

Generating Symbolic Interpolants for Scattered Data with Normal Vectors

Ming Li¹, Xiao-Shan Gao^{2*} and Jin-San Cheng²

¹School of Computer Science, Cardiff University, Cardiff, CF24 3AA, UK

²KLMM, AMSS, Academia Sinica, Beijing 100080, China

Email: m.li@cs.cf.ac.uk; xgao@mmrc.iss.ac.cn; jcheng@amss.ac.cn

Abstract

Algorithms to generate a triangular or a quadrilateral interpolant with G^1 -continuity are given in this paper for arbitrary scattered data with associated normal vectors over a prescribed triangular or quadrilateral decomposition. The interpolants are constructed with a general method to generate surfaces from moving Bézier curves under geometric constraints. With the algorithm, we may obtain interpolants in complete symbolic parametric forms, leading to a fast computation of the interpolant. A dynamic interpolation solid modelling software package DISM is implemented based on the algorithm which can be used to generate and manipulate solid objects in an interactive way.

Keywords: Solid modelling, Hermite interpolation surface, Bézier curve, sweeping curve

* Partially supported by a National Key Basic Research Project of China (NO. G1998030600) and by a US NSF grant CCR-0201253.

1 Introduction

Given a set of scattered data with a prescribed triangular or quadrilateral decomposition in space, a smooth interpolation surface (or interpolant) is often needed to pass through these points. Such a problem arises in numerous areas of applications such as reverse engineering, solid modelling, medical imaging and scientific visualization. Additionally, the tangent planes or normal vectors at the interpolated points might be prescribed by the user simultaneously to give a local control for the final interpolation surface. The surface passing through a set of spatial points with prescribed normal vectors at these points is called an *Hermite Interpolation Surface (HIS)*.

Numerous schemes have been proposed to deal with the problem of scattered data interpolation with or without normal vectors being prescribed. A survey of scattered data interpolation methods

for electronic imaging systems can be found in [1]. Besides the subdivision method reported in [2, 3, 4, 5], various interpolation schemes could be mainly categorized into two classes: the implicit schemes, with interpolants expressed as $f(x, y, z) = 0$, and the parametric schemes, with interpolants expressed as $\mathbf{S}(u, v) = (x(u, v), y(u, v), z(u, v))$. As an implicit scheme, Dahmen used piecewise quadratic implicit patches in Bernstein-Bézier form to construct an HIS [6]. Various improvements for this work were then proposed later in order to decrease the number of the interpolation patches and overcome some shape control problems such as multi-sheets and self-intersections [7, 8, 9, 10, 11]. Gao and Li used blending methods to generate an implicit HIS [12]. Turk and O'Brien constructed an implicit interpolation surface with a variational scattered data interpolation approach [13].

It has been studied for a long time to construct an interpolation surface in parametric form. The interpolants involved in these schemes mainly consist of the tensor product patches, the Bézier triangles, the NURBS and the Coons patches. Some of these parametric interpolation scheme can be found in [14] and in surveys [15, 16, 17]. Mann et al surveyed several parametric interpolation schemes for triangulated data [17]. Most of these parametric schemes proceeded by first building boundary curves for a triangle face, and then filling the interior of the face with one or more surface patches. However, the *twist compatibility* problem might appear in some of these schemes. Furthermore, as

investigated in [17], most of these schemes might introduce shape defects. An interpolant with G^1 -continuity was derived from a spline surface covering the prescribed boundary curves in [18]. Nielson took a side-vertex method to generate a surface by passing a curve from one vertex to its opposite side [19]. Loop presented an algorithm to construct an approximation smooth spline surface for an irregular control mesh [20]. Pottmann et al. presented an active contour model for curve and surface approximation [21]. Fasshauer and Schumaker gave a parametric spline surface with minimum certain natural energy to interpolate the prescribed 3D scattered data [22]. Bajaj and Xu proposed a symbolic technique to construct boundary curves with minimum bending energy and some numerical methods are then used to construct surfaces to interpolate these wire frames smoothly [23]. Wiltsche focused on the construction of a C^k -continuous ($k = 0, 1, 2$) interpolating spline surface for a given data set [24].

In this paper, a general method is proposed to generate a surface from moving Bézier curves under some geometric constraints. We call this a *sweeping curve*. It can be used to construct an HIS, to convert a polygonal model into a parametric surface, to deform a 3D object, to form blending surfaces and to fill N-sided hole. Here we mainly use it to construct an HIS. A basic idea behind the method is that *a surface is the locus of a curve that is moving through space and thereby changing its shape* [25]. To generate surfaces following this idea,

we need answer some important questions: how to control the shape of the surface generated from a sweeping curve? Specifically, how to give a precise mathematical expression for the sweeping curve? How to control the variation of the sweeping curve to satisfy some prescribed geometric constraints? The answer to the questions will be illustrated in detail in our construction of an HIS in this paper.

In our method, we treat a surface as a sweeping cubic Bézier curve determined by its four moving control points, which are called *control curves*. Based on the above idea, a triangular or quadrilateral parametric interpolant with pleasing shape will be given over an arbitrary spatial data with a prescribed triangular or quadrilateral decomposition. All our computations are taken in vector forms, leading to a complete symbolic expression for the final interpolant.

The algorithm consists of three steps. First, boundary curves are constructed with a minimum bending energy in cubic Bézier form with one free parameter left to control the shape. A normal vector function along the boundary curve is given in a quadratic form. Second, the interior between two boundary curves is filled with a sweeping curve interpolating the corresponding points on the boundary curves, resulting a *pre-interpolant* in a bicubic Bézier form. The pre-interpolant can be regarded as a moving cubic Bézier curve with some control curves as its control points. ***The final interpolant is taken as a convex combination of the pre-interpolants, ending with***

a complete symbolic G^1 parametric representation. All our constructions in this scheme, including the boundary curve and pre-interpolant construction, can be regarded as the construction of an interpolation cubic Bézier curve from two points with associated normal vectors. In the end, a triangular and quadrilateral interpolant are obtained respectively from these steps. **However, in the process of the triangular interpolant generation, an unexpected degenerate case appears, which might result in impractical inpotalants although it also provides nice theory analysis for the interpolant construction. While the quadrilateral interpolant involves less patches and the degenerated bicubic surface will not appear in its construction, a quadrilateral decomposition is preferred to a triangle one.**

Based on the method, a dynamic interpolation solid modelling software package DISM is implemented which can be used to generate and manipulate solid objects dynamically. The software has the following features. (1) 3D points and vectors at points could be generated in an interactive way; (2) after a triangular or quadrilateral decomposition is given by the user, the interpolating surface can be generated automatically; (3) when the user changes the position of the points or the normal vectors, the interpolation surfaces will change continuously. This process can be considered as a dynamic deformation from one object to another. DISM provides a convenient tool to generate and

manipulate solid objects.

The rest of the paper is organized as follows. We introduce the problem in section 2. The construction of the boundary curves and its associated normal vectors is included in section 3. The main result, a quadrilateral interpolant and a triangular one, is illustrated in section 4. The software is shown in section 5 with some examples given. In the end, we conclude this paper in section 6.

2 Preliminaries

2.1 Problem Description

The following notations will be used in the paper.

$\mathcal{P} := \{\mathbf{p}_i \in \mathcal{R}^3, i = 1, \dots, k\}$, a set of points.

$\mathcal{N} := \{\mathbf{n}_i \in \mathcal{R}^3, i = 1, \dots, k\}$, a set of unit vectors.

$[\mathcal{P}, \mathcal{N}] := \{(\mathbf{p}_i, \mathbf{n}_i), \mathbf{p}_i \in \mathcal{P}, \mathbf{n}_i \in \mathcal{N}\}$.

$\Lambda := \{(u, v, w) \in \mathcal{R}^3, 0 \leq u, v, w \leq 1 \text{ and } u + v + w = 1\}$.

$\Gamma := \{(s, t) \in \mathcal{R}^2, 0 \leq s, t \leq 1\}$.

The dot product of two vectors is denoted as $\mathbf{v}_0 \cdot \mathbf{v}_1$ and the cross product of two vectors is denoted as $\mathbf{v}_0 \times \mathbf{v}_1$.

Without confusion, we will not take specific distinction between points and vectors throughout this paper. When a point is taken as a vector, we refer to the vector starting from the origin point and ending at this point.

A quadrilateral interpolant and a triangular interpolant will be given in this paper, which are stated as follows:

Quadrilateral interpolant: Let $q = (i, j, k, l)$ be a prescribed quadrilateral decomposition. Find a quadrilateral spline surface

$$\begin{aligned} \mathbf{S}_q : [\mathcal{P}, \mathcal{N}]^4 \times \Gamma &\rightarrow \mathcal{R}^3 \\ (\mathbf{p}_i, \mathbf{n}_i, \mathbf{p}_j, \mathbf{n}_j, \mathbf{p}_k, \mathbf{n}_k, \mathbf{p}_l, \mathbf{n}_l, s, t) &\rightarrow \mathbf{S}_q[i, j, k, l](s, t) \end{aligned}$$

such that the following conditions are satisfied

1. Interpolation conditions

$$\begin{aligned} \mathbf{S}_q(0, 0) &= \mathbf{p}_i, \quad \mathbf{S}_q(1, 0) = \mathbf{p}_j, \\ \mathbf{S}_q(1, 1) &= \mathbf{p}_k, \quad \mathbf{S}_q(0, 1) = \mathbf{p}_l, \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathbf{S}_q(0,0)}{\partial \mu} \cdot \mathbf{n}_i &= 0, \quad \frac{\partial \mathbf{S}_q(1,0)}{\partial \mu} \cdot \mathbf{n}_j = 0, \\ \frac{\partial \mathbf{S}_q(1,1)}{\partial \mu} \cdot \mathbf{n}_k &= 0, \quad \frac{\partial \mathbf{S}_q(0,1)}{\partial \mu} \cdot \mathbf{n}_l = 0, \quad \mu = s, t. \end{aligned}$$

2. For two adjacent points $\mathbf{p}_i, \mathbf{p}_j$ in \mathcal{P} , two patches passing through $\mathbf{p}_i, \mathbf{p}_j$ must meet with G^1 -continuity.

Without confusion, we also denote $\mathbf{S}_q[i, j, k, l]$ by \mathbf{S}_q .

Triangular interpolant: Let $I = (i, j, k)$ be a prescribed triangulation. Find a triangular spline surface

$$\begin{aligned} \mathbf{S}_I : [\mathcal{P}, \mathcal{N}]^3 \times \Lambda &\rightarrow \mathcal{R}^3 \\ (\mathbf{p}_i, \mathbf{n}_i, \mathbf{p}_j, \mathbf{n}_j, \mathbf{p}_k, \mathbf{n}_k, u, v, w) &\rightarrow \mathbf{S}_I[i, j, k](u, v, w) \end{aligned}$$

such that the following conditions are satisfied

1. Interpolation conditions

$$\mathbf{S}_I(1, 0, 0) = \mathbf{p}_i,$$

$$\mathbf{S}_I(0, 1, 0) = \mathbf{p}_j, \quad \mathbf{S}_I(0, 0, 1) = \mathbf{p}_k,$$

$$\frac{\partial \mathbf{S}_I(1, 0, 0)}{\partial \mu} \cdot \mathbf{n}_i = 0,$$

$$\frac{\partial \mathbf{S}_I(0, 1, 0)}{\partial \mu} \cdot \mathbf{n}_j = 0, \quad \frac{\partial \mathbf{S}_I(0, 0, 1)}{\partial \mu} \cdot \mathbf{n}_k = 0,$$

where $\mu = u, v$.

2. For two adjacent points $\mathbf{p}_i, \mathbf{p}_j$ in \mathcal{P} , two patches passing through $\mathbf{p}_i, \mathbf{p}_j$ must meet with G^1 -continuity.

Without confusion, we also denote $\mathbf{S}_{[i,j,k]}$ by \mathbf{S}_I .

2.2 Bézier Curves

Let $B_i^n(t) = C_n^i t^i (1-t)^{n-i}$ denote the *Bernstein polynomials*. A Bézier curve of degree n with control points \mathbf{b}_i can be defined as $\mathbf{b}^n(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t)$ with some properties listed below [25].

Derivative: Let Δ be the difference operator defined as $\Delta \mathbf{b}_i = \mathbf{b}_{i+1} - \mathbf{b}_i$, then the derivative of $\mathbf{b}^n(t)$ is given by

$$\frac{d}{dt} \mathbf{b}^n(t) = n \sum_{i=0}^{n-1} \Delta \mathbf{b}_i B_i^{n-1}(t). \quad (1)$$

Integrals: For any $0 \leq i \leq n$,

$$\int_0^1 B_i^n(t) dt = \frac{1}{n+1}. \quad (2)$$

Theorem 2.1 $\mathbf{b}^m(t) = \sum_{i=0}^m \mathbf{d}_i B_i^m(t)$ is normal to $\mathbf{b}^n(t)$ if and only if

$$\sum_{i+j=k} \frac{C_n^i C_m^j}{C_{n+m}^k} \mathbf{b}_i \cdot \mathbf{d}_j = 0, \quad 0 \leq k \leq m+n.$$

Proof: $\mathbf{b}^m(t)$ is normal to $\mathbf{b}^n(t)$ if and only if their dot product is zero. Since,

$$\mathbf{b}^m(t) \cdot \mathbf{b}^n(t) = \sum_{k=0}^{m+n} \left(\sum_{i+j=k} \frac{C_n^i C_m^j}{C_{n+m}^k} \mathbf{b}_i \cdot \mathbf{d}_j \right) B_k^{m+n}(t) \quad (3)$$

and $B_k^{m+n}(t), 0 \leq k \leq m+n$ is the bases of a polynomial space of degree $m+n$, the corresponding coefficients must be identically zero. \blacksquare

The following lemma is known:

Lemma 2.2 Let $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$ and \mathbf{f} be any space vectors and we have

$$(\mathbf{n}_0 \times \mathbf{n}_1 \cdot \mathbf{n}_2) \mathbf{f} = (\mathbf{f} \cdot \mathbf{n}_0)(\mathbf{n}_1 \times \mathbf{n}_2) + (\mathbf{f} \cdot \mathbf{n}_1)(\mathbf{n}_2 \times \mathbf{n}_0) + (\mathbf{f} \cdot \mathbf{n}_2)(\mathbf{n}_0 \times \mathbf{n}_1). \quad (4)$$

3 Boundary Constructions

The boundary constructions includes the construction of a boundary curve $\mathbf{g}_v(t)$ and a normal vector $\mathbf{g}_n(t)$.

Boundary constructions: Suppose that two adjacent points $(\mathbf{p}_0, \mathbf{n}_0), (\mathbf{p}_1, \mathbf{n}_1) \in [\mathcal{P}, \mathcal{N}]$ are given. Find an interpolation curve $\mathbf{g}_v(t)$ and its corresponding normal vector $\mathbf{g}_n(t)$,

$$\begin{aligned} \mathbf{g}_v : [\mathcal{P}, \mathcal{N}]^2 \times [0, 1] &\rightarrow \mathcal{R}^3 \\ (\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1, t) &\rightarrow \mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t) \end{aligned}$$

$$\begin{aligned} \mathbf{g}_n : [\mathcal{P}, \mathcal{N}]^2 \times [0, 1] &\rightarrow \mathcal{R}^3 \\ (\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1, t) &\rightarrow \mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t) \end{aligned}$$

such that

$$\begin{aligned} \mathbf{g}_v(0) &= \mathbf{p}_0, \quad \mathbf{g}_v(1) = \mathbf{p}_1, \\ \mathbf{g}'_v(0) \cdot \mathbf{n}_0 &= 0, \quad \mathbf{g}'_v(1) \cdot \mathbf{n}_1 = 0, \end{aligned}$$

$$\begin{aligned} \mathbf{g}_n(0) &= c_0 \mathbf{n}_0, \quad \mathbf{g}_n(1) = c_1 \mathbf{n}_1, \quad c_0, c_1 \neq 0 \in \mathcal{R} \\ \mathbf{g}'_v(t) \cdot \mathbf{g}_n(t) &= 0, \quad t \in [0, 1]. \end{aligned}$$

We denote $\mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t)$ and $\mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t)$ by $\mathbf{g}_v(t)$ and $\mathbf{g}_n(t)$ without confusion.

3.1 Boundary Curve Construction

The following lemma shows the limitation of a quadratic curve in the construction of a boundary curve.

Lemma 3.1 *If \mathbf{n}_0 and \mathbf{n}_1 are linearly dependent, there exists a quadratic interpolation curve $\mathbf{g}_v(t)$ if and only if $(\mathbf{p}_1 - \mathbf{p}_0) \cdot \mathbf{n}_0 = 0$. In this case $\mathbf{g}_v(t)$ is degenerated to the line $\mathbf{p}_0\mathbf{p}_1$.*

Proof. Since a quadratic curve $\mathbf{g}_v(t)$ is just a parabola, whose derivative is a monotonic function, there does not exist two points on it which share the same tangent direction. ■



Figure 1: $(\mathbf{p}_0, \mathbf{n}_0), (\mathbf{p}_1, \mathbf{n}_1)$ are in general cases

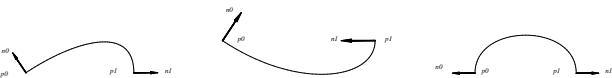


Figure 2: $(\mathbf{p}_0, \mathbf{n}_0), (\mathbf{p}_1, \mathbf{n}_1)$ are in special cases

From lemma 3.1, instead of a quadratic curve, a cubic Bézier curve in the following form will be constructed

$$\mathbf{g}_v(t) = \sum_{i=0}^3 \mathbf{b}_i B_i^n(t).$$

It is evident that $\mathbf{b}_0 = \mathbf{p}_0$ and $\mathbf{b}_3 = \mathbf{p}_1$. We will construct \mathbf{b}_1 and \mathbf{b}_2 in what follows, where we convert the point-normal problem to the point-tangent problem according to the algorithm described in [25]. The resulted boundary curve is a cubic Bézier curve with minimum bending energy with one free parameter left to control its shape. The construction algorithm will be illustrated in two cases: the general cases and the special cases according to the relation of the points and their corresponding normals vectors. Figure 1 shows some general cases that neither \mathbf{n}_0 nor \mathbf{n}_1 is parallel to $\mathbf{p}_0\mathbf{p}_1$ and Figure 2 shows special cases that one or both of \mathbf{n}_0 and \mathbf{n}_1 are parallel to $\mathbf{p}_0\mathbf{p}_1$.

1. Tangent direction of the boundary curve

General cases: Project \mathbf{b}_3 into the plane passing through \mathbf{b}_0 and with the normal direction \mathbf{n}_0 . We get the projection point

$$\tilde{\mathbf{b}}_1 = \mathbf{b}_3 + \frac{(\mathbf{b}_0 - \mathbf{b}_3) \cdot \mathbf{n}_0}{\|\mathbf{n}_0\|^2} \mathbf{n}_0.$$

In a similar way, we get

$$\tilde{\mathbf{b}}_2 = \mathbf{b}_0 + \frac{(\mathbf{b}_3 - \mathbf{b}_0) \cdot \mathbf{n}_1}{\|\mathbf{n}_1\|^2} \mathbf{n}_1.$$

Define the tangent directions at points \mathbf{b}_0 and \mathbf{b}_3 as

$$\mathbf{T}_0 = \frac{\tilde{\mathbf{b}}_1 - \mathbf{b}_0}{\|\tilde{\mathbf{b}}_1 - \mathbf{b}_0\|}, \quad \mathbf{T}_3 = \frac{\mathbf{b}_3 - \tilde{\mathbf{b}}_2}{\|\mathbf{b}_3 - \tilde{\mathbf{b}}_2\|}. \quad (5)$$

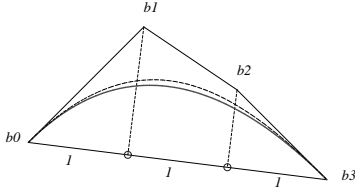


Figure 3: Parameter selection

Special cases: In these cases one or both of $\tilde{\mathbf{b}}_1 - \mathbf{b}_0$ and $\mathbf{b}_3 - \tilde{\mathbf{b}}_2$ is zero. A plane \mathbf{p} is first selected according to the following criteria and \mathbf{n} is defined as the unit normal vector of \mathbf{p} .

- If just one of \mathbf{n}_0 or \mathbf{n}_1 is parallel to $\mathbf{p}_0\mathbf{p}_1$ and suppose it is \mathbf{n}_0 . Let \mathbf{p} be the plane containing $\mathbf{p}_0, \mathbf{p}_1$ and paralleling to \mathbf{n}_1 .
- Suppose that both \mathbf{n}_0 and \mathbf{n}_1 are parallel to $\mathbf{p}_0\mathbf{p}_1$. If $\mathbf{b}_0\mathbf{b}_3$ is shared by two triangles, let \mathbf{p} be a plane passing through $\mathbf{p}_0\mathbf{p}_1$ and forming the same angle with the two triangles; otherwise let \mathbf{p} be a plane perpendicular to the triangle containing $\mathbf{p}_0\mathbf{p}_1$.

Let $\mathbf{T}_0 = \mathbf{n} \times \mathbf{n}_0$ and $\mathbf{T}_3 = \mathbf{n} \times \mathbf{n}_1$.

Now the point-normal interpolation problem with respect to $(\mathbf{p}_0, \mathbf{n}_0)$ $(\mathbf{p}_1, \mathbf{n}_1)$ is converted into the point-tangent problem with respect to $(\mathbf{b}_0, \mathbf{T}_0)$ and $(\mathbf{b}_3, \mathbf{T}_3)$.

2. Parameter selection

We can set $\mathbf{b}_1 = \mathbf{b}_0 + \alpha_0 \mathbf{T}_0$ and $\mathbf{b}_2 = \mathbf{b}_3 - \alpha_3 \mathbf{T}_3$ for arbitrary $\alpha_0, \alpha_3 \neq 0$. **However, if the value for α_0 or α_3 is too small, there could exist corners at the endpoints. On the other hand, if the**

value is too large, there could exist loops. Proper values of $\bar{\alpha}_0$ and $\bar{\alpha}_3$ are first given as below following an idea from [25] (see Figure 3):

For $i = 0, 3$, let θ_i be the angle between \mathbf{T}_i and the vector $\mathbf{b}_3 - \mathbf{b}_0$ and we have that

$$\cos \theta_i = \frac{(\mathbf{b}_3 - \mathbf{b}_0) \cdot \mathbf{T}_i}{\|\mathbf{b}_3 - \mathbf{b}_0\|}.$$

Let

$$d_i = \frac{\|\mathbf{b}_3 - \mathbf{b}_0\|}{3 \cos \theta_i} = \frac{\|\mathbf{b}_3 - \mathbf{b}_0\|^2}{3(\mathbf{b}_3 - \mathbf{b}_0) \cdot \mathbf{T}_i},$$

and take

$$\bar{\alpha}_i = \begin{cases} d_i & \text{if } \theta_i \leq 60^\circ, \\ \frac{2}{3}\|\mathbf{b}_3 - \mathbf{b}_0\|, & \text{otherwise.} \end{cases}$$

Let $\tilde{\alpha}_i = \omega \bar{\alpha}_i$, where $\omega > 0$ is a free parameter influencing the shape of the final interpolation curve, which can be used to deform an object as shown in our later examples. We take $\omega = 1$ as a default selection. Taking $\mathbf{b}_1 = \mathbf{b}_0 + \tilde{\alpha}_0 \mathbf{T}_0$ and $\mathbf{b}_2 = \mathbf{b}_3 - \tilde{\alpha}_3 \mathbf{T}_3$, we get the curve $\mathbf{g}_v(t)$ which is the dotted curve in Figure 3.

3. Minimum bending energy

The parameters $\tilde{\alpha}_0$ and $\tilde{\alpha}_3$ selected as above generally result in a fine curve shape, but we will improve it by enforcing the resulted curve to have a minimum bending energy.

Take $\alpha_0 = s \tilde{\alpha}_0$ and $\alpha_3 = (2 - s) \tilde{\alpha}_3$ and let s minimize the bending energy $\int_0^1 \|\mathbf{g}_v'(t)\|^2 dt$ as follows:

$$\Delta \mathbf{b}_0 = \alpha_0 \mathbf{T}_0,$$

$$\Delta \mathbf{b}_1 = (\mathbf{b}_3 - \mathbf{b}_0) - (\alpha_0 \mathbf{T}_0 + \alpha_3 \mathbf{T}_3),$$

$$\Delta \mathbf{b}_2 = \alpha_3 \mathbf{T}_3.$$

From (1) and (3), we get

$$\begin{aligned} \mathbf{g}_v'(t) &= 3(\Delta \mathbf{b}_0 B_0^2(t) + \Delta \mathbf{b}_1 B_1^2(t) + \Delta \mathbf{b}_2 B_2^2(t)), \\ \|\mathbf{g}_v'(t)\|^2 &= 9 \sum_{k=0}^4 \left(\sum_{i+j=k} \frac{C_2^i C_2^j}{C_4^k} \Delta \mathbf{b}_i \cdot \Delta \mathbf{b}_j \right) B_k^4(t). \end{aligned}$$

From (2) we have

$$\begin{aligned} \int_0^1 \|\mathbf{g}_v'(t)\|^2 dt &= \frac{9}{5} \sum_{k=0}^4 \sum_{i+j=k} \frac{C_2^i C_2^j}{C_4^k} \Delta \mathbf{b}_i \cdot \Delta \mathbf{b}_j \\ &= \frac{9}{5} (\|\Delta \mathbf{b}_0\|^2 + \Delta \mathbf{b}_0 \cdot \Delta \mathbf{b}_1 + \\ &\quad \frac{1}{3} (2\|\Delta \mathbf{b}_1\|^2 + \Delta \mathbf{b}_0 \cdot \Delta \mathbf{b}_2) + \Delta \mathbf{b}_1 \cdot \Delta \mathbf{b}_2 + \|\Delta \mathbf{b}_2\|^2). \end{aligned}$$

Let $\frac{d}{ds} \int_0^1 \|\mathbf{g}_v'(t)\|^2 dt = 0$ and we get a unique s that minimizes the bending energy

$$s = \frac{8\tilde{\alpha}_3^2 + \tilde{\alpha}_0(\mathbf{b}_3 - \mathbf{b}_0) \cdot \mathbf{T}_0 + 2\tilde{\alpha}_0\tilde{\alpha}_3\mathbf{T}_0 \cdot \mathbf{T}_3 + \tilde{\alpha}_3(\mathbf{b}_0 - \mathbf{b}_3) \cdot \mathbf{T}_3}{2(2\tilde{\alpha}_0^2 + \tilde{\alpha}_0\tilde{\alpha}_3\mathbf{T}_0 \cdot \mathbf{T}_3 + 2\tilde{\alpha}_3^2)}. \quad (6)$$

We can also minimize $\int_0^1 \|\mathbf{g}_v''(t)\|^2 dt$ in a similar way and get

$$s = \frac{4\tilde{\alpha}_3^2 + \tilde{\alpha}_0(\mathbf{b}_3 - \mathbf{b}_0) \cdot \mathbf{T}_0 - 2\tilde{\alpha}_0\tilde{\alpha}_3\mathbf{T}_0 \cdot \mathbf{T}_3 + \tilde{\alpha}_3(\mathbf{b}_0 - \mathbf{b}_3) \cdot \mathbf{T}_3}{2(\tilde{\alpha}_0^2 - \tilde{\alpha}_0\tilde{\alpha}_3\mathbf{T}_0 \cdot \mathbf{T}_3 + \tilde{\alpha}_3^2)}. \quad (7)$$

It can be seen from (6) or (7) that $s = 1$ when $\theta_0 = \theta_3$, i.e. when $(\mathbf{b}_3 - \mathbf{b}_0) \cdot \mathbf{T}_0 = (\mathbf{b}_3 - \mathbf{b}_0) \cdot \mathbf{T}_3$. When $s = 0$ or $s = 2$, the endpoint \mathbf{b}_0 or \mathbf{b}_3 will be a singular point and we set $s = 1$ for the cases. The solid curve in Figure 3 is a cubic curve with minimum bending energy with s expressed as (7). Figures 1 and 2 show the resulted curves $\mathbf{g}_v(t)$ generated in this way.

Lemma 3.2 The boundary curve $\mathbf{g}_v(t) = \sum_{i=0}^3 \mathbf{b}_i B_i^3(t)$ can be defined as follows

$$\mathbf{b}_0 = \mathbf{p}_0, \mathbf{b}_1 = \mathbf{b}_0 + \alpha_0 \mathbf{T}_0, \mathbf{b}_2 = \mathbf{b}_3 - \alpha_3 \mathbf{T}_3, \mathbf{b}_3 = \mathbf{p}_1$$

with $\alpha_0, \alpha_3 \neq 0$. Furthermore, for any given ω there exists a unique value of $\alpha_0 = s\tilde{\alpha}_0$ and $\alpha_3 = (2 - s)\tilde{\alpha}_3$ with s expressed as (6) or (7) to minimize the bending energy $\int_0^1 \|\mathbf{g}_v'(t)\|^2 dt$ or $\int_0^1 \|\mathbf{g}_v''(t)\|^2 dt$.

From the construction of $\mathbf{g}_v(t)$ and the properties of Bézier curves, it can be verified that

Lemma 3.3 $\mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t) = \mathbf{g}_v[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_0, \mathbf{n}_0](1 - t)$.

Figure 4 shows the boundary curves for the sampling points from a torus and a sphere.

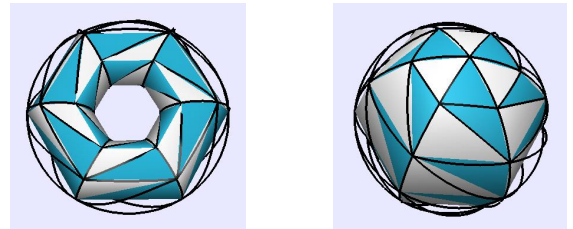


Figure 4: Boundary curves

3.2 Normal Vector Construction

The normal vector is to be given in a quadratic form in this subsection. It can be obtained by solving a system of linear equations, but with such methods it is difficult to control the variation of the resulted normal vectors from \mathbf{n}_0 to \mathbf{n}_1 . The normal vector in our algorithm will be constructed

geometrically according to two cases: $\mathbf{n}_0 = \mathbf{n}_1$ and $\mathbf{n}_0 \neq \mathbf{n}_1$.

When $\mathbf{n}_0 = \mathbf{n}_1$, the curve $\mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t)$ is contained in a plane with the normal vector \mathbf{n} . The normal vector $\mathbf{g}_n(t)$ is defined as follows:

$$\mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t) = \frac{\mathbf{n} \times \mathbf{g}'_v(t) \|\mathbf{g}'_v(t)\|}{\|\mathbf{g}'_v(t)\|^2} \quad (8)$$

When $\mathbf{n}_0 \neq \mathbf{n}_1$, we define $\mathbf{g}_n(t)$ in the following rational quadratic form

$$\mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t) = (1 - \kappa(t))\mathbf{n}_0 + \kappa(t)\mathbf{n}_1, \quad (9)$$

where $\kappa(t)$ is a rational quadratic function as follows

$$\kappa(t) = \frac{\mathbf{n}_0 \cdot \mathbf{g}'_v(t)}{(\mathbf{n}_0 - \mathbf{n}_1) \cdot \mathbf{g}'_v(t)}.$$

Lemma 3.4 Equation (8) and (9) define the normal vector $\mathbf{g}_n(t)$.

Proof: When $\mathbf{n}_0 = \mathbf{n}_1$, since $\mathbf{g}'_v(0) = \alpha_0 \mathbf{T}_0$ and $\mathbf{T}_0 = \mathbf{n}_0 \times \mathbf{n}$, we have

$$\mathbf{n}_0 = \mathbf{n} \times \mathbf{T}_0 = \frac{\mathbf{n} \times \mathbf{g}'_v(0)}{\alpha_0} = \frac{\mathbf{n} \times \mathbf{g}'_v(0)}{\|\mathbf{g}'_v(0)\|} = \mathbf{g}_n(0).$$

It can be verified in a similar way that $\mathbf{g}_n(1) = \mathbf{n}_1$.

Furthermore we have

$$\mathbf{g}_n(t) \cdot \mathbf{g}'_v(t) = \frac{\mathbf{n} \times \mathbf{g}'_v(t)}{\|\mathbf{g}'_v(t)\|} \cdot \mathbf{g}'_v(t) = 0.$$

When $\mathbf{n}_0 \neq \mathbf{n}_1$, since $\mathbf{g}'_v(0) \cdot \mathbf{n}_0 = 0$ and $\mathbf{g}'_v(1) \cdot \mathbf{n}_1 = 0$, we get $\kappa(0) = 0$ and $\kappa(1) = 1$ and then

$$\mathbf{g}_n(0) = (1 - \kappa(0))\mathbf{n}_0 + \kappa(0)\mathbf{n}_1 = \mathbf{n}_0,$$

$$\mathbf{g}_n(1) = (1 - \kappa(1))\mathbf{n}_0 + \kappa(1)\mathbf{n}_1 = \mathbf{n}_1.$$

Substitute $\kappa(t)$ into $\mathbf{g}_n(t)$

$$\begin{aligned} \mathbf{g}_n(t) \cdot \mathbf{g}'_v(t) &= ((1 - \kappa(t))\mathbf{n}_0 + \kappa(t)\mathbf{n}_1) \cdot \mathbf{g}'_v(t) \\ &= \frac{(\mathbf{n}_0 \cdot \mathbf{g}'_v(t)) \mathbf{n}_1 - (\mathbf{n}_1 \cdot \mathbf{g}'_v(t)) \mathbf{n}_0}{(\mathbf{n}_0 - \mathbf{n}_1) \cdot \mathbf{g}'_v(t)} \cdot \mathbf{g}'_v(t) = 0. \end{aligned}$$

■

The construction of $\mathbf{g}_n(t)$ will always succeed except for a particular case that $\mathbf{n}_0 \neq \mathbf{n}_1$ and $\mathbf{n}_0 \cdot \mathbf{T}_3 = 0$, where \mathbf{T}_3 is expressed in (5). We have the following lemmas for that case.

Lemma 3.5 When $\mathbf{n}_0 \cdot \mathbf{T}_3 = 0$ and $\mathbf{n}_0 \neq \mathbf{n}_1$, the cubic Bézier curve $\mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t)$ will be contained in a plane. Furthermore the case will arise if and only if $\mathbf{n}_0 \cdot \mathbf{n}_1 = 0$, that is \mathbf{n}_0 is perpendicular to the plane passing through $\mathbf{p}_0\mathbf{p}_1$ with normal direction \mathbf{n}_1 .

Proof: From the construction of \mathbf{T}_0 , it is evident that $\mathbf{n}_0 \cdot \mathbf{T}_0 = 0$. Additionally since $\mathbf{n}_0 \cdot \mathbf{T}_3 = 0$ we have that both \mathbf{T}_0 and \mathbf{T}_3 are parallel to the tangent plane going through \mathbf{p}_0 with its normal vector parallel to \mathbf{n}_0 . Write this plane as \mathbf{F}_0 . We have that $\mathbf{p}_0, \tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2$ and \mathbf{p}_1 lie in \mathbf{F}_0 , so does the curve $\mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1]$. Furthermore since $\tilde{\mathbf{b}}_2 = \mathbf{p}_0 + \frac{(\mathbf{p}_1 - \mathbf{p}_0) \cdot \mathbf{n}_1}{\|\mathbf{n}_1\|^2} \mathbf{n}_1$ and $(\tilde{\mathbf{b}}_2 - \mathbf{p}_0) \cdot \mathbf{n}_0 = 0$, so we have $\mathbf{n}_1 \cdot \mathbf{n}_0 = 0$.

■



Figure 5: Normal of general cases



Figure 6: Normal of special cases

Lemma 3.6 When $\mathbf{n}_0 \cdot \mathbf{T}_3 = 0$, $\mathbf{n}_0 \neq \mathbf{n}_1$, there might exist no quadratic normal vector $\mathbf{g}_n(t)$ for its corresponding boundary curve $\mathbf{g}_v(t)$.

Proof: From the definition of $\mathbf{g}_n(t)$, a quadratic $\mathbf{g}_n(t)$ can be written in the form $\mathbf{g}_n(t) = \mathbf{n}_{00}B_0^2(t) + \mathbf{n}_{01}B_1^2(t) + \mathbf{n}_{02}B_2^2(t)$, where $\mathbf{n}_{00} = c_0\mathbf{n}_0$, $\mathbf{n}_{02} = c_2\mathbf{n}_1$ are nonzero constants. From Theorem 2.1, $\mathbf{g}_n(t) \cdot \mathbf{g}'_v(t) = 0$ if and only if

$$\begin{aligned} \mathbf{d}_0 \cdot \mathbf{n}_{01} + \mathbf{d}_1 \cdot \mathbf{n}_{00} &= 0, \\ \mathbf{d}_0 \cdot \mathbf{n}_{02} + 4\mathbf{d}_1 \cdot \mathbf{n}_{01} + \mathbf{d}_2 \cdot \mathbf{n}_{00} &= 0, \\ \mathbf{d}_1 \cdot \mathbf{n}_{02} + \mathbf{d}_2 \cdot \mathbf{n}_{01} &= 0, \end{aligned} \quad (10)$$

with $\mathbf{g}_v(t) = \sum_{i=0}^3 \mathbf{b}_i B_i^3(t)$ and $\mathbf{d}_i = \Delta \mathbf{b}_i = \mathbf{b}_{i+1} - \mathbf{b}_i$, $i = 0, 1, 2$. Since

$$\begin{aligned} \lambda_2 &= \mathbf{d}_2 \cdot \mathbf{n}_{01} = -\mathbf{d}_1 \cdot \mathbf{n}_{02}, \\ \lambda_1 &= \mathbf{d}_1 \cdot \mathbf{n}_{01} = -\frac{\mathbf{d}_0 \cdot \mathbf{n}_{02} + \mathbf{d}_2 \cdot \mathbf{n}_{00}}{4}, \end{aligned}$$

we have

$$\mathbf{n}_{01} = \mu_2 \mathbf{d}_2 + \mu_1 \mathbf{d}_1 + \alpha \mathbf{d}_2 \times \mathbf{d}_1 \quad (11)$$

for arbitrary α with

$$\begin{aligned} \mu_2 &= \frac{\lambda_1 \mathbf{d}_2 \cdot \mathbf{d}_1 - \lambda_2 \|\mathbf{d}_1\|^2}{(\mathbf{d}_2 \cdot \mathbf{d}_1)^2 - \|\mathbf{d}_2\|^2 \|\mathbf{d}_1\|^2}, \\ \mu_1 &= \frac{\lambda_2 \mathbf{d}_2 \cdot \mathbf{d}_1 - \lambda_1 \|\mathbf{d}_2\|^2}{(\mathbf{d}_2 \cdot \mathbf{d}_1)^2 - \|\mathbf{d}_2\|^2 \|\mathbf{d}_1\|^2}. \end{aligned}$$

Substitute (11) into the last equation of (10) and note that $\mathbf{d}_0 \times \mathbf{d}_1 \cdot \mathbf{d}_2 = 0$ and $\mathbf{d}_1 \cdot \mathbf{n}_{00} = 0$. It must be satisfied that $\mathbf{d}_0 \cdot \mathbf{n}_{01} = \mu_2 \mathbf{d}_0 \cdot \mathbf{d}_2 + \mu_1 \mathbf{d}_0 \cdot \mathbf{d}_1 = 0$. However this is not always true. \blacksquare

From the above lemma, a proper cubic curve $\mathbf{g}_v(t)$ must be reset to avoid this unexpected case so that when $\mathbf{n}_0 \neq \mathbf{n}_1$ we will not end with a plane curve $\mathbf{g}_v(t)$. We reset $\mathbf{T}_0 = \frac{3}{4}\mathbf{T}_0 + \frac{1}{4}\mathbf{n}_1$ and we have that $\mathbf{T}_0 \cdot \mathbf{n}_0 = (\frac{3}{4}\mathbf{T}_0 + \frac{1}{4}\mathbf{n}_1) \cdot \mathbf{T}_0 = 0$ and $\mathbf{T}_0 \cdot \mathbf{n}_1 = (\frac{3}{4}\mathbf{T}_0 + \frac{1}{4}\mathbf{n}_1) \cdot \mathbf{n}_1 = \frac{1}{4} \neq 0$.

Since the essence of the normal vector is its direction, we can take $\mathbf{g}_n(t)$ in the following quadratic

form throughout this paper.

$$\mathbf{g}_n(t) = \begin{cases} \mathbf{n} \times \mathbf{g}'_v(t) & \mathbf{n}_0 = \mathbf{n}_1, \\ (\mathbf{n}_0 \cdot \mathbf{g}'_v(t)) \mathbf{n}_1 - (\mathbf{n}_1 \cdot \mathbf{g}'_v(t)) \mathbf{n}_0 & \mathbf{n}_0 \neq \mathbf{n}_1. \end{cases}$$

We have the following lemma

Lemma 3.7 *We have $\mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t) = \mathbf{g}_n[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_0, \mathbf{n}_0](1-t)$.*

Figures 5 and 6 show the figures of the curves $\mathbf{g}(t) = \mathbf{g}_v(t) + \frac{\mathbf{g}_n(t)}{\|\mathbf{g}_n(t)\|}$ for the curve $\mathbf{g}_v(t)$ as shown in Figures 1 and 2. You can get a unit normal vector locating at $\mathbf{g}_v(t)$ with a vector starting from $\mathbf{g}_v(t)$ and ending at $\mathbf{g}(t)$. Figure 7 shows boundary curves $\mathbf{g}_v(t)$ with the normal vectors $\mathbf{g}_n(t)$ varying along with them for a quadrilateral face and a triangular face.

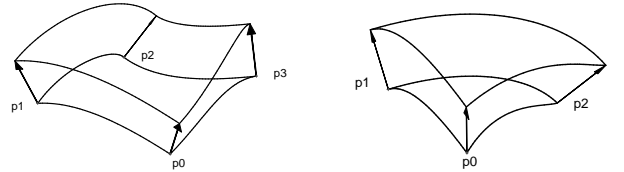


Figure 7: Boundary constructions

4 Surface Interpolants

In this section, we will illustrate the construction of the surface interpolants including a quadrilateral and a triangular interpolant. Before that construction the control curve will be first constructed.

4.1 Control Curve

Control Curve: Suppose $(\mathbf{p}_0, \mathbf{n}_0), (\mathbf{p}_1, \mathbf{n}_1) \in [\mathcal{P}, \mathcal{N}]$ are given. Find a control curve

$$\begin{aligned} \mathbf{g}_c : [\mathcal{P}, \mathcal{N}]^2 \times [0, 1] &\rightarrow \mathcal{R}^3, \\ (\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1, t) &\rightarrow \mathbf{g}_c[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t), \end{aligned}$$

such that

$$(\mathbf{g}_c(t) - \mathbf{g}_v(t)) \cdot \mathbf{g}_n(t) = 0$$

where $\mathbf{g}_v(t)$ and $\mathbf{g}_n(t)$ are given in section 3. Without confusion, we denote $\mathbf{g}_c[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t)$ by $\mathbf{g}_c(t)$. In fact $\mathbf{g}_c(t)$ is a space curve with the property that the vector from $\mathbf{g}_v(t)$ to $\mathbf{g}_c(t)$ is perpendicular to $\mathbf{g}_n(t)$. See Figure 8.

The control curve is to be constructed in a symbolic form with minimum bending energy. Suppose $\mathbf{g}_v(t), \mathbf{g}_n(t)$ and $\mathbf{g}_c(t)$ are given in the following forms with $\mathbf{d}_i, i = 0, 1, 2, 3$, to be determined,

$$\begin{aligned} \mathbf{g}_v(t) &= \sum_{i=0}^3 \mathbf{b}_i B_i^3(t), \\ \mathbf{g}_n(t) &= \sum_{i=0}^2 \mathbf{n}_{0i} B_i^2(t), \quad \mathbf{g}_c(t) = \sum_{i=0}^3 \mathbf{d}_i B_i^3(t). \end{aligned}$$

It is clear that

$$\mathbf{b}_0 = \mathbf{p}_0, \mathbf{b}_3 = \mathbf{p}_1, \quad \mathbf{n}_{00} = c_0 \mathbf{n}_0, \mathbf{n}_{02} = c_1 \mathbf{n}_1,$$

where c_0 and c_1 are constant parameters.

From Theorem 2.1 and $(\mathbf{g}_c(t) - \mathbf{g}_v(t)) \cdot \mathbf{g}_n(t) = 0$, we have

$$\sum_{i+j=k} \frac{C_3^i C_2^j}{C_5^k} (\mathbf{d}_i - \mathbf{b}_i) \cdot \mathbf{n}_{0j} = 0, \quad 0 \leq k \leq 5. \quad (12)$$

Fix \mathbf{d}_0 and \mathbf{d}_3 by the following constraints which come from (12) by setting $k = 0$ and 5,

$$(\mathbf{d}_0 - \mathbf{p}_0) \cdot \mathbf{n}_0 = 0, \quad (\mathbf{d}_3 - \mathbf{p}_1) \cdot \mathbf{n}_1 = 0. \quad (13)$$

We rewrite $\mathbf{g}_c[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](t)$ as $\mathbf{g}_c[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1, \mathbf{d}_0, \mathbf{d}_3](t)$ to show its dependence on \mathbf{d}_0 and \mathbf{d}_3 when needed. Specific values for \mathbf{d}_0 and \mathbf{d}_3 will be given for some requirements, which will be shown in the constructions of the pre-interpolants in section 4. Write $\mathbf{f}_i = \mathbf{d}_i - \mathbf{b}_i$, $i = 0, 1, 2, 3$, and we get from (12) the following equations

$$\begin{aligned} 2\mathbf{f}_0 \cdot \mathbf{n}_{01} + 3\mathbf{f}_1 \cdot \mathbf{n}_{00} &= 0, \\ \mathbf{f}_0 \cdot \mathbf{n}_{02} + 6\mathbf{f}_1 \cdot \mathbf{n}_{01} + 3\mathbf{f}_2 \cdot \mathbf{n}_{00} &= 0, \\ 3\mathbf{f}_1 \cdot \mathbf{n}_{02} + 6\mathbf{f}_2 \cdot \mathbf{n}_{01} + \mathbf{f}_3 \cdot \mathbf{n}_{00} &= 0, \\ 3\mathbf{f}_2 \cdot \mathbf{n}_{02} + 2\mathbf{f}_3 \cdot \mathbf{n}_{01} &= 0. \end{aligned} \quad (14)$$

A symbolic expression for \mathbf{d}_1 and \mathbf{d}_2 will be obtained from (14) according to the three cases below:

(1) When $\mathbf{n}_{00} \times \mathbf{n}_{02} \cdot \mathbf{n}_{01} \neq 0$, we get from (14) that

$$\begin{aligned} \lambda_{10} &= \mathbf{f}_1 \cdot \mathbf{n}_{00} = -\frac{2\mathbf{f}_0 \cdot \mathbf{n}_{01}}{3}, \\ \lambda_{11} &= \mathbf{f}_1 \cdot \mathbf{n}_{02} = -\frac{\mathbf{f}_3 \cdot \mathbf{n}_{00} + 6\alpha_2}{3}, \\ \lambda_{20} &= \mathbf{f}_2 \cdot \mathbf{n}_{00} = -\frac{\mathbf{f}_0 \cdot \mathbf{n}_{02} + 6\alpha_1}{3}, \\ \lambda_{21} &= \mathbf{f}_2 \cdot \mathbf{n}_{02} = -\frac{2\mathbf{f}_3 \cdot \mathbf{n}_{01}}{3}, \end{aligned}$$

where α_i are free parameters with geometrical meanings that $\alpha_i = \mathbf{f}_i \cdot \mathbf{n}_{01}$, $i = 1, 2$.

From Lemma 4, we have

$$\begin{aligned} (\mathbf{n}_{00} \times \mathbf{n}_{02} \cdot \mathbf{n}_{01}) \mathbf{f}_1 &= (\mathbf{f}_1 \cdot \mathbf{n}_{00})(\mathbf{n}_{02} \times \mathbf{n}_{01}) + \\ &+ (\mathbf{f}_1 \cdot \mathbf{n}_{02})(\mathbf{n}_{01} \times \mathbf{n}_{00}) + (\mathbf{f}_1 \cdot \mathbf{n}_{01})(\mathbf{n}_{00} \times \mathbf{n}_{02}), \end{aligned}$$

so

$$\begin{aligned} \mathbf{d}_i &= \mathbf{b}_i + \\ &+ \frac{\lambda_{i0}(\mathbf{n}_{02} \times \mathbf{n}_{01}) + \lambda_{i1}(\mathbf{n}_{01} \times \mathbf{n}_{00}) + \alpha_i(\mathbf{n}_{00} \times \mathbf{n}_{02})}{\mathbf{n}_{00} \times \mathbf{n}_{02} \cdot \mathbf{n}_{01}}, \end{aligned} \quad (15)$$

where $i = 1, 2$. Select values for α_1 and α_2 to minimize the bending energy $\int_0^1 \|\mathbf{g}_c''(t)\|^2 dt$, we have

$$\begin{aligned} \alpha_1 &= -\frac{\eta(3\mathbf{b}_1 - 2\mathbf{d}_0 - \mathbf{d}_3) + 3\lambda_{10}\mathbf{N}_{21} + 3\lambda_{11}\mathbf{N}_{10} \cdot \mathbf{N}_{02}}{3}, \\ \alpha_2 &= -\frac{\eta(3\mathbf{b}_2 - 2\mathbf{d}_3 - \mathbf{d}_0) + 3\lambda_{20}\mathbf{N}_{21} + 3\lambda_{21}\mathbf{N}_{10} \cdot \mathbf{N}_{02}}{3}, \end{aligned}$$

where $\mathbf{N}_{21} = \mathbf{n}_{02} \times \mathbf{n}_{01}$, $\mathbf{N}_{10} = \mathbf{n}_{01} \times \mathbf{n}_{00}$, with $\mathbf{N}_{02} = \mathbf{n}_{00} \times \mathbf{n}_{02}$ and $\eta = \mathbf{n}_{00} \times \mathbf{n}_{02} \cdot \mathbf{n}_{01}$.

In fact, we always have $\mathbf{n}_{00} \times \mathbf{n}_{02} \cdot \mathbf{n}_{01} = 0$ for $\mathbf{g}_n(t)$ constructed in Section 2, which can be seen from the following two cases:

(2) When $\mathbf{n}_0 = \mathbf{n}_1$, from $\mathbf{g}_n(t) = \mathbf{n} \times \mathbf{g}'_v(t)$ we have $\mathbf{n}_{00} = \mathbf{n}_{02}$. It is evident that $\mathbf{n}_{00} \times \mathbf{n}_{02} \cdot \mathbf{n}_{01} = 0$. From (14), it is easy to get

$$\begin{aligned}\lambda_{10} &= \mathbf{f}_1 \cdot \mathbf{n}_{00} = -\frac{2\mathbf{f}_0 \cdot \mathbf{n}_{01}}{3}, \\ \lambda_{11} &= \mathbf{f}_1 \cdot \mathbf{n}_{01} = \frac{2\mathbf{f}_3 \cdot \mathbf{n}_{01} - \mathbf{f}_0 \cdot \mathbf{n}_{00}}{6}, \\ \lambda_{20} &= \mathbf{f}_2 \cdot \mathbf{n}_{00} = -\frac{2\mathbf{f}_3 \cdot \mathbf{n}_{01}}{3}, \\ \lambda_{21} &= \mathbf{f}_2 \cdot \mathbf{n}_{01} = \frac{2\mathbf{f}_0 \cdot \mathbf{n}_{01} - \mathbf{f}_3 \cdot \mathbf{n}_{00}}{6},\end{aligned}$$

so

$$\mathbf{d}_i = \mathbf{b}_i + \mu_{i0}\mathbf{n}_{00} + \mu_{i1}\mathbf{n}_{01} + \beta_i(\mathbf{n}_{00} \times \mathbf{n}_{01}), \quad (16)$$

where

$$\begin{aligned}\mu_{i0} &= \frac{\lambda_{i1}\mathbf{n}_{00} \cdot \mathbf{n}_{01} - \lambda_{i0}\|\mathbf{n}_{01}\|^2}{(\mathbf{n}_{00} \cdot \mathbf{n}_{01})^2 - (\|\mathbf{n}_{00}\|\|\mathbf{n}_{01}\|)^2}, \\ \mu_{i1} &= \frac{\lambda_{i0}\mathbf{n}_{00} \cdot \mathbf{n}_{01} - \lambda_{i1}\|\mathbf{n}_{00}\|^2}{(\mathbf{n}_{00} \cdot \mathbf{n}_{01})^2 - (\|\mathbf{n}_{00}\|\|\mathbf{n}_{01}\|)^2},\end{aligned}$$

and β_i are free parameters with geometrical representation: $\beta_i = \frac{\mathbf{n}_{00} \times \mathbf{n}_{01} \cdot \mathbf{f}_i}{\|\mathbf{n}_{00} \times \mathbf{n}_{01}\|^2}$, $i = 1, 2$. Set β_i to minimize the bending energy $\int_0^1 \|\mathbf{g}_c''(t)\|^2 dt$ and we get

$$\begin{aligned}\beta_1 &= \frac{(2\mathbf{d}_0 + \mathbf{d}_3 - 3\mathbf{b}_1) \cdot (\mathbf{n}_{00} \times \mathbf{n}_{01})}{3\|\mathbf{n}_{00} \times \mathbf{n}_{01}\|^2}, \\ \beta_2 &= \frac{(\mathbf{d}_0 + 2\mathbf{d}_3 - 3\mathbf{b}_2) \cdot (\mathbf{n}_{00} \times \mathbf{n}_{01})}{3\|\mathbf{n}_{00} \times \mathbf{n}_{01}\|^2}.\end{aligned}$$

(3) Case $\mathbf{n}_0 \neq \mathbf{n}_1$. Since

$$\begin{aligned}\mathbf{n}_{00} &= \mathbf{g}_n(0) = -(\mathbf{n}_1 \cdot \mathbf{g}'_v(0))\mathbf{n}_0, \\ \mathbf{n}_{02} &= \mathbf{g}_n(1) = (\mathbf{n}_0 \cdot \mathbf{g}'_v(1))\mathbf{n}_1, \\ \mathbf{n}_{01} &= \frac{\mathbf{g}_n'(0)}{2} + \mathbf{n}_{00} = \lambda_0\mathbf{n}_{00} + \lambda_2\mathbf{n}_{02},\end{aligned}$$

with

$$\lambda_0 = 1 + \frac{\mathbf{n}_1 \cdot \mathbf{g}_v''(0)}{2\mathbf{n}_1 \cdot \mathbf{g}_v'(0)}, \quad \lambda_2 = \frac{\mathbf{n}_0 \cdot \mathbf{g}_v''(0)}{2\mathbf{n}_0 \cdot \mathbf{g}_v'(1)},$$

it can be seen that $\mathbf{n}_{00} \times \mathbf{n}_{02} \cdot \mathbf{n}_{01} = 0$.

Substituting $\mathbf{n}_{01} = \lambda_0\mathbf{n}_{00} + \lambda_2\mathbf{n}_{02}$ into (14), we get

$$\begin{aligned}\lambda_{10} &= \mathbf{f}_1 \cdot \mathbf{n}_{00} = -\frac{2\mathbf{f}_0 \cdot \mathbf{n}_{01}}{3}, \\ \lambda_{12} &= \mathbf{f}_1 \cdot \mathbf{n}_{02} = \frac{\mathbf{f}_3 \cdot \mathbf{n}_{00} + (8\lambda_0^2\mathbf{f}_0 - 4\lambda_2\mathbf{f}_3) \cdot \mathbf{n}_{01} - 2\lambda_0\mathbf{f}_0 \cdot \mathbf{n}_{02}}{3(4\lambda_0\lambda_2 - 1)}, \\ \lambda_{20} &= \mathbf{f}_2 \cdot \mathbf{n}_{00} = \frac{\mathbf{f}_0 \cdot \mathbf{n}_{02} + (8\lambda_2^2\mathbf{f}_3 - 4\lambda_0\mathbf{f}_0) \cdot \mathbf{n}_{01} - 2\lambda_2\mathbf{f}_3 \cdot \mathbf{n}_{00}}{3(4\lambda_0\lambda_2 - 1)}, \\ \lambda_{22} &= \mathbf{f}_2 \cdot \mathbf{n}_{02} = -\frac{2\mathbf{f}_3 \cdot \mathbf{n}_{01}}{3}.\end{aligned}$$

When $4\lambda_0\lambda_2 = 1$, it can be seen from (13) that

$$\begin{aligned}\mathbf{f}_3 \cdot \mathbf{n}_{00} + (8\lambda_0^2\mathbf{f}_0 - 4\lambda_2\mathbf{f}_3) \cdot \mathbf{n}_{01} - 2\lambda_0\mathbf{f}_0 \cdot \mathbf{n}_{02} \\ = \frac{32\lambda_2^5\mathbf{f}_3 \cdot \mathbf{n}_{02} - \mathbf{f}_0 \cdot \mathbf{n}_{00}}{8\lambda_2^3} = 0, \\ \mathbf{f}_0 \cdot \mathbf{n}_{02} + (8\lambda_2^2\mathbf{f}_3 - 4\lambda_0\mathbf{f}_0) \cdot \mathbf{n}_{01} - 2\lambda_2\mathbf{f}_3 \cdot \mathbf{n}_{00} \\ = \frac{32\lambda_0^5\mathbf{f}_0 \cdot \mathbf{n}_{00} - \mathbf{f}_3 \cdot \mathbf{n}_{02}}{8\lambda_0^3} = 0.\end{aligned}$$

We can see that (14) is satisfied for any value of $\lambda_{12}, \lambda_{20}$. We set $\lambda_{12} = \lambda_{20} = 0$ in this case.

Similar to case (2), we get

$$\mathbf{d}_i = \mathbf{b}_i + \mu_{i0}\mathbf{n}_{00} + \mu_{i2}\mathbf{n}_{02} + \gamma_i(\mathbf{n}_{00} \times \mathbf{n}_{02}), \quad (17)$$

where

$$\begin{aligned}\mu_{i0} &= \frac{\lambda_{i2}\mathbf{n}_{00} \cdot \mathbf{n}_{02} - \lambda_{i0}\|\mathbf{n}_{02}\|^2}{(\mathbf{n}_{00} \cdot \mathbf{n}_{02})^2 - (\|\mathbf{n}_{00}\|\|\mathbf{n}_{02}\|)^2}, \\ \mu_{i2} &= \frac{\lambda_{i0}\mathbf{n}_{00} \cdot \mathbf{n}_{02} - \lambda_{i2}\|\mathbf{n}_{00}\|^2}{(\mathbf{n}_{00} \cdot \mathbf{n}_{02})^2 - (\|\mathbf{n}_{00}\|\|\mathbf{n}_{02}\|)^2},\end{aligned}$$

and γ_i are free parameters with geometrical representation $\gamma_i = \frac{\mathbf{f}_i \cdot (\mathbf{n}_{00} \times \mathbf{n}_{02})}{\|\mathbf{n}_{00} \times \mathbf{n}_{02}\|^2}$, $i = 1, 2$. Finding values for γ_i to minimize the bending energy $\int_0^1 \|\mathbf{g}_c''(t)\|^2 dt$, we get

$$\begin{aligned}\gamma_1 &= \frac{(2\mathbf{d}_0 + \mathbf{d}_3 - 3\mathbf{b}_1) \cdot (\mathbf{n}_{00} \times \mathbf{n}_{02})}{3\|\mathbf{n}_{00} \times \mathbf{n}_{02}\|^2}, \\ \gamma_2 &= \frac{(\mathbf{d}_0 + 2\mathbf{d}_3 - 3\mathbf{b}_2) \cdot (\mathbf{n}_{00} \times \mathbf{n}_{02})}{3\|\mathbf{n}_{00} \times \mathbf{n}_{02}\|^2}.\end{aligned}$$

Lemma 4.1 See Figure 8. Let \mathbf{d}_0 and \mathbf{d}_3 be two fixed points satisfying $(\mathbf{d}_0 - \mathbf{p}_0) \cdot \mathbf{n}_0 = 0$, $(\mathbf{d}_3 - \mathbf{p}_1) \cdot \mathbf{n}_1 = 0$. Then there exists a unique cubic control curve

$\mathbf{g}_c[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1, \mathbf{d}_0, \mathbf{d}_3](t) = \sum_{i=0}^3 \mathbf{d}_i B_i^3(t)$ with minimum bending energy with \mathbf{d}_1 and \mathbf{d}_2 expressed as (16) when $\mathbf{n}_0 = \mathbf{n}_1$ or \mathbf{d}_1 and \mathbf{d}_2 expressed as (17) when $\mathbf{n}_0 \neq \mathbf{n}_1$.

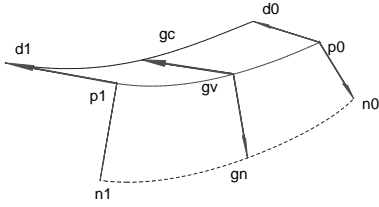


Figure 8: Control curve

4.2 Quadrilateral Interpolant

Without loss of generality suppose that four points $(\mathbf{p}_i, \mathbf{n}_i) \in [\mathcal{P}, \mathcal{N}]$, $0 \leq i \leq 3$. Let (s, t) , $0 \leq s, t \leq 1$, be the local coordinate for a point with respect to the quadrilateral $\mathbf{p}_0 \mathbf{p}_1 \mathbf{p}_2 \mathbf{p}_3$ (Figure 9).

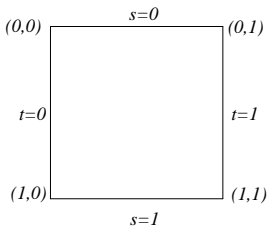


Figure 9: Quadrilateral domain

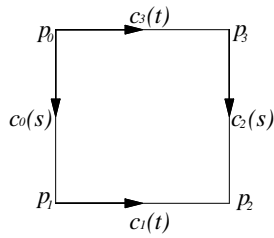


Figure 10: Quadrilateral boundary curves

Set up the boundary curves $\mathbf{c}_i(s)$ or $\mathbf{c}_i(t)$ and the normal fields $\mathbf{n}_i(s)$ or $\mathbf{n}_i(t)$ as follows (Figure 10):

$$\begin{aligned} \mathbf{c}_0(s) &= \mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](s), & \mathbf{n}_0(s) &= \mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](s), \\ \mathbf{c}_2(s) &= \mathbf{g}_v[\mathbf{p}_3, \mathbf{n}_3, \mathbf{p}_2, \mathbf{n}_2](s), & \mathbf{n}_2(s) &= \mathbf{g}_n[\mathbf{p}_3, \mathbf{n}_3, \mathbf{p}_2, \mathbf{n}_2](s), \\ \mathbf{c}_1(t) &= \mathbf{g}_v[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_2, \mathbf{n}_2](t), & \mathbf{n}_1(t) &= \mathbf{g}_n[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_2, \mathbf{n}_2](t), \\ \mathbf{c}_3(t) &= \mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_3, \mathbf{n}_3](t), & \mathbf{n}_3(t) &= \mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_3, \mathbf{n}_3](t). \end{aligned}$$

We will first create a pre-interpolant \mathbf{S}_s such that it interpolates the boundary curves $\mathbf{c}_i(s)$, $i = 0, 2$, $\mathbf{c}_i(t)$, $i = 1, 3$ and interpolates the normal fields $\mathbf{n}_1(t)$ and $\mathbf{n}_3(t)$ along $\mathbf{c}_1(t)$ and $\mathbf{c}_3(t)$ respectively. The problem of the quadrilateral pre-interpolant is formulated as follows:

Quadrilateral pre-interpolant: Let $\mathbf{c}_i(t)$, $\mathbf{c}_i(s)$, $\mathbf{n}_i(t)$, $\mathbf{n}_i(s)$ be stated as above. Find a quadrilateral surface interpolant $\mathbf{S}_s(s, t)$ such that

1. $\mathbf{S}_s(s, t)$ interpolates the boundary curves:

$$\mathbf{S}_s(s, 0) = \mathbf{c}_0(s), \quad \mathbf{S}_s(s, 1) = \mathbf{c}_2(s),$$

$$\mathbf{S}_s(0, t) = \mathbf{c}_3(t), \quad \mathbf{S}_s(1, t) = \mathbf{c}_1(t).$$

2. The tangent field determined by $\frac{\partial \mathbf{S}_s(s, t)}{\partial s}$ and $\frac{\partial \mathbf{S}_s(s, t)}{\partial t}$ are orthogonal to $\mathbf{n}_1(t)$ and $\mathbf{n}_3(t)$ along $\mathbf{c}_1(t)$ and $\mathbf{c}_3(t)$:

$$\frac{\partial \mathbf{S}_s(0, t)}{\partial \mu} \cdot \mathbf{n}_3(t) = 0, \quad \frac{\partial \mathbf{S}_s(1, t)}{\partial \mu} \cdot \mathbf{n}_1(t) = 0, \quad \mu = s, t.$$

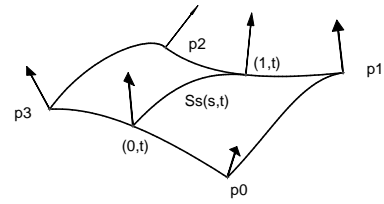


Figure 11: Pre-interpolant construction

We can fill the interior of $\mathbf{S}_s(s, t)$ by passing from every points on $\mathbf{c}_3(t)$ to points on $\mathbf{c}_1(t)$ with a curve $\mathbf{g}_v(s, t)$. See also Figure 11.

Theorem 4.2 A quadrilateral pre-interpolant can be constructed as follows:

$$\mathbf{S}_s(s, t) = \mathbf{g}_v[\mathbf{c}_3(t), \mathbf{n}_3(t), \mathbf{c}_1(t), \mathbf{n}_1(t)](s).$$

Proof:

1. The boundary interpolation is verified as follows:

$$\begin{aligned}
\mathbf{S}_s(s, 0) &= \mathbf{g}_v[\mathbf{c}_3(0), \mathbf{n}_3(0), \mathbf{c}_1(0), \mathbf{n}_1(0)](s) \\
&= \mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](s) = \mathbf{c}_0(s), \\
\mathbf{S}_s(s, 1) &= \mathbf{g}_v[\mathbf{c}_3(1), \mathbf{n}_3(1), \mathbf{c}_1(1), \mathbf{n}_1(1)](s) \\
&= \mathbf{g}_v[\mathbf{p}_3, \mathbf{n}_3, \mathbf{p}_2, \mathbf{n}_2](s) = \mathbf{c}_2(s), \\
\mathbf{S}_s(0, t) &= \mathbf{g}_v[\mathbf{c}_3(t), \mathbf{n}_3(t), \mathbf{c}_1(t), \mathbf{n}_1(t)](0) = \mathbf{c}_3(t), \\
\mathbf{S}_s(1, t) &= \mathbf{g}_v[\mathbf{c}_3(t), \mathbf{n}_3(t), \mathbf{c}_1(t), \mathbf{n}_1(t)](1) = \mathbf{c}_1(t).
\end{aligned}$$

2. The interpolation of the normal fields at $t = 0$ and $t = 1$ is verified as follows:

$$\begin{aligned}
\frac{\partial \mathbf{S}_s(0, t)}{\partial s} \cdot \mathbf{n}_3(t) &= \\
\mathbf{g}'_v[\mathbf{c}_3(t), \mathbf{n}_3(t), \mathbf{c}_1(t), \mathbf{n}_1(t)](0) \cdot \mathbf{n}_3(t) &= 0, \\
\frac{\partial \mathbf{S}_s(0, t)}{\partial t} \cdot \mathbf{n}_3(t) &= \\
\mathbf{g}_v[\mathbf{c}'_3(t), \mathbf{n}'_3(t), \mathbf{c}'_1(t), \mathbf{n}'_1(t)](0) \cdot \mathbf{n}_3(t) &= \\
= \mathbf{c}'_3(t) \cdot \mathbf{n}_3(t) &= 0.
\end{aligned}$$

It can be verified in a similar way that

$$\frac{\partial \mathbf{S}_s(1, t)}{\partial s} \cdot \mathbf{n}_1(t) = 0, \quad \frac{\partial \mathbf{S}_s(1, t)}{\partial t} \cdot \mathbf{n}_1(t) = 0.$$

■

From the theorem, it can be seen that the pre-interpolant $\mathbf{S}_s(s, t)$ can be obtained just as the construction of an interpolation curve from two points. In fact we can get any point on $\mathbf{S}_s(s, t)$ with the de Casteljau algorithm [25]. The theorem above is therefore very attractive for a numerical generation of an interpolation surface. $\mathbf{S}_t(s, t)$ can be defined in a similar way. On the other hand, since $\mathbf{g}_v[\mathbf{c}_3(t), \mathbf{n}_3(t), \mathbf{c}_1(t), \mathbf{n}_1(t)]$ involves some operations such as point projection, the resulted surface $\mathbf{S}_s(s, t)$ will have a complex expression in t .

Treating the loci of the control points as control curves, we now give a new quadrilateral pre-interpolant with $\mathbf{g}_c(t)$.

Theorem 4.3 Suppose the control points of $\mathbf{c}_i(s)$ or $\mathbf{c}_i(t)$ are \mathbf{c}_{ij} , $0 \leq i \leq 3$, $0 \leq j \leq 3$ (Figure 12.) and let

$$\begin{aligned}
\mathbf{b}_0(t) &= \mathbf{c}_3(t), \mathbf{b}_1(t) = \mathbf{g}_c[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_3, \mathbf{n}_3, \mathbf{c}_{01}, \mathbf{c}_{21}](t), \\
\mathbf{b}_3(t) &= \mathbf{c}_1(t), \mathbf{b}_2(t) = \mathbf{g}_c[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_2, \mathbf{n}_2, \mathbf{c}_{02}, \mathbf{c}_{22}](t).
\end{aligned}$$

A quadrilateral pre-interpolant $\mathbf{S}_s(s, t)$ can be given in the following bicubic form:

$$\mathbf{S}_s(s, t) = \sum_{i=0}^3 \mathbf{b}_i(t) B_i^3(s).$$

Proof:

1. The boundary interpolation is verified as follows:

$$\begin{aligned}
\mathbf{S}_s(s, 0) &= \sum_{i=0}^3 \mathbf{b}_i(0) B_i^3(s) \\
&= \mathbf{p}_0 B_0^3(s) + \mathbf{c}_{01} B_1^3(s) + \mathbf{c}_{02} B_2^3(s) + \mathbf{p}_1 B_3^3(s) \\
&= \mathbf{c}_0(s), \\
\mathbf{S}_s(s, 1) &= \sum_{i=0}^3 \mathbf{b}_i(1) B_i^3(s) \\
&= \mathbf{p}_3 B_0^3(s) + \mathbf{c}_{21} B_1^3(s) + \mathbf{c}_{22} B_2^3(s) + \mathbf{p}_2 B_3^3(s) \\
&= \mathbf{c}_2(s), \\
\mathbf{S}_s(0, t) &= \mathbf{b}_0(t) = \mathbf{c}_3(t), \\
\mathbf{S}_s(1, t) &= \mathbf{b}_3(t) = \mathbf{c}_1(t).
\end{aligned}$$

2. The interpolation of the normal fields at $t = 0$ and $t = 1$ is verified as follows:

$$\begin{aligned}
\frac{\partial \mathbf{S}_s(0, t)}{\partial s} \cdot \mathbf{n}_3(t) &= 3(\sum_{i=0}^2 \Delta \mathbf{b}_i(t) B_i^2(0)) \cdot \mathbf{n}_3(t) \\
&= 3(\mathbf{b}_1(t) - \mathbf{b}_0(t)) \cdot \mathbf{n}_3(t) = 0, \\
\frac{\partial \mathbf{S}_s(0, t)}{\partial t} \cdot \mathbf{n}_3(t) &= (\sum_{i=0}^3 \mathbf{b}'_i(t) B_i^3(0)) \cdot \mathbf{n}_3(t) \\
&= \mathbf{b}'_0(t) \cdot \mathbf{n}_3(t) = \mathbf{c}'_3(t) \cdot \mathbf{n}_3(t) \\
&= 0.
\end{aligned}$$

It can be verified in a similar way that

$$\frac{\partial \mathbf{S}_s(1, t)}{\partial s} \cdot \mathbf{n}_1(t) = 0, \quad \frac{\partial \mathbf{S}_s(1, t)}{\partial t} \cdot \mathbf{n}_1(t) = 0.$$

■

A pre-interpolant $\mathbf{S}_t(s, t)$ can be given in a similar way such that it interpolates the boundary curves $\mathbf{c}_i(s)$ or $\mathbf{c}_i(t)$, $0 \leq i \leq 3$, and interpolates the normal fields $\mathbf{n}_1(t)$ and $\mathbf{n}_3(t)$ along $\mathbf{c}_1(t)$ and $\mathbf{c}_3(t)$. Let

$$\begin{aligned} \mathbf{b}_0(s) &= \mathbf{c}_0(s), \quad \mathbf{b}_1(s) = \mathbf{g}_c[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1, \mathbf{c}_{31}, \mathbf{c}_{11}](s), \\ \mathbf{b}_3(s) &= \mathbf{c}_2(s), \quad \mathbf{b}_2(s) = \mathbf{g}_c[\mathbf{p}_3, \mathbf{n}_3, \mathbf{p}_2, \mathbf{n}_2, \mathbf{c}_{32}, \mathbf{c}_{12}](s). \end{aligned}$$

and define

$$\mathbf{S}_t(s, t) = \sum_{i=0}^3 \mathbf{b}_i(s) B_i^3(t).$$

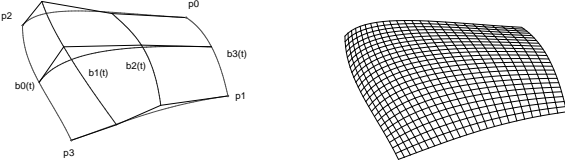


Figure 12: Quadrilateral pre-interpolant

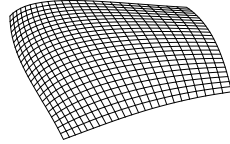


Figure 13: Quadrilateral interpolant

Theorem 4.4 *The quadrilateral interpolant (Figure 13)*

$$\mathbf{S}_q = \frac{s(1-s)\mathbf{S}_s + t(1-t)\mathbf{S}_t}{s(1-s) + t(1-t)}.$$

interpolates all the boundary curves $\mathbf{c}_i(s)$ and $\mathbf{c}_i(t)$, and interpolates the normal fields $\mathbf{n}_i(s)$ or $\mathbf{n}_i(t)$ along $\mathbf{c}_i(s)$ or $\mathbf{c}_i(t)$, $0 \leq i \leq 3$.

Proof: Both \mathbf{S}_s and \mathbf{S}_t interpolate the boundary curves, so does their convex combination \mathbf{S}_q . Furthermore, \mathbf{S}_s or \mathbf{S}_t interpolates the normal fields $\mathbf{n}_1(t), \mathbf{n}_3(t)$ along $\mathbf{c}_1(t), \mathbf{c}_3(t)$ or $\mathbf{n}_0(s), \mathbf{n}_2(s)$ along $\mathbf{c}_0(s), \mathbf{c}_2(s)$ respectively. When $s = 0$ we have

$$\begin{aligned} \frac{\partial \mathbf{S}_q}{\partial s} &= \frac{\partial \mathbf{S}_t}{\partial s} + \frac{(1-2s)}{t(1-t)}(\mathbf{S}_s - \mathbf{S}_t) = \frac{\partial \mathbf{S}_t}{\partial s}, \\ \frac{\partial \mathbf{S}_q}{\partial t} &= \frac{\partial \mathbf{S}_t}{\partial t}. \end{aligned}$$

Since \mathbf{S}_t interpolates the normal fields $\mathbf{n}_3(t)$ along $\mathbf{c}_3(t)$ so does \mathbf{S}_q . The other cases can be proved similarly and \mathbf{S}_q is a quadrilateral interpolant we need. \blacksquare

Theorem 4.5

$$\begin{aligned} \lim_{s \rightarrow 0^+, t \rightarrow 0^+} \mathbf{S}_q &= \mathbf{p}_0, \quad \lim_{s \rightarrow 1^-, t \rightarrow 0^+} \mathbf{S}_q = \mathbf{p}_1, \\ \lim_{s \rightarrow 1^-, t \rightarrow 1^-} \mathbf{S}_q &= \mathbf{p}_2, \quad \lim_{s \rightarrow 0^+, t \rightarrow 1^-} \mathbf{S}_q = \mathbf{p}_3. \end{aligned}$$

Proof: We will give a proof only for $\lim_{s \rightarrow 0^+, t \rightarrow 0^+} \mathbf{S}_q = \mathbf{p}_0$. Write

$$\omega(s, t) = \frac{s(1-s)}{s(1-s) + t(1-t)}.$$

It can be seen that $0 < \omega(s, t) < 1$ when $0 < s, t < 1$. Since $\mathbf{S}_s(0, 0) = \mathbf{p}_0$ and $\mathbf{S}_t(0, 0) = \mathbf{p}_0$, it is evident that

$$\lim_{s \rightarrow 0^+, t \rightarrow 0^+} \mathbf{S}_s = \mathbf{p}_0, \quad \lim_{s \rightarrow 0^+, t \rightarrow 0^+} \mathbf{S}_t = \mathbf{p}_0.$$

For any given $\epsilon > 0$, there exists a $\delta > 0$ such that $\|\mathbf{S}_s(s, t) - \mathbf{p}_0\| < \epsilon$ and $\|\mathbf{S}_t(s, t) - \mathbf{p}_0\| < \epsilon$ when $\sqrt{s^2 + t^2} < \delta$ and $0 < s, t < 1$. Then

$$\begin{aligned} \|\mathbf{S}_q - \mathbf{p}_0\| &= \|\omega(s, t)(\mathbf{S}_s - \mathbf{p}_0) + (1 - \omega(s, t))(\mathbf{S}_t - \mathbf{p}_0)\| \\ &\leq \omega(s, t)\|\mathbf{S}_s - \mathbf{p}_0\| + (1 - \omega(s, t))\|\mathbf{S}_t - \mathbf{p}_0\| \\ &< \omega(s, t)\epsilon + (1 - \omega(s, t))\epsilon \\ &= \epsilon. \end{aligned}$$

The other cases can be proved similarly. \blacksquare

Figure 14 shows three quadrilateral interpolants for the sampling points from a torus with the weights ω in $\mathbf{g}_v(t)$ being 0.15, 0.3 and 0.45 respectively.

4.3 Triangular Interpolant

A triangular interpolant will be given in a similar way as the construction of a quadrilateral interpolant. Suppose the boundary curves $\mathbf{c}_i(t)$ with

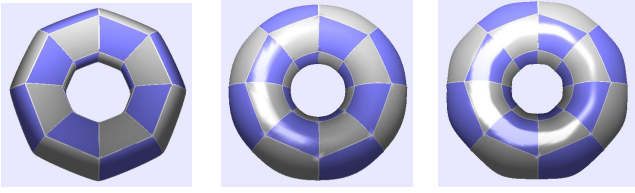


Figure 14: Quadrilateral interpolant

the normal fields $\mathbf{n}_i(t)$, $i = 0, 1, 2$, are given over a triangular domain with vertices \mathbf{p}_i , $i = 0, 1, 2$. The barycentric coordinate of a point is set with respect to the triangle $\mathbf{p}_0\mathbf{p}_1\mathbf{p}_2$ (Figures 15 and 16). Since variable t in $\mathbf{c}_i(t)$ and $\mathbf{n}_i(t)$ may be changed into $(1-t)$, we may assume, without loss of generality, that t increases from $(1, 0, 0)$ to $(0, 1, 0)$, from $(0, 1, 0)$ to $(0, 0, 1)$, and from $(0, 0, 1)$ to $(1, 0, 0)$. Hence the variable of \mathbf{c}_0 and \mathbf{n}_0 is w , \mathbf{c}_1 and \mathbf{n}_1 is u , \mathbf{c}_2 and \mathbf{n}_2 is v , with the following expressions:

$$\begin{aligned} \mathbf{c}_0(w) &= \mathbf{g}_v[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_2, \mathbf{n}_2](w), \\ \mathbf{c}_1(u) &= \mathbf{g}_v[\mathbf{p}_2, \mathbf{n}_2, \mathbf{p}_0, \mathbf{n}_0](u), \\ \mathbf{c}_2(v) &= \mathbf{g}_v[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](v), \\ \mathbf{n}_0(w) &= \mathbf{g}_n[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_2, \mathbf{n}_2](w), \\ \mathbf{n}_1(u) &= \mathbf{g}_n[\mathbf{p}_2, \mathbf{n}_2, \mathbf{p}_0, \mathbf{n}_0](u), \\ \mathbf{n}_2(v) &= \mathbf{g}_n[\mathbf{p}_0, \mathbf{n}_0, \mathbf{p}_1, \mathbf{n}_1](v). \end{aligned}$$

Similar to the construction of a quadrilateral interpolant, we will create a pre-interpolant corresponding to each vertex such that it interpolates the three boundary curves and the normal fields along two of the three ones. In fact, the triangular pre-interpolant can be regarded as a degenerated case for the quadrilateral pre-interpolant. We formulate the problem of the triangular pre-interpolant as follows:

Triangular pre-interpolant: Let $\mathbf{c}_i(t), \mathbf{n}_i(t)$, $i = 0, 1, 2$, be stated as above.

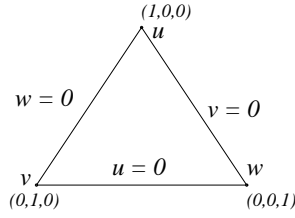


Figure 15: Triangular domain

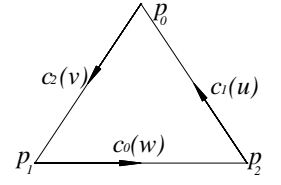


Figure 16: Triangular boundary curves

Find a triangular interpolant $\mathbf{S}_u(u, v, w)$ such that

1. $\mathbf{S}_u(u, v, w)$ interpolates the boundary curves:
$$\begin{aligned} \mathbf{S}_u(0, 1-w, w) &= \mathbf{c}_0(w), \\ \mathbf{S}_u(u, 0, 1-u) &= \mathbf{c}_1(u), \\ \mathbf{S}_u(1-v, v, 0) &= \mathbf{c}_2(v). \end{aligned}$$
2. The tangent field determined by the span of $(\frac{\partial \mathbf{S}_u}{\partial u}, \frac{\partial \mathbf{S}_u}{\partial v})$ are orthogonal to both $\mathbf{n}_1(u)$ and $\mathbf{n}_2(v)$ along the curves $\mathbf{c}_1(u)$ and $\mathbf{c}_2(v)$ respectively:

$$\frac{\partial \mathbf{S}_u(u, 0, 1-u)}{\partial \mu} \cdot \mathbf{n}_1(u) = 0, \quad \frac{\partial \mathbf{S}_u(1-v, v, 0)}{\partial \mu} \cdot \mathbf{n}_2(v) = 0,$$

where $\mu = u, v$.

Let

$$\begin{aligned} \tilde{\mathbf{c}}_2(u) &= \mathbf{g}_v[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_0, \mathbf{n}_0](u), \\ \tilde{\mathbf{n}}_2(u) &= \mathbf{g}_n[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_0, \mathbf{n}_0](u). \end{aligned}$$

Noting that $u = 1-v$ and from lemmas 3.3 and 3.7, it is known that $\tilde{\mathbf{c}}_2(u) = \mathbf{c}_2(v)$ and $\tilde{\mathbf{n}}_2(u) = \mathbf{n}_2(v)$.

In a similar way, we can also fill the interior of $\mathbf{S}_u(u, v, w)$ with a curve passing from every points on $\tilde{\mathbf{c}}_2(u)$ to points on $\mathbf{c}_1(u)$ (Figure 17). Noting that w is changing from 0 to $1-u$, we get the following lemma.

Theorem 4.6 *A triangular pre-interpolant $\mathbf{S}_u(u, v, w)$ can be set as follows:*

$$\mathbf{S}_u(u, v, w) = \mathbf{g}_v[\tilde{\mathbf{c}}_2(u), \tilde{\mathbf{n}}_2(u), \mathbf{c}_1(u), \mathbf{n}_1(u)]\left(\frac{w}{1-u}\right).$$

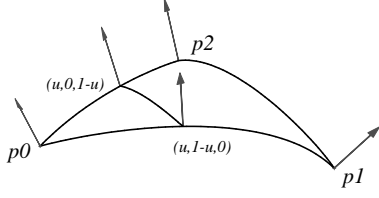


Figure 17: Pre-interpolant construction

Proof:

1. The boundary interpolation is verified as follows:

$$\begin{aligned}
\mathbf{S}_u(0, 1-w, w) &= \mathbf{g}_v[\tilde{\mathbf{c}}_2(0), \tilde{\mathbf{n}}_2(0), \mathbf{c}_1(0), \mathbf{n}_1(0)](w) \\
&= \mathbf{g}_v[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_2, \mathbf{n}_2](w) = \mathbf{c}_0(w), \\
\mathbf{S}_u(u, 0, 1-u) &= \mathbf{g}_v[\tilde{\mathbf{c}}_2(u), \tilde{\mathbf{n}}_2(u), \mathbf{c}_1(u), \mathbf{n}_1(u)](1) \\
&= \mathbf{c}_1(u), \\
\mathbf{S}_u(1-v, v, 0) &= \mathbf{g}_v[\tilde{\mathbf{c}}_2(u), \tilde{\mathbf{n}}_2(u), \mathbf{c}_1(u), \mathbf{n}_1(u)](0) \\
&= \tilde{\mathbf{c}}_2(u) = \mathbf{c}_2(v).
\end{aligned}$$

2. The interpolation of the normal fields at $v = 0$, $w = 0$ is verified as follows: From

$$\begin{aligned}
&\frac{\partial \mathbf{S}_u}{\partial v}(u, v, w) \\
&= \frac{-1}{1-u} \mathbf{g}'_v[\tilde{\mathbf{c}}_2(u), \tilde{\mathbf{n}}_2(u), \mathbf{c}_1(u), \mathbf{n}_1(u)](\frac{w}{1-u}), \\
&\frac{\partial \mathbf{S}_u}{\partial u}(u, v, w) = \mathbf{g}_v[\tilde{\mathbf{c}}'_2(u), \tilde{\mathbf{n}}'_2(u), \mathbf{c}'_1(u), \mathbf{n}'_1(u)](\frac{w}{1-u}) \\
&+ \frac{v}{(1-u)} \frac{\partial \mathbf{S}_u}{\partial v}(u, v, w),
\end{aligned}$$

when $v = 0$, we get

$$\begin{aligned}
&\frac{\partial \mathbf{S}_u(u, 0, 1-u)}{\partial v} \cdot \mathbf{n}_1(u) \\
&= \frac{-1}{1-u} \mathbf{g}'_v[\tilde{\mathbf{c}}_2(u), \tilde{\mathbf{n}}_2(u), \mathbf{c}_1(u), \mathbf{n}_1(u)](1) \cdot \mathbf{n}_1(u) \\
&= 0, \\
&\frac{\partial \mathbf{S}_u(u, 0, 1-u)}{\partial u} \cdot \mathbf{n}_1(u) \\
&= (\mathbf{g}_v[\tilde{\mathbf{c}}'_2(u), \tilde{\mathbf{n}}'_2(u), \mathbf{c}'_1(u), \mathbf{n}'_1(u)](1) + \\
&\quad \frac{\partial \mathbf{S}_u(u, 0, 1-u)}{\partial v}) \cdot \mathbf{n}_1(u) \\
&= \mathbf{c}'_1(u) \cdot \mathbf{n}_1(u) = 0.
\end{aligned}$$

when $w = 0$, we get

$$\begin{aligned}
&\frac{\partial \mathbf{S}_u(1-v, v, 0)}{\partial v} \cdot \mathbf{n}_2(v) \\
&= \frac{-1}{1-u} \mathbf{g}'_v[\tilde{\mathbf{c}}_2(u), \tilde{\mathbf{n}}_2(u), \mathbf{c}_1(u), \mathbf{n}_1(u)](0) \cdot \mathbf{n}_2(v) \\
&= \frac{-1}{1-u} \mathbf{g}'_v[\tilde{\mathbf{c}}_2(u), \tilde{\mathbf{n}}_2(u), \mathbf{c}_1(u), \mathbf{n}_1(u)](0) \cdot \tilde{\mathbf{n}}_2(u) \\
&= 0, \\
&\frac{\partial \mathbf{S}_u(1-v, v, 0)}{\partial u} \cdot \mathbf{n}_2(v) \\
&= (\mathbf{g}_v[\tilde{\mathbf{c}}'_2(u), \tilde{\mathbf{n}}'_2(u), \mathbf{c}'_1(u), \mathbf{n}'_1(u)](0) + \\
&\quad \frac{\partial \mathbf{S}_u(1-v, v, 0)}{\partial v}) \cdot \mathbf{n}_2(v) \\
&= \mathbf{g}_v[\tilde{\mathbf{c}}'_2(u), \tilde{\mathbf{n}}'_2(u), \mathbf{c}'_1(u), \mathbf{n}'_1(u)](0) \cdot \tilde{\mathbf{n}}_2(u) \\
&= \tilde{\mathbf{c}}'_2(u) \cdot \tilde{\mathbf{n}}_2(u) \\
&= 0.
\end{aligned}$$

We can also give the following triangular pre-interpolant from the control curves $\mathbf{g}_c(u)$.

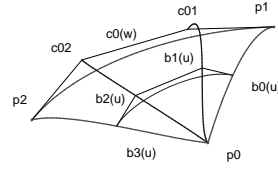


Figure 18: Pre-interpolant

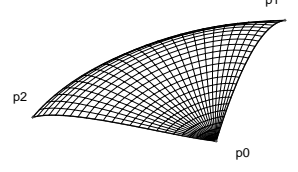


Figure 19: Triangular interpolant

Theorem 4.7 Suppose $\mathbf{c}_{00}, \mathbf{c}_{01}, \mathbf{c}_{02}$ and \mathbf{c}_{03} are the control points of $\mathbf{c}_0(w)$ (Figure 18.) and let

$$\begin{aligned}
\mathbf{b}_0(u) &= \tilde{\mathbf{c}}_2(u) = \mathbf{g}_v[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_0, \mathbf{n}_0](u), \\
\mathbf{b}_1(u) &= \mathbf{g}_c[\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_0, \mathbf{n}_0, \mathbf{c}_{01}, \mathbf{p}_0](u), \\
\mathbf{b}_3(u) &= \mathbf{c}_1(u) = \mathbf{g}_c[\mathbf{p}_2, \mathbf{n}_2, \mathbf{p}_0, \mathbf{n}_0](u), \\
\mathbf{b}_2(u) &= \mathbf{g}_c[\mathbf{p}_2, \mathbf{n}_2, \mathbf{p}_0, \mathbf{n}_0, \mathbf{c}_{02}, \mathbf{p}_0](u).
\end{aligned}$$

A triangular pre-interpolant \mathbf{S}_u can be defined as follows

$$\mathbf{S}_u(u, v, w) = \sum_{i=0}^3 \mathbf{b}_i(u) B_i^3(\frac{w}{1-u}).$$

Proof:

1. The proof of the boundary interpolation can be done similarly to that of Theorem 4.6.
2. The interpolation of the normal fields at $v = 0$, $w = 0$ is verified as follows: From

$$\frac{\partial \mathbf{S}_u}{\partial v}(u, v, w) = \frac{-3}{1-u} \sum_{i=0}^2 \Delta \mathbf{b}_i(u) B_i^2(\frac{w}{1-u}),$$

$$\frac{\partial \mathbf{S}_u}{\partial u}(u, v, w) = \sum_{i=0}^3 \mathbf{b}'_i(u) B_i^3(\frac{w}{1-u}) + \frac{v}{1-u} \frac{\partial \mathbf{S}_u}{\partial v}(u, v, w),$$

when $v = 0$, it can be verified that

$$\begin{aligned}
& \frac{\partial \mathbf{S}_{\mathbf{u}}(u, 0, 1-u)}{\partial v} \cdot \mathbf{n}_1(u) \\
&= \frac{-3}{1-u} (\mathbf{b}_3(u) - \mathbf{b}_2(u)) \cdot \mathbf{n}_1(u) \\
&= \frac{3}{1-u} ((\mathbf{g}_c - \mathbf{g}_v) \cdot \mathbf{g}_n) [\mathbf{p}_2, \mathbf{n}_2, \mathbf{p}_0, \mathbf{n}_0](u) \\
&= 0, \\
& \frac{\partial \mathbf{S}_{\mathbf{u}}(u, 0, 1-u)}{\partial u} \cdot \mathbf{n}_1(u) \\
&= \mathbf{b}'_3(u) \cdot \mathbf{n}_1(u) = \mathbf{c}'_1(u) \cdot \mathbf{n}_1(u) = 0;
\end{aligned}$$

when $w = 0$, it can be verified that

$$\begin{aligned}
& \frac{\partial \mathbf{S}_{\mathbf{u}}(1-v, v, 0)}{\partial v} \cdot \mathbf{n}_2(v) \\
&= \frac{-3}{1-u} (\tilde{\mathbf{b}}_1(u) - \mathbf{b}_0(u)) \cdot \tilde{\mathbf{n}}_2(u) \\
&= \frac{-3}{1-u} ((\mathbf{g}_c - \mathbf{g}_v) \cdot \mathbf{g}_n) [\mathbf{p}_1, \mathbf{n}_1, \mathbf{p}_0, \mathbf{n}_0](u) \\
&= 0 \\
& \frac{\partial \mathbf{S}_{\mathbf{u}}(1-v, v, 0)}{\partial u} \cdot \mathbf{n}_2(v) \\
&= \mathbf{b}'_0(u) \cdot \mathbf{n}_2(v) = \mathbf{c}'_2(v) \cdot \mathbf{n}_2(v) = 0.
\end{aligned}$$

□

Theorem 4.8 $\lim_{u \rightarrow 1^-, v \rightarrow 0^+, w \rightarrow 0^+} \mathbf{S}_{\mathbf{u}}(u, v, w) = \mathbf{p}_0$.

Proof: We first have that when $0 < u, v, w < 1$,

$$0 < B_i^3\left(\frac{w}{1-u}\right) < 1, \quad \sum_{i=1}^3 B_i^3\left(\frac{w}{1-u}\right) = 1.$$

Since $\mathbf{b}_i(1) = \mathbf{p}_0$, it is evident that $\lim_{u \rightarrow 1^-} \mathbf{b}_i(u) = \mathbf{p}_0$. For any given $\epsilon > 0$, there exists a $\delta > 0$ such that $\|\mathbf{b}_i(u) - \mathbf{p}_0\| < \epsilon$ when $1 - \delta < u < 1$ for $0 \leq i \leq 3$. Then

$$\begin{aligned}
& \|\mathbf{S}_{\mathbf{u}}(u, v, w) - \mathbf{p}_0\| \\
&= \left\| \sum_{i=0}^3 \mathbf{b}_i(u) B_i^3\left(\frac{w}{1-u}\right) - \mathbf{p}_0 \right\| \\
&= \left\| \sum_{i=0}^3 (\mathbf{b}_i(u) - \mathbf{p}_0) B_i^3\left(\frac{w}{1-u}\right) \right\| \\
&\leq \sum_{i=0}^3 (\|\mathbf{b}_i(u) - \mathbf{p}_0\| B_i^3\left(\frac{w}{1-u}\right)) \\
&< \epsilon \sum_{i=0}^3 B_i^3\left(\frac{w}{1-u}\right) \\
&= \epsilon.
\end{aligned}$$

□

Constructing $\mathbf{S}_{\mathbf{v}}(u, v, w)$ and $\mathbf{S}_{\mathbf{w}}(u, v, w)$ in a similar way as Theorem 4.7, we can now give a triangular interpolant.

Theorem 4.9 *Figure 19.* A triangular interpolant $\mathbf{S}_{\mathbf{I}}$ can be written as their convex combination as follows:

$$\mathbf{S}_{\mathbf{I}} = u\mathbf{S}_{\mathbf{u}} + v\mathbf{S}_{\mathbf{v}} + w\mathbf{S}_{\mathbf{w}}$$

Proof: Since $\mathbf{S}_{\mathbf{I}}$ is a convex combination of $\mathbf{S}_{\mathbf{u}}$, $\mathbf{S}_{\mathbf{v}}$ and $\mathbf{S}_{\mathbf{w}}$ and all of them interpolate the three boundary curves, so does $\mathbf{S}_{\mathbf{I}}$.

When $u = 0$, since both $\mathbf{S}_{\mathbf{v}}$ and $\mathbf{S}_{\mathbf{w}}$ interpolate the normal field $\mathbf{n}_0(w)$ along $\mathbf{c}_0(w)$ and

$$\begin{aligned}
\frac{\partial \mathbf{S}_{\mathbf{I}}(0, v, w)}{\partial u} &= \mathbf{S}_{\mathbf{u}} + u \frac{\partial \mathbf{S}_{\mathbf{u}}}{\partial u} + v \frac{\partial \mathbf{S}_{\mathbf{v}}}{\partial u} - \mathbf{S}_{\mathbf{w}} + w \frac{\partial \mathbf{S}_{\mathbf{w}}}{\partial u} \\
&= v \frac{\partial \mathbf{S}_{\mathbf{v}}}{\partial u} + w \frac{\partial \mathbf{S}_{\mathbf{w}}}{\partial u}, \\
\frac{\partial \mathbf{S}_{\mathbf{I}}(0, v, w)}{\partial v} &= \mathbf{S}_{\mathbf{v}} + u \frac{\partial \mathbf{S}_{\mathbf{u}}}{\partial v} + v \frac{\partial \mathbf{S}_{\mathbf{v}}}{\partial v} - \mathbf{S}_{\mathbf{w}} + w \frac{\partial \mathbf{S}_{\mathbf{w}}}{\partial v} \\
&= v \frac{\partial \mathbf{S}_{\mathbf{v}}}{\partial v} + w \frac{\partial \mathbf{S}_{\mathbf{w}}}{\partial v},
\end{aligned}$$

it can be seen that $\mathbf{S}_{\mathbf{I}}$ interpolates the normal field $\mathbf{n}_0(w)$ along $\mathbf{c}_0(w)$.

The case of $w = 0$ and $v = 0$ can be proved in a similar way.

□

5 DISM: A Dynamic Interpolation Solid Modelling Software Package

As compared with previous interpolation algorithms, our algorithm has the following characters.

1. Fast generation of interpolation surfaces.

Since a complete symbolic and parametric representation is given for the interpolation surface, each point on the surface can then be

generated in a fast way. In fact, any surface point can be generated with the de Casteljau algorithm of Bézier curves.

2. A general modelling method. In the method, a surface can be generated dynamically from a moving Bézier curve under some additional geometric constraints. This general method can be used to construct an interpolation surface, to convert a polygonal model into a parametric surface, to deform a 3D object, to form blends between objects and to fill N-sided hole. Some details of these applications will be illustrated in our later works.
3. Fine shape: Both the boundary curves and the control curves are constructed with minimum bending energy, so an interpolation surface with fine shape can be given. A fine shape will be definitely ensured if we take the pre-interpolant from Theorem 4.2 or 4.6.
4. Applicable for arbitrary scattered data over a triangle or quadrilateral decomposition.
5. Easy shape control: Since a free parameter ω with obvious geometrical meaning is left in $\mathbf{g}_v(t)$, we can control the shape of the boundary curves and therefore the final interpolation surface easily.

Based on the algorithm, a 3D dynamic interpolation solid modelling software package DIMS is implemented (Figure 20), which can be used to generate and manipulate 3D objects in an interactive way. It has the following features.

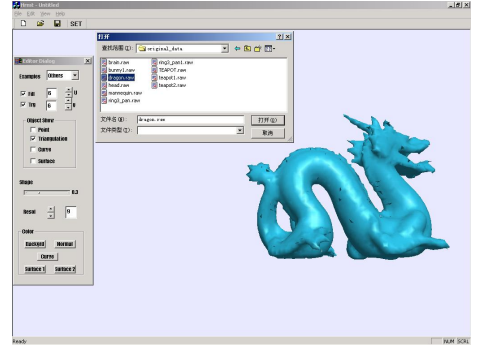


Figure 20: Software interface

1. 3D points and vectors at points could be generated in an interactive way from dialog inputs or mouse inputs.
2. After a triangle or quadrilateral decomposition is given by the user, the interpolation surface can be generated automatically. We plan to add the triangle or quadrilateral decomposition algorithms later.
3. When the user changes the points or the vectors, the interpolation surfaces will change locally and continuously. With this feature we can change one object to another object by changing certain points, which provides a convenient tools for object deformation.

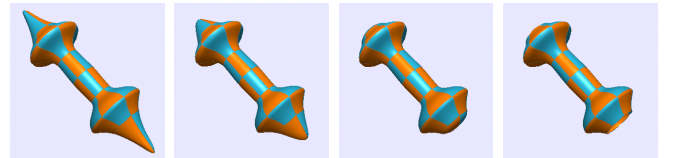


Figure 21: One point changed

Figure 21 shows different shapes of one object with two controlling points being changed. Figure 22 shows the deformation process from a sphere to a dumbbell by dragging several

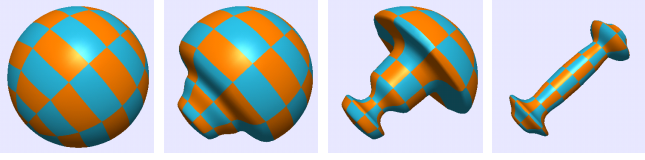


Figure 22: From sphere to dumbbell

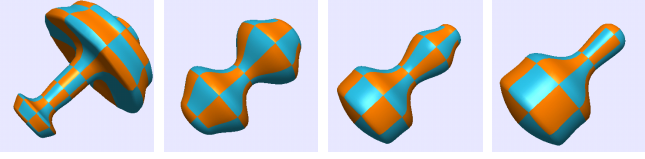


Figure 23: Objects from a sphere

points and normal vectors on them. Figure 23 shows several shapes obtained from a sphere by dragging the controlling points on the sphere.

4. When the weight ω in \mathbf{g}_v is changed, the resulted surface will change correspondingly. Figure 24 shows four different results with $\omega = 0.15, 0.3, 0.4$ and 0.5 respectively for a teapot model. In fact, we can change the weight ω for one boundary curve locally and keep the weights of the others unchanged, which can then be used to generate a bump for an object.

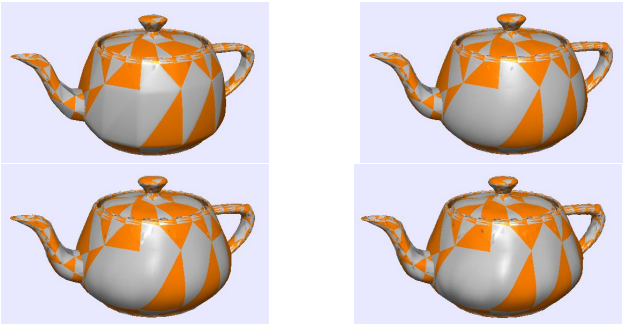


Figure 24: Different weights

time, we may moving control points continuously along certain loci and generate the interpolation surfaces for each position of the moving points. In this way, we may generate the effect of animation.

6. With this algorithm, we also provide a method to compress objects. Specifically, when we want to save a free-form object, all we need to save is the scattered data with its corresponding triangulation. The object can then be generated from the interpolation algorithm in a fast way.

Two more complicated examples generated by our programm are shown below. Figure 25 shows the triangulation, the boundary curves, and the final result for the sampling points from a teapot model with 695 vertices. Figures 26 shows the result of the sampling points from a human head model with 1528 vertices.

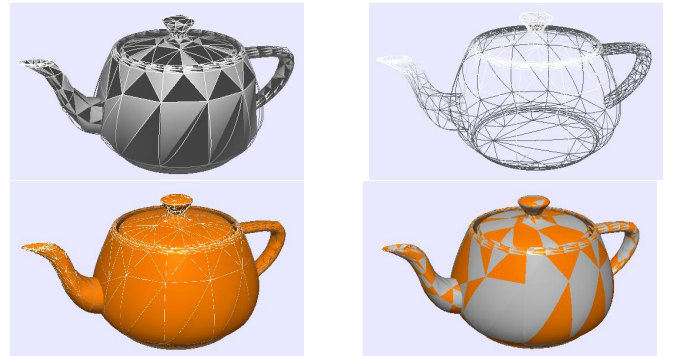


Figure 25: Teapot

5. Since the interpolants can be generated in real

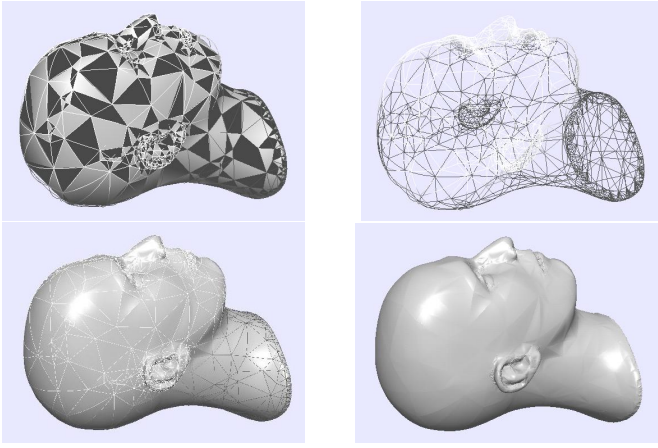


Figure 26: Human head

6 Conclusion

The purpose of this paper is to introduce a tool to generate and to manipulate solid models in an interactive way. In order to achieve real time deformation, we introduce a triangular interpolant and a quadrilateral interpolant with a general method to generate surface with sweeping curves. These interpolants are in complete symbolic parametric forms, leading to a fast computation. Another idea of the dynamic tool: free dragging of points and other geometric objects, comes from dynamic geometry [26, 27]. By dragging the control points on a solid object, we may deform one object to another in a continuous way.

The method to construct the interpolants introduced in this paper could have other applications such as to convert a polygonal model into a parametric surface, to deform a 3D object, to form blends between objects and to fill N-sided hole. Further study on this problem should be done. For instance, we may ask how to obtain in-

terpolants with lower degree. In our case, since the quadrilateral interpolant involves less patches and a degenerated bicubic surface will not appear in its construction, a quadrilateral decomposition is preferred to a triangle one.

References

- [1] I. Amidror, Scattered Data Interpolation Methods for Electronic Imaging Systems: a survey. *Journal of Electronic Imaging*, 11(2), 157-176, 2002.
- [2] H. Biermann, A. Levin and D. Zorin, Piecewise Smooth Subdivision Surface with Normal Control. *Proceedings of ACM SIGGRAPH 2000*, 113-120, 2000.
- [3] G. Li and H. Li, Vertex and Normal Interpolation of Surfaces Based on Control Net Generated by Mixed Subdivisions, *Journal of Computer-Aided Design & Computer Graphics* 13, 6, 537-544, 2001.
- [4] A. Nasri, Surface Interpolation on Irregular Networks with Normal Conditions. *Computer Aided Geometric Design*, 8, 89-96, 1991.
- [5] D. Zorin, P. Schröder and W. Sweldens, Interpolating Subdivision for Meshes with Arbitrary Topology. *Proceedings of ACM SIGGRAPH 1996*, 189-192, 1996.
- [6] W. Dahmen, Smooth Piecewise Quadric Surfaces. *Mathematical Methods in Com-*

- puter Aided Geometric Design*, T. Lyche and L. Schumaker (editors), Academic Press, Boston, 181-193, 1989.
- [7] C. Bajaj, J. Chen and G. Xu, Modelling with Cubic A-Patches. *ACM Trans. on Graphics* 14, 2, 103-133, 1995.
- [8] W. Dahmen and T. Thamm-Schaar, Cubicoids: Modelling and Visualization. *Computer Aided Geometric Design* 10, 89-108, 1993.
- [9] B. Guo, Quadric and Cubic Bitetrahedral Patches. *The Visual Computer*, 11, 253-262, 1995.
- [10] W. Wang and K. Qin, On the existence and computation of rational quartic spherical curves for Hermite Interpolation. *The Visual Computer*, vol. 16, no. 3/4, (2000), 187-196.
- [11] G. Xu, H. Huang and C. Bajaj, C^1 Modelling with A-patches from Rational Trivariate Functions. *Computer Aided Geometric Design* 18, 221-243, 2001.
- [12] X. Gao and M. Li, Construct Piecewise Hermite Interpolation Surface with Blending Methods. *Proceedings of Geometric Modeling and Processing: Theory and Applications*, H. Suzuki and R. Martin (eds), IEEE Computer Society, 53-59, 2002.
- [13] G. Turk and J. O' Brien, Modelling with Implicit Surfaces that Interpolate. *ACM Transactions on Graphics*, 21(4), 855-873, 2002.
- [14] H. Qin and Ying He, Surface Reconstruction with Triangular B-splines, *Proceedings of Geometric Modeling and Processing*, 13-15 April, 2004, Beijing.
- [15] P. Alfeld, Scattered Data Interpolation in Three or More Variables. *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche, L. Schumaker (editors), Academic Press, 1-35, 1989.
- [16] M. Lounsberry, S. Mann and T. DeRose, Parametric Surface Interpolation. *IEEE Comp. Graphics Appl.*, 12, 45-52, 1992.
- [17] S. Mann, C. Loop, M. Lounsberry, D. Meyers, J. Painter, T. DeRose and K. Sloan, A Survey of Parametric Scattered Data Fitting Using Triangular Interpolants. *Curve and Surface Design*, H. Hagen (editor), Springer-Verlag, 1992.
- [18] G. Herron, Smooth Closed Surfaces with Discrete Triangular Interpolants. *Computer Aided Geometric Design*, 2, 297-306, 1985.
- [19] G. Nielson, A Transfinite, Visually Continuous, Triangular Interpolant. *Geometric Modelling: Algorithms and New Trends*, G. Farin (editor), 235-246, SIAM, Philadelphia, 1987.
- [20] C. Loop, Smooth Spline Surfaces over Irregular Meshes. *Proceedings of ACM SIGGRAPH 1994*, 303-309, 1994.
- [21] H. Pottmann, S. Leopoldseder and M. Hofer, Approximation with Active B-spline Curve

and Surfaces, *Proceedings of Pacific Graphics 2002*, 9-11 October 2002, Beijing.

- [22] G. Fasshauer and L. Schumaker, Minimal Energy Surfaces Using Parametric Splines, *Computer Aided Geometric Design*, 13, 45-79, 1996.
- [23] C. Bajaj and G. Xu, Spline Approximation of Real Algebraic Surfaces. *Journal of Symbolic Computation*, 23, 315-333, 1997.
- [24] A. Wiltche, C^1 - and C^2 -continuous Spline-interpolation of a Regular Triangular Net of Points. *Computers & Graphics*, Artical in Press.
- [25] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 4th edition, Academic Press, 1997.
- [26] J.M. Laborde, *GABRI Geometry II*, Texas Instruments, Dallas, 1994.
- [27] X. Gao, J.Z. Zhang and S.C. Chou, *Geometry Expert* (in Chinese), Nine Chapter Pub., Taipai, Taiwan, 1998.