Graphical Models 72 (2010) 1–6

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Hyperbolic polynomial uniform B-spline curves and surfaces with shape parameter

Liu Xumin [a,*], Xu Weixiang [b], Guan Yong [a], Shang Yuanyuan [a]

[a] College of Information Engineering, Capital Normal University, Beijing 100048, China
[b] School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

ARTICLE INFO

Article history:
Received 31 December 2006
Received in revised form 5 July 2009
Accepted 13 October 2009
Available online 17 October 2009

Keywords:
Computer-aided geometric design
Computer graphics
Surface modeling
Hyperbolic polynomial uniform B-spline with shape parameter

ABSTRACT

In this paper, we study in depth a new modeling method for parameter curves and surface-hyperbolic polynomial uniform B-spline surfaces with shape parameter. Based on this model, we give an example of free-form curve modeling, and analyze the effect that different shape parameters have on the curve shape. In light of computer graphics theories, we develop a space free-form surface modeling system on the Microsoft Windows operating system. By means of the prototype system, we perform free-form surface modeling, give examples for its application, and discuss how the adjustment of the shape parameter affects the shape of free-form surfaces.

1. Introduction

Adjusting shape parameters to amend the shape of curves or surfaces is a hot topic in the study of computer-aided geometrical design. Research on how change of shape parameters affects spline curves, and surfaces is necessary to increase the curve and surface modeling flexibility. In engineering, the weight of NURBS model can be used to adjust the shape of curves and surfaces; however, this kind of adjusting method has some limitations. For instance, too many weights make the choice irregular and make the model unable to represent transcendental curves [1].

Previous studies propose several spline models with shape parameters, with the attempt to improve the adjustment of the shape of curves or surfaces and to obtain satisfactory graphics. These models are aimed to inherit the merit and overcome the limit of NURBS model. Some of these models include cubic Beta spline [2,3], C–B spline [4,5], uniform trigonometric polynomial B-spline [1], uniform hyperbolic polynomial B-spline [6], etc. In more recent literature, models with shape parameter are proposed, such as uniform B-spline with shape parameter [7], hyperbolic polynomial uniform B-spline with shape parameter (HPUBSP) [8] and trigonometric polynomial uniform B-spline with shape parameter [9]. Most of these models replace polynomials of B-spline curve with nonpolynomials, so that rational representation of the model is avoided and the properties of the polynomial B-spline basis are preserved. Apart from the models mentioned earlier, there are also some mathematical models [10–12].

The models with shape parameters share something in common in that they are all expansion of B-splines and have shape parameters, which can be used to slightly modify curves and surfaces. When the shape parameter is assigned certain values, the basic function of these models has properties like positivity, local support, partition of unity, linear independence, symmetry and continuity, and its curve has properties including convex, geometry invariability, locality and symmetry. Besides, changing the value of shape parameters may alter how close the curves mentioned earlier approach their control polygons.

Meanwhile, these models differ due to their different basic functions. (1) In terms of curve features, Beta spline models can only approximate circles; C–B spline can represent circle and ellipse segments with great accuracy; uniform B-splines with shape parameter can only approach

* Corresponding author.
E-mail address: liuxumin@126.com (L. Xumin).

1524-0703/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.gmod.2009.10.001

circle segments but cannot represent them; trigonometric polynomial uniform B-spline models with shape parameter can accurately represent circles, ellipses and helixes. (2) In terms of curve continuity, splines in tension have $[1, t, \sinh \alpha t, \cosh \alpha x t]$ as their bases; exponential splines, which take $[1, t, e^{\alpha t}, e^{-\alpha t}]$ as their bases, have equal value as splines in tension. Splines in tension can represent hyperbolas accurately but cannot represent high degree polynomial curve. However, k-order ($k \geqslant 2$) hyperbolic polynomial uniform B-spline model can represent high order models and achieve C^{k-2} continuity, which makes the mathematical models more comprehensive theoretically and makes their application in CAGD surface modeling more convenient. (3) In terms of the number of shape parameters and adjusting scope, apart from Beta spline curves that have two adjustable shape parameters, the other spline curves have only one. However, their adjusting scopes vary. For C–B spline curves, the adjusting scope of shape parameter is $0 < \alpha \leqslant \pi$ [4]; for fourth-order B-spline curves with shape parameter, the adjusting scope is $-2 \leqslant \lambda \leqslant 1$ [7]; for fourth-order trigonometric polynomial uniform B-spline with shape parameter curve, the adjusting scope is $-1 \leqslant \lambda \leqslant 1$ [9]; for fourth-order HPUBSP curve, the adjusting scope is $-\mathrm{cth}^2(1/2) \leqslant \lambda \leqslant \mathrm{cth}^2(1/2)$ [8]. Due to these different properties, we need to choose the proper model according to our need and requirement.

Compared with the NURBS model, the advantages of hyperbolic polynomial B-spline model lie in that one shape parameter is needed to slightly adjust the shape parameter, the designer is not required much mathematical knowledge and the adjusting range is easier to choose and control.

In this paper, we discuss the HPUBSP model, which is proposed in Ref. [8]. Based on the mathematical definition of the model, we go deep into the formula derivation, design the modeling example and analyze the effect of the shape parameter on the curve shape. Furthermore, using computer graphics theory, we design a series of surface modeling arithmetic. We also develop a space free-form surface modeling system (SFSMS) using the VC++ programming language. Through an example of space free-form surface modeling, we analyze how the adjustment of the shape parameter changes the surface shape and present the adjustment range of the shape parameter.

2. HPUBSP model

The HPUBSP curve mentioned in Ref. [8] can be adjusted by changing the value of the shape parameter if the control polygon is kept unchanged. The curve can be adjusted to be close to its control polygon by changing the value of the shape parameter. Different C^2 consecutive curves can be obtained at different positions according to their shape parameter values. Each curve segment is built by four-control vertices one after the other. This is not only simple structure but the same structure as Cubic B-spline curve. So, it can be used conveniently.

2.1. The construction of basis function

Basis functions are an important component of the geometry of spline curve, so it is necessary to discuss the basis function of HPUBSP surfaces.

Definition 1. The construction of HPUBSP is as follows:

Suppose that the nonnegative function $h(\lambda, t), t \in [0, 2]$ meets the following conditions:

(1) $h(\lambda, 0) = h(\lambda, 2) = 0, \quad t \in [0, 2], h(\lambda, t) = h(\lambda, 2 - t)$
(2) $h(\lambda, t)$ is a C^0 continuous piecewise quadratic function.

$$h(\lambda) = a \sinh t + b \sinh 2t, \qquad t \in [0, 1]$$

(3) $\int_0^2 h(\lambda, t) dt = 1$

Let $t = 2 - t'$ in (2). Through the substitution of variables, we get the function $h(\lambda, t')$, in which $t' \in [1, 2]$.

From (1) and (2), we get:

$$h(\lambda, t) = \begin{cases} a \sinh t + b \sinh 2t & (0 \leqslant t \leqslant 1) \\ a \sinh(2 - t) + b \sinh 2(2 - t) & (1 \leqslant t \leqslant 2) \end{cases}$$

Replacing the value of $h(\lambda, t)$ in (3), we get:

$$\int_0^2 h(\lambda, t) dt = \int_0^1 (a \sinh t + b \sinh 2t) dt + \int_1^2 (a \sinh(2 - t) + b \sinh 2(2 - t)) dt = 1a \frac{(e-1)^2}{e} + b \frac{(e^2-1)^2}{2e^2} = 1$$

Let $a = \frac{e(1+\lambda)}{(e-1)^2}$, then $b = -\frac{2e^2\lambda}{(e^2-1)^2}$.

Let the initial value of i be 0, we define the second-order HPUBSP in the first nonzero interval as follow:

$$H_{0,2}(t) = \begin{cases} \frac{e}{(e-1)^2}\left[(1+\lambda)\sinh t - \frac{2e}{(e+1)^2}\lambda \sinh 2t\right] & (0 \leqslant t \leqslant 1) \\ \frac{e}{(e-1)^2}\left[(1+\lambda)\sinh(2-t) - \frac{2e}{(e+1)^2}\lambda \sinh(2-t)\right] & (1 \leqslant t < 2) \\ 0 & \text{elsewhere} \end{cases} \tag{1}$$

where $-cth^2 \frac{1}{2} \leqslant \lambda \leqslant cth^2 \frac{1}{2}$

We define the second-order HPUBSP in the ith nonzero interval as follows:

$$H_{i,2}(t) = H_{0,2}(t - i), \qquad t = 0, \pm 1, \pm 2, \ldots$$

If $k \geqslant 3$, then the k-order HPUBSP in the first nonzero interval is defined as follows:

$$H_{0,k}(t) = \int_{t-1}^{t} H_{0,k-1}(x) dx, \quad i = 0, \pm 1, \pm 2, \ldots \tag{2}$$

We define the k-order HPUBSP in the ith nonzero interval as follows:

$$H_{i,k}(t) = H_{0,k}(t - i) \tag{3}$$

From the definition mentioned earlier, we can get the basis function of fourth-order HPUBSP as follows:

$$H_{0,4}(t) = \begin{cases} a((1+\lambda)\sinh(t) - \frac{1}{2}b\lambda\sinh(2t)) - a(1+\lambda-b\lambda)t & (0 \leqslant t < 1) \\ -2a((1+\lambda)\sinh(t-1) - \frac{1}{2}b\lambda\sinh(2t-2)) & \\ +a((1+\lambda)\sinh(2-t) - \frac{1}{2}b\lambda\sinh(4-2t)) & \\ +a(1+\lambda-b\lambda)(t-1) + 2a((1+\lambda)\cosh(1) & \\ -b\lambda\cosh(2))(t-1) - a(1+\lambda-b\lambda) & (1 \leqslant t < 2) \\ -2a((1+\lambda)\sinh(3-t) - \frac{1}{2}b\lambda\sinh(6-2t)) & \\ -a(\frac{1}{2}b\lambda\sinh(2t-4) - (1+\lambda)\sinh(t-2)) & \\ +a(b\lambda - (1+\lambda))(t-2) + 2a((1+\lambda)\cosh(1) & \\ -b\lambda\cosh(2))(3-t) & (2 \leqslant t < 3) \\ a((1+\lambda)\sinh(4-t) - \frac{1}{2}b\lambda\sinh(2(4-t))) - a(b\lambda & \\ -(1+\lambda))(t-1) + 3a(b\lambda - (1+\lambda)) & (3 \leqslant t < 4) \\ 0 & (\text{elsewhere}) \end{cases} \tag{4}$$