uv}, z_{vv})$, where x, y and z are the x-, y- and z-components of $S(u, v)$. Different Qs define different energy approximation forms.

It is of interest to evaluate these energy approximation methods, compared with the exact energy form. Since the membrane model (27) is not a good smoothing tool (it tends to generate interpolating surfaces with visible peaks and dips[27] and the thin plate model (25) is a special case of (27), it is only necessary to compare (23) with (27). (27) can be computed as follows.

Using the representations of $x(u, v), y(u, v)$ and $z(u, v)$ in (21), *DELTA* can be expressed as

$$\delta = (P^x, P^y, P^z) \cdot \begin{pmatrix} W_{uu} & W_{uv} & W_{vv} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_{uu} & W_{uv} & W_{vv} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_{uu} & W_{uv} & W_{vv} \end{pmatrix} \tag{28}$$

where W_{uu}, W_{uv} and W_{vv} are the second partial derivatives of W with respect to u and v, and 0 is a zero vector of the same dimension as W. Hence, the energy form (27) can be expressed as

$$E(S) = (P^x, P^y, P^z) \cdot M \cdot \begin{pmatrix} (P^x)^{\mathrm{T}} \\ (P^y)^{\mathrm{T}} \\ (P^z)^{\mathrm{T}} \end{pmatrix} \tag{29}$$

where M is a 3×3 block matrix

$$M = \begin{pmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \\ M_{3,1} & M_{3,2} & M_{3,3} \end{pmatrix}$$

with each block being an $mn \times mn$ matrix defined as follows:

$$\begin{aligned}
M_{1,1} &= Q_{1,1}A_{1,1} + Q_{2,1}A_{2,1} + Q_{3,1}A_{3,1} + Q_{1,2}A_{1,2} \\
&\quad + Q_{2,2}A_{2,2} + Q_{3,2}A_{3,2} + Q_{1,3}A_{1,3} + Q_{2,3}A_{2,3} + Q_{3,3}A_{3,3} \\
M_{2,1} &= Q_{4,1}A_{1,1} + Q_{5,1}A_{2,1} + Q_{6,1}A_{3,1} + Q_{4,2}A_{1,2} + Q_{5,2}A_{2,2} \\
&\quad + Q_{6,2}A_{3,2} + Q_{4,3}A_{1,3} + Q_{5,3}A_{2,3} + Q_{6,3}A_{3,3} \\
M_{3,1} &= Q_{7,1}A_{1,1}, + Q_{8,1}A_{2,1} + Q_{9,1}A_{3,1} + Q_{7,2}A_{1,2} + Q_{8,2}A_{2,2} \\
&\quad + Q_{9,2}A_{3,2} + Q_{7,3}A_{1,3} + Q_{8,3}A_{2,3} + Q_{9,3}A_{3,3} \\
M_{1,2} &= Q_{1,4}A_{1,1} + Q_{2,4}A_{2,1} + Q_{3,4}A_{3,1} + Q_{1,5}A_{1,2} + Q_{2,5}A_{2,2} \\
&\quad + Q_{3,5}A_{3,2} + Q_{1,6}A_{1,3} + Q_{2,6}A_{2,3} + Q_{3,6}A_{3,3} \\
M_{2,2} &= Q_{4,4}A_{1,1} + Q_{5,4}A_{2,1} + Q_{6,4}A_{3,1} + Q_{4,5}A_{1,2} + Q_{5,5}A_{2,2} \\
&\quad + Q_{6,5}A_{3,2} + Q_{4,6}A_{1,3} + Q_{5,6}A_{2,3} + Q_{6,6}A_{3,3} \\
M_{3,2} &= Q_{7,4}A_{1,1}, + Q_{8,4}A_{2,1} + Q_{9,4}A_{3,1} + Q_{7,5}A_{1,2} + Q_{8,5}A_{2,2} \\
&\quad + Q_{9,5}A_{3,2} + Q_{7,6}A_{1,3} + Q_{8,6}A_{2,3} + Q_{9,6}A_{3,3} \\
M_{1,3} &= Q_{1,7}A_{1,1} + Q_{2,7}A_{2,1} + Q_{3,7}A_{3,1} + Q_{1,8}A_{1,2} + Q_{2,8}A_{2,2} \\
&\quad + Q_{3,8}A_{3,2} + Q_{1,9}A_{1,3} + Q_{2,9}A_{2,3} + Q_{3,9}A_{3,3} \\
M_{2,3} &= Q_{4,7}A_{1,1} + Q_{5,7}A_{2,1} + Q_{6,7}A_{3,1} + Q_{4,8}A_{1,2} + Q_{5,8}A_{2,2} \\
&\quad + Q_{6,8}A_{3,2} + Q_{4,9}A_{1,3} + Q_{5,9}A_{2,3} + Q_{6,9}A_{3,3} \\
M_{3,3} &= Q_{7,7}A_{1,1} + Q_{8,7}A_{2,1} + Q_{9,7}A_{3,1} + Q_{7,8}A_{1,2} + Q_{8,8}A_{2,2} \\
&\quad + Q_{9,8}A_{3,2} + Q_{7,9}A_{1,3} + Q_{8,9}A_{2,3} + Q_{9,9}A_{3,3}
\end{aligned}$$

where $Q_{i,j}$ is the (i,j)th entry of Q and

$$\begin{aligned}
A_{1,1} &= \int\int_D W_{uu} \cdot W_{uu}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v, \\
A_{2,1} &= \int\int_D W_{uv} \cdot W_{uu}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v, \\
A_{3,1} &= \int\int_D W_{vv} \cdot W_{uu}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v \\
A_{1,2} &= \int\int_D W_{uu} \cdot W_{uv}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v, \\
A_{2,2} &= \int\int_D W_{uv} \cdot W_{uv}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v, \\
A_{3,2} &= \int\int_D W_{vv} \cdot W_{uv}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v, \\
A_{1,3} &= \int\int_D W_{uu} \cdot W_{vv}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v, \\
A_{2,3} &= \int\int_D W_{uv} \cdot W_{vv}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v, \\
A_{3,3} &= \int\int_D W_{vv} \cdot W_{vv}^{\mathrm{T}}\mathrm{d}u\mathrm{d}v
\end{aligned}$$

(29) is a *quadratic form* of its control points. Each component of a control point is treated as a variable in this quadratic form. Consequently, one can again use the

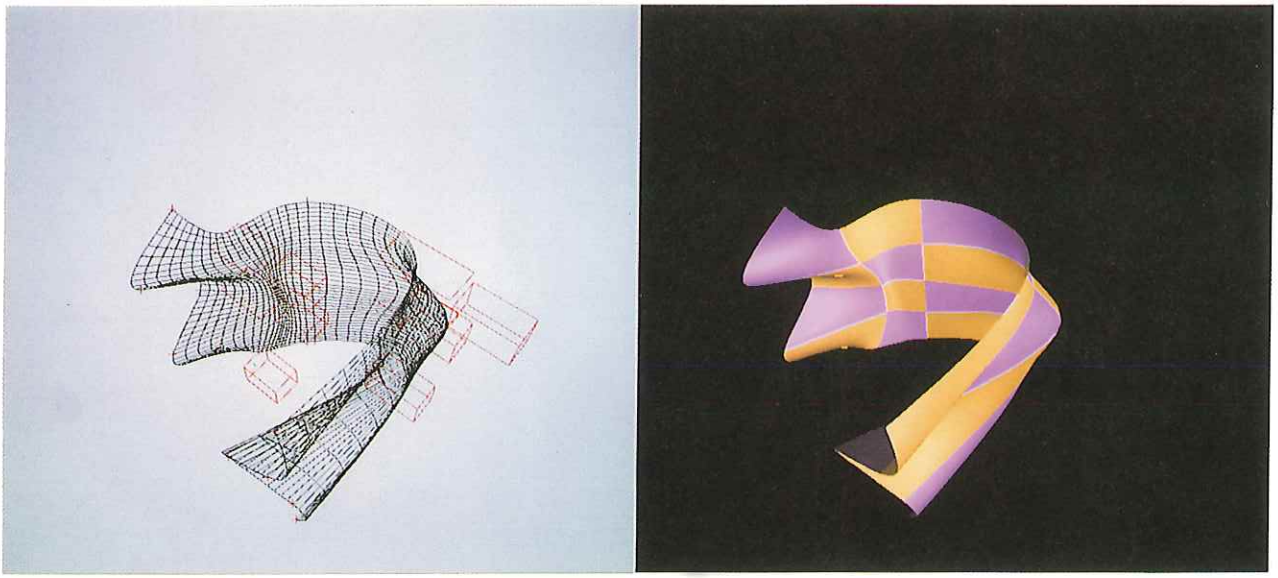


Figure 15 Case (c): combination of energy form (19) and average centripetal model

flatter regions are expected. The computation times shown in *Table 3* indicate that minimizing the exact energy form is not that terrible for surface interproximation either.

6. SUMMARY

A study of the cubic curve interproximation process with different energy forms and parametrization techniques is performed. The results of the study show that the energy form plays a much more significant role than the parametrization technique on the shape and smoothness of the resulting curves, and the curves generated by the approximated energy forms tend to have flatter regions and sharper turns than, and are far from being good approximation to, those generated by the exact energy form. Hence, one should avoid using approximated energy forms in a shape optimization process where minimization of the curve energy is required.

An interproximation technique for bicubic B-spline surfaces is also presented. A bicubic B-spline surface that interproximates the given data (points or regions) and has a relatively smooth shape is generated by minimizing the strain energy of the surface. The exact surface energy form and a general approximated surface energy form have both been used in the energy minimization process with three different parametrization techniques. The test results of the surface interproximation process show the same conclusion as the curve interproximation process.

The 'fairness' functional considered in this paper is the strain energy of a curve or surface. Several new fairness functionals have been used recently[11,20]. Moreton and Sequin's approach[20], which adds a measure on curvature variation, produces the most impressive surfaces. The only problem with their approach is its expensiveness. However, with the advance of hardware and software, this certainly could be a possible direction for surface interproximation to look at in the future.

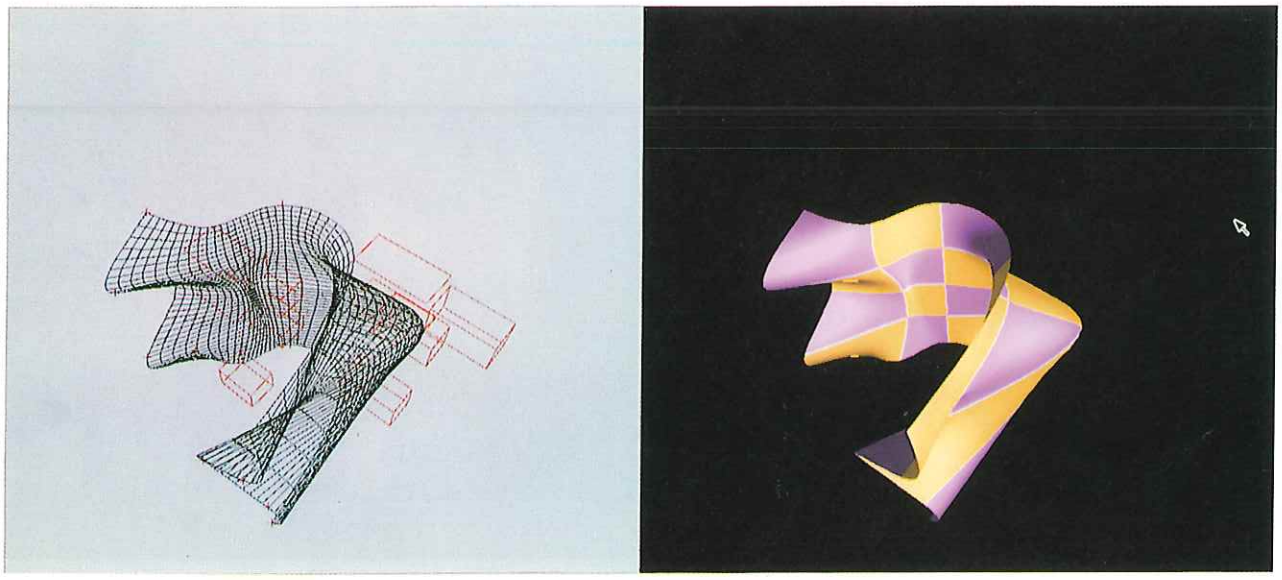
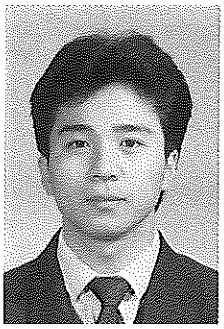
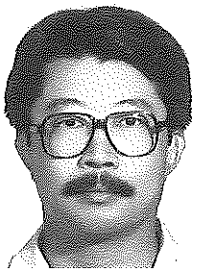


Figure 16 Case (d): combination of energy form (3.29) and average centripetal model

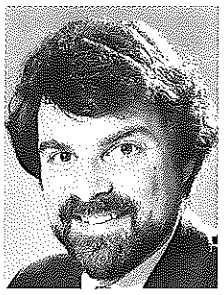
15. Kjellander, J. A. P., Smoothing of cubic parametric splines. *Computer-Aided Design*, 1983, **15**(5), 175–179.
16. Kjellander, J. A. P., Smoothing of bicubic parametric surfaces. *Computer-Aided Design*, 1983, **15**(5), 288–293.
17. Lee, E. T. Y., Energy, fairness and a counterexample. *Computer-Aided Design*, 1990, **22**(1), 37–40.
18. Lee, E. T. Y., On choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 1989, **21**(6), 363–370.
19. Lounsbery, M., Mann, S. and DeRose, T., Parametric surface interpolation. *IEEE Computer Graphics & Applications*, 1992, **12**(5), 45–52.
20. Moreton, H. P. and Sequin, C. H., Functional optimization for fair surface design. *SIGGRAPH '92*, July 1992, 167–176.
21. *NAG FORTRAN Library Manual, Mark 16*, 1st Edition, Sept. 1993.
22. Piegl, L., On NURBS: a survey. *IEEE CG&A*, 1991, **11**(1), 55–71.
23. Powell, M. J. D., Variable metric methods for constrainted optimization. In *Mathematical Programming: The State of the Art*, eds. A. Bachem, M. Grotschel and B. Korte. Springer-Verlag, Berlin, 1983, pp. 288–311.
24. Quak, E. and Schumaker, L. L., Calculation of the energy of a piecewise polynomial surface. In *Algorithms for Approximation II*, eds. M. G. Cox and J. C. Mason. Clarendon Press, Oxford, 1989, pp. 134–143.
25. Quak, E. and Schumaker, L. L., Cubic spline fitting using data dependent triangulations. *Computer-Aided Geometric Design*, 1990, **7**, 293–301.
26. Ritter, K., Generalized spline interpolation and nonlinear programming. *Approximation with Special Emphasis on Spline Functions*, ed. I. J. Schoenberg. Academic Press, London, 1969, pp. 75–118.
27. Szeliski, R., Fast surface interpolation using hierarchial basis functions. *IEEE Trans. Pattern Anal. Machine Intell.*, 1990, **12**, 513–528.



Xuefu Wang is a post doctoral fellow in the Department of Computer Science at the University of Kentucky, USA. He attended Tsinghua University, Beijing, China, in 1986, and received his BE, MSc, and PhD degrees in computer science in 1991, 1994 and 1994, respectively. His research interests include computer graphics, computer-aided geometric modeling and image processing.



Fuhua (Frank) Cheng is an associate professor of computer science at the University of Kentucky, USA, where he joined the faculty in 1986. He has held visiting positions at the University of Tokyo and the University of Aizu, Japan. He received a BS and an MS in mathematics from the National Tsing Hua University in Taiwan in 1973 and 1975, respectively, an MS in mathematics, an MS in computer science, and a PhD in applied mathematics and computer science from the Ohio State University, USA, in 1978, 1980 and 1982, respectively. His research interests include computer-aided geometric modeling, computer graphics, and parallel computing in geometric modeling and computer graphics.



Brian A Barsky is Professor of Computer Science and Affiliate Professor of Optometry and Vision Science at the University of California at Berkeley, where he joined the faculty in 1981. He is a member of the Bioengineering Graduate Group, an interdisciplinary and intercampus program, between UC Berkeley and UC San Francisco. He attended McGill University where he received a DCS in engineering in 1973 and a BSc in mathematics and computer science in 1976. He studied computer graphics and computer science at Cornell University where he earned an MS degree in 1978. His PhD degree is in computer science from the University of Utah in 1981. His research interests include computer-aided geometric design and modeling, interactive three-dimensional computer graphics, visualization in scientific computing, computer-aided cornea modeling and visualization. Dr Barsky is an area editor of CVGIP: Graphical Models and Image Processing.