Graphical Models 70 (2008) 76–86

Contents lists available at ScienceDirect

Graphical Models

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

Polynomial splines over hierarchical T-meshes

Jiansong Deng *, Falai Chen, Xin Li, Changqi Hu, Weihua Tong, Zhouwang Yang, Yuyu Feng

Department of Mathematics, University of Science and Technology of China, 96 Jinzhai Road, Hefei 230026, Anhui, PR China

ARTICLE INFO

Article history:
Received 17 October 2006
Received in revised form 3 February 2008
Accepted 17 March 2008
Available online 8 April 2008

Keywords:
T-mesh
T-splines
NURBS
Local refinement
Surface fitting
Shape simplification

ABSTRACT

In this paper, we introduce a new type of splines—polynomial splines over hierarchical T-meshes (called *PHT-splines*) to model geometric objects. PHT-splines are a generalization of B-splines over hierarchical T-meshes. We present the detailed construction process of spline basis functions over T-meshes which have the same important properties as B-splines do, such as nonnegativity, local support and partition of unity. As two fundamental operations, cross insertion and cross removal of PHT-splines are discussed. With the new splines, surface models can be constructed efficiently and adaptively to fit open or closed mesh models, where only linear systems of equations with a few unknowns are involved. With this approach, a NURBS surface can be efficiently simplified into a PHT-spline which dramatically reduces the superfluous control points of the NURBS surface. Furthermore, PHT-splines allow for several important types of geometry processing in a natural and efficient manner, such as conversion of a PHT-spline into an assembly of tensor-product spline patches, and shape simplification of PHT-splines over a coarser T-mesh. PHT-splines not only inherit many good properties of Sederberg's T-splines such as adaptivity and locality, but also extend T-splines in several aspects except that they are only C^1 continuous. For example, PHT-splines are polynomial instead of rational; cross insertion/removal of PHT-splines is local and simple.

1. Introduction

A serious weakness with NURBS models is that they contain a large number of superfluous control points. T-splines [21] overcome this weakness by allowing T-junctions in the control meshes, and thus provide adaptiveness and flexibility in modeling geometric objects. In this paper, we introduce a new type of splines—polynomial splines over hierarchical T-meshes, called *PHT-splines*. A PHT-spline is a piecewise bicubic polynomial over a hierarchical T-mesh. The basis functions of PHT-splines have the same important properties as B-splines do, such as nonnegativity, local support and partition of unity. Thus PHT-splines are a generalization of B-splines over hierarchical T-meshes.

With the new splines, surface models can be constructed very efficiently and adaptively to fit open or closed meshes (with genus zero), where only linear systems of equations with a few unknowns are involved. Fig. 1 illustrates an example where the female head mesh has 19,231 points and 38,388 triangular faces. Provided the mesh parameterization is given, which is not time-consuming with an up-to-date parameterization algorithm, a PHT-spline surface can be constructed to fit the mesh model in about two seconds with an ordinary personal computer. Further examples in Section 5 show that the complexity of our surface fitting algorithm is almost linear with respect to the mesh size.

The surface fitting algorithm can be easily adapted to approximate a NURBS model with a PHT-spline. Thus conversion from NURBS models to PHT-splines is very effective. As a comparison, it is reported by Zheng that fitting a z-map model of about 10,000 points with T-splines takes about 30 min [24].

* Corresponding author.
E-mail addresses: dengjs@ustc.edu.cn (J. Deng), chenfl@ustc.edu.cn (F. Chen).

1524-0703/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.gmod.2008.03.001

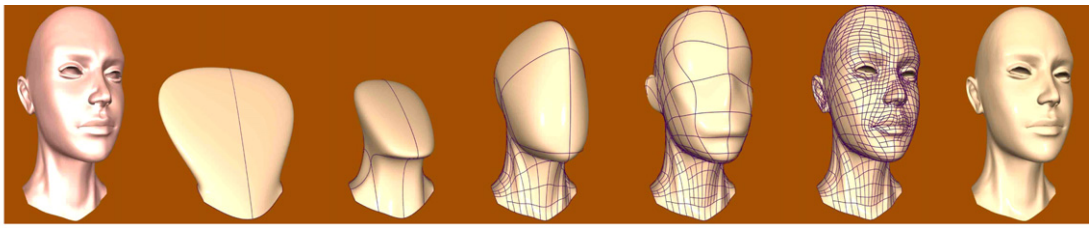

Fig. 1. Fitting a mesh model (courtesy of Open3DProject) with polynomial splines over a hierarchical T-mesh. The curves on the surfaces from b to f are the images of the hierarchical T-mesh.

Fig. 2 is another example which compares the NURBS model of an ear (from http://www.b3dscann.com/samples.htm) and the fitting results of the ear model with PHT-splines. Fig. 2a shows a part of the NURBS model which has wrinkles, gaps and self-intersections, while Fig. 2b is a part of the PHT-spline model which looks much smoother and fairer than the original NURBS model.

The local refinement of PHT-splines is another important operation in PHT-spline theory. Due to the nature of hierarchical T-meshes, the local refinement of PHT-splines can be achieved by *cross insertion*, i.e., dividing a cell into four subcells with a cross. The inverse process is called *cross removal*. Many geometry processing algorithms such as surface fitting, conversion between NURBS and PHT-splines, conversion of PHT-splines to tensor-product (TP for short) splines, shape simplification, etc. all heavily depend on cross insertion/removal operation. We will present a local and simple cross insertion/removal algorithm and apply it in several geometry processing algorithms.

1.1. Related work

In geometric modeling, the representation of surfaces is a fundamental research topic. Computer graphics and computer aided design communities prefer parametric surfaces, especially TP B-spline surfaces. However, TP B-spline surfaces suffer from the weakness that the control points must lie topologically in a rectangular grid. Thus local refinement with a TP B-spline surface is difficult. To solve this problem, [7] invented hierarchical B-splines by introducing two concepts: local refinement using an efficient representation and multi-resolution editing. Other related work on hierarchical B-splines includes CHARMS [11], multilevel B-splines [15,16], TP splines with knot segments [22]. An innovation along this direction is the invention of T-splines by Sederberg [21]. A T-spline is a point-based spline defined over a T-mesh. For every vertex, a basis function is defined. Each of the basis functions comes from some TP spline space. T-splines support many valuable operations within a consistent framework, and most importantly, they can eliminate most superfluous control points in NURBS representations and they permit local refinement. However, the local refinement of T-splines depends on the structure of the T-mesh. In the worst case, the algorithm would extend all partial rows of control points to cross the entire surface [20]. The reason leading to this problem is that the spline function over each cell of the T-mesh is not a polynomial, but a piecewise function. On the other hand, since the basis functions do not form a partition of unity, T-splines are rational, which leads to complicated computations in subsequent geometry operations. Sederberg [20] put forward the problem on how to construct polynomial T-splines.

1.2. Overview

The main contribution of this paper is the invention of a new type of splines—polynomial splines over hierarchical T-meshes (called PHT-splines) in geometric modeling. PHT-splines not only inherit the main properties of T-splines such as adaptivity, but also exhibit several advantages over T-splines. For example, unlike T-splines, PHT-splines are polynomial instead of rational; The local refinement algorithm of PHT-splines is local and simple. The conversion between NURBS and PHT-splines is very fast, while conversion between NURBS and T-splines is a

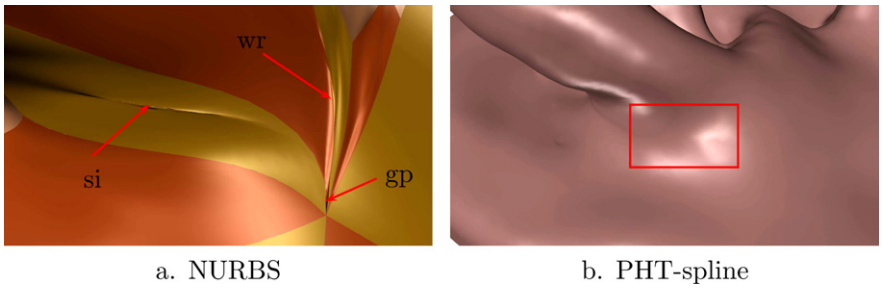

Fig. 2. Comparison of fitting results (si, self-intersection; gp, gap; wr, wrinkle). (a) The original NURBS model. (b) The PHT-spline fitting model. The part in the red frame in (b) corresponds to (a).

bottleneck of T-splines in practical applications. Compared with T-splines and hierarchical B-splines, PHT-splines are only C^1 continuous. However, PHT splines have a set of basis functions, which is a necessity in some theoretical analysis and applications, while hierarchical B-splines have a redundant set of 'basis functions'. On the other hand, hierarchical B-splines require a very special hierarchical T-mesh structure due to their refinement scheme, while PHT-splines work over arbitrary hierarchical T-meshes.

The remainder of the paper is organized as follows. In Section 2, we introduce polynomial spline spaces over T-meshes and the dimension formula proved in [2]. Section 3 describes in detail the construction of the basis functions of a spline space over a hierarchical T-mesh. The properties of the basis functions are discussed and PHT-spline surfaces are introduced. Section 4 presents two important operations—cross insertion and removal in PHT-spline theory. In Section 5, we propose a surface fitting scheme to fit open and closed meshes of genus-zero with PHT-splines. In Section 6, some geometry processing algorithms such as shape conversion and shape simplification are discussed. Section 7 concludes the paper with a summary and some future work.

2. Polynomial splines over T-meshes

In this section, we briefly review the definition of T-meshes, and then introduce polynomial spline spaces over T-meshes. The dimension formula of the spline space is given.

2.1. T-meshes

Given a rectangular domain, a *T-mesh* is a partition of the domain and it is basically a rectangular grid that allows T-junctions [2,21]. It is assumed that the end points of each grid line in the T-mesh must be on two other grid lines, and each cell or facet in the grid must be a rectangle. Fig. 3 shows an example of a T-mesh. A grid point in a T-mesh is also called a *vertex* of the T-mesh. If a vertex is on the boundary of the domain, then is called a *boundary vertex*. Otherwise, it is called an *interior vertex*. For example, $\mathbf{b}_i$, $i = 1, \ldots, 10$, in Fig. 3 are boundary vertices, while all the other vertices $\mathbf{v}_i$, $i = 1, \ldots, 5$, are interior vertices. Interior vertices have two types. One is crossing, for example, $\mathbf{v}_2$ in Fig. 3; and the other is T-junctional, for example, $\mathbf{v}_1$ in Fig. 3. They are called *crossing vertices* and *T-vertices*, respectively. The line segment connecting two adjacent vertices on a grid line is called an *edge* of the T-mesh.

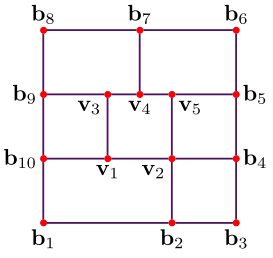

Fig. 3. An example of a T-mesh.

2.2. Hierarchical T-meshes

Instead of considering general T-meshes, we restrict our attention to hierarchical T-meshes in the paper, since such meshes do not lose the main property—adaptivity of general T-meshes.

A *hierarchical T-mesh* is a special type of T-mesh which has a natural level structure. It is defined in a recursive fashion. One generally starts from a TP mesh (level 0). From level k to level $k+1$, one subdivide a cell at level k into four subcells which are cells at level $k+1$. For simplicity, we subdivide each cell by connecting the middle points of the opposite edges with two straight lines. Fig. 4 illustrates the process of generating a hierarchical T-mesh.

Hierarchical T-meshes have appeared in many research disciplines in computer science, computational mathematics, and so on. For example, adaptive finite elements [25, Chapter 15] and hierarchical B-splines [7] are defined over hierarchical T-meshes.

2.3. Spline spaces over T-meshes

Given a T-mesh $\mathscr{T}$, $\mathscr{F}$ denotes all the cells in $\mathscr{T}$ and Ω the region occupied by all the cells in $\mathscr{T}$. Define

$$\mathscr{S}(m,n,\alpha,\beta,\mathscr{T}) := \{s(x,y) \in C^{\alpha,\beta}(\Omega) | s(x,y)|_\phi \in \mathbb{P}_{mn} \text{ for any } \phi \in \mathscr{F}\},$$

where $\mathbb{P}_{mn}$ is the space of all the polynomials of bi-degree (m,n), and $C^{\alpha,\beta}(\Omega)$ is the space consisting of all the bivariate functions which are continuous in Ω with order α along x direction and with order β along y direction. It follows that $\mathscr{S}(m,n,\alpha,\beta,\mathscr{T})$ is a linear space. It is called the *spline space over the given T-mesh* $\mathscr{T}$.

For a given T-mesh $\mathscr{T}$, it is easy to see that T-splines (in non-rational form) and hierarchial B-splines form a proper subset of the spline space $\mathscr{S}(3,3,2,2,\mathscr{T}')$ in general, where $\mathscr{T}'$ is a new T-mesh obtained by inserting some edges into $\mathscr{T}$. In this sense, the splines over a T-mesh are a generalization of T-splines and hierarchical B-splines. This generalization makes full use of the current domain partition and provides more flexibility in geometric modeling than T-splines and hierarchical B-splines in practice.

Theorem 4.2 in [2] provides a dimension formula for the spline space $\mathscr{S}(m,n,\alpha,\beta,\mathscr{T})$ for $m \geqslant 2\alpha+1$ and $n \geqslant 2\beta+1$. Specifically, we have

$$\dim \mathscr{S}(3,3,1,1,\mathscr{T}) = 4(V^b + V^+), \tag{1}$$

where V^b and V^+ represent the number of boundary vertices and interior crossing vertices, respectively. The current paper focuses on the spline space $\mathscr{S}(3,3,1,1,\mathscr{T})$ for a hierarchical T-mesh $\mathscr{T}$, though the results are also valid over general spline spaces $\mathscr{S}(2\alpha+1,2\beta+1,\alpha,\beta,\mathscr{T})$ for $\alpha, \beta \geqslant 1$.

The dimension formula gives us a hint on how to construct basis functions for the spline space, i.e., each boundary vertex or interior crossing vertex associates with four basis functions. This observation will be further explored in the construction of the basis functions in the next

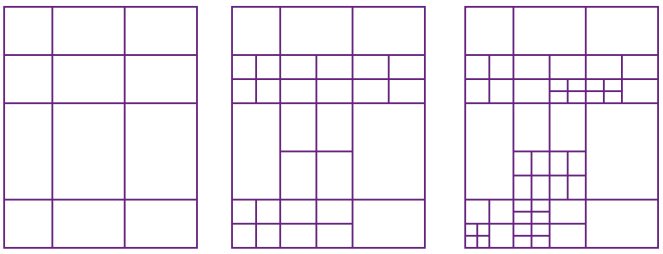

Fig. 4. A hierarchical T-mesh.

section. Hence later on if a vertex is a boundary vertex or an interior crossing vertex, we call it a *basis vertex*.

3. Basis functions of PHT-splines

In this section, we construct a set of basis functions over hierarchical T-meshes which have the same important properties as B-splines do, such as nonnegativity, local support and partition of unity. The basis functions directly lead to the construction of polynomial splines over hierarchical T-meshes.

Before constructing the basis functions, we point out some basic facts. Let $t_0, t_0 = t_1, t_1 < t_2, t_2 < \ldots < t_{p-2}, t_{p-2} < t_{p-1}, t_{p-1} = t_p, t_p$ be the knot vector of a C^1 continuous cubic spline, where every interior knot is with multiplicity two. For any interior knot t_i, only two B-spline basis functions have support $[t_{i-1}, t_{i+1}]$. These two basis functions are associated with knot vectors $(t_{i-1}, t_{i-1}, t_i, t_i, t_{i+1})$ and $(t_{i-1}, t_i, t_i, t_{i+1}, t_{i+1})$, respectively. Except for the two basis functions, all the other B-spline basis functions and their derivatives vanish at t_i. Generalize this fact to C^1 continuous bicubic TP B-splines, for any interior knot vertex (s_i, t_j), where s_i and t_j are the knots in two directions respectively, only four B-spline basis functions have support $[s_{i-1}, s_{i+1}] \times [t_{i-1}, t_{i+1}]$. We say these four basis functions are *associated with* the vertex (s_i, t_j). The function values, the first order partial derivatives and the mixed derivatives of all the other B-spline basis functions vanish at (s_i, t_j).

The basic strategy to construct the basis functions for the spline space $\mathscr{S}(3,3,1,1,\mathscr{T})$ is level by level. Denote the T-mesh at level k by $\mathscr{T}_k$. For level 0, the standard TP B-spline basis functions are taken to be the basis functions over the initial TP mesh $\mathscr{T}_0$, where every basis vertex is associated with four basis functions according to the analysis in the beginning of the subsection. Obviously, these basis functions have the properties such as nonnegativity, local support and partition of unity. Now suppose the basis functions $\{b_j^k(u,v)\}$, $j = 1, \ldots, d_k$ at level k have been constructed, and these basis functions has the same properties. Then the basis functions at level $k+1$ consist of two parts. The first part comes from the modification of the basis functions at level k. The second part comes from the basis functions associated with the new basis vertices at level $k+1$. The basis functions generated at level $k+1$ have the same properties as the previous level.

Now we describe the details of constructing the basis functions at level $k+1$. For brevity, a basis function is represented by specifying its 16 Bézier ordinates in every cell within the support of the basis function.

3.1. Modification of the basis functions at level k

Suppose, among all the cells at level k, the cells θ_i^k, $i = 1, \ldots, C_k$, are subdivided. For each j, if the basis function $b_j^k(u,v)$ does not vanish in some cells of $\{\theta_i^k\}$, then subdivide $b_j^k(u,v)$ into these cells at level $k+1$ according to formula (14.15) in [4]. See Fig. 5a and b for an example. It

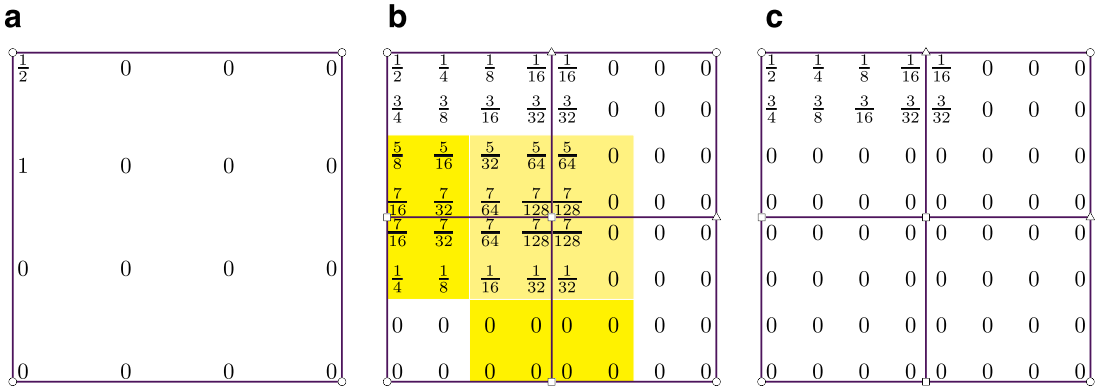

Fig. 5. Modification of a basis function. (a) A function $f(s,t)$ is defined with the given Bézier ordinates. (b) $f(s,t)$ is subdivided into four subcells. The square shaped vertices are new basis vertices. (c) The Bézier ordinates around the new basis vertices are reset to zero. Though $f(s,t)$ is changed, it still lies in $\mathscr{S}(3,3,1,1,\mathscr{T}_{k+1})$.

should be noted that the function $b_j^k(u,v)$ has not changed, but is now defined over the mesh $\mathscr{T}_{k+1}$.

From $\mathscr{T}_k$ to $\mathscr{T}_{k+1}$, some new basis vertices appear. Denote them as ξ_i^{k+1}, $i=1,\ldots,V_{k+1}$. In each cell at level $k+1$, the 16 Bézier ordinates are divided into four parts. Each part is associated with a cell corner vertex as shown in Fig. 6. Then all the basis functions $\{b_j^k(u,v)\}$ at level k are modified to $\{\bar{b}_j^k(u,v)\}$ in the following fashion: for each j, reset all the associated Bézier ordinates with the new basis vertices to zero. See Fig. 5c for an example, where there are three new vertices.

One can show that $\{\bar{b}_j^k(u,v)\}$ are in $\mathscr{S}(3,3,1,1,\mathscr{T}_{k+1})$.

For example, as shown in Fig. 7, Column (a) presents the same parts of $\mathscr{T}_k$ and $\mathscr{T}_{k+1}$, respectively. Here, from $\mathscr{T}_k$ to $\mathscr{T}_{k+1}$, a cross is inserted into a cell in the selected part. Columns (b) shows the shapes of a basis function associated with the crossing vertex $(1,1)$ before and after the modification, respectively.

3.2. Adding new basis functions at level $k+1$

For each new basis vertex (s_i,t_i), its neighboring cells must lie in a position as shown in Fig. 8. Then the four basis functions associated with (s_i,t_i) are defined to be $M_{ik}^3(s)N_{il}^3(t)$, $k,l=1,2$, where $M_{i1}^3(s)$, $M_{i2}^3(s)$, $N_{i1}^3(t)$, and $N_{i2}^3(t)$ are the cubic B-spline basis functions associated with the knot vectors $(s_{i-1}, s_{i-1}, s_i, s_i, s_{i+1})$, $(s_{i-1}, s_i, s_i, s_{i+1}, s_{i+1})$, $(t_{i-1}, t_{i-1}, t_i, t_i, t_{i+1})$ and $(t_{i-1}, t_i, t_i, t_{i+1}, t_{i+1})$, respectively. If (s_i,t_i) is a boundary vertex, then either $s_{i-1}=s_i$, or $s_{i+1}=s_i$, or $t_{i-1}=t_i$ or $t_{i+1}=t_i$.

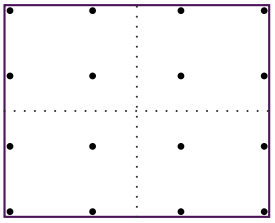

Fig. 6. The Bézier ordinates are associated with the four corner vertices.

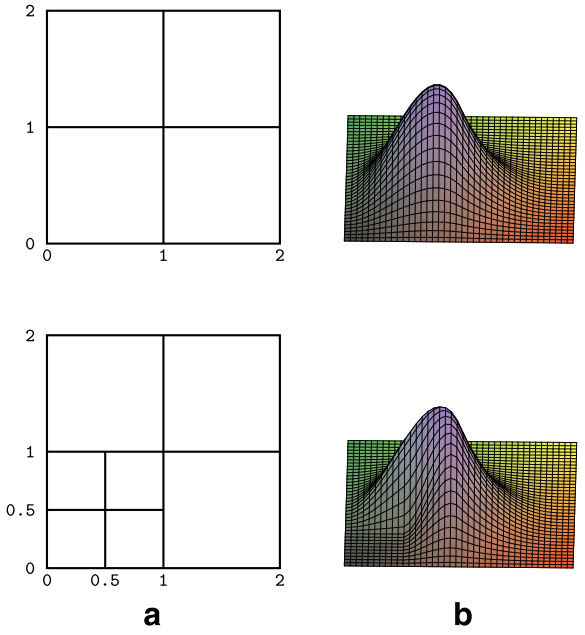

Fig. 7. Modification of a basis function.

These four basis functions are in $\mathscr{S}(3,3,1,1,\mathscr{T}_{k+1})$ and they have the same support $[s_{i-1},s_{i+1}]\times[t_{i-1},t_{i+1}]$. Denote all the basis functions associated with the new basis vertices as $\{\tilde{b}_j^{k+1}(u,v)\}_{j=1}^{4V_{k+1}}$.

From the previous construction process, it follows that $\{\tilde{b}_j^{k+1}(u,v)\}_{j=1}^{4V_{k+1}}$ and $\{\bar{b}_j^k(u,v)\}_{j=1}^{d_k}$ are linearly independent. Since $d_{k+1}=d_k+4V_{k+1}$ according to the dimension formula (1), these functions form a basis for the spline space $\mathscr{S}(3,3,1,1,\mathscr{T}_{k+1})$, and are denoted as $\{b_j^{k+1}(u,v)\}_{j=1}^{d_{k+1}}$.

3.3. Properties of the basis functions

The basis functions constructed in the last subsection have the following properties:

- *Nonnegativity:* This follows from the fact that each basis function has nonnegative Bézier ordinates.
- *Partition of unity:* Define

$$s_1(u,v)=\sum_{j=1}^{4V_{k+1}}\tilde{b}_j^{k+1}(u,v),\quad s_2(u,v)=\sum_{j=1}^{d_k}\bar{b}_j^k(u,v).$$

Check the Bézier ordinates of $s_1(u,v)$ and $s_2(u,v)$ in every cell. For $s_1(u,v)$, the Bézier ordinates associated with the new basis vertices are one, and all the others are zero; for $s_2(u,v)$, the Bézier ordinates associated with the new basis vertices are zero, and all the others are one. Hence it follows that $s_1(u,v)+s_2(u,v)\equiv 1$.

- *Local support.* From the construction of the basis functions, the support for each basis function is inside a rectangle, and its size decreases level by level. In fact, the rectangular support for each basis function has the following properties: (1) It is bounded by edges in the mesh. (2) Suppose the basis vertex associating with the basis function is **v**. If **v** is an interior vertex, then it lies in the interior of the rectangle. Otherwise it lies on the boundary of the rectangle. (3) The rectangle has the minimal size among all the rectangles with the previous two properties.

Next we study another important property of the basis functions. For any basis function $b(u,v)$, we define a *geometric information* (the function value, the first order

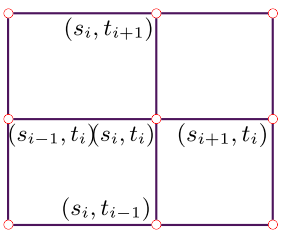

Fig. 8. A new basis vertex and its neighboring cells.

partial derivatives and the mixed partial derivative) operator by

$$\mathscr{L}b(u,v) = (b(u,v), b_u(u,v), b_v(u,v), b_{uv}(u,v)). \quad (2)$$

It follows that $\mathscr{L}(\cdot)$ is a linear operator and satisfies the following properties:

- For any basis function $b(u,v)$ and any basis vertex (u_0,v_0) in $\mathscr{T}$, $\mathscr{L}b(u_0,v_0) = \mathbf{0}$ holds for all the basis functions, except for the four basis functions associated with the basis vertex (u_0,v_0).
- For a basis vertex (u_0,v_0) which appears in the hierarchical T-mesh since level k_0, $\mathscr{L}b_j^k(u_0,v_0)$ remains unchanged for any $k \geqslant k_0$. Here $b_j^k(u,v)$ is a basis function constructed at level k.

This is true since the Bézier ordinate modification happens only around the new basis vertices. These observations are important for the surface fitting algorithm developed in Section 5.

3.4. PHT-spline surfaces

Let $\mathscr{T}$ be a hierarchical T-mesh, and $\{b_j(u,v)\}$, $j = 1,\ldots,d$ be the basis functions constructed in Section 3. Then the polynomial spline surface over $\mathscr{T}$ (called *PHT-spline surface*) is defined by

$$\mathbf{S}(u,v) = \sum_{j=1}^{d} \mathbf{C}_j b_j(u,v), \quad (u,v) \in [0,1]\times[0,1], \quad (3)$$

where $\mathbf{C}_j$ are control points. PHT-spline surfaces have similar properties with B-spline surfaces such as convex-hull property, affine invariant, local support, etc.

To efficiently manipulate and evaluate a PHT-spline surface, one should maintain the Bézier representation of the PHT-spline surface in every cell, since the basis functions are represented in Bézier forms in every cell. The geometry processing algorithms in later sections rely on such representation.

In the following we discuss how to determine the control points of a PHT-spline surface according to its geometric information at the basis vertices. Since the operator $\mathscr{L}(\cdot)$ in Eq. (2) is linear, for any fixed basis vertex (u_0,v_0) with which four basis functions with indices j_1,j_2,j_3,j_4 are associated, one has

$$\mathscr{L}\mathbf{S}(u_0,v_0) = \sum_{j=1}^{d} \mathbf{C}_j \mathscr{L}b_j(u_0,v_0) = \sum_{i=1}^{4} \mathbf{C}_{j_i} \mathscr{L}b_{j_i}(u_0,v_0) = \mathbf{C}\cdot\mathbf{b},$$

where $\mathbf{C} = (\mathbf{C}_{j_1}, \mathbf{C}_{j_2}, \mathbf{C}_{j_3}, \mathbf{C}_{j_4})$ is a 3×4 matrix, $\mathbf{b} = (\mathscr{L}b_{j_1}(u_0, v_0), \mathscr{L}b_{j_2}(u_0,v_0), \mathscr{L}b_{j_3}(u_0,v_0), \mathscr{L}b_{j_4}(u_0,v_0))$ is a 4×4 matrix, and $\mathscr{L}\mathbf{S}(u_0,v_0)$ is the geometric information of $\mathbf{S}(u,v)$ at (u_0,v_0).

Next we show how to obtain the matrix $\mathbf{b}$, and we use the computation of $\mathscr{L}b_{j_1}(u_0,v_0)$ as an illustration. 9, the four neighboring cells around the basis vertex are with size $3\Delta u_1\times 3\Delta v_1$, $3\Delta u_2\times 3\Delta v_1$, $3\Delta u_1\times 3\Delta v_2$, and $3\Delta u_2\times 3\Delta v_2$, respectively, and the black dots denotes the Bézier ordinates. The basis function $b_{j_1}(u,v)$ has knots $(u_0-3\Delta u_1, u_0-3\Delta u_1, u_0, u_0, u_0+3\Delta u_2)\times(v_0-3\Delta v_1, v_0-3\Delta v_1, v_0, v_0, v_0+3\Delta v_2)$. Hence the geometric information of $b_{j_1}(u,v)$ at (u_0,v_0) can be easily calculated, which is the first column in the matrix $\mathbf{b}$:

$$\mathbf{b} = \begin{pmatrix} (1-\lambda)(1-\mu) & -\alpha(1-\mu) & -\beta(1-\lambda) & \alpha\beta \\ \lambda(1-\mu) & \alpha(1-\mu) & -\beta\lambda & -\alpha\beta \\ (1-\lambda)\mu & -\alpha\mu & \beta(1-\lambda) & -\alpha\beta \\ \lambda\mu & \alpha\mu & \beta\lambda & \alpha\beta \end{pmatrix},$$

where

$$\alpha = \frac{1}{\Delta u_1+\Delta u_2}, \quad \beta = \frac{1}{\Delta v_1+\Delta v_2}, \quad \lambda = \alpha\Delta u_1, \quad \mu = \beta\Delta v_1.$$

The other three columns of $\mathbf{b}$ correspond to the geometric information of the other three basis functions.

The matrix $\mathbf{b}$ is invertible and one has

$$\mathbf{C} = \mathscr{L}\mathbf{S}(u_0,v_0)\cdot\mathbf{b}^{-1}. \quad (4)$$

The above formula reflects the relationship between the control points and the geometric information at the basis vertices of a spline surface over a (hierarchical) T-mesh. An efficient surface fitting scheme will be proposed in Section 5 based on Eq. (4).

4. Cross insertion and removal

Knot insertion and removal operations play an important role in curve and surface modeling with standard B-splines. For PHT-splines, cross insertion and removal operations play the same role as knot insertion and removal with B-splines. Many geometry processing algorithms such as surface fitting, NURBS simplification (Section 5), conversion to TP surfaces, and shape simplification (Section 6) all heavily depend on these two operations.

4.1. Cross insertion

Suppose we are given a PHT-spline surface $\mathbf{S}(u,v)$ over a hierarchical T-mesh $\mathscr{T}$ as defined in (3), and a cell θ from $\mathscr{T}$ is subdivided into four subcells by inserting a cross. Denote the new T-mesh after subdivision by $\widetilde{\mathscr{T}}$. We want to find the representation of $\mathbf{S}(u,v)$ over the new T-mesh $\widetilde{\mathscr{T}}$.

Suppose the level of the cell θ in $\mathscr{T}$ is k_1, and the maximal level of the adjacent cells of θ is k_2. If $k_1 \geqslant k_2 - 1$, then the old control points $\mathbf{C}_j$ are unchanged, and for each new basis vertex in $\widetilde{\mathscr{T}}$, the associated basis functions can be constructed as described in Section 3, and the corresponding control points are determined according to Eq. (4).

If $k_1 < k_2 - 1$, the process is a bit more complicated. We need to remove temporarily the cells adjacent to θ whose

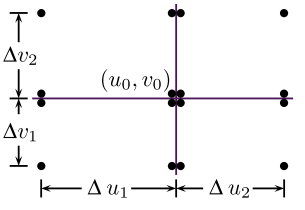

Fig. 9. The measurement of the T-mesh $\mathscr{T}$ around a new basis vertex (u_0,v_0).

levels are larger than $k_1 + 1$, and then perform cross insertion for θ. After this process, we perform cross insertion algorithm to recover those removed cells level by level.

It should be noted that all the operations take place in θ and its adjacent cells, hence the operation is local. Furthermore, $k_1 \geqslant k_2 - 1$ is the most common case in geometry processing applications.

4.2. Cross removal

Cross removal is the inverse operation of cross insertion. For simplicity, we require that the mesh after cross removal is still a hierarchical T-mesh. Like knot removal, cross removal can be achieved only approximately in general.

Let $\mathbf{S}(u,v)$ be a PHT-spline surface over a hierarchical T-mesh $\mathscr{T}$, and $\widetilde{\mathscr{T}}$ be the mesh after removing a cross from $\mathscr{T}$. Let θ be the cell in $\widetilde{\mathscr{T}}$ from which the cross is removed. Our idea is to find a PHT-spline surface $\widetilde{\mathbf{S}}(u,v)$ defined over $\widetilde{\mathscr{T}}$ such that $\mathbf{S}(u,v)$ and $\widetilde{\mathbf{S}}(u,v)$ share the same geometric information at the basis vertices of $\widetilde{\mathscr{T}}$. This requirement guarantees that $\widetilde{\mathbf{S}}(u,v) \equiv \mathbf{S}(u,v)$ whenever $\mathbf{S}(u,v)$ is obtained from $\widetilde{\mathbf{S}}(u,v)$ by inserting the cross in θ.

The basic approach is as follows. First, eliminate those terms in $\mathbf{S}(u,v)$ whose corresponding basis functions are associated with the basis vertices in $\mathscr{T}$ but not in $\widetilde{\mathscr{T}}$ to obtain $\overline{\mathbf{S}}(\mathbf{u},\mathbf{v})$. Second, modify the basis function $b_i(u,v)$ in each term of $\overline{\mathbf{S}}(\mathbf{u},\mathbf{v})$ to $\tilde{b}_i(u,v)$ while keeping the corresponding control points unchanged. The requirement is that $\{\tilde{b}_i(u,v)\}$ form a basis for the spline space $\mathscr{S}(3,3,1,1,\widetilde{\mathscr{T}})$, and that $b_i(u,v)$ and $\tilde{b}_i(u,v)$ share the same geometric information at the basis vertices of $\widetilde{\mathscr{T}}$. Over the cell θ as shown in Fig. 10, the Bézier representation of $\tilde{b}_i(u,v)$ can be easily obtained from the 16 black Bézier ordinates which come from $b_i(u,v)$. In the adjacent cells of θ, the modification is similar. Note that only those basis functions need to be modified which do not vanish over θ or its adjacent cells.

5. Surface fitting

Surface fitting is an investigated problem which has been addressed by many papers. The reader is referred to [1,9] for surveys. In this paper, we are going to present a very efficient scheme to fit a mesh model with a PHT-spline. The scheme shares the similar adaptive strategy with those in hierarchical B-splines, multilevel B-splines, and so on [8,16]. It directly leads to an efficient method to convert a NURBS surface into a PHT-spline, and a NURBS model into a few PHT-spline patches which are C^1 continuous globally. It should be noted that if one tries to solve the fitting problem by the least-squares method, a system of linear equations of huge size would be involved.

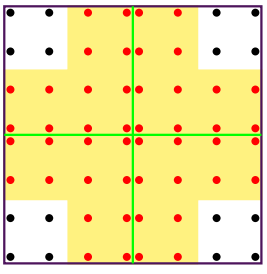

Fig. 10. Cross removal.

5.1. Fitting open meshes

Suppose we are given an open mesh model with vertices $\mathbf{P}_i$, $i = 1,2,\ldots,N$ in 3D space, and their corresponding parameter values (s_i,t_i), $i = 1,2,\ldots,N$ obtained from some parameterization of the mesh (The reader is referred to [5,6] for a survey on mesh parameterization). The parameter domain is assumed to be $[0,1]\times[0,1]$.

According to Eq. (4), in order to construct a PHT-spline to fit the given mesh, one only has to estimate $\mathscr{L}\mathbf{S}(u,v)$ at every basis vertex. This can be done in the following two steps. First, compute $\mathscr{L}\mathbf{S}(s_i,t_i)$. To do so, we find a set of points in the neighborhood $\mathscr{N}(\mathbf{P}_i)$ of $\mathbf{P}_i$, and then fit the set of points with a quadratic surface patch. The required information can be obtained by evaluating the surface patch at (s_i,t_i). Next, the geometric information $\mathscr{L}\mathbf{S}(u,v)$ at every basis vertex can be calculated by a simple linear interpolation to $\mathscr{L}\mathbf{S}(s_i,t_i)$, $i = 1,2,\ldots,N$.

The surface fitting scheme repeats the following two steps until the fitting error in each cell is less than some tolerance ε:

1. Keep unchanged the control points associating with the old basis functions, and compute the control points for the new basis functions (in the beginning, every basis function is new) according to Eq. (4);
2. Determine the cells whose fitting errors are greater than ε, and then subdivide these cells into subcells to form a new mesh, and construct basis functions for the new mesh.

Here the fitting error over a cell θ is defined to be

$$\max_{(u,v)\in\theta} \|\mathbf{P}(u,v) - \mathbf{S}(u,v)\|,$$

where $\mathbf{P}(u,v)$ is the (piecewise linear) parametric equation of the mesh model. In practice, the fitting error is calculated as the maximum of $\|\mathbf{P}(u_i,v_i) - \mathbf{S}(u_i,v_i)\|$ for some sample points (u_i,v_i) in θ.

Three examples are provided to illustrate the above surface fitting scheme in Figs. 1 and 11. In all these examples, the initial T-meshes are the square $[0,1]\times[0,1]$, and the parameterizations are obtained with the discrete harmonic mappings proposed by [3]. The tolerance of the fitting error is $\varepsilon = 0.1\%$, which refers to the size of the bounding box of the corresponding model. Table 1 shows the computational time for the three examples, where CP stands for control points. The computation is performed on a PC with Pentium 4 CPU, 3.20 GHz and 1.0 GB RAM. From the examples, we can conclude that the computational complexity is almost linear in the mesh size.

5.2. Fitting closed meshes

Many geometric models are described by closed meshes of genus-zero. For such models, the unit sphere is the most

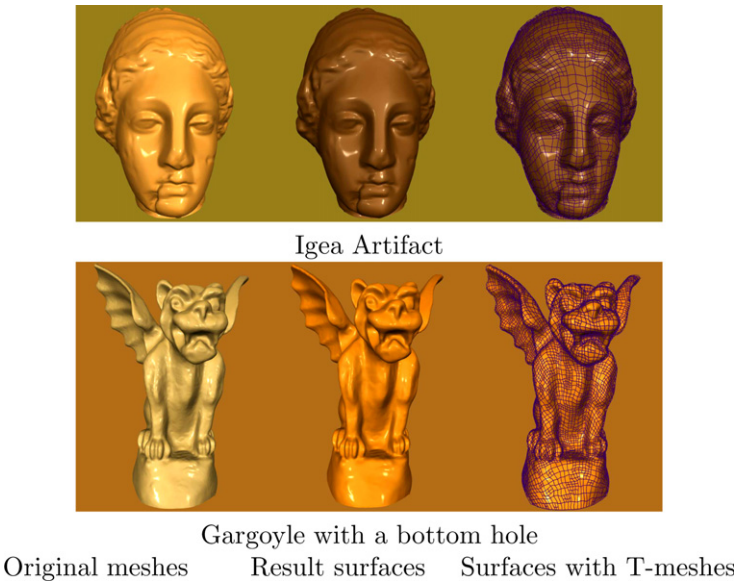

Fig. 11. Two examples of fitting open meshes.

natural parameterization domain. In this section, we propose a method to fit a closed mesh model with a family of PHT-splines which are pieced together with C^1 continuity. The method relies on the spherical parameterization of closed meshes. The reader is referred to [10,12,19] for some references on this topic.

The basic idea is as follows. Find an inscribed cube for the sphere of the parameterization domain, and project the six faces of the cube onto the sphere by the central projection from the center of the sphere. The sphere is divided into six parts corresponding to the six faces, and correspondingly the mesh model is divided into six parts by the spherical parameterization. This induces a planar parameterization for each part of the mesh model. Fitting each part of the mesh by a PHT-spline results in a surface model for the closed mesh.

To guarantee the smoothness between two adjacent PHT-splines, one should ensure that the two corresponding T-mesh domains share the same boundary vertices along their common edges when subdividing cells in the fitting process, and that the geometric information at boundary basis vertices is the average information of the two parts along the common edges. In particular, the G^1 geometric continuity conditions in [17] are used to maintain the smoothness around the corner. In this fashion, the final PHT-spline surface is C^1 continuity except at the corners, where G^1 smoothness is obtained.

Fig. 12 illustrates two examples for fitting closed meshes with PHT-splines, and Table 2 shows the execution time. Again the computational complexity is linear in the mesh size.

Table 1
Computation time for fitting open meshes

Mesh	# Points	# Levels	# CP	Time (s)
Female head	19,231	10	4432	2.31
Igea artifact	44,992	9	17,744	10.73
Gargoyle	91,279	17	35,764	33.23

5.3. NURBS approximation

The surface fitting scheme provided in the previous subsections can be easily adapted to approximate a NURBS surface with a PHT-spline surface. Since the local geometric information can be directly obtained from the NURBS surface, the algorithm is much faster. Fig. 13 demonstrates an example. The conversion process takes 0.875 s from a to b/c, and 0.125 s from a to d/e.

6. Geometry processing

PHT-splines allow for many types of geometry processing. For example, in the same approach as in [21], two or more PHT-splines can be stitched together. The surface fitting scheme for closed meshes in Section 5.2 provides another approach for merging two PHT-spline surfaces. On the other hand, similar to [7], the local details can be added

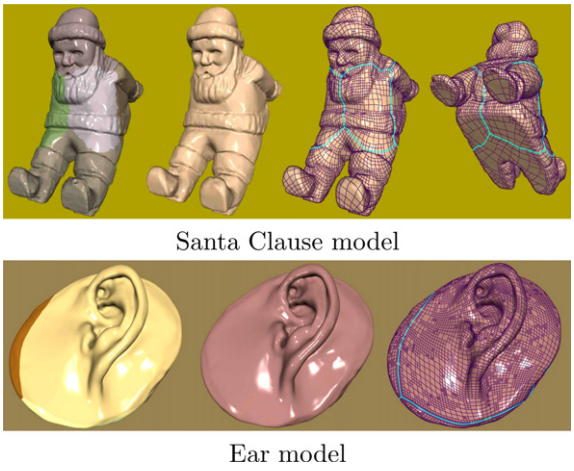

Fig. 12. Two examples of fitting closed meshes (Courtesy of Berding 3D Scanning (http://www.b3dscann.com/samples.htm)).

Table 2
Computation time for fitting closed meshes

Mesh	# Points	# Levels	#CP	Time (s)
Santa claus	46,048	11	23,552	8.52
Gargoyle	100,002	11	50,652	23.22
Ear	1,407,202	9	58,876	304.25

onto the shape of the PHT-spline surfaces by mesh refinements.

In this section, more types of geometry processing on PHT-splines will be explored briefly. They include conversion of a PHT-spline to an assembly of TP B-splines and shape simplification.

6.1. Conversion of PHT-splines to TP-splines

Given a PHT-spline defined over a hierarchical T-mesh, the PHT-spline can be easily converted into a family of TP B-spline patches. This conversion makes it easy for PHT-splines to be conveniently imported into the current surface modeling system.

Given a nonnegative integer σ and a hierarchical T-mesh $\mathscr{T}$, one can organize the cells in $\mathscr{T}$ into an assembly of rectangles such that the level differences of the cells in each rectangle are not greater than σ. This organization is usually not unique, but any solution leads to a conversion of the PHT-spline surface in the following fashion: in each rectangle, perform some cross insertions such that the T-mesh in the rectangle becomes a TP mesh, and then form a TP B-spline surface over the rectangle from the Bézier representation of the PHT-spline surface in each cell. See Fig. 14 for an example.

Fig. 15 shows the conversion results of the PHT-spline surface from Fig. 1g. Here the control nets of the TP B-spline patches are displayed in brown, and the patch boundaries are in light blue. It takes 1.44, 0.33 and 0.14 s to generate the results in Fig. 15a, b and c, respectively.

6.2. Shape simplification

Shape simplification of a PHT-spline surface over a T-mesh $\mathscr{T}$ is to find a coarser mesh $\widetilde{\mathscr{T}}$ (obtained from $\mathscr{T}$ by removing some crosses) of $\mathscr{T}$ such that the difference between the original PHT-spline surface and the new PHT-spline surface defined over $\widetilde{\mathscr{T}}$ is smaller than some given tolerance, and the two surfaces share the same geometric information at the same basis vertices. The result of shape simplification can be applied in level-of-detail operation in computer graphics.

Given a PHT-spline surface $\mathbf{S}(u,v)$ over a hierarchical T-mesh $\mathscr{T}$ with level k_0 and a tolerance ϵ, the procedure of shape simplification can be described as follows:

- Set $k = k_0 - 1$ and $\bar{\mathbf{S}}(u,v) = \mathbf{S}(u,v)$.
- For each cell θ in $\mathscr{T}$ at level k which has a cross inside θ (and each subcell of θ does not have a cross inside itself), perform cross removal operation for $\bar{\mathbf{S}}(u,v)$. Denote the PHT-spline surface after cross removal by $\tilde{\mathbf{S}}(u,v)$, and the mesh after cross removal by $\widetilde{\mathscr{T}}$. If $\|\mathbf{S}(u,v) - \tilde{\mathbf{S}}(u,v)\| < \epsilon$ over θ and its adjacent cells, then replace $\bar{\mathbf{S}}(u,v)$ by $\tilde{\mathbf{S}}(u,v)$, and $\mathscr{T}$ by $\widetilde{\mathscr{T}}$.
- Set $k = k - 1$. If $k \geqslant 0$, go to the previous step.

To speed up computation, we use the Bézier representations for the PHT-spline surfaces. The Bézier representation for $\tilde{\mathbf{S}}(u,v)$ over $\widetilde{\mathscr{T}}$ can be easily obtained from the Bézier representation of $\bar{\mathbf{S}}(\mathbf{u},\mathbf{v})$ by the smoothness conditions. The difference between $\mathbf{S}(u,v)$ and $\tilde{\mathbf{S}}(u,v)$ can be measured by the control points of the two surfaces. Fig. 16 shows the surface simplification result of the female head model, where the original mesh has 1528 cells. The simplification operation can be accomplished in real-time.

7. Conclusions and future work

The paper presents the theoretical foundation for PHT-splines—polynomial splines over hierarchical T-meshes. The construction of basis functions is simple and straight-

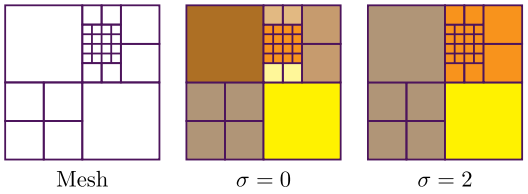

Fig. 14. Examples of hierarchical T-mesh segmentation.

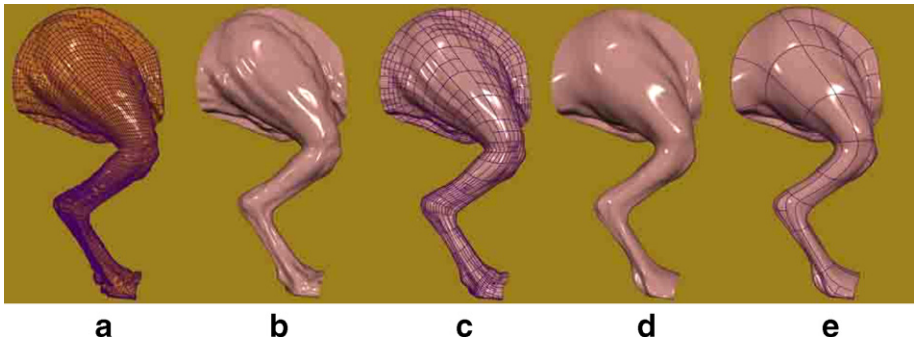

Fig. 13. NURBS approximation. (a) The original NURBS surface with 8835 control points. (b and c) A PHT-spline approximation with 4404 control points $\varepsilon = 0.35\%$. (e and f) A PHT-spline approximation with 468 control points $\varepsilon = 1.1\%$.

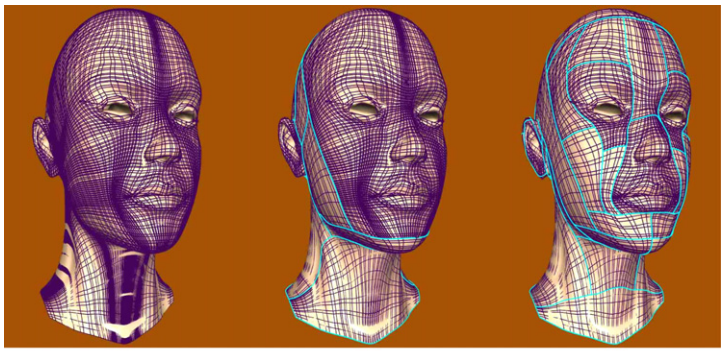

Fig. 15. Conversion of PHT-spline surfaces to TP-spline surfaces.

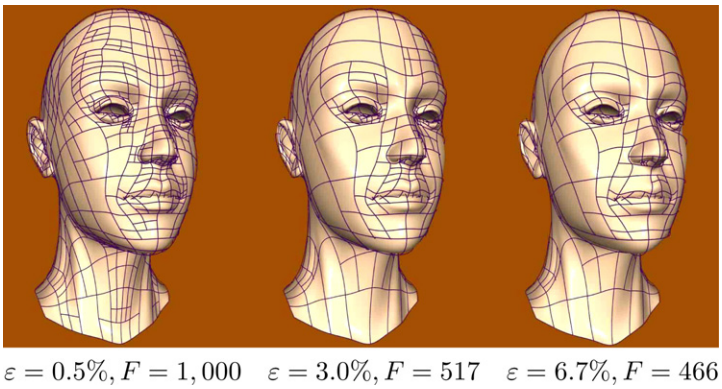

Fig. 16. Shape simplification.

forward to implement. The cross insertion and removal operations are discussed as well. With PHT-splines, an efficient and adaptive surface fitting scheme are presented. Several geometry processing algorithms are studied, such as conversion of a PHT-spline surface into a family of NURBS patches, and shape simplification of PHT-splines. The approach can be applied to building up the theoretical foundation for the spline spaces $\mathscr{S}(m,n,\alpha,\beta,\mathscr{T})$, where $\mathscr{T}$ is a hierarchical T-mesh, $m \geqslant 2\alpha+1$ and $n \geqslant 2\beta+1$.

PHT-splines inherit many important properties of T-splines such as adaptivity and locality. They also exhibit several advantages over T-splines. For example, unlike T-splines, PHT-splines are polynomial instead of rational. Thus geometric computation with PHT-splines is simpler and less costly. The local refinement algorithm of PHT-splines is local and simple while the complexity of knot insertion with T-splines may be uncertain. Finally, conversion between NURBS and PHT-splines is very fast, while conversion between NURBS and T-splines is a bottleneck of T-splines in practical applications.

One main drawback of our new splines is that they are only C^1 continuous. However, having a set of basis functions should be helpful for theoretic analysis and many other applications. For example, in fitting scattered data points with implicit splines [13,14,23], a set of basis functions are a necessity. Thus the implicit form of our new splines (i.e., the zero level-set of PHT-splines) can be applied in adaptive curve/surface fitting of scattered data. We will explore this possibility in future papers.

As a comparison, hierarchical B-splines do not provide a set of basis functions and they work with a redundant set of generators. Furthermore, hierarchical B-splines require a very special hierarchical T-mesh structure due to their refinement scheme. They form a proper subspace of C^2 continuous bicubic spline space over a T-mesh.

It is a challenge problem to obtain the dimension formula for the spline space $\mathscr{S}(3,3,2,2,\mathscr{T})$, since it may involve more complicated topological quantities. Please refer to [18] for some initial progress in this direction.

There are still many interesting research problems about PHT-splines for further research. For example,

- How to compute the dimensions for spline spaces over general T-meshes? And how to construct the basis functions for the spline spaces? These are in general very difficult problems. Currently, we are working on the spline spaces $\mathscr{S}(2,2,1,1,\mathscr{T})$ and $\mathscr{S}(3,3,2,2,\mathscr{T})$.
- Is there a more efficient cross insertion algorithm whenever $k_1 < k_2 - 1$?

- How to fit a model with complicated topology with PHT-splines?

There are problems worthy of further study.

Acknowledgments

The authors are support by the National Key Basic Research Project of China (No. 2004CB318000), the NSF of China (Nos. 60225002, 60533060, 10671192 and 10701069), One Hundred Talent Project supported by CAS, the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20060358055), the 111 Project (No. b07033) and NSF of Anhui Province of China (No. 070416230).

References

[1] P. Chinvate, A. Jablokow, Review of surface representation and fitting for reverse engineering, Computer Integrated Manufacturing Systems 8 (3) (1995) 193–204.
[2] J. Deng, F. Chen, Y. Feng, Dimensions of spline spaces over T-meshes, Journal of Computational and Applied Mathematics 194 (2) (2006) 267–283.
[3] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, W. Stuetzle, Multiresolution analysis of arbitrary meshes, in: SIGGRAPH'95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM Press, New York, NY, USA, 1995, pp. 173–182.
[4] G. Farin, Curves and Surfaces for CAGD—A Practical Guide, 5th ed., Morgan Kaufmann Publishers, 2002.
[5] M.S. Floater, K. Hormann, Parameterization of triangulations and unorganized points, in: Armin Iske, Ewald Quak, Michael S. Floater (Eds.), Tutorials on Multiresolution in Geometric Modelling, Springer-Verlag, 2002, pp. 287–316.
[6] M.S. Floater, K. Hormann, Surface parameterization: A tutorial and survey, in: N.A. Dodgson, M.S. Floater, M.A. Sabin (Eds.), Advances in Multiresolution for Geometric Modelling, Springer-Verlag, Heidelberg, 2004, pp. 157–186.
[7] D.R. Forsey, R.H. Bartels, Hierarchical B-spline refinement, Computer Graphics 22 (4) (1988) 205–212.
[8] D.R. Forsey, R.H. Bartels, Surface fitting with hierarchical splines, ACM Transactions on Graphics 14 (2) (1995) 134–161.
[9] R. Franke, G.M. Nielson, Scattered data interpolation and applications: a tutorial and survey, in: H. Hagen, D. Roller (Eds.), Geometric Modeling, Springer, Berlin, 1991, pp. 131–161.
[10] C. Gotsman, X. Gu, A. Sheffer, Fundamentals of spherical parameterization for 3D meshes, ACM Transactions on Graphics 22 (3) (2003) 358–363.
[11] E. Grinspun, P. Krysl, P. Schröder, Charms: a simple framework for adaptive simulation, ACM Transactions on Graphics 21 (3) (2002) 281–290.
[12] X. Gu, Y. Wang, T.F. Chan, P.M. Thompson, S.-T. Yau, Genus zero surface conformal mapping and its application to brain surface mapping, IEEE Transaction on Medical Imaging (7) (2004) 949–958.
[13] B. Jüttler, Least-squares fitting of algebraic spline curves via normal vector estimation, in: Proceedings of the 9th IMA Conference on the Mathematics of Surfaces, Springer-Verlag, 2000, pp. 263–280.
[14] B. Jüttler, A. Felis, Least-squares fitting of algebraic spline surfaces, Advances in Computational Mathematics 17 (2002) 135–152.
[15] R. Kraft, Adaptive and linearly independent multilevel B-splines, in: A.L. Méhauté, C. Rabut, L.L. Schumaker (Eds.), Surface Fitting and Multiresolution Methods, Vanderbilt University Press, Nashville, 1997, pp. 209–218.
[16] S. Lee, G. Wolberg, S.Y. Shin, Scattered data interpolation with multilevel B-splines, IEEE Transactions on Visualization and Computer Graphics 3 (3) (1997) 228–244.
[17] S.L. Lee, A.A. Majid, Closed smooth piecewise bicubic surfaces, ACM Transactions on Graphics 10 (4) (1991) 342–365.
[18] C.J. Li, R.H. Wang, F. Zhang, Improvement on the dimension of spline on T-mesh, Journal of Information and Computational Science 3 (2) (2006) 235–244.
[19] E. Praun, H. Hoppe, Spherical parametrization and remeshing, ACM Transaction on Graphics 22 (3) (2003) 340–349.
[20] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, ACM Transactions on Graphics 23 (3) (2004) 276–283.
[21] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Transactions on Graphics 22 (3) (2003) 161–172.
[22] F. Weller, H. Hagen, Tensor product spline spaces with knot segments, in: M. Daehlen, T. Lyche, L.L. Schumaker (Eds.), Mathematical Methods for Curves and Surfaces, Vanderbilt University Press, Nashville, 1995, pp. 563–572.
[23] Z. Yang, J. Deng, F. Chen, Fitting point clouds with active implicit B-spline curves, The Visual Computer (Special Issue for Pacific Graphics 2005) 21 (8–10) (2005) 831–839.
[24] J. Zheng, Y. Wang, H.S. Seah, Adaptive T-spline surface fitting to z-map models, in: GRAPHITE'05: Proceedings of the 3rd international conference on computer graphics and interactive techniques in Australasia and South East Asia, ACM Press, New York, NY, USA, 2005, pp. 405–411.
[25] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method, Volume 1: The Basis, 5th ed., Elsevier (Singapore) Pte Ltd., 2004.