ELSEVIER

Computer-Aided Design 33 (2001) 975–987

COMPUTER-AIDED
DESIGN

www.elsevier.com/locate/cad

Streamline modeling with subdivision surfaces on the Gaussian sphere

K.T. Miura[a,*], L. Wang[a], F. Cheng[b]

[a]*Department of Mechanical Engineering, Shizuoka University, 3-5-1, Johoku, Hamamatsu, 432-8561 Japan*
[b]*Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA*

Received 1 September 1999; revised 16 August 2000; accepted 24 August 2000

Abstract

Curvature and variation of curvature are the essential factors in determining the fairness of a surface. Unfortunately, most of the traditional surface representation schemes do not provide users with direct manipulation techniques of these quantities. Streamline modeling, a recently proposed free-form surface design methodology, is aimed at overcoming this shortcoming by allowing a user to control tangent vectors (and, consequently, curvature and variation of curvature) of the surface to be designed directly. A free-form surface is regarded as a set of streamlines: iso-parametric lines defined by blending directions of tangent vectors instead of blending positions of control points. This new surface design methodology can generate high quality smooth surfaces but requires much processing power for tangent vector blending. In this paper, we present subdivision based blending techniques of tangent vectors. These techniques can be used to develop subdivision techniques for curves and surfaces on the Gaussian sphere, such as Doo–Sabin, Catmull–Clark, and Kobbelt subdivisions. We also present new streamline modeling techniques based on the new tangent vector blending techniques. The new techniques reduce the processing time for the integration process required in streamline modeling. A prototype system based on the new techniques shows that free-form surface design using the streamline modeling methodology can achieve real-time performance. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Surface modeling; Streamline modeling; Subdivision surface; Gaussian sphere; Tangent vector blending

1. Introduction

Bézier and non-uniform rational B-spline (NURBS) surfaces are the most popular surface representations used in graphics and various CAD/CAM applications due to their nice characteristics such as *convex hull/local control properties*, small *energy*, and *numerical stability* [6]. The main idea of these surface construction schemes is to blend positions of control points to define surface shapes. However, they have potential instability on curvature distribution and variation of curvature: these quantities may change dramatically due to a simple reposition of the control points, and they may have complicated curvature or variation of curvature profiles because they are not parametrized according to arc length and the curvature formulation has a square root term in its denominator even though they are polynomial surfaces.

Streamline modeling is a free-form surface design methodology proposed by Miura et al. [10] to overcome the instability problem of traditional shape representation schemes. The new method is intended to provide the users with direct control of the curvature and variation of curvature of a surface, the essential factors in determining the fairness of a surface.

In streamline modeling, directions of the tangent vectors not positions of the control points are blended to define arc length parametrized fair curves called *streamlines* and a surface is constructed as a set of streamlines. The streamline modeling concept itself seems to be promising in that it can generate high-quality smooth surfaces stably, but the system described in Ref. [10] failed to be efficiently interactive because the blending technique of tangent vectors needs much processing power for the integration process to obtain final shapes. More efficient and powerful blending techniques of tangent vectors are needed.

Subdivision surfaces are becoming more popular in modeling and graphics communities because of their scalability, numerical stability and code simplicity, as well as their ability to represent arbitrary topology faces [3]. By developing subdivision based blending techniques of tangent vectors, we are not only able to develop subdivision techniques for curves and surfaces on the Gaussian sphere, a unit sphere in three dimensional space whose points are unit tangent vectors, but also reduce the processing time for the integration process required in streamline modeling. These

[*] Corresponding author. Tel: +81-53-478-1074; fax: +81-53-478-1074.
E-mail address: ktmiura@eng.shizuoka.ac.jp (K.T. Miura).

PII: S0010-4485(00)00134-2

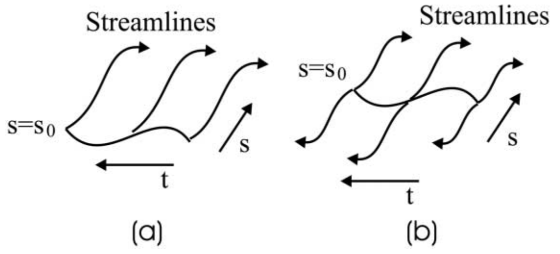


Fig. 1. Streamlines as iso-parametric curves.

techniques will be presented in this paper. A prototype system based on these techniques shows that free-form surface design using the streamline modeling approach can achieve real-time performance.

The remainder of this paper is organized as follows. The basic concept of streamline modeling is reviewed in Section 2. Properties of curves and surfaces on the Gaussian sphere and new subdivision techniques for them are discussed and presented in Sections 3 and 4. Implementation issues of *streamline modeling* with subdivision surfaces on the Gaussian sphere are addressed in Section 5. Several surface examples designed with our prototype system are shown in Section 6. Concluding remarks and comments on future research directions are given in Section 7.

2. Basic concept of streamline modeling

The basic idea of streamline modeling is to construct a parametric surface as a set of iso-parametric curves called *streamlines*, as shown in Fig. 1, which are defined by blending tangent vectors instead of positions. Each streamline has a constant parameter value t_0 and starts from a point at $\boldsymbol{S}(s_0, t_0)$ in the positive or negative s direction. As shown in Fig. 1(a), s_0 may be equal to the minimum value $s_{\min}$, the maximum value $s_{\max}$ of s, or a value in between, as shown in Fig. 1(b). Fig. 2 shows examples of surfaces designed based on the streamline modeling concept [13]. Each streamline is defined by blending nine control tangent vectors displayed as yellow arrows. The surfaces are deformed by changing the directions of their tangent vectors.

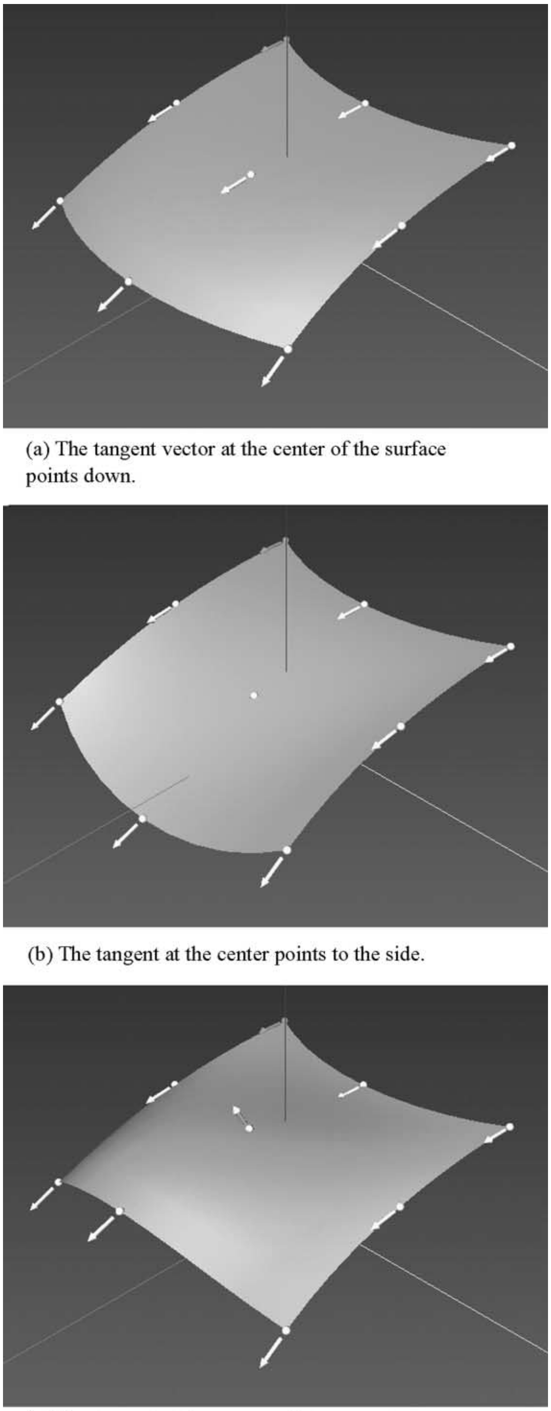
Given a surface $\boldsymbol{S}(s, t)$, if the first partial derivative of the surface with respect to the arc length parameter s, $\partial\boldsymbol{S}(s,t)/\partial s = \boldsymbol{s}(s,t)$ is known, then the surface can be defined by

$$\boldsymbol{S}(s,t) = \boldsymbol{S}(s_0,t) + \int_{s_0}^{s} \boldsymbol{s}(u,t)\mathrm{d}u \tag{1}$$

where s_0 is a fixed point in s parameter space and the curve $\boldsymbol{S}(s_0, t_0)$ is called an *initial curve*. If $s < s_0$, the above equation should be interpreted as

$$\boldsymbol{S}(s,t) = \boldsymbol{S}(s_0,t) - \int_{s}^{s_0} \boldsymbol{s}(u,t)\mathrm{d}u. \tag{2}$$

Eq. (1) shows that if an initial curve in parameter t with s fixed, $s = s_0$, and the first partial derivative $\boldsymbol{s}(s,t)$ with respect to s over the entire parameter space are given, then one can construct a surface by integrating the first derivative as does (1). Since parameter s is the arc length, the length of each streamline depends on the maximum


(a) The tangent vector at the center of the surface points down.

(b) The tangent at the center points to the side.

(c) The tangent at the center points upward.

Fig. 2. Surface examples of streamline modeling (QI surfaces). (a) The tangent vector at the center of the surface points down; (b) the tangent at the center points to the side; (c) the tangent at the center points upward. Red and green lines are x and y axes, respectively.

value of s only. The maximum value of s can be made t-dependent.

Eq. (1) shows that the first derivative $\boldsymbol{S}(s, t)$ with respect to t, $\partial \boldsymbol{S}(s,t)/\partial t = \boldsymbol{s}(s,t)$, is given by

$$\boldsymbol{t}(s,t) = \frac{\partial \boldsymbol{S}(s,t)}{\partial t} = \boldsymbol{t}_0 + \int_{s_0}^{s} \frac{\partial \boldsymbol{s}(u,t)}{\partial t} \mathrm{d}u, \tag{3}$$

where $\boldsymbol{t_0} = \partial \boldsymbol{S}(s_0, t)/\partial t$. Note that while parameter s is defined by arc length, t is not, because the norm of $\boldsymbol{t}(s, t)$ is not guaranteed to be equal to 1.

Note also that $\boldsymbol{s}(s, t)$ in Eq. (1) is a surface on the Gaussian sphere. Its points are unit tangent vectors of the surface to be designed. That is where the concept of blending tangent vectors to generate free-form surfaces comes in.

3. Curves on the Gaussian sphere

Interpolations of tangent vectors is a key issue in streamline modeling, because it determines not only the quality of the generated surfaces but also the processing speed of the display and modification process. Hence, a basic problem in streamline modeling is how to approximate or interpolate given tangent vectors to generate smooth curves on the Gaussian sphere. For notational convenience, we shall use unit quaternions to represent rotations about arbitrary axes in the rest of this section. We need to review some basic properties of quaternions first.

3.1. Quaternions

3.1.1. Definition of quaternion

Similar to the definition of a complex number $z = a + bi$ where a and b are real numbers and i is the imaginary number, a *quaternion* q is given by

$$q = a + bi + cj + dk, \tag{4}$$

where a, b, c, d are real numbers and i, j and k are different imaginary units. For given two quaternions $q_0 = a_0 + b_0 i + c_0 j + d_0 k$ and $q_1 = a_1 + b_1 i + c_1 j + d_1 k$ the addition rule is defined by

$$q_0 + q_1 = (a_0 + a_1) + (b_0 + b_1)i + (c_0 + c_1)j + (d_0 + d_1)k. \tag{5}$$

Subtraction is defined as an inverse operation of addition.

Multiplication among i, j and k is defined by

$$\begin{aligned} &i^2 = -1, \quad j^2 = -1, \quad k^2 = -1, \quad ij = k, \\ &ji = -k, \quad jk = i, \quad kj = -i, \quad ki = j, \\ &ik = -j. \end{aligned} \tag{6}$$

Hence

$$\begin{aligned} q_0 q_1 = {} & (a_0 a_1 - b_0 b_1 - c_0 c_1 - d_0 d_1) \\ & + (a_0 b_1 + b_0 a_1 + c_0 d_1 - d_0 c_1)i \\ & + (a_0 c_1 + c_0 a_1 + d_0 b_1 - b_0 d_1)j \\ & + (a_0 d_1 + d_0 a_1 + b_0 c_1 - c_0 b_1)k. \end{aligned} \tag{7}$$

$q_0 q_1$ is called the *product* of q_0 and q_1. As the above equation indicates, generally $q_0 q_1 \neq q_1 q_0$. Division is defined as an inverse operation of multiplication. Since multiplication of quaternions is noncommutative, there exist right and left quotients.

The *conjugate* $\bar{q}$ of $q = a + bi + cj + dk$ is defined as follows:

$$\bar{q} = a - bi - cj - dk. \tag{8}$$

It is easy to see that

$$q\bar{q} = \bar{q}q = |q|^2 = a^2 + b^2 + c^2 + d^2. \tag{9}$$

$|q|$ is called the norm of q. Quaternion norm is multiplication-invariant, i.e.

$$|q_0 q_1| = |q_0||q_1|. \tag{10}$$

The inverse $q - {}^{1}$ of a quaternion $q = a + bi + cj + dk$ satisfies the condition $qq - {}^{1} = 1$ and can be expressed as

$$q^{-1} = \frac{1}{|q|^2}(a - bi - cj - dk) = \frac{\bar{q}}{|q|^2}. \tag{11}$$

Multiplication of a quaternion by its inverse does not depend on the order. It is easy to verify that $q - {}^{1}q = 1$.

Eq. (7) shows that the product of two pure vector quaternions (whose real parts are equal to 0) $q_0 = b_0 i + c_0 j + d_0 k$ and $q_1 = b_1 i + c_1 j + d_1 k$ is given by

$$q_0 q_1 = -(q_0, q_1) + [q_0, q_1] \tag{12}$$

where

$$(q_0, q_1) \equiv b_0 b_1 + c_0 c_1 + d_0 d_1 \tag{13}$$

and

$$[q_0, q_1] \equiv (c_0 d_1 - d_0 c_1)i + (d_0 b_1 - b_0 d_1)j + (b_0 c_1 - c_0 b_1)k. \tag{14}$$

$(\mathrm{q}_0, \mathrm{q}_1)$ and $[\mathrm{q}_0, \mathrm{q}_1]$ are identical to the scalar and cross products in three dimensional space, respectively if i, j and k correspond to x, y, and z axes.

3.1.2. Rotation

Given a unit quaternion $q \in S^3$ a 3D rotation $RQ \in SO(3)$ is defined as follows:

$$R_q(\boldsymbol{v}) = q\boldsymbol{v}q^{-1}, \quad \text{for} \quad \boldsymbol{v} \in R^3, \tag{15}$$

where $\boldsymbol{v} = (x, y, z)$ is interpreted as a quaternion $(0,\ x,\ y,\ z)$ and quaternion multiplication is assumed for the product. If $q = \cos\theta + \hat{\boldsymbol{v}}\sin\theta \in S^3$ for an angle θ and a unit vector

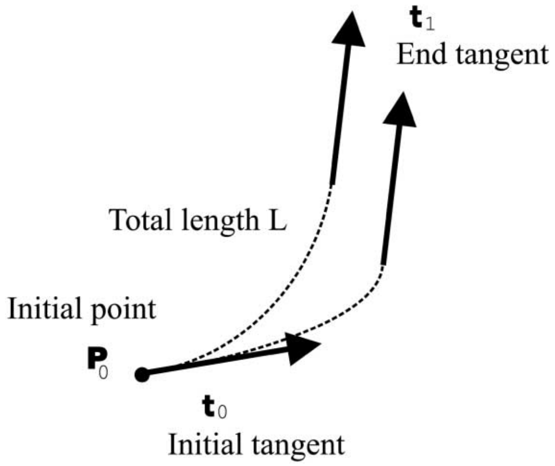


Fig. 3. Interpolation problem: finding a smooth curve whose initial point, tangent vectors at its initial and end points, and total length are given.

$\hat{\boldsymbol{v}} \in S^3$, then R_q is a rotation by an angle of 2θ about the axis $\hat{\boldsymbol{v}}$ [7]. When a smooth rotation curve $r_{q(u)} \in SO(3)$ has an angular derivative $2\omega(u) \in R^3$ (if u is time, then it is an angular velocity), the unit quaternion curve $q(u) \in S^2$ satisfies:

$$\frac{\mathrm{d}q(u)}{\mathrm{d}u} = q(u)\omega(u). \tag{16}$$

The unit quaternion multiplication is not commutative; therefore, the order of multiplication is thus very important. Miura [11]used the global frame because it is convenient to define curves in a fixed frame. In addition, the global frame is essential for our new unit quaternion curve construction scheme (see Section 4).

Given a vector $\boldsymbol{v} = \theta\hat{\boldsymbol{v}} \in R^3$ with $\hat{\boldsymbol{v}} \in S^2$, the exponential of $\boldsymbol{v}$, defined as follows

$$\exp(\boldsymbol{v}) = \sum_{i=0}^{\infty} \frac{\boldsymbol{v}^i}{i!} = \cos\theta + \hat{\boldsymbol{v}}\sin\theta \in S^3, \tag{17}$$

is a unit quaternion, which represents a rotation by an angle of 2θ about the axis $\hat{\boldsymbol{v}}$, where $\boldsymbol{v}^i$ is computed using the quaternion multiplication. The exponential function *exp* can be interpreted as a mapping from an angular derivative vector (measured in R^3) to a unit quaternion which represents a rotation.

Given two unit quaternions q_0 and q_1, the geodesic curve, or the spherical interpolation function (*slerp*) $\gamma_{\mathrm{q0,\ q1}}(u) =$ *slerp*$(q_0, q_1, u) \in \mathrm{S}^3$ (which connects q_0 and q_1) is given by

$$\gamma_{q_0,q_1}(u) = slerp(q_0, q_1, u) = q_0\exp\left(\log\left(q_0^{-1}q_1\right)u\right)$$
$$= q_0\left(q_0^{-1}q_1\right)^u. \tag{18}$$

3.2. Interpolation of tangent vectors

A simple tangent vector interpolation problem: 'finding a smooth curve in 3D space whose initial point, tangent vectors at its initial and end points and total length are given,' will be discussed here (Fig. 3). This problem is different from that of interpolating positions of points. The most natural and desirable answer is a circular arc, because if we rotate its tangent with an angle proportional to arc length, the resulting curve becomes a circular arc, which has constant curvature. The interpolation of two tangent vectors by circular arc achieves the minimum variational energy of variation of curvature because the variation of curvature for a circular arc is always zero.

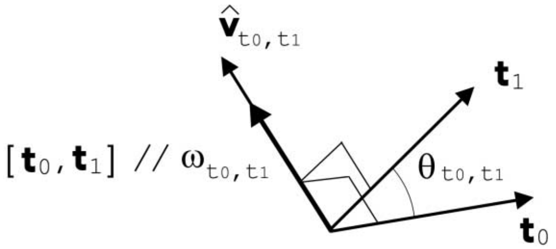

$\hat{\mathbf{v}}_{t0,t1}$
$\mathbf{t}_1$
$[\mathbf{t}_0, \mathbf{t}_1] \ // \ \omega_{t0,t1}$
$\theta_{t0,t1}$
$\mathbf{t}_0$

Fig. 4. The axis of rotation.

For two given unit tangent vectors $\boldsymbol{t}_0 = q_0\hat{\boldsymbol{v}}q_0^{-1}$ and $\boldsymbol{t}_1 = q_1\hat{\boldsymbol{v}}_0q_1^{-1}$ where q_0 and q_1 are quaternions,[1] the geodesic curve $\gamma_{t_0},t_1(u)$ on the Gaussian sphere, connecting the end points of the two unit tangent vectors $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$, is a great arc between the two end points and is given by

$$\gamma_{t_0,t_1}(u) = q_{t_0,t_1}(u)\boldsymbol{t}_{0q}q_{t_0,t_1}^{-1}(u), \tag{19}$$

where

$$q_{t_0,t_1}(u) = \cos\left(\frac{\theta_{t_0,t_1}}{2}u\right) + \hat{\boldsymbol{v}}\sin\left(\frac{\theta_{t_0,t_1}}{2}u\right) = \exp\left(\omega_{t_0,t_1}u\right). \tag{20}$$

ω_{t_0},t_1 is equivalent to an angular velocity of the rotation if parameter u is regarded as time, and its direction is equal to the cross product of $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$, as shown in Fig. 4 [10]. $\boldsymbol{\omega}_{t_0,t_1}$ is given by

$$\boldsymbol{\omega}_{t_0,t_1} = \frac{1}{2}\sin^{-1}(|\boldsymbol{t}_0 \times \boldsymbol{t}_1|)\frac{\boldsymbol{t}_0 \times \boldsymbol{t}_1}{|\boldsymbol{t}_0 \times \boldsymbol{t}_1|} = \frac{1}{2}\theta_{t_0,t_1}\hat{\boldsymbol{v}}_{t_0,t_1}, \tag{21}$$

where θ_{t_0,t_1} is the angle between $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$, $|\boldsymbol{t}_0 \times \boldsymbol{t}_1|$ is the norm of $\boldsymbol{t}_0 \times \boldsymbol{t}_1$ and $\hat{\boldsymbol{v}}_{t_0,t_1}$ is a unit vector whose direction is equal to $\boldsymbol{t}_0 \times \boldsymbol{t}_1$.

The above $\boldsymbol{\omega}_{t_0,t_1}$ makes $\gamma_{t_0,t_1}(u)$ satisfy $\gamma_{t_0},t_1(0) = \boldsymbol{t}_0$ and $\gamma_{t_0},t_1(1) = \boldsymbol{t}_1$. The resulting curve, whose tangent is given by $q_{t_0,t_1}(u)\boldsymbol{t}_0q_{t_0,t_1}^{-1}(u)$, is a circular arc. Furthermore, $\boldsymbol{\omega}_{t_0},t_1$ depends on $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ only, it does not depend on the choices of q_0 or q_1 that are not unique for $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$. If one uses $\boldsymbol{\omega}_{t_0},t_1 = \log(q_0^{-1})$ as used in Ref. [8], the resulting curve in general is not a circular arc.

The above argument shows that a simple linear

[1] Note that for a given tangent vector $\boldsymbol{t}$, q does not uniquely satisfy $\boldsymbol{t} = q\hat{\boldsymbol{v}}_0q^{-1}$, as infinitely many rotations are possible to change $\hat{\boldsymbol{v}}_0$ to $\boldsymbol{t}$. Assume $\hat{\boldsymbol{v}}_0 = (1, 0, 0)$. q is given by, for example, $\cos\beta + \hat{\boldsymbol{v}}\sin\beta = (\cos\beta, 0, \cos\alpha\sin\beta, \sin\alpha\sin\beta)$.

combination of tangent vectors is not suitable for constructing a curve that is a locus of tangent vectors for streamline modeling. Miura et al. [10] proposed a unit quaternion curve based on the approach of Kim et al. [8] for streamline modeling as follows:

$$q(u) = \left(\prod_{i=n}^{1} \exp(\omega_i \bar{B}_{i,k}(u))\right) q_0, \tag{22}$$

where $\tilde{B}_{i,k}(u)$ is a cumulative form of the B-spline basis function $\tilde{B}_{i,k}$ of order k defined by

$$\tilde{B}_{i,k}(u) = \sum_{j=1}^{n} B_{i,k}(u) \tag{23}$$

$$= \begin{cases} \displaystyle\sum_{j=i}^{i+k-1} B_{j,k}(u) & \text{if} \quad u_i < u < u_{u_{i+k-1}} \\ 1 & \text{if} \quad u \geq u_{i+k-1} \\ 0 & \text{if} \quad u \leq u_i. \end{cases} \tag{24}$$

The $\boldsymbol{\omega}_i$ used in Eq. (22) is defined by

$$\omega_i = \frac{1}{2}\sin^{-1}(|\boldsymbol{t}_{i-1} \times \boldsymbol{t}_i|)\frac{\boldsymbol{t}_{i-1} \times \boldsymbol{t}_i}{|\boldsymbol{t}_{i-1} \times \boldsymbol{t}_i|}. \tag{26}$$

The above $\boldsymbol{\omega}_i$ makes sense in the global frame because it is defined by vectors in the global frame. The integral curve whose tangent is given by $q(u)v_0q(u) - {}^1$ was proposed as the unit quaternion integral (QI) curve in Ref. [10].

3.3. Linear combination of tangent vectors

Although the unit quaternion curve described above produces a smooth curve on the Gaussian sphere, it suffers two serious problems. One is that is does not produce an identical curve when the same tangent vectors $\boldsymbol{t}_i, i = 0, \ldots, n$ are specified in reverse order: the resulting unit quaternion curves $q_\mathrm{o}(u)$ and $q_\mathrm{r}(u)$ are

$$q_\mathrm{o} = q_\mathrm{o}[\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_n](u) = \left(\prod_{i=n}^{1} \exp(\boldsymbol{\omega}_i \tilde{B}_{i,k}(u))\right) q_\mathrm{o}$$

$$q_\mathrm{r} = q_r[\boldsymbol{t}_n, \boldsymbol{t}_{n-1}, \ldots, \boldsymbol{t}_0](u) = \left(\prod_{i=n}^{1} \exp(-\boldsymbol{\omega}_i \tilde{B}_{i,k}(u))\right) \times \left(\prod_{i=n}^{1} \exp(\boldsymbol{\omega}_i(u))\right) q_\mathrm{o} \tag{25}$$

with

$$r_r(u) \neq q_\mathrm{o}(1 - u) \tag{27}$$

The other problem is that the streamline calculation process for the unit quaternion curves needs more processing power. With numerical integration being the only choice, it takes a long time to compute a streamline if the integrand involves curve formulae of exponential functions.

To overcome these problems, we shall use subdivision curves and surfaces on the Gaussian sphere. The basic idea is to define a smooth curve or surface as the limit of a sequence of successive refinements. For each refinement, new points that are linear combinations of old points are inserted. In order to apply the subdivision concept to S^2 space (equivalent to the Gaussian sphere), one needs to define linear combination of tangent vectors first. In R^3, the mid-point $\boldsymbol{p}_\mathrm{m}$ of two points $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ is defined by

$$\boldsymbol{p}_\mathrm{m} = \frac{1}{2}(\boldsymbol{p}_0 + \boldsymbol{p}_1) \tag{28}$$

In S^2, however, the mid-tangent $\boldsymbol{t}_\mathrm{m}$ of two tangents $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ can not be defined by

$$\boldsymbol{t}_\mathrm{m} = \frac{1}{2}(\boldsymbol{t}_0 + \boldsymbol{t}_1) \tag{29}$$

because the norm of $\boldsymbol{t}_\mathrm{m}$ would not be of unit length even though the norms of both $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ are equal to 1. It should be defined as

$$\boldsymbol{t}_\mathrm{m} = q\left(\frac{1}{2}\right)\boldsymbol{t}_0 q^{-1}\left(\frac{1}{2}\right) \tag{30}$$

where

$$q(u) = \exp\left(\boldsymbol{\omega}_{t_0,t_1} u\right) \tag{31}$$

and $\boldsymbol{\omega}_{t_0,t_1}$ is defined in Eq. (21). Although Eq. (30) uses quaternion calculus, it can be calculated by addition of two tangent vectors and its normalization in a shorter processing time as follows:

$$\boldsymbol{t}_\mathrm{m} = \frac{\boldsymbol{t}_0 + \boldsymbol{t}_1}{|\boldsymbol{t}_0 \times \boldsymbol{t}_1|} \tag{32}$$

We define a linear interpolation of two tangent vectors $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ as

$$\boldsymbol{t}(u) = (1 - u)\boldsymbol{t}_0 \oplus u\boldsymbol{t}_1 = q(u)\boldsymbol{t}_0 q^{-1}(u). \tag{33}$$

Note that the above interpolation is commutative, i.e.

$$(1 - u)\boldsymbol{t}_0 \oplus u\boldsymbol{t}_1 = u\boldsymbol{t}_1 \oplus (1 - u)\boldsymbol{t}_0. \tag{34}$$

3.4. General formulation of linear combination

As mentioned in the previous subsection, the unit quaternion curve $q_0(u)$ representing a blend of two tangent vectors and its reversed curve $q_\mathrm{r}(u)$ are identical in such a sense that $q_\mathrm{r}(u) = q_\mathrm{o}(1 - u)$. Similarly, the result of successive rotations about a fixed axis does not depend on the order of its rotations and a unit quaternion curve representing such rotations is identical with its reversed curve. However the result of successive general rotations about different axes depends on their order and the corresponding original curve and its reverse are not identical. In this subsection, we will propose a method to make the reversed curve identical with the original curve.

First, consider a general linear combination of points in R^3. A general linear combination $\boldsymbol{p}$ of $n + 1$ points $\boldsymbol{p}_i$,

$I = 0, \ldots, n$ in R^3 is defined by

$$\boldsymbol{p} = c_0\boldsymbol{p}_0 + c_1\boldsymbol{p}_1 + \cdots + c_n\boldsymbol{p}_n \tag{35}$$

where c_i, $i = 0, \ldots, n$, are coefficients. Since our method depends on whether n is even or odd, we will discuss the cases that $n = 2$ and $n = 3$ first, and then deal with the general case.

If $n = 2$, Eq. (35) can be rewritten as follows:

$$\boldsymbol{p} = \boldsymbol{p}[\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2] = c_0\boldsymbol{p}_0 + c_1\boldsymbol{p}_1 + c_2\boldsymbol{p}_2. \tag{36}$$

For subdivision, because of the requirement of invariance under affine transformation, the sum of the coefficients $\sum_0^n c_i$ should be equal to 1. Hence,

$$c_0 + c_1 + c_2 = 1. \tag{37}$$

By considering the identify of the reversed curve with the original, we can rewrite Eq. (37) as follows:

$$\begin{aligned}\boldsymbol{p} &= \left(c_0\boldsymbol{p}_0 + \frac{1}{2}c_1\boldsymbol{p}_1\right) + \left(\frac{1}{2}c_1\boldsymbol{p}_1 + c_2\boldsymbol{p}_2\right) \\ &= \left(c_0 + \frac{1}{2}c_1\right)\left(\frac{c_0}{c_0 + \frac{1}{2}c_1}\boldsymbol{p}_0 + \frac{\frac{1}{2}c_1}{c_0 + \frac{1}{2}c_1}\boldsymbol{p}_1\right) \\ &\quad + \left(\frac{1}{2}c_1 + c_2\right)\left(\frac{\frac{1}{2}c_1}{\frac{1}{2}c_1 + c_2}\boldsymbol{p}_1 + \frac{c_2}{\frac{1}{2}c_1 + c_2}\boldsymbol{p}_2\right).\end{aligned} \tag{38}$$

For example, if $c_0 = 1/7$, $c_1 = 2/7$, $c_2 = 4/7$, then,

$$\boldsymbol{p} = \frac{2}{7}\left(\frac{1}{2}\boldsymbol{p}_0 + \frac{1}{2}\boldsymbol{p}_1\right) + \frac{5}{7}\left(\frac{1}{5}\boldsymbol{p}_1 + \frac{4}{5}\boldsymbol{p}_2\right). \tag{39}$$

Based on the above transformation, a linear combination $\boldsymbol{t}$ of 3 tangents $\boldsymbol{t}_0$, $\boldsymbol{t}_1$, and $\boldsymbol{t}_2$ in S^2 can be defined as repetitions of the binary operator $\oplus$:

$$\begin{aligned}\boldsymbol{t} &= \boldsymbol{t}[\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2] = c_0\boldsymbol{t}_0 \oplus c_1\boldsymbol{t}_1 \oplus c_2\boldsymbol{t}_2 \\ &\equiv \left(c_0 + \frac{1}{2}c_1\right)\left(\frac{c_0}{c_0 + \frac{1}{2}c_1}\boldsymbol{t}_0 \oplus \frac{\frac{1}{2}c_1}{c_0 + \frac{1}{2}c_1}\boldsymbol{t}_1\right) \\ &\quad \oplus \left(\frac{1}{2}c_1 + c_2\right)\left(\frac{\frac{1}{2}c_1}{\frac{1}{2}c_1 + c_2}\boldsymbol{t}_1 \oplus \frac{c_2}{\frac{1}{2}c_1 + c_2}\boldsymbol{t}_2\right).\end{aligned} \tag{40}$$

In the above definition, if the coefficients c_i are functions of parameter u (i.e. $c_0 = c_0(u)$, $c_1 = c_1(u)$, and $c_2 = c_2(u)$ and furthermore, $c_2(u) = c_0(1 - u)$ and $c_1(u) = c_1(1 - u)$ (these conditions are satisfied if the coefficients are Bernstein basis functions), then the following equation is satisfied:

$$\boldsymbol{t}_r(u) = \boldsymbol{t}_o(1 - u) \tag{41}$$

where

$$\boldsymbol{t}_o(u) = \boldsymbol{t}_{\mathrm{o}}[\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2](u), \tag{42}$$

$$\boldsymbol{t}_{\mathrm{r}}(u) = \boldsymbol{t}_{\mathrm{r}}[\boldsymbol{t}_2, \boldsymbol{t}_1, \boldsymbol{t}_0](u). \tag{43}$$

When $n = 3$, Eq. (35) is transformed into the following form, which is simpler than the $n = 2$ case

$$\begin{aligned}\boldsymbol{t} &\equiv (c_0 + c_1)\left(\frac{c_0}{c_0 + c_1}\boldsymbol{t}_0 \oplus \frac{c_1}{c_0 + c_1}\boldsymbol{t}_1\right) \\ &\quad \oplus (c_2 + c_3)\left(\frac{c_2}{c_2 + c_3}\boldsymbol{t}_2 \oplus \frac{c_3}{c_2 + c_3}\boldsymbol{t}_3\right).\end{aligned} \tag{44}$$

In this case the reversed curve is identical with the original curve as well.

In the general case, if n is even, Eq. (35) is subdivided into two portions, similar to the $n = 2$ case, as follows:

$$\begin{aligned}\boldsymbol{t} &= c_0\boldsymbol{t}_0 + c_1\boldsymbol{t}_1 + \cdots + c_n\boldsymbol{t}_n = d_0\left(\frac{c_0}{d_0}\boldsymbol{t}_0 + \cdots + \frac{\frac{1}{2}c_{\frac{n}{2}}}{d_0}\boldsymbol{t}_{\frac{n}{2}}\right) \\ &\quad \oplus d_1\left(\frac{\frac{1}{2}c_{\frac{n}{2}}}{d_1}\boldsymbol{t}_0 + \cdots + \frac{c_n}{d_1}\boldsymbol{t}_n\right)\end{aligned} \tag{45}$$

where $d_0 = c_0 + \cdots + (1/2)c_{n/2}$, $d_1 = (1/2)c_{n/2} + \cdots + c_n$, and $d_0 + d_1 = 1$. If n is odd, similar to the $n = 3$ case $\boldsymbol{t}$ is written as

$$\begin{aligned}\boldsymbol{t} &= c_0\boldsymbol{t}_0 + c_1\boldsymbol{t}_1 + \cdots + c_n\boldsymbol{t}_n = d_0\left(\frac{c_0}{d_0}\boldsymbol{t}_0 + \cdots + \frac{c_{\frac{n-1}{2}}}{d_0}\boldsymbol{t}_{\frac{n-1}{2}}\right) \\ &\quad \oplus d_1\left(\frac{c_{\frac{n+1}{2}}}{d_1}\boldsymbol{t}_{\frac{n+1}{2}} + \cdots + \frac{\frac{1}{2}c_{\frac{n}{2}}}{d_1}\boldsymbol{t}_n\right)\end{aligned} \tag{46}$$

where $d_0 = c_0 + \cdots + c_{(n-1)/2}$, $d_1 = c_{(n+1)/2} + \cdots + c_n$. In either case, we have $d_0 + d_1 = 1$. This follows from the requirement of invariance under affine transformations. One then simply repeats the same process recursively on each portion of the subdivided expression, and the linear combination of tangent vectors is defined.

In the linear combination, the sum of the operands involving a $\oplus$ operator is always equal to 1. For instance, in Eq. (4), the sum of the operands sandwiching the first $\oplus$ is

$$\frac{c_0}{c_0 + \frac{1}{2}c_1} + \frac{\frac{1}{2}c_1}{c_0 + \frac{1}{2}c_1} = 1. \tag{47}$$

The sum of the operands of the second $\oplus$ is

$$\left(c_0 + \frac{1}{2}c_1\right) + \left(\frac{1}{2}c_1 + c_2\right) = c_0 + c_1 + c_2 = 1. \tag{48}$$

This follows from the requirement of invariance under affine transformation. Hence, the above definition is well defined.

This definition of linear combination of tangent vectors makes the reversed curve identical with the original curve. This definition also allows the linear combination to be calculated without quaternion calculus in a shorter time by the repetition of rotation calculations.

3.5. Chaikin's algorithm

Chaikin's algorithm [2] generates a quadratic B-spline curve from a polygon by successively cutting its corners. Each subdivision generates two new points on each polygon leg at (1/4, 3/4). For a polygon with $n+1$ vertices $\boldsymbol{p}_i^j$ at the subdivision length j, two new points defined as follows are inserted into the polygon leg $\boldsymbol{p}_i^j\boldsymbol{p}_{i+1}^j$

$$\boldsymbol{p}_{2i}^{j+1} := \frac{3}{4}\boldsymbol{p}_i^j + \frac{1}{4}\boldsymbol{p}_{i+1}^j, \tag{49}$$

$$\boldsymbol{p}_{2i+1}^{j+1} := \frac{1}{4}\boldsymbol{p}_i^j + \frac{3}{4}\boldsymbol{p}_{i+1}^j. \tag{50}$$

The above equations use only two points to generate a new point. It is straightforward to extend them to curves on the Gaussian sphere using the linear interpolation process given by Eq. (33), as follows:

$$\boldsymbol{t}_{2i}^{j+1} := \frac{3}{4}\boldsymbol{t}_i^j \oplus \frac{1}{4}\boldsymbol{t}_{i+1}^j, \tag{51}$$

$$\boldsymbol{t}_{2i+1}^{j+1} := \frac{1}{4}\boldsymbol{t}_i^j \oplus \frac{3}{4}\boldsymbol{t}_{i+1}^j, \tag{52}$$

Note that Eqs. (51) and (52) can be calculated very quickly because, for example, Eq. (51) can be rewritten as follows:

$$\boldsymbol{t}_{2i}^{j+1} := \frac{1}{2}\boldsymbol{t}_i^j \oplus \frac{1}{2}\left(\frac{1}{2}\boldsymbol{t}_i^j \oplus \frac{1}{2}\boldsymbol{t}_{i+1}^j\right). \tag{53}$$

It can be calculated by performing the addition of two vectors and its normalization twice.

3.6. B-spline subdivision

B-spline division is a generalization of Chaikin's algorithm and it obeys a refinement equation [14]. The refinement equation for B-splines of degree l is given by

$$B_l(t) = \frac{1}{2^l}\sum_{k=0}^{l+1}\begin{pmatrix} l+1 \\ k \end{pmatrix} B_l(2t-k). \tag{54}$$

Quadratic B-spline division is identical to Chaikin's algorithm and cubic B-spline subdivision is given by

$$\boldsymbol{p}_{2i}^{j+1} := \frac{1}{8}\left(\boldsymbol{p}_i^j + 6\boldsymbol{p}_{i+1}^j + \boldsymbol{p}_{i+2}^j\right), \tag{55}$$

$$\boldsymbol{p}_{2i+1}^{j+1} := \frac{1}{2}\left(\boldsymbol{p}_{i+1}^j + \boldsymbol{p}_{i+2}^j\right). \tag{56}$$

To extend it to curves on the Gaussian sphere, one can use our tangent blending method with the following equations:

$$\boldsymbol{t}_{2i}^{j+1} := \frac{1}{2}\left(\frac{1}{4}\boldsymbol{t}_i^j \oplus \frac{3}{4}\boldsymbol{t}_{i+1}^j\right) \oplus \frac{1}{2}\left(\frac{3}{4}\boldsymbol{t}_{i+1}^j \oplus \frac{1}{4}\boldsymbol{t}_{i+2}^j\right), \tag{57}$$

$$\boldsymbol{t}_{2i+1}^{j+1} := \frac{1}{2}\boldsymbol{t}_{i+1}^j \oplus \frac{1}{2}\boldsymbol{t}_{i+2}^j, \tag{58}$$

Similar to Chaikin's algorithm, the above expressions can be calculated by performing the addition of two vectors and its normalization five times and once for Eqs. (57) and (58), respectively.

3.7. 4-point interpolatory subdivision

A modified 4-point interpolatory subdivision refines polygons with

$$\boldsymbol{p}_{2i}^{j+1} := \boldsymbol{p}_i^j \tag{59}$$

$$\boldsymbol{p}_{2i+1}^{j+1} := \frac{8+\omega}{16}\left(\boldsymbol{p}_i^j + \boldsymbol{p}_{i+1}^j\right) - \frac{\omega}{16}\left(\boldsymbol{p}_{i-1}^j + \boldsymbol{p}_{i+2}^j\right) \tag{60}$$

where $0 < \omega < 2(\sqrt{5}-1)$ is sufficient to ensure convergence to a smooth limit curve [5]. The standard value is $\omega = 1$ with which the scheme has an order three precision.

On the Gaussian sphere, using the symmetric definition of $\oplus$ operation Eq. (60) becomes

$$\boldsymbol{t}_{2i+1}^{j+1} := \frac{1}{2}\left(-\frac{\omega}{8}\boldsymbol{t}_{i-1}^j \oplus \frac{8+\omega}{8}\boldsymbol{t}_i^j\right) \oplus \frac{1}{2}\left(\frac{8+\omega}{8}\boldsymbol{t}_{i+1}^j \oplus -\frac{\omega}{8}\boldsymbol{t}_{i+2}^j\right). \tag{61}$$

Coefficient $-\omega/8$ is negative and $(8+\omega)/8$ is larger than 1, but Eq. (33) is well defined for a parameter u smaller than 0 or larger than 1. It can be used in Eq. (61) for curves on the Gaussian sphere.

Fig. 5(a) shows a 4-point interpolatory subdivision curve. Its subdivision depth (level of recursive subdivision) is 3. We can see its interpolatory nature in the figure. For comparison, Fig. 5(b) shows a B-spline unit quaternion curve proposed by Miura et al. [10]. The curve approximates the tangent vectors only.

Table 1 shows processing times of tangent calculations for Chaikin's, cubic B-spline, and 4-point interpolatory subdivisions and B-spline unit quaternion curves of several orders. The number of the control tangents is the same for every curve although the numbers of calculated tangents for Chaikin's and cubic B-spline subdivisions are different from others. Each subdivision generates different number of tangents by one subdivision step, for example if the number of the original control tangents is equal to 5, then Chaikin's algorithm generates 8 tangents, cubic B-spline and 4-point interpolatory subdivisions generate 7 and 9 tangents, respectively. For B-spline unit quaternion curves, any number of tangents can be calculated and their processing times are shown for the same number of tangents' calculations as the 4-point interpolatory case.

Chaikin's algorithm is much faster even though the number of calculated tangents is 1.5 times of those of others. This is because Eqs. (51) and (52) can be calculated by performing the addition of two vectors and its normalization twice, as mentioned in Subsection 3.5. By the same reason, cubic B-spline subdivision is also fast, but slower than Chaikin's algorithm because of more complex subdivision formulae. Even the 4-point interpolatory subdivision is four or five times faster than the B-spline quaternion curves.

4. Surfaces on the Gaussian sphere

4.1. Doo–Sabin and Catmull–Clark subdivision

Doo and Sabin [4] extended Chaikin's idea for curves to generate surfaces. A surface is generated from a polyhedral network by successively cutting off its corners and edges. An algorithm may be given as follows [12]:

1. For every vertex V_i^j of the polyhedron P^j, a new vertex V_i^{j+1}, called an *image*, is generated on each face adjacent to V_i^j.
2. For each face F_i^j of P^j, a new face, called an *F-face*, is constructed by connecting the *images* vertices V_i^{j+1} generated in Step 1.
3. For each edge E_i, common to two faces F_k^j and F_l^j and a new 4-sided face, called an *E- face*, is constructed by connected the *images* of the end vertices of E_i^j on the faces F_k^j and F_l^j.
4. For each vertex V_i^{j+1}, where n faces meet, a new face, called a *V-face*, is constructed by connecting the images of V_i^j on the faces meeting at V_i^j.

An *image* vertex V_i^{j+1} generated in Step 1 depends only on the vertices of P^j and is given by

$$V_i^{j+1} := \sum_{k=0}^{n-1} a_{ik} V_k^j \tag{62}$$

where V_k^j are vertices of the old faces and a_{ij} are coefficients defined as follows:

$$a_{ik} = \frac{n+5}{4n}, \qquad \text{for} \qquad i = k. \tag{63}$$

$$a_{ik} = \frac{3 + 2\cos\left(2\pi\left(\frac{i-k}{n}\right)\right)}{4n}, \qquad \text{for} \qquad i \neq j.$$

The Catmull–Clark subdivision method [1] is similar to the Doo–Sabin method, but is based on the tensor product bicubic splines. It produces surfaces that are C^2 everywhere except at extraordinary vertices, where they are C^1. Similar to the Doo–Sabin scheme, all coefficients of the subdivision equations of the Catmull–Clark are positive and our blending method is applicable as well.

4.2. Kobbelt subdivision

The Kobbelt surface scheme is a simple extension of the 4-point interpolatory subdivision to surfaces [9]. The vertices of a quadrilateral mesh are indexed locally so that each face can be represented as a sequence of vertices: $s_{i,k}^j = \{\boldsymbol{p}_{i,k}^j, \boldsymbol{p}_{i+1,k}^j, \boldsymbol{p}_{i+1,k+1}^j, \boldsymbol{p}_{i,k+1}^j,\}$. The points $\boldsymbol{p}_{i,k}^{j+1}$ of the refined net can be classified into three disjointed groups. The *vertex-points* $\boldsymbol{p}_{2i,2}^{j+1} := \boldsymbol{p}_{i+1,k}^j$ are fixed due to the interpolatory property. The *edge-points* $\boldsymbol{p}_{2i+1,2k}^{j+1}$ and $\boldsymbol{p}_{2i,2k+1}^{j+1}$ are computed by applying the 4-point rule in the corresponding grid direction, e.g.

$$\boldsymbol{p}_{2i+1,2k}^{j+1} := \frac{8+\omega}{16}\left(\boldsymbol{p}_{i,k}^j + \boldsymbol{p}_{i+1,k}^j\right) - \frac{\omega}{16}\left(\boldsymbol{p}_{i-1,k}^j + \boldsymbol{p}_{i+2}^j\right). \tag{64}$$

The *face-points* $\boldsymbol{p}_{2i+1,2k+1}^{j+1}$ are computed by applying the 4-point rule to four consecutive edge-points $\boldsymbol{p}_{2i+1,2k-2}^j, \ldots, \boldsymbol{p}_{2i+1,2k+4}^j$ or $\boldsymbol{p}_{2i-2,2k+1}^j, \ldots, \boldsymbol{p}_{2i+4,2k+1}^j$.

Following Eq. (61), subdivision can be performed in the surface case as well, but mainly on two choices of the blending order: one is to blend tangents in the s parameter direction first and then in the t parameter direction; the other is to blend t first and then s. As effects of different blending orders are not clear and their processing loads are the same, the users are allowed to select the blending order of their own choice in our prototype system.

5. Streamline modeling with subdivision surfaces on the Gaussian sphere

The main design task in streamline modeling is to specify tangent vectors in a parameter direction $\boldsymbol{s}(s, t)$ as well as the initial curve $\boldsymbol{S}(s_0, t)$ in Eq. (1). Subdivision surfaces on the Gaussian sphere are used to specify $\boldsymbol{s}(s, t)$.

As shown in Fig. 6, an open quadrilateral polygonal mesh is used in the parameter space to specify the original topological structure of a subdivision surface on the Gaussian

Table 1
Processing times

Curve type	Order	Number of control tangents	Number of calculated tangents	Time (ms)
Subdivision (Chaikin)		5	386 (depth = 7)	0.31
Subdivision (cubic B-spline)		5	259 (depth = 7)	0.63
Subdivision (4 interpol.)		5	257 (depth = 6)	1.09
B-spline	3	5	257	4.69
B-spline	4	5	257	4.84
B-spline	5	5	257	5.16

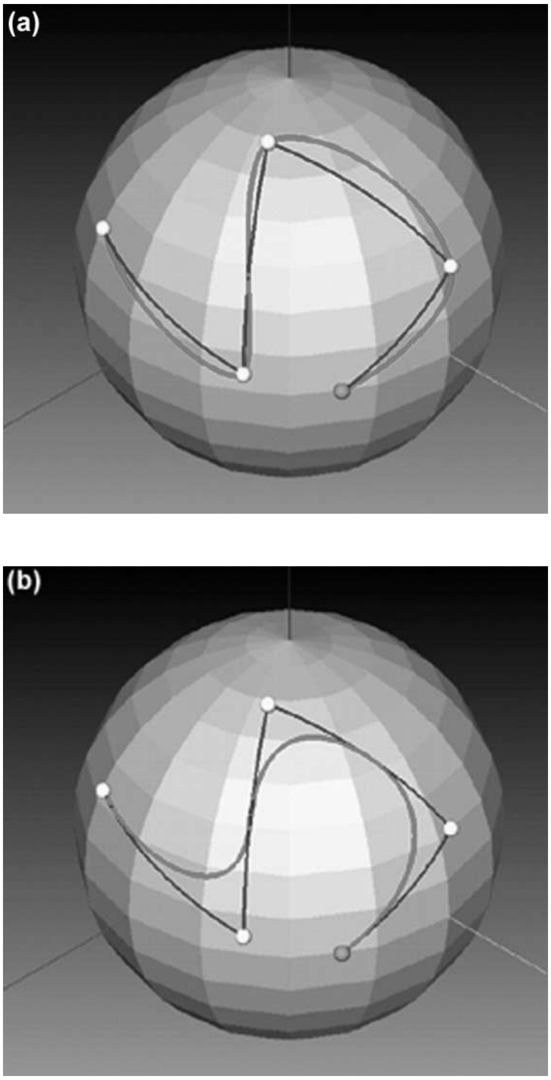


Fig. 5. Comparison of 4-point interpolatory subdivision and B-spline quaternion curves on the Gaussian sphere. (a) 4-point interpolatory subdivision curve (red curve, $\omega = 1$, subdivision length = 3). Red ball corresponds to the first vertex (initial tangent vector) and yellow balls correspond to vertices of the subdivision curve on the Gaussian sphere. Blue curves are great arcs between two points on the sphere; (b) B-spline unit quaternion curve proposed by Miura et al. [10] (red curve). Blue curves are great arcs between two points on the sphere.

sphere as well as its vertex distribution. Each horizontal line of the mesh with a constant parameter value t corresponds to a streamline. In the figure L_t is the total length of the initial curve. The total length L_s in s parameter direction can be specified as a function of t. This means that each streamline can have a proper total length of its own. After the specification of the topological structure of the subdivision surface, the next task for the designer is to specify $L_s(t)$ and tangent vectors corresponding to its vertices. At this initial stage, a streamline whose t value is equal to t_0 is generated by using a tangent vector sequence on a line of the subdivision surface as follows: if the tangent vector sequence is given by $(\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_n)$ and its t parameter for the corresponding vertices in the parameter space is $(s_0, s_1, \ldots, s_n)$, then the vertices of the streamline p_i $(i = 0, \ldots, n+1)$ are calculated by

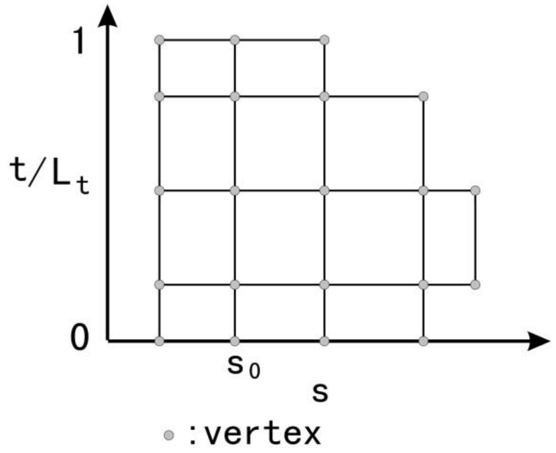


Fig. 6. Specification of the topological structure of a division surface on the Gaussian sphere.

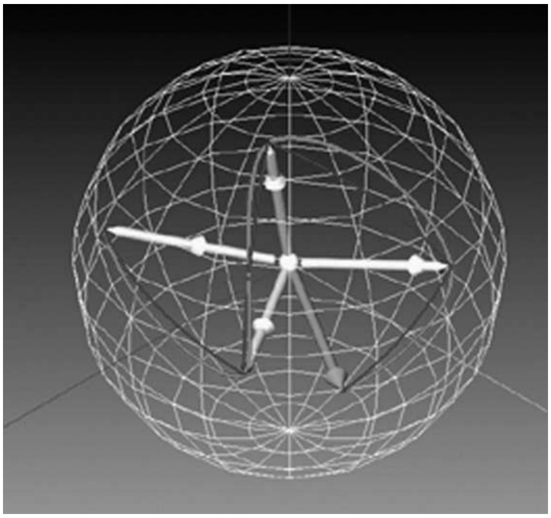


Fig. 7. Arrows corresponding to tangent vectors. Red curve is a 4-point interpolatory subdivision curve ($\omega = 1$, subdivision depth = 3). Red arrow corresponds to the first vertex (initial tangent vector) and yellow arrows correspond to vertices of the subdivision curve on the Gaussian sphere. Blue curves are great arcs between two points on the sphere.

$$\boldsymbol{p}_0 = \boldsymbol{S}(s_0, t_a), \tag{65}$$

$$\boldsymbol{p}_1 = \boldsymbol{p}_0 + \frac{1}{2}(s_1 - s_0)\boldsymbol{t}_0,$$

$$\boldsymbol{p}_{i+1} = \boldsymbol{p}_i + \frac{1}{2}(s_{i+1} - s_{i-1})\boldsymbol{t}_i, \qquad \text{for} \qquad i = 1, \ldots, n-1$$

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n + \frac{1}{2}(s_n - s_{n-1})\boldsymbol{t}_{n.}$$

By carrying out the subdivision steps, new streamlines are generated and for some types of subdivision surfaces, the

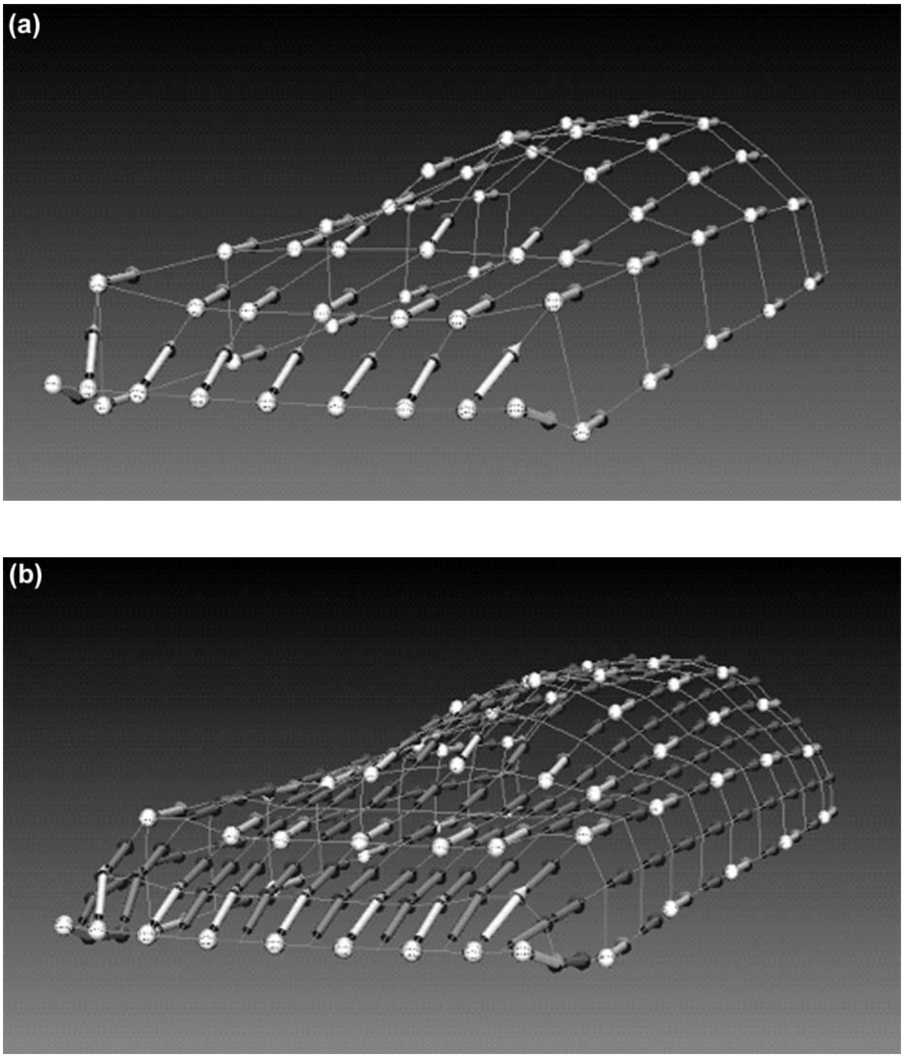


Fig. 8. Car model no. 1.1. (a) Surface defined with original mesh (subdivision depth = 0). Yellow arrows are tangent vectors corresponding to vertices of the subdivision surfaces on the Gaussian sphere. (b) With subdivision surface (subdivision depth = 1). Red arrows are tangent vectors corresponding to refined vertices.

streamlines that already exist are refined (for example, the Kobbelt surface). The tangent vector sequence of the newly generated streamlines are obtained as new vertices of the subdivision surface on the Gaussian sphere. The corresponding t and s sequence values are obtained by the same type of subdivision of the mesh in the parameter space. Note that the t value must be consistent for a streamline and its value is taken from the initial curve $\boldsymbol{S}(s_0, t)$. The vertices of the new and refined streamlines are calculated using the approach described above.

The vertices in Fig. 6 are allocated on vertical lines of the mesh to make the figure simple. It is, however, not necessary for adjacent streamlines to have the same length values s_i for vertically adjacent vertices. If a larger interval is specified, say $s_{i+1} - s_i$ is made larger, then the tangent of the streamline varies slowly there and it, hence, has smaller curvature. In contrast, if a smaller interval is specified, the streamline would have larger curvature instead. It is possible to perform multi-resolution tangent editing after some subdivision steps in a way similar to the ordinary subdivision surfaces.

6. Surface examples of streamline modeling

Several surfaces designed with subdivision surfaces on the Gaussian sphere are shown in this section. These surfaces are generated using Kobbelt type subdivision because its interpolation nature makes the manipulation process more intuitive for the users. Approximation subdivision schemes like B-spline subdivision can produce smoother surfaces, but are not so intuitive for the users in the streamline modeling process because modeling with tangent vectors itself is not as intuitive as with positions.

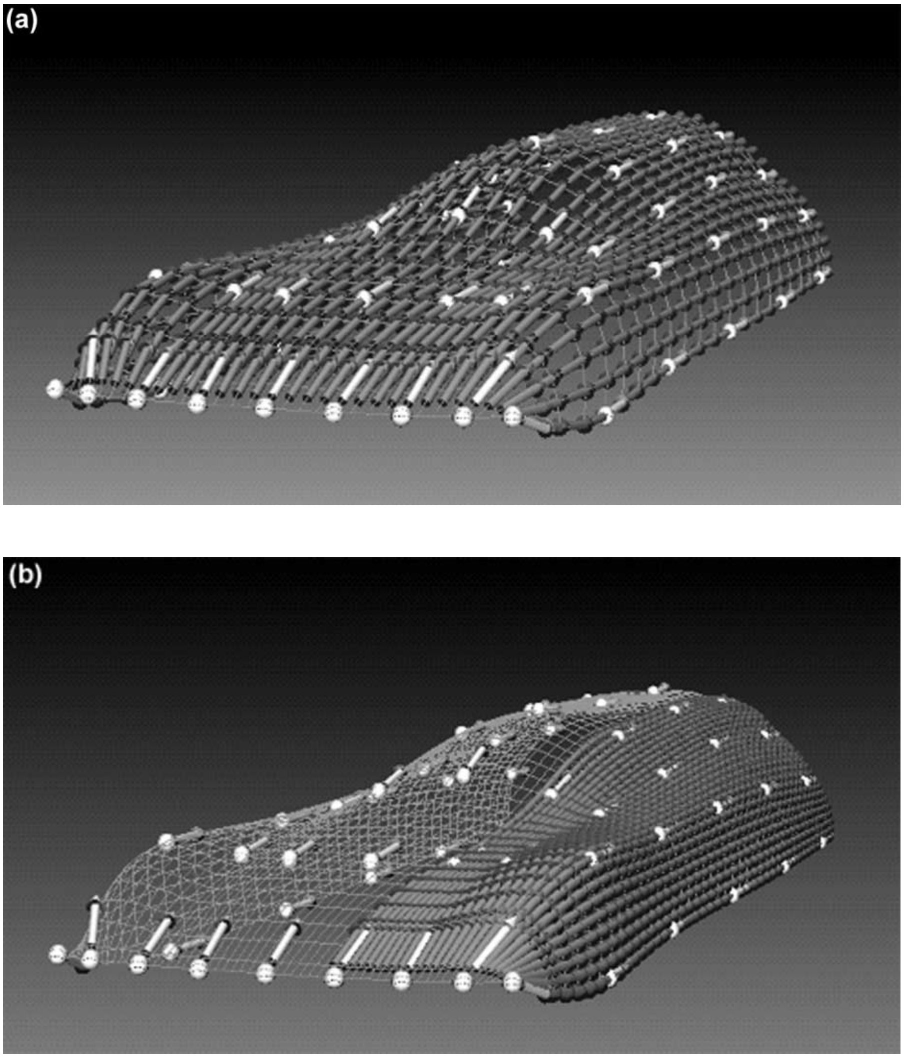


Fig. 9. Car model no. 1.2. (a) With subdivision surface (subdivision depth = 2); (b) with subdivision surface (subdivision depth = 3). Only half of the red arrows are shown.

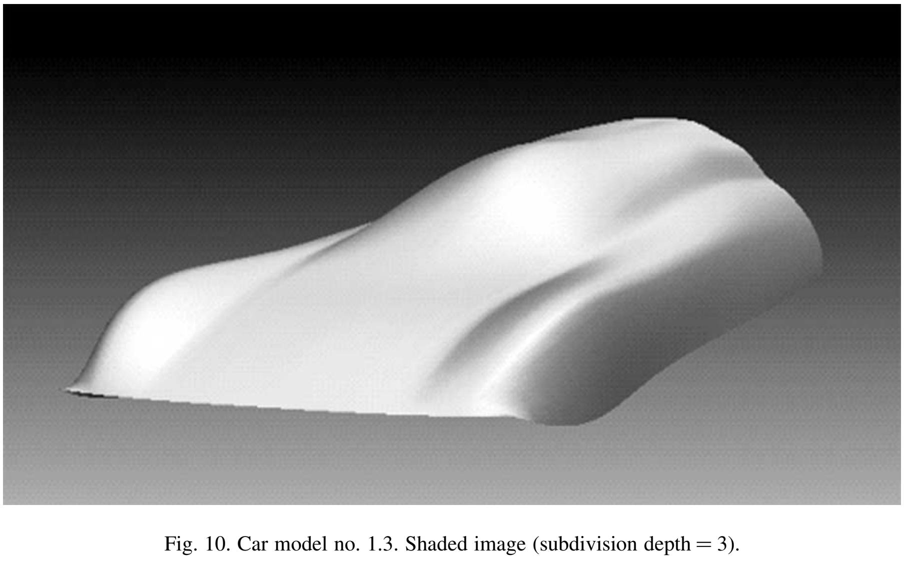


Fig. 10. Car model no. 1.3. Shaded image (subdivision depth = 3).

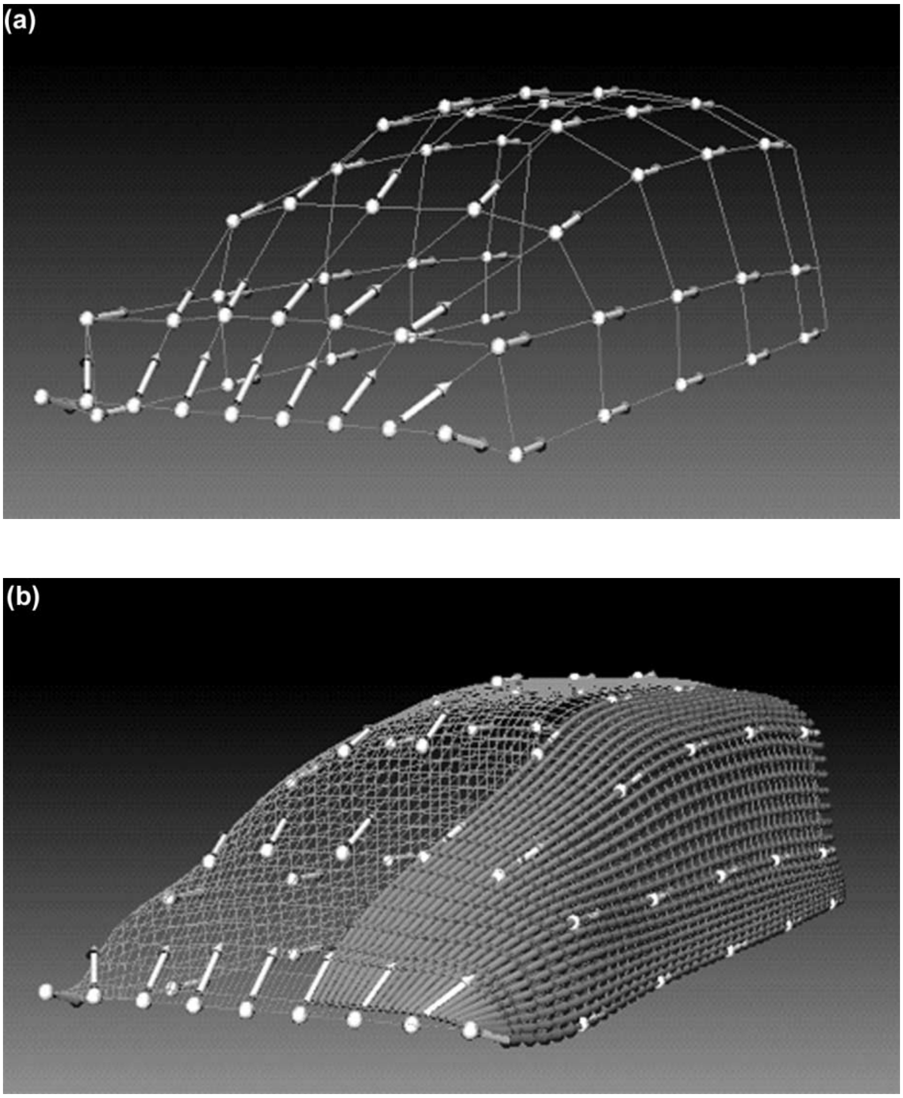


Fig. 11. Car model no. 2. (a) Surface defined with original mesh (subdivision depth = 0). Yellow arrows are tangent vectors corresponding to vertices of the subdivision surfaces on the Gaussian sphere; (b) with subdivision surface (subdivision depth = 3). Red arrows are tangent vectors corresponding to refined vertices. Only half of the red arrows are shown.

To design a streamline, one must specify several unit tangent vectors. Specifying a unit vector is equivalent to specifying a point on the Gaussian sphere. Tools have been provided in our prototype system for the user to select and drag a point of the Gaussian sphere (Fig. 5) or select and drag an arrow (Fig. 7). The arrows have specially shaped heads according to their index numbers. Hence, the user can select any of them, even though several arrows are oriented in the same direction.

Slider tools are also provided for the user to change the x, y, or z coordinate of a tangent vector so as to keep its unit length and to rotate a tangent vector about an arbitrary axis. The user is also allowed to change the length coefficient of a tangent vector and, hence, the s_i value, as mentioned in the previous section.

The initial curves can be specified by any type of free-form curve representation. In our system, a unit QI curve [10] is used as the initial curve, because a QI curve is *streamline modeling* based and its shape is controlled by the tangent vectors. Using a QI curve as the initial curve does not hamper the performance of the system even though it takes slightly more processing power since only one initial curve is needed to define a surface.

Figs. 8–10 show the refining process of streamline modeling with subdivision surfaces. The original mesh of the subdivision surface has 9 × 6 vertices (Fig. 8(a)). Surfaces defined with refined subdivision surfaces are shown in Fig. 8(b), (subdivision depth = 1), Fig. 9(a), (subdivision depth = 2), and Fig. 9(b), (subdivision depth = 3). A shaded image of the surface shown in Fig. 9(b) is shown in Fig. 10. Fig. 11 shows a reshaped model.

7. Conclusions

New approximation and interpolation methods for points on the Gaussian sphere, which correspond to unit tangent vectors, have been developed. Subdivision surfaces generated from polygons on the Gaussian sphere are used in the approximation and interpolation process.

New streamline modeling techniques using subdivision surfaces on the Gaussian sphere have also been proposed. A prototype design system based on these techniques is developed. The new techniques inherit virtues of the original subdivision system and provide a new operability to streamline modeling. Advantages of the new techniques include real-time performance achieved by integration process over coarse subdivision surfaces and easiness in generating high quality fair, smooth surfaces based on streamline modeling.

Several areas require further study in the future. One interesting topic is to deal with arbitrary topology in streamline modeling. This is one of the most important properties of subdivision surfaces. One can also think about merger and separation of streamlines. Currently, streamlines can be specified in fixed parameter direction. However, it seems possible to switch parameters from one (s) to the other (t). Future research is necessary on where and how should the parameters be switched. Research in this area could lead to the design of arbitrary topology faces.

Another topic of interest is the multi-resolution representation for streamline modeling. This is a desirable, and yet difficult, job because of the integration process.

References

[1] Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-aided Design 1978;10(6):350–5.

[2] Chaikin G. An algorithm for high-speed curve generation. Computer Graphics and Image Processing 1974;3:346–9.

[3] DeRose T, Kass M, Truong T. Subdivision surfaces in character animation. Computer Graphics (Proc. Of SIGGRAPH '98) July; 1998;32:85–94.

[4] Doo D, Sabin M. Behaviour of recursive division faces near extraordinary points. Computer-aided Design 1978:10(6)356–60.

[5] Dyn N, Levin D. Interpolating subdivision schemes for the generation of curves and surfaces. In: Hausmann W, Jetter K,, editors. Multivar. approx. and interp, Basel: Birkhauser Verlag, 1990.

[6] Farin G. Curves and surfaces for computer aided geometric design. Academic Press, 1988.

[7] Kantor IL, Solodovnikov AS. Hypercomplex numbers. 1989: Springer-Verlag, 1989.

[8] Kim M-J, Kim M-S, Shin SY. A general construction for unit quaternion curves with simple high order derivatives. Computer Graphics (Proc. of SIGGRAPH '95), 1995. pp. 369–376.

[9] Kobbelt L. Interpolary subdivision on open quadrilateral nets with arbitrary topology. Computer Graphics Forum 15 (1996), Eurographiscs '96, pp. 409–420.

[10] Miura KT, Sakiyama N, Kaneko T, Kubo. Streamline modeling: designing free-form surfaces by blending tangent vectors. Sixth IFIP WG5.2 Workshops on Geometric Modeling, 1998. pp. 168–179.

[11] Miura KT. Unit quaternion integral curve: a new type of fair free-form curves. Computer Aided Geometric Design 2000;17(1):39–58.

[12] Nasri AH. Polyhedral subdivision methods for free-form surfaces. Transation on Graphics 1987;6(1):29–73.

[13] Sakiyama N, Miura KT, Takahashi T, Kaneko T, Kubo T. Fair curve and surface design system using tangent control. Pacific Graphics '98, 1998. pp. 27–36.

[14] Schroder P, Zorin D. Subdivision for modeling and animation, SIGGRAPH 98 Course Note, 1998.

Kenjiro T. Miura is an Associate Professor of the Department of Mechanical Engineering, Shizuoka University in Japan where he joined the faculty in 1997. He received a BEng and an MEng in precision machinery from the University of Tokyo in Japan in 1982 and 1984, respectively, and a PhD in mechanical engineering from Cornell University in 1991. His research interests include computer aided geometric design, and mesh generation.

Lazhu Wang is a Foreign Researcher at Shizuoka University in Japan. He received his PhD from Beijing University of Aeronautics and Astronautics in China. His research fields are in geometric modeling and the development of CAD/CAM systems.

Fuhua (Frank) Cheng is Professor of Computer Science and Supervisor of the Graphics and Geometric Modeling Lab at the University of Kentucky where he joined the faculty in 1986. He is also an Adjunct Professor of Applied Mathematics at the Shandong University of Technology, Jinan, China. He received a BS and an MS in mathematics from the National Tsing Hua University in Taiwan in 1973 and 1975, respectively, an MS in mathematics, an MS in computer science from the Ohio State University in 1978, 1980, and 1982, respectively. Dr Cheng has held visiting positions at the University of Tokyo and the University of Aizu, Japan. His research interests include computer aided geometric modeling, computer graphics, parallel computing in geometric modeling and computer graphics, and collaborative CAD.