Neural, Parallel & Scientific Computations 5 (1997) 37-54

SURFACE DESIGN BASED ON HERMITE SPLINE INTERPOLATION WITH TENSION CONTROL AND OPTIMAL TWIST VECTORS

Xuefu Wang and Fuhua (Frank) Cheng
Department of Computer Science, University of Kentucky, Lexington KY 40506-0046

ABSTRACT: A new approach to the problem of constructing a smooth surface to interpolate a grid of data points is presented. A network of cubic Hermite spline curves which interpolate the data points is constructed first. Tension control is allowed in the interpolation mode for these curves. A bicubic Hermite spline surface which interpolates the network of curves is then constructed. The surface is constructed by minimizing its strain energy to generate optimal twist vectors at the data points. A new representation scheme is used for the Hermite spline curves and surfaces. The new representation scheme facilitates the energy minimization process. This work also shows that it is not necessary to require C^2 patch boundaries to produce an overall C^1 surface.

Keywords - Hermite spline curves/surfaces, tension, geometric/parametric continuity, twist vectors, strain (elastic) energy, interpolation

1. INTRODUCTION

Constructing a smooth surface to interpolate a grid of 3D data points is one of the popular methods in shape modeling and design. Another popular method is constructing a smooth surface to interpolate a network of curves. Both problems have been extensively studied (Bartels, 1987; Farin, 1988).

The first approach usually uses B-spline type surface schemes for the interpolation process. One problem with this approach is the selection of parameter space. Several models are available (Piegl, 1991), such as the *average centripetal model*, the *average chord length model*, and the *uniform model*. However, since the u knots and v knots simply can not be selected to reflect the distribution of the data points as in the curve case (Lee, 1989), one sometime would get large tangent vectors for data points relatively close to each other. This situation is especially severe when the data points are unevenly distributed and, consequently, causes the *rollback* (or, *folding*) and *bump* phenomena. For instance, a bicubic B-spline surface which interpolates a given data set (Figure 4) is shown in Figure 5. The u knots and v knots of this surface are defined by the average centripetal model. This surface carries both the *rollback* and *bump* phenomena. The rollback phenomenon, in the form of a hidden overlapping area, occurs in the patches bounded by curves A, B, C, and D; one such patch is shown in Figure 6. This is because while the u tangent vectors at the lower portions of curves A and B acting as shearing forces to cause oscillation of the surface between these curves, the increasingly closer horizontal distance between these curves forces the oscillation area to fall within a narrower and narrower region and, eventually, touch each other and form a banded overlapping area.

The second approach is usually preferable than the first approach, in that it gives designer a better control of the final shape of the surface by specifying its cross-sectional sha beforehand. A problem with this approach is the setting of the *twist vectors* at the data po (Farin, 1988). The key is to avoid the *wrinkle phenomenon* (Kallay, 1990) or *flat spots* (Fa 1988) associated with zero twist vectors. Several approaches for estimating the values of twist vectors are available (Akima, 1978; Barnhill, 1978; Barnhill, 1988; Carmichael, 19 Dube, 1975) (see (Farin, 1988) for a review of some of these methods). An increasingly pop method is to use twist vectors that minimize some surface energy form (Cheng, 1994; Frie 1986; Hagen, 1987; Kallay, 1990; Moreton, 92; Szeliski, 1990).

A capability usually required in the shape design process is shape control of the resul surface. The shape control process should be easy to use for (non-mathematician) end-design Among all the possible shape parameters existing in various surface representations such knots, weights, tension, bias, and curvature, only the tension effect seems to be intuitive (Bl 1995). A shape generation process based on surface interpolation technique with the capabi of intuitive tension control at the data points appears to be the most ergonomic design envir ment. Several curve/surface schemes, such as *Beta-spline* and *Gamma-spline*, allow tens control in the design mode (Barsky, 81; Bartels, 1987; Blanc, 1995; Boehm, 1985). But only *spline* seems to allow this in the interpolation mode (Blanc, 1995). The problem with X-spl curves and surfaces is that the geometric meaning of shape parameters is not so intuitive.

In this paper, we present a surface design process which has the characteristics of both face interpolation techniques mentioned above. That is, a user is allowed to design the shape a surface by specifying a rectangular grid of data points for a surface interpolation proce However, the surface is generated by generating a network of bicubic Hermite spline curv which interpolate the data points first (see, e.g., Figure 7). The shapes of these curves can edited by adjusting tension parameters at the data points. A bicubic Hermite spline surfa which interpolates this network of curves is then constructed. The shape of the surface smoothed by using twist vectors that minimize the strain energy of the surface.

The principal contributions of this paper are:

- It presents a new representation scheme for the Hermite spline curves and surfaces.
- It presents a tension control technique for a cubic Hermite spline curve. The Herm spline curve is g^2-continuous.
- It demonstrates how the energy minimization process for a bicubic Hermite spline surfa to construct optimal twist vectors can be performed as the process of solving a system linear equations. The Hermite spline surface is C^1-continuous.
- It shows that it is not necessary to require C^2-continuous patch boundaries to produce overall C^1-continuous piecewise surface (Farin, 1988).

Since the tension parameters are associated with the data points, not the control points (Barsk 1981; Bartels, 1987; Boehm, 1985), an intuitive setting of the tension parameters can also b adopted to avoid the rollback and bump phenomena even before the network curves are co structed. Details of the paper are shown in the subsequent sections.

2. CONSTRUCTION OF NETWORK CURVES WITH TENSION CONTROL

Given a grid of data points $\mathbf{P}_{i,j}$ and associated tension parameters $\{\alpha_{i,j}, \beta_{i,j}\}$ where $i = 0, ..., m$ and $j = 0, ..., n$, we will construct a network of cubic, g^2 piecewise polynomial curves $\mathbf{C}^{(j)}(u)$ and $\mathbf{D}^{(i)}(v)$ such that $\mathbf{C}^{(j)}(i) = \mathbf{P}_{i,j}$ for $i = 0, 1, ..., m$ and $\mathbf{D}^{(i)}(j) = \mathbf{P}_{i,j}$ for $j = 0, 1, ..., n$. The tension parameters are used to control the shape of $\mathbf{C}^{(j)}(u)$ and $\mathbf{D}^{(i)}(v)$. A curve is said to be g^2-continuous (or V^2-continuous (Farin, 1982)) if it has continuous first derivative and curvature. A piecewise polynomial curve $\mathbf{X}(t)$ is g^2-continuous if and only if there exists a constant ν_i for each knot t_i such that the second derivative of X at t_i satisfies the following "jump" condition (Farin, 1985):

$$\mathbf{X}_{tt}(t_i+) - \mathbf{X}_{tt}(t_i-) = \nu_i \, \mathbf{X}_t(t_i) \tag{2.1}$$

where '+' and '−' denote the right and left limits, and $\mathbf{X}_t$ and $\mathbf{X}_{tt}$ are the first and second derivatives of $\mathbf{X}$ with respect to t. The following curve schemes: Fowler-Wilson splines (Fritsch, 1986), ν-splines (Nielson, 1974), Manning's interpolants (Manning, 1974), Barsky's β-splines (Barsky, 1981), Farin's splines (Farin, 82), Bartels and Beatty's β-splines (Bartels, 1984) and Boehm's γ-splines (Boehm, 1985) are all g^2 splines according to this characterization. (2.1) was first derived by Nielson as one of the characterization conditions for ν-splines (Nielson, 1974). The fact that ν-splines satisfy a "jump" condition has also been pointed out by Fritsch (Fritsch, 1986).

We will show the construction of $\mathbf{C}^{(0)}(u)$ only; the construction of other $\mathbf{C}^{(j)}(u)$ and $\mathbf{D}^{(i)}(v)$ is similar. $\mathbf{C}^{(0)}(u)$ will be constructed as a cubic *Hermite spline curve* defined as follows.

Given a set of 2D (3D) points $\{\mathbf{Q}_0, \mathbf{Q}_1, ..., \mathbf{Q}_m\}$ and a set of 2D (3D) tangent vectors $\{\mathbf{T}_0, \mathbf{T}_1, ..., \mathbf{T}_m\}$, a *Hermite spline curve* $\theta(t)$ is defined as follows

$$\begin{aligned}\theta(t) &= \sum_{i=0}^{m} h_{i,1}(t)\mathbf{Q}_i + \sum_{i=0}^{m} h_{i,2}(t)\mathbf{T}_i \\ &= \sum_{i=0}^{m} h_1(t-i)\mathbf{Q}_i + \sum_{i=0}^{m} h_2(t-i)\mathbf{T}_i, \qquad 0 \le t \le m\end{aligned} \tag{2.2}$$

where

$$h_1(t) = \begin{cases} 1 - 3t^2 - 2t^3, & -1 \le t < 0 \\ 1 - 3t^2 + 2t^3, & 0 \le t < 1 \\ 0, & \textit{otherwise} \end{cases} \tag{2.3}$$

and

$$h_2(t) = \begin{cases} t + 2t^2 + t^3, & -1 \le t < 0 \\ t - 2t^2 + t^3, & 0 \le t < 1 \\ 0, & \textit{otherwise} \end{cases} . \tag{2.4}$$

$h_{i,1}(t)$ and $h_{i,2}(t)$ are called the *Hermite basis functions* of *type one* and *type two*, respectively. They are compositions of *cubic Hermite polynomials* (Farin, 1988; p.67). The shapes of $a_i(t)$ and $b_i(t)$ are shown in Figure 1. $\theta(t)$ interpolates both $\mathbf{Q}_i$ and $\mathbf{T}_i$, i.e., if $\theta_t(i)$ denotes the derivative of θ with respect to t at i then we have both $\theta(i) = \mathbf{Q}_i$ and $\theta_t(i) = \mathbf{T}_i$, $i = 0, 1, ..., m$. Each segment of $\theta(t)$ is a Hermite curve segment.

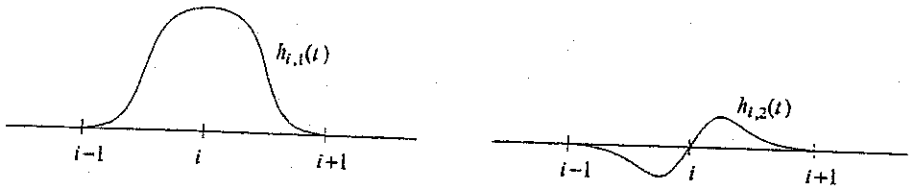

Figure 1. Shapes of $h_{i,1}(t)$ and $h_{i,2}(t)$.

$\mathbf{C}^{(0)}(u)$ will be represented as a cubic Hermite spline curve, as follows.

$$\mathbf{C}^{(0)}(u) = \sum_{i=0}^{m} h_{i,1}(u)\,\mathbf{P}_{i,0} + \sum_{i=0}^{m} h_{i,2}(u)\,\mathbf{P}^{u}_{i,0}, \qquad 0 \le u \le m \tag{2.5}$$

with $\mathbf{P}^{u}_{i,0}$, the derivatives of $\mathbf{C}^{(0)}(u)$ at $u = i$, to be determined. From our experience with cubic Bezier curves, it is easy to see that $\mathbf{C}^{(0)}(u)$ will be a C^2-continuous curve if $\mathbf{P}^{u}_{i,0}$ satisfy the following conditions (Faux, 1979)

$$\mathbf{P}^{u}_{i-1,0} + 4\mathbf{P}^{u}_{i,0} + \mathbf{P}^{u}_{i+1,0} = 3(\mathbf{P}_{i+1,0} - \mathbf{P}_{i-1,0}), \qquad i = 1, 2, ..., m-1.$$

However, instead of the above conditions, $\mathbf{P}^{u}_{i,0}$ are required to satisfy the following conditions

$$\mathbf{P}^{u}_{i-1,0} + 4\alpha_{i,0}\mathbf{P}^{u}_{i,0} + \mathbf{P}^{u}_{i+1,0} = 3(\mathbf{P}_{i+1,0} - \mathbf{P}_{i-1,0}), \qquad i = 1, 2, ..., m-1. \tag{2.6}$$

The role of $\alpha_{i,0}$ here is for the user to control the magnitude of the tangent vector $\mathbf{P}^{u}_{i,0}$ and, consequently, the height of the curve in the vicinity of $\mathbf{P}_{i,0}$ (Figure 2). A Hermite spline curve which satisfies (2.6) is g^2-continuous. The following simple derivation shows that it satisfies (2.1).

$$\frac{d}{du^2}\mathbf{C}^{(0)}(i+) = -6\mathbf{P}_{i,0} + 6\mathbf{P}_{i+1,0} - 4\mathbf{P}^{u}_{i,0} - 2\mathbf{P}^{u}_{i+1,0}$$

$$= 6\mathbf{P}_{i-1,0} - 6\mathbf{P}_{i,0} + 2\mathbf{P}^{u}_{i-1,0} + 4\mathbf{P}^{u}_{i,0} + 8(\alpha_{i,0}-1)\mathbf{P}^{u}_{i,0}$$

$$= \frac{d}{du^2}\mathbf{C}^{(0)}(i-) + 8(\alpha_{i,0}-1)\frac{d}{du}\mathbf{C}^{(0)}(i)$$

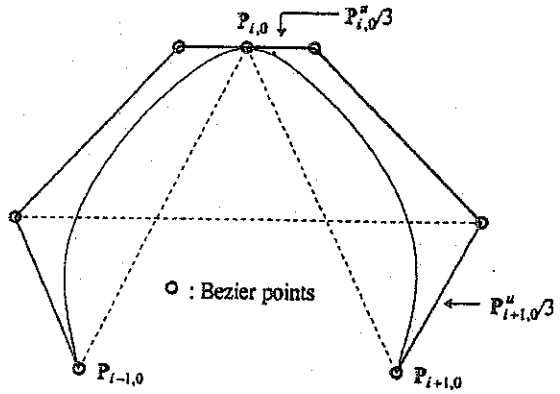

Figure 2. Effect of $\alpha_{i,0}$ when $\alpha_{i,0} > 1$.

The $m-1$ tangent vectors $\{ \mathbf{P}^u_{i,0} \mid i = 1, ..., m-1 \}$ of $\mathbf{C}^{(0)}(u)$ can be found by solving (2.6) with appropriate boundary conditions. Three boundary conditions are possible: given tangent vectors at the endpoints, modified natural boundary condition, and smoothly joined closed curve. For example, in the second case, one may use the following extra conditions to solve (2.6) for the tangent vectors.

$$(1+\alpha_{0,0})\mathbf{P}^u_{0,0} + \mathbf{P}^u_{1,0} = -3\mathbf{P}_{0,0} + 3\mathbf{P}_{2,0}\,; \quad \mathbf{P}^u_{m-1,0} + (1+\alpha_{m,0})\mathbf{P}^u_{m,0} = -3\mathbf{P}_{m-2,0} + 3\mathbf{P}_{m,0} \tag{2.7}$$

These conditions also allow $\mathbf{P}^u_{0,0}$ and $\mathbf{P}^u_{m,0}$ to be shortened or lengthened by adjusting the values of $\alpha_{0,0}$ and $\alpha_{m,0}$.

It can be shown that

$$\lim_{\alpha_{i,0}\to\infty} \mathbf{P}^u_{i,0} = 0 \tag{2.8}$$

Hence, when the tension parameters $\alpha_{i,0}$ and $\alpha_{i+1,0}$ tend to infinity, the tangent vectors $\mathbf{P}^u_{i,0}$ and $\mathbf{P}^u_{i+1,0}$ are forced to converge to 0 and, consequently, the curve segment $\theta_i(u)$ is pulled towards the line segment $\mathbf{P}_{i,0}\mathbf{P}_{i+1,0}$. The proof of (2.8) is shown in the Appendix. Note that the reason one can manipulate the smoothness of $\mathbf{C}^{(0)}(u)$ is because that $h_{i,1}(t)$ and $h_{i,2}(t)$ are not C^2-continuous.

3. CONSTRUCTION OF INTERPOLATING SURFACE WITH OPTIMAL TWIST VECTORS

In this section we will construct a C^1-continuous piecewise surface to interpolate the network curves constructed in Section 2. The surface, denoted $\mathbf{S}(u,v)$, is a bicubic Hermite spline surface with parameter space $[0, m]\times[0, n]$. $\mathbf{S}(u,v)$ interpolates the network curves $\mathbf{C}^{(j)}(u)$ and $\mathbf{D}^{(i)}(v)$ at the isoparametric curves $\mathbf{S}(u,j)$ and $\mathbf{S}(i,v)$, i.e., for each fixed j, $j = 0, 1, ..., n$, $\mathbf{S}(u,j)$ coincides with $\mathbf{C}^{(j)}(u)$ and for each fixed i, $i = 0, 1, ..., m$, $\mathbf{S}(i,v)$ coincides with $\mathbf{D}^{(i)}(v)$. $\mathbf{S}(u,v)$

has optimal twist vectors at the interpolation points $\mathbf{P}_{ij}$. The optimal twist vectors are generated by minimizing the energy of the surface and, hence, guarantee the smoothness of the surface. The definition of a bicubic Hermite Spline surface and the construction of $S(u,v)$ are shown below. Techniques on piecewise Hermite surface interpolation can be found in (Farin, 1988).

Given a set of 3D data points $\mathbf{Q}_{i,j}$, and u tangent vectors $\mathbf{U}_{i,j}$, v tangent vectors $\mathbf{V}_{i,j}$, and *twist vectors* $\mathbf{T}_{i,j}$ at these points, $i = 0, 1, ..., m$, $j = 0, 1, ..., n$, a bicubic *Hermite spline surface* $\Theta(u, v)$ is defined as follows

$$\begin{aligned}\Theta(u,v) = &\sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,1}(u)h_{j,1}(v)\mathbf{Q}_{i,j} + \sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,2}(u)h_{j,1}(v)\mathbf{U}_{i,j} \\ &+ \sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,1}(u)h_{j,2}(v)\mathbf{V}_{i,j} + \sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,2}(u)h_{j,2}(v)\mathbf{T}_{i,j}, \\ &0 \le u \le m,\ 0 \le v \le n\end{aligned} \tag{3.1}$$

where $h_{k,1}(\cdot)$ and $h_{k,2}(\cdot)$ are Hermite basis functions of type one and type two. $\Theta(u,v)$ interpolates $\mathbf{Q}_{i,j}$, $\mathbf{U}_{i,j}$, $\mathbf{V}_{i,j}$ and $\mathbf{T}_{i,j}$, i.e., if Θ_u, Θ_v, and Θ_{uv} denote the partial derivatives of Θ with respect to u and v then we have $\Theta(i,j) = \mathbf{Q}_{i,j}$, $\Theta_u(i,j) = \mathbf{U}_{i,j}$, $\Theta_v(i,j) = \mathbf{V}_{i,j}$ and $\Theta_{uv}(i,j) = \mathbf{T}_{i,j}$ for $i = 0, 1, ..., m$ and $j = 0, 1, ..., n$. Each patch of $\Theta(u,v)$ is a bicubic *Hermite surface patch*. A bicubic Hermite spline surface is at least C^1-continuous in both u and v parameters.

The surface $S(u,v)$ to be constructed will be represented as a bicubic Hermite spline surface, as follows.

$$\begin{aligned}S(u,v) = &\sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,1}(u)h_{j,1}(v)\mathbf{P}_{i,j} + \sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,2}(u)h_{j,1}(v)\mathbf{P}^{u}_{i,j} \\ &+ \sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,1}(u)h_{j,2}(v)\mathbf{P}^{v}_{i,j} + \sum_{i=0}^{m}\sum_{j=0}^{n} h_{i,2}(u)h_{j,2}(v)\mathbf{P}^{uv}_{i,j}\end{aligned} \tag{3.2}$$

with $\mathbf{P}^{uv}_{i,j}$, the twist vectors of $S(u,v)$ at $(u, v) = (i, j)$, to be determined. $\mathbf{P}^{u}_{i,j}$ and $\mathbf{P}^{v}_{i,j}$, the partial derivatives of $S(u,v)$ with respect to u and v at $(u, v) = (i, j)$, are known from Section 2. It is easy to see that the $S(u, v)$ defined in (3.2) interpolates all the network curves $\mathbf{C}^{(j)}(u)$ and $\mathbf{D}^{(i)}(v)$ constructed in Section 2.

We will choose twist vectors that minimize the strain energy of the surface, defined as follows.

$$E(S) = \iint_S (\kappa_1^2 + \kappa_2^2)\, d\sigma \tag{3.3}$$

where κ_1 and κ_2 are the principal curvatures of the surface and $d\sigma$ is the surface area measure. This is a standard approach for surface smoothing in engineering (Kallay, 1990; Nowacki, 1982). 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, one can get the energy form in terms of the first and the second surface derivatives as follows.

$$E(\mathbf{S}) = \iint_{\mathbf{D}} [(\frac{LG-2MF+NE}{EG-F^2})^2 - 2\frac{LN-M^2}{EG-F^2}](EG-F^2)^{1/2}\, dudv \tag{3.4}$$

where

$$E = \mathbf{S}_u \cdot \mathbf{S}_u \quad F = \mathbf{S}_u \cdot \mathbf{S}_v \quad G = \mathbf{S}_v \cdot \mathbf{S}_v$$

$$L = \mathbf{n} \cdot \mathbf{S}_{uu} \quad M = \mathbf{n} \cdot \mathbf{S}_{uv} \quad N = \mathbf{n} \cdot \mathbf{S}_{vv}$$

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

Since direct evaluation of (3.4) is a complicated process, several approximation methods have been used. These include the following simplified energy form of Hagen and Schulze (Hagen, 1987),

$$E(\mathbf{S}) = \iint_{\mathbf{D}} [\frac{L^2G^2}{(EG)^{3/2}} + \frac{E^2N^2}{(EG)^{3/2}} + \frac{2M^2}{(EG)^{1/2}}]\, dudv \tag{3.5}$$

the *thin plate model* (Cheng, 1994; Quak, 1989; Quak, 1990) which is a small deflection approximation of the surface curvature,

$$E(\mathbf{S}) = \iint_{\mathbf{D}} (\|\mathbf{S}_{uu}\|^2 + 2\|\mathbf{S}_{uv}\|^2 + \|\mathbf{S}_{vv}\|^2)\, dudv, \tag{3.6}$$

the *membrane model* (Szeliski, 1990) which is a small deflection approximation of the surface area,

$$E(\mathbf{S}) = \iint_{\mathbf{D}} (\|\mathbf{S}_u\|^2 + \|\mathbf{S}_v\|^2)\, dudv \tag{3.7}$$

and a generalized form of the thin plate model (Kallay, 1990)

$$E(\mathbf{S}) = \iint_{\mathbf{D}} (\Delta \cdot Q \cdot \Delta^T)\, dudv \tag{3.8}$$

where Q is a 9×9 symmetric, positive-definite coefficient matrix, Δ is a vector of second partial derivatives of $\mathbf{S}$ defined as follows

$$\Delta = (X_{uu}, X_{uv}, X_{vv}, Y_{uu}, Y_{uv}, Y_{vv}, Z_{uu}, Z_{uv}, Z_{vv}) \tag{3.9}$$

with X, Y and Z being the x-, y- and z- components of $\mathbf{S}(u,v)$, and Δ^T is the transpose of Δ.

Hagen and Schulze's energy form is the result of assuming that $\mathbf{S}_u$ and $\mathbf{S}_v$ are orthogonal ($F = \mathbf{S}_u \cdot \mathbf{S}_v = 0$). This assumption might not hold sometime. The membrane model is not a good smoothing tool due to the fact that it tends to generate interpolating surfaces with visible peaks and dips (Szeliski, 1990). Kallay and Ravani's model (3.8) seems to be a good choice in that it not just covers the thin plate model as a special case, but also provides the user with a capability of setting up an appropriate energy approximation form for a specific application by adjusting the entries of the matrix Q. We will use this model in the minimization process**.

For each (i, j) in {(1, 1), (1, 2), (2, 1), (2, 2)}, let $\mathbf{A}^{(i,j)}$ be a vector of $(m+1)(n+1)$ components defined as follows.

$$\mathbf{A}^{(i,j)} = \begin{pmatrix} h_{0,i}(u)h_{0,j}(v) \\ \vdots \\ h_{0,i}(u)h_{n,j}(v) \\ \vdots \\ h_{m,i}(u)h_{0,j}(v) \\ \vdots \\ h_{m,i}(u)h_{n,j}(v) \end{pmatrix} \tag{3.10}$$

where $h_{k,i}(u)$ and $h_{k,j}(v)$ are Hermite basis functions of type i and j. Then $S(u, v)$ can be expressed as follows

$$S(u, v) = \mathbf{P} \cdot \mathbf{A}^{(1,1)} + \mathbf{P}^u \cdot \mathbf{A}^{(2,1)} + \mathbf{P}^v \cdot \mathbf{A}^{(1,2)} + \mathbf{P}^{uv} \cdot \mathbf{A}^{(2,2)} \tag{3.11}$$

where

$$\mathbf{P} = (\mathbf{P}_{0,0}, \mathbf{P}_{0,1}, \ldots, \mathbf{P}_{0,n}, \mathbf{P}_{1,0}, \mathbf{P}_{1,1}, \ldots, \mathbf{P}_{1,n}, \ldots, \mathbf{P}_{m,0}, \mathbf{P}_{m,1}, \ldots, \mathbf{P}_{m,n})$$
$$\mathbf{P}^u = (\mathbf{P}^u_{0,0}, \mathbf{P}^u_{0,1}, \ldots, \mathbf{P}^u_{0,n}, \mathbf{P}^u_{1,0}, \mathbf{P}^u_{1,1}, \ldots, \mathbf{P}^u_{1,n}, \ldots, \mathbf{P}^u_{m,0}, \mathbf{P}^u_{m,1}, \ldots, \mathbf{P}^u_{m,n})$$
$$\mathbf{P}^v = (\mathbf{P}^v_{0,0}, \mathbf{P}^v_{0,1}, \ldots, \mathbf{P}^v_{0,n}, \mathbf{P}^v_{1,0}, \mathbf{P}^v_{1,1}, \ldots, \mathbf{P}^v_{1,n}, \ldots, \mathbf{P}^v_{m,0}, \mathbf{P}^v_{m,1}, \ldots, \mathbf{P}^v_{m,n})$$
$$\mathbf{P}^{uv} = (\mathbf{P}^{uv}_{0,0}, \mathbf{P}^{uv}_{0,1}, \ldots, \mathbf{P}^{uv}_{0,n}, \mathbf{P}^{uv}_{1,0}, \mathbf{P}^{uv}_{1,1}, \ldots, \mathbf{P}^{uv}_{1,n}, \ldots, \mathbf{P}^{uv}_{m,0}, \mathbf{P}^{uv}_{m,1}, \ldots, \mathbf{P}^{uv}_{m,n})$$

For these vectors, let $\mathbf{X}$, $\mathbf{X}^u$, $\mathbf{X}^v$, and $\mathbf{X}^{uv}$ be their x-components, $\mathbf{Y}$, $\mathbf{Y}^u$, $\mathbf{Y}^v$, and $\mathbf{Y}^{uv}$ be their

** Other criterion, such as minimal curvature variation (Moreton, 1992), has been used in the minimization process as well. However, this is not our concern here.

y-components, and $\mathbb{Z}$, $\mathbb{Z}^u$, $\mathbb{Z}^v$, and $\mathbb{Z}^{uv}$ be their z-components. Furthermore, for each (i,j) in $\{(1,1), (1,2), (2,1), (2,2)\}$, define a $3(m+1)(n+1)\times 9$ matrix $W_{i,j}$ as follows

$$W_{i,j} = \begin{bmatrix} \mathbf{A}_{uu}^{(i,j)} & \mathbf{A}_{uv}^{(i,j)} & \mathbf{A}_{vv}^{(i,j)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{A}_{uu}^{(i,j)} & \mathbf{A}_{uv}^{(i,j)} & \mathbf{A}_{vv}^{(i,j)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{A}_{uu}^{(i,j)} & \mathbf{A}_{uv}^{(i,j)} & \mathbf{A}_{vv}^{(i,j)} \end{bmatrix}$$

where $\mathbf{A}_{uu}^{(i,j)}$, $\mathbf{A}_{uv}^{(i,j)}$, and $\mathbf{A}_{vv}^{(i,j)}$ are the second partial derivatives of the vector $\mathbf{A}^{(i,j)}$ defined in (3.10), and 0 is a zero matrix of dimension $(m+1)(n+1)\times 1$. Then Δ defined in (3.9) can be expressed in the following form.

$$\Delta = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})\cdot W_{1,1} + (\mathbf{X}^u, \mathbf{Y}^u, \mathbf{Z}^u)\cdot W_{2,1} + (\mathbf{X}^v, \mathbf{Y}^v, \mathbf{Z}^v)\cdot W_{1,2} + (\mathbf{X}^{uv}, \mathbf{Y}^{uv}, \mathbf{Z}^{uv})\cdot W_{2,2}$$

Consequently,

$$\Delta Q \Delta^T = \delta + 2[(\mathbf{X}, \mathbf{Y}, \mathbf{Z})\cdot W_{1,1} + (\mathbf{X}^u, \mathbf{Y}^u, \mathbf{Z}^u)\cdot W_{2,1} + (\mathbf{X}^v, \mathbf{Y}^v, \mathbf{Z}^v)\cdot W_{1,2}]\cdot$$

$$\cdot Q\cdot (W_{2,2})^T\cdot \begin{bmatrix} (\mathbf{X}^{uv})^T \\ (\mathbf{Y}^{uv})^T \\ (\mathbf{Z}^{uv})^T \end{bmatrix} + (\mathbf{X}^{uv}, \mathbf{Y}^{uv}, \mathbf{Z}^{uv})\cdot W_{2,2}\cdot Q\cdot (W_{2,2})^T\cdot \begin{bmatrix} (\mathbf{X}^{uv})^T \\ (\mathbf{Y}^{uv})^T \\ (\mathbf{Z}^{uv})^T \end{bmatrix} \quad (3.12)$$

where δ is the sum of six terms independent of $\mathbf{P}^{uv}$. Therefore, by substituting (3.12) into (3.8) we have

$$\iint_{\mathbf{D}} (\Delta Q \Delta^T) du dv = c + 2\mathbf{B}^T\cdot \mathbf{G} + \mathbf{G}^T\cdot A\cdot \mathbf{G} \quad (3.13)$$

for some constant c with

$$\mathbf{G}^T = (\mathbf{X}^{uv}, \mathbf{Y}^{uv}, \mathbf{Z}^{uv}), \qquad A = \iint_{\mathbf{D}} (W_{2,2}\cdot Q\cdot (W_{2,2})^T) du dv, \quad (3.14)$$

and

$$\begin{aligned} \mathbf{B} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})\cdot [\iint_{\mathbf{D}} (W_{1,1}\cdot Q\cdot (W_{2,2})^T) du dv] \\ + (\mathbf{X}^u, \mathbf{Y}^u, \mathbf{Z}^u)\cdot [\iint_{\mathbf{D}} (W_{2,1}\cdot Q\cdot (W_{2,2})^T) du dv] \\ + (\mathbf{X}^v, \mathbf{Y}^v, \mathbf{Z}^v)\cdot [\iint_{\mathbf{D}} (W_{1,2}\cdot Q\cdot (W_{2,2})^T) du dv]. \end{aligned} \quad (3.15)$$

To minimize (3.13), it is sufficient to minimize

$$f(\mathbf{G}) \equiv \mathbf{B}^T\mathbf{G} + \frac{1}{2}\mathbf{G}^T \cdot A \cdot \mathbf{G}. \tag{3.16}$$

(3.16) can be minimized using quadratic programming method. However, note that (3.13) is a concave-up elliptic paraboloid, its minimum always exists. Hence, by setting the derivative of (3.13) with respect to $\mathbf{G}$ to zero, we have

$$A \cdot \mathbf{G} + \mathbf{B} = 0, \tag{3.17}$$

a system of $3(m+1)(n+1)$ linear equations in $3(m+1)(n+1)$ variables with A being a symmetric positive-definite matrix. Therefore, the minimum of (3.13) exists when

$$\mathbf{G} = -A^{-1} \cdot \mathbf{B}. \tag{3.18}$$

4. IMPLEMENTATION AND RESULTS

The new approach has been implemented on the following platform: HPUX level 9.05 on a Hewlett Packard HP 735 machine using an IBM RISC System/6000 machine as the display device. The optimization process is performed by using the "f07adf" subroutine of the *NAG Fortran Library* (NAG, 1993) to compute the LU factorization of the matrix $-A^{-1}$ in (3.18), and then using the "f07aef" subroutine to solve the system of linear equations (3.18). Several test cases are shown in Figures 8 - 15.

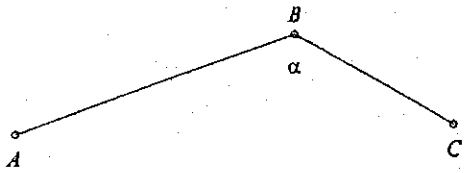

Figure 3. Three adjacent data points.

The rollback and bump phenomena can be removed by adjusting the tension parameters in the regions where they occur, or one can avoid these problems by appropriately setting the tension parameters beforehand. In the second case, the tension parameters for each data point must reflect the distribution of adjacent data points. For instance, to set the tension parameter α for the point B in Figure 3, we can either set α following the *chord length model*:

$$\alpha = \frac{max(\|A-B\|, \|B-C\|)}{min(\|A-B\|, \|B-C\|)}$$

or the *centripetal model* (Lee, 1989):

$$\alpha = \left[\frac{max(\|A-B\|, \|B-C\|)}{min(\|A-B\|, \|B-C\|)}\right]^{1/2}$$

These models are special cases of the following general form:

$$\alpha = \left[\frac{max(\|A - B\|, \|B - C\|)}{min(\|A - B\|, \|B - C\|)} \right]^{\beta}$$

where β is positive. However, according to our test results, $\beta = 1/2$ (i.e., the centripetal model) gives the best results in the sense that this β value always generates the smallest strain energy. In figure 8, a bicubic Hermite surface which interpolates the same data set (Figure 4) using our technique is shown. The tension parameters are set using the centripetal model ($\beta = 1/2$) and the twist vectors are optimized. The patch corresponding to the one shown in Figure 6 is shown in Figure 9. Note that the rollback problem does not exist here.

Figure 10 through Figure 13 are examples of tension control. In Figure 10, all the tension parameters, in both u and v directions, are set to 10. In figure 11, all the u direction tension parameters are 10 but v direction tension parameters are set by the centripetal model. Figure 12 is the opposite, i.e, all the v direction tension parameters are 10 but u direction tension parameters are set by the centripetal model. In Figure 13, all the tension parameters are set by the centripetal model but the tension parameters at data points A and B, where the tension parameters are set to 10. Twist vectors in all these cases are optimized.

In Figure 14 the Gaussian curvature distribution of the patch identified in Figure 8 but with zero twist vectors is shown. The Gaussian curvature distribution of the patch with optimized twist vectors is shown Figure 15. Note that in Figure 15, the curvature is distributed more uniformly on the left half of the patch than Figure 14. The upper right portion of Figure 15, however, is only slightly better than that of Figure 15. This is because the twist force needed for the patch to change its direction from perpendicular to the xz plane to be parallel to the xz plane is especially large at the upper right corner of the patch which, consequently, makes the effect of the optimization process not so obvious there.

5. SUMMARY AND FUTURE WORK

We have described a new approach to the surface interpolation problem. The new approach consists of two steps: (1) constructing a rectangular network of g^2 cubic Hermite spline curves with tension control to interpolate the data points, and (2) constructing a C^1 bicubic Hermite spline surface with optimal twist vectors to interpolate the network of curves. The cubic Hermite spline curves and bicubic Hermite spline surface are represented in new forms. Tension control of the Hermite spline curves is achieved through the introduction of a parameter to control the magnitude of each tangent vector. The optimal twist vectors are generated by minimizing the strain energy of the surface.

The new representation schemes of the Hermite spline curves and surface have proven effective in tension control and energy minimization; they makes the tension control process natural and the energy computation process simpler.

Our method provides a user with easy control on the shape of the interpolating surface, and the smoothness of the interpolating surface is automatically guaranteed. Our work also proves

that it is not necessary to require C^2 patch boundaries to produce an overall C^1 surface, as claimed by Farin (Farin, 1988; p 227). Note that, although it was proven by Nielson that g^2, G^2 and C^2 are equivalent for piecewise polynomial curves (Nielson, 1986), these concepts are not equivalent for piecewise polynomial surfaces.

A number of areas of future research remain, including: (1) investigate implementations on parallel architectures, (2) study the possibility of generating network curves of more general topological type, and (3) extend the current surface generation technique to more general network curves such as network curves of arbitrary topological type.

ACKNOWLEDGMENT

This work is supported by NSF grant DMI-9400823. The first author is also partially supported by the Center for Robotics and Manufacturing Systems, University of Kentucky.

REFERENCES

1. H. Akima, A method of bivariate interpolation and smooth surface fitting based on local procedure, *CACM* 17,1 (1978), 18-20.
2. R.E. Barnhill, J.H. Brown and I.M. Klucewicz, A new twist in computer aided design, *Computer Graphics and Image Processing* 8 (1978), 78-91.
3. R.E. Barnhill, G. Farin, L. Fayard, and H. Hagen, Twists, curvature and surface interrogation, *Computer Aided Design* 20 (1988), 341-346.
4. B.A. Barsky, The beta-spline, A local representation based on shape parameters and fundamental geometric measures, *PhD Thesis*, Dept. of Computer Science, University of Utah, 1981.
5. B.A. Barsky and T.D. DeRose, Geometric Continuity of Parametric Curves: Three Equivalent Characterizations, *IEEE CG&A* 9,6 (November 1989), 60-68.
6. R.H. Bartels and J.C. Beatty, Beta-splines with a difference, Tech. Rep. CS-83-40, Dept. of Computer Science, Univ. of Waterloo, Ontario, 1984.
7. R. Bartels, J. Beatty, B. Barsky, "An Introduction to Splines for Computer Graphics and Geometric Modeling", Morgan Kaufmann, 1987.
8. C. Blanc and C. Schlick, X-Splines: A Spline Model Designed for the End-User, *SIGGRAPH 95 Conference proceedings*, August 6-11, 1995, Los Angeles, CA, 377-386.
9. W. Boehm, Curvature continuous curves and surfaces, *Computer Aided Geometric Design* 2,4 (1985), 313-323.
10. R.L. Carmichael, A Collection of Procedures for Defining Airplane Surfaces for Input to PANIAR, *Computer-Aided Geometry Modeling*, J.N. Shoosmith and R.E. Fulton, eds. NASA Conference Publication 2272, 1984.
11. C. Cheng, and Y.F. Zheng, Thin plate spline surface approximation using Coons' patches, *Computer Aided Geometric Design*, 11 (1994), 269-287.

12. R.P. Dube, Local schemes for computer aided geometric design, Ph.D. Thesis, Department of Mathematics, University of Utah, 1975.
13. G. Farin, Visually $C^{(2)}$ Cubic Splines, *CAD* 14,3 (May 1982), 137-139.
14. G. Farin, Some remarks on V^2-splines, *Computer Aided Geometric Design* 2 (1985), 325-328.
15. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, San Diego, 1988.
16. I.D. Faux and M.J. Pratt, *Computational Geometry for Design and Manufacture*, John Wiley & Sons, New York, 1979.
17. F.N. Fritsch, The Wilson-Fowler spline is a v-spline, *Computer Aided Geometric Design* 3 (1986), 155-162.
18. H. Hagen and G. Schulze, Automatic smoothing with geometric surface patches, *Computer Aided Geometric Design* 4 (1987), 231-236.
19. M. Kallay and B. Ravani, Optimal twist vectors as a tool for interpolating a network of curves with a minimum energy surface, *Computer Aided Geometric Design* 7 (1990), 465-473.
20. E.T.Y. Lee, On choosing nodes in parametric curve interpolation, *Computer Aided Design*, 21,6 (July/August, 1989), 363-370.
21. J.R. Manning, Continuity Conditions for Spline Curves, *Comput. J.* 17 (1974), 181-186.
22. Henry P. Moreton, Carlo H. Sequin, Functional Optimization for Fair Surface Design, Siggraph'92, 167-176.
23. *NAG FORTRAN Library Manual. Mark 16*, 1st Edition (Sept. 1993).
24. G. M. Nielson, Some piecewise polynomial alternatives to spline under tension, *Computer-Aided Geometric Design*, eds., R. E. Barnhill and R. F. Riesenfeld, Academic Press, New York (1974), 209-235.
25. G. M. Nielson, A Locally Controllable Spline with Tension for Interactive Curve Design, *Computer Aided Geometric Design* 1,3 (1984), 199-205.
26. G. M. Nielson, Rectangular v-Splines, *IEEE Computer Graphics and Applications* 6, 2 (February 1986), 58-67.
27. H. Nowacki and D. Reese, Design and fairing of ship surfaces, in R. Barnhill and W. Boehm, eds., *Surfaces in Computer Aided Geometric Design*, North-Holland (1982), 121-134.
28. J. Peters, Smooth Interpolation of a Mesh of Curves, Constructive Approximation 7(1991), 221-247.
29. L. Piegl and W. Tiller, Curve and Surface Constructions for Computer Aided Design Using Rational B-splines, *Computer Aided Design* 19(9):485-498, November, 1987.
30. L. Piegl, On NURBS: A Survey. IEEE CG&A, 11, 1 (1991), 55-71.
31. E. Quak, and L.L. Schumaker, 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), 134-143.

32. E. Quak, and L.L. Schumaker, Cubic spline fitting using data dependent triangulations, *Computer Aided Geometric Design*, 7 (1990), 293-301.
33. R. Szeliski, Fast surface interpolation using hierarchical basis functions, *IEEE Trans. Pattern Anal. Machine Intell.* 12 (1990), 513-528.

APPENDIX

We prove (2.8) for the second boundary condition: modified natural boundary condition. The proof for the other two boundary conditions can be proved similarly. The proof of (2.10) involves the solution process of $\mathbf{P}^u_{i,0}$.

From (2.6) and (2.7), we get a system of equations $A\,\mathbf{Q} = \mathbf{P}$ where

$$A = \begin{bmatrix} (1+\alpha_{0,0}) & 1 & & & & \\ 1 & 4\alpha_{1,0} & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 4\alpha_{m-1,0} & 1 & \\ & & & 1 & (1+\alpha_{m,0}) \end{bmatrix},$$

$\mathbf{Q} = [\mathbf{P}^u_{0,0}, \mathbf{P}^u_{1,0}, \ldots, \mathbf{P}^u_{m,0}]$, and $\mathbf{P} = [3(\mathbf{P}_{2,0}-\mathbf{P}_{0,0}), 3(\mathbf{P}_{2,0}-\mathbf{P}_{0,0}), 3(\mathbf{P}_{3,0}-\mathbf{P}_{1,0}), \ldots, 3(\mathbf{P}_{m-1,0}-\mathbf{P}_{m-3,0}), 3(\mathbf{P}_{m,0}-\mathbf{P}_{m-2,0}), 3(\mathbf{P}_{m,0}-\mathbf{P}_{m-2,0})]$. Matrix A is diagonally dominant and positive definite if $\alpha_{0,0} > 0$, $\alpha_{m,0} > 0$ and $\alpha_{i,0} > 1/2$ for $i = 1, 2, \ldots, m-1$. To solve this system for $\mathbf{P}^u_{i,0}$, one first reduces $A\,\mathbf{Q} = \mathbf{P}$ into an upper triangular system $\underline{A}\,\mathbf{Q} = \mathbf{V}$ with

$$\underline{A} = \begin{bmatrix} 1 & w_0 & & & \\ & 1 & w_1 & & \\ & & \ddots & \ddots & \\ & & & 1 & w_{m-1} \\ & & & & 1 \end{bmatrix}$$

and $\mathbf{V} = [\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_{m-1}, \mathbf{V}_m]^T$ where

$$\begin{cases} w_0 = 1/(1+\alpha_{0,0}) \\ w_i = 1/(4\alpha_{i,0}-w_{i-1}), \quad i = 1, 2, \ldots, m \end{cases} \tag{A.1}$$

$$\begin{cases} \mathbf{V}_0 = 3w_0(\mathbb{P}_{2,0}-\mathbb{P}_{0,0}) \\ \mathbf{V}_i = [3(\mathbb{P}_{i+1,0}-\mathbb{P}_{i-1,0}) - \mathbf{V}_{i-1}]w_i\,, \quad i = 1, 2, ..., m \end{cases} \tag{A.2}$$

One then solves $\underline{A}\,\mathbf{Q} = \mathbf{V}$ backward as follows.

$$\begin{cases} \mathbf{P}^u_{m,0} = \mathbf{V}_m \\ \mathbf{P}^u_{i,0} = \mathbf{V}_i + w_i\mathbf{P}^u_{i+1,0}\,, \quad i = m-1, ..., 0 \end{cases} \tag{A.3}$$

When $\alpha_{i,0}$ tends to infinity, w_i and $\mathbf{V}_i$ tend to zero (see (A.1) and (A.2)) and, consequently, $\mathbf{P}^u_{i,0}$ tends to zero (see (A.3)).

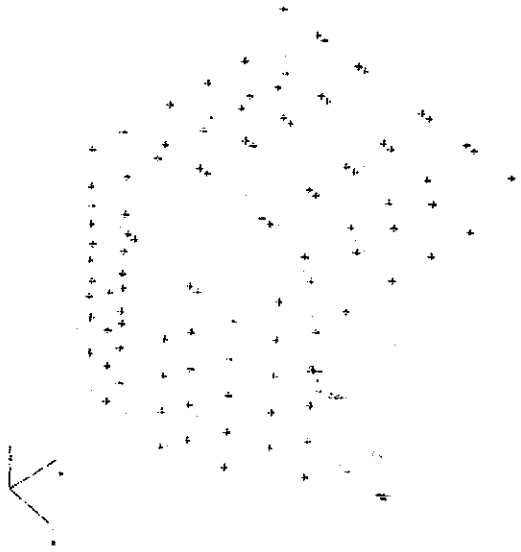

Figure 4. Data points.

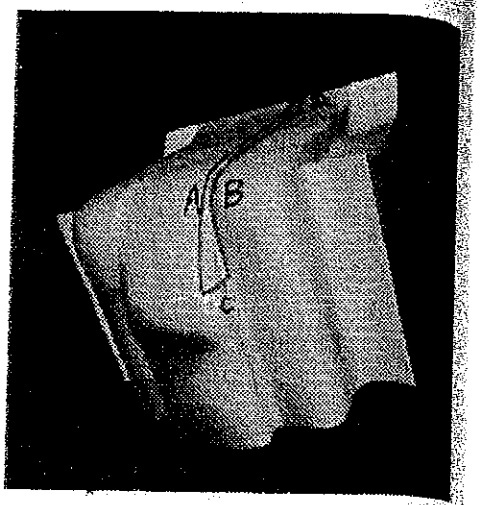

Figure 5. B-spline surface interpolation.

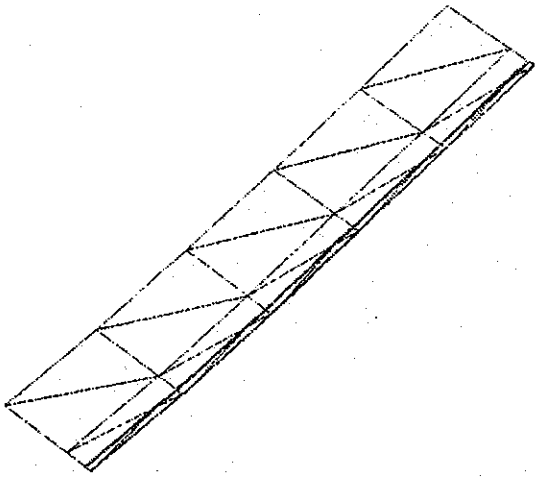

Figure 6. Rollback phenomenon.

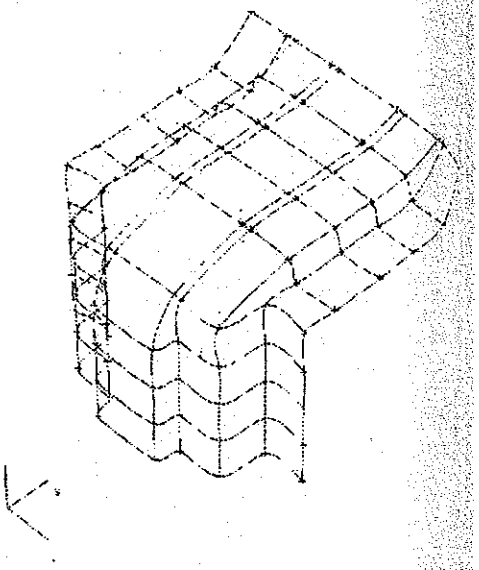

Figure 7. Network curves.

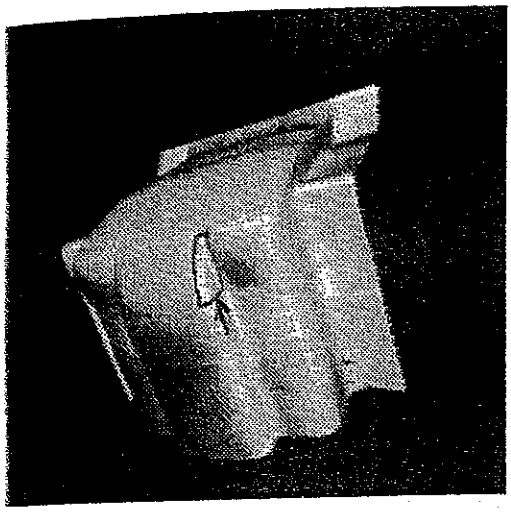

Figure 8. Tension control with $\beta = 1/2$.

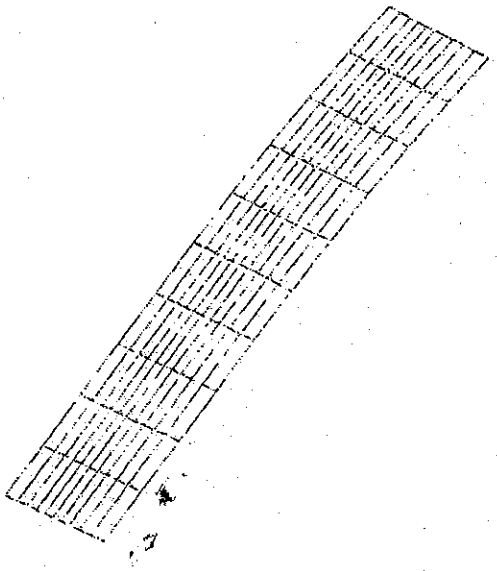

Figure 9. Removal of rollback phenomenon.

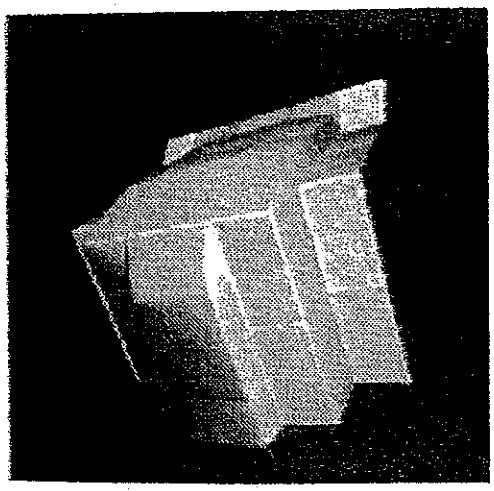

Figure 10. Global tension control.

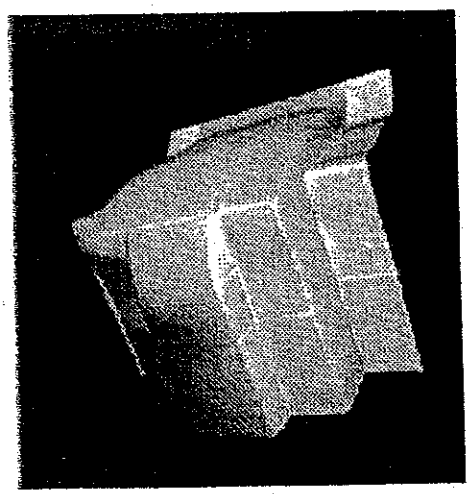

Figure 11. u tension control.

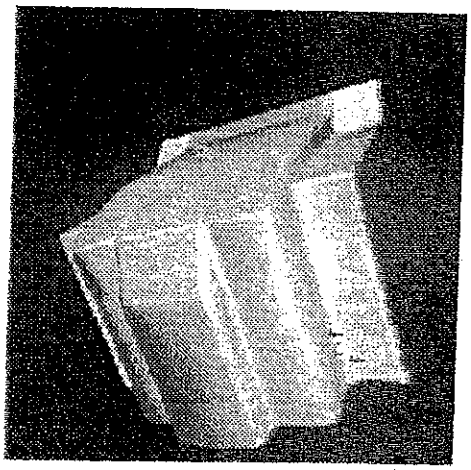

Figure 12. r tension control

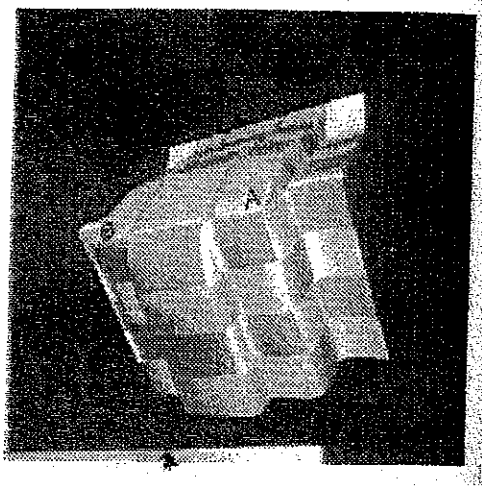

Figure 13. Local tension control.

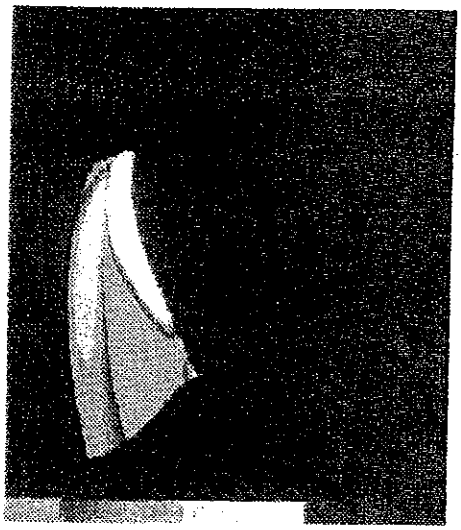

Figure 14. Curvature distribution I.

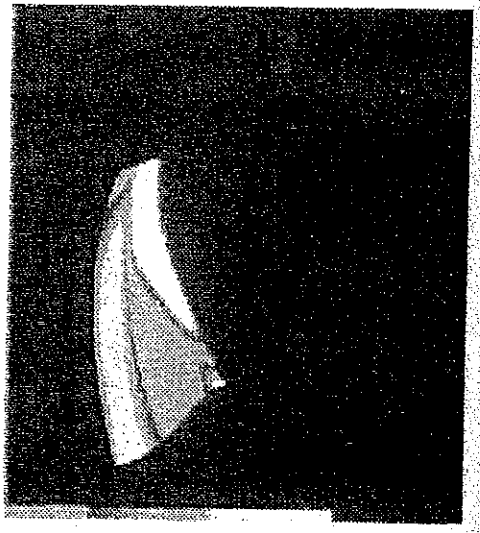

Figure 15. Curvature distribution II.