Submesh Splines over Hierarchical T-meshes

Liangbing Jin Jiansong Deng Falai Chen
Department of Mathematics, University of Science and Technology of China, Hefei, Anhui, 230026, China
Email: `lbjin@mail.ustc.edu.cn; dengjs|chenfl@ustc.edu.cn`

Abstract

In this paper we propose a new type of splines — biquadratic submesh splines over hierarchical T-meshes. The biquadratic submesh splines are in rational form consisting of some biquadratic B-splines defined over tensor-product submeshes of a hierarchical T-mesh, where every submesh is around a cell in the crossing-vertex relationship graph of the T-mesh. We provide an effective algorithm to locate the valid tensor-product submeshes. A local refinement algorithm is presented and the application of submesh splines in surface fitting is provided.

Key words: *hierarchical T-mesh, submesh splines, local refinement, surface fitting*

1. Introduction

Tensor-product B-spline surfaces are a standard representation for free-form surfaces in the disciplines of computer graphics and geometric modeling ([5], [10]). One of their major weaknesses is that the control points must lie topologically in a rectangular grid and the local refinement by knot insertion influences entire rows or columns of the control points. To overcome this inflexibility, Forsey and Bartels [7] introduced hierarchical B-splines. Hierarchical B-splines were also studied by Kraft [9]. They constructed a multilevel spline space which is a linear span of tensor-product B-splines on different, hierarchically ordered grid levels.

T-splines, proposed by Sederberg ([11], [12]), are another innovation in this direction. A T-spline is a type of point-based spline defined over a T-mesh which is a rectangular grid that allows T-junctions. T-splines can eliminate most superfluous control points in NURBS representation. A T-spline is a piecewise rational polynomial within each cell of the T-mesh. This fact makes the local refinement algorithm of T-splines would extend all partial rows of control points to cross the entire surface in the worst case. In order to be compatible with the standard defining fashion, two of the present authors introduced the concept of spline spaces over T-meshes [1]. The dimension formula was proved with the B-net method [1] and the smoothing cofactor method [8] for the spline space $\mathbf{S}\ m, n, \alpha, \beta, \mathcal{T})$ for $m \geqslant 2\alpha + 1$ and $n \geqslant 2\beta + 1$. Then in [2] we provided an approach to define the basis functions of the C^1 continuous bicubic splines over hierarchical T-meshes and discussed its applications in surface fitting.

In practice, we prefer splines with highest possible smoothness, e.g. splines in $\mathbf{S}\ m, n, m-1, n-1, \mathcal{T})$. Unfortunately, there is a lack of theoretic foundations for such spline spaces. For example, we do not know the dimension formula for such splines space for $m \geqslant 3$ [3]. We do not know how to construct a set of basis functions neither. In this paper, we propose a new type of splines — submesh splines, which are defined in term of some tensor-product B-splines. A submesh spline is a single rational polynomial within each cell of a T-mesh. Hierarchical B-splines require a very special hierarchical T-mesh structure due to its refinement scheme, but a submesh spline is suitable for any hierarchical T-mesh. Compared with PHT-splines in [2], submesh splines have higher order of smoothness and are more adaptable to applications.

In this paper, we mainly discuss biquadratic submesh splines over hierarchical T-meshes. In [3] we have shown that the dimension of biquadratic spline spaces over hierarchical T-meshes is equal to the cell number of the corresponding crossing-vertex relationship graph. According to this conclusion, we can define submesh functions over valid tensor-product submeshes. The submesh functions have some good properties, such as nonnegativity, local support and partition of unity. We present a local refinement algorithm for submesh splines, which is achieved by cross insertion, i.e., dividing a cell into four subcells by inserting a cross. In some situations, the local refinement requires some additional divided cells in order to retain the shape of the submesh spline surfaces. Using submesh splines, surface models can be constructed adaptively to fit open mesh models with disk topology. Examples in Section 5 show that our surface fitting method needs less control points compared with NURBS and PHT-splines.

This paper is organized as follows. Section 2 recalls some preliminary knowledge about T-meshes and spline spaces over T-meshes, and defines the crossing-vertex relationship graph of hierarchical T-meshes. Section 3 introduces the biquadratic submesh spline spaces over hierarchical T-meshes and describes a method to find valid submeshes. The local refinement algorithm is provided in Section 4. Section 5 discusses the surface fitting with the biquadratic submesh splines over hierarchical T-meshes. Section 6 concludes the paper with a summary and some future work.

978-1-4244-3701-6/09/$25.00 ©2009 IEEE

2. Polynomial Splines over T-meshes and Crossing-vertex Relationship Graph

In this section, we review the definition of T-meshes, hierarchical T-meshes and splines spaces over T-meshes, and then introduce the crossing-vertex relationship graph of hierarchical T-meshes.

2.1. T-meshes and Hierarchical T-meshes

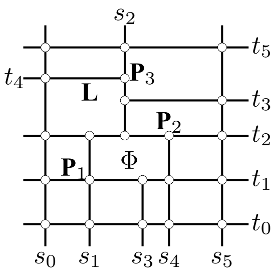
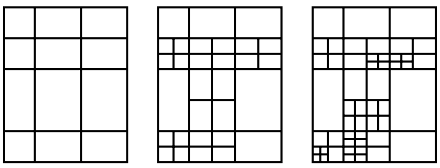


Figure 1. A T-mesh

A **T-mesh** in $\mathbb{R}^2$ is basically a rectangular grid that allows T-junctions. We use the compatible definitions of vertices, edges, cells with those in [1]. Figure 1 shows a T-mesh in (s,t) parameter space, where s_i denote s coordinates, and t_i denote t coordinates. Thus, each vertex has a knot coordinate. For example, $\mathbf{P}_1$ has knot coordinates (s_1,t_1) and $\mathbf{P}_2$ has knot coordinates (s_4,t_2). Similarly, we can define edge's knot coordinates and cell's knot coordinates. For example, $([s_0,s_2],t_4)$ denotes the edge $\mathbf{L}$ whose two endpoints are (s_0,t_4) and (s_2,t_4). The cell Φ has knot coordinates $([s_1,s_4],[t_1,t_2])$.

We classify T-vertices into two types: **horizontal T-vertices** and **vertical T-vertices**. In Figure 1, $\mathbf{P}_2$ is a vertical T-vertex, $\mathbf{P}_3$ is a horizontal T-vertex. A **horizontal/vertical l-edge** is a contiguous line segment which consists of some horizontal/vertical interior edges and whose two endpoints are boundary vertices or horizontal/vertical T-vertices. Obviously, an l-edge is the longest possible line segment in the T-mesh.



a. Level 0 b. Level 1 c. Level 2

Figure 2. A hierarchical T-mesh

A hierarchical T-mesh [2] is a special type of T-mesh which has a natural nested structure. It is defined in a recursive fashion . Figure 2 illustrates the process of generating a hierarchical T-mesh. For a hierarchical T-mesh $\mathcal{T}$, in order to emphasis its level structure in some cases, we denote the T-mesh of level k to be $\mathcal{T}^k$.

2.2. Spline Spaces over T-meshes

Given a regular T-mesh $\mathcal{T}$, $\mathcal{F}$ denotes all the cells in $\mathcal{T}$ and Ω the region occupied by all the cells in $\mathcal{T}$. In [1], the following spline space is defined

$$\begin{aligned}&\mathbf{S}(m,n,\alpha,\beta,\mathcal{T})\\&:=\{s(x,y)\in C^{\alpha,\beta}(\Omega):s(x,y)|_\phi\in\mathbb{P}_{mn},\forall\phi\in\mathcal{F}\},\end{aligned}$$

where $\mathbb{P}_{mn}$ is the space of all the polynomials with bi-degree (m,n), and $C^{\alpha,\beta}$ 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 $\mathbf{S}(m,n,\alpha,\beta,\mathcal{T})$ is a linear space. It is called the **spline space over T-mesh** $\mathcal{T}$.

In this paper, we are interested mainly in the spline space $\mathbf{S}(2,2,1,1,\mathcal{T})$, where $\mathcal{T}$ is a hierarchical T-mesh.

2.3. Crossing-vertex Relationship Graph

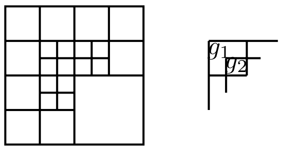


Figure 3. A T-mesh $\mathcal{T}$ and its crossing-vertex relationship graph $\mathcal{G}$

Given a hierarchical T-mesh $\mathcal{T}$, we keep all the interior crossing vertices and the edges connecting them, and remove all the other vertices and the edges in $\mathcal{T}$, then we can get a new mesh $\mathcal{G}$. $\mathcal{G}$ is called the **Crossing-vertex Relationship Graph** of the hierarchical T-mesh $\mathcal{T}$.

Obviously, the crossing-vertex relationship graph is not a rectangular grid, since there are L-vertices, hanging edges and hanging vertices in the crossing-vertex relationship graph. The valence of a vertex can be 1, 2, 3 and 4. Figure 3 shows the crossing-vertex relationship graph $\mathcal{G}$ of a hierarchical T-mesh $\mathcal{T}$, where there are two cells in $\mathcal{G}$, denote as g_1 and g_2.

3. Biquadratic Submesh Spline Spaces over Hierarchical T-meshes

In this section, we first introduce the concept of submeshes in T-meshes, and then define the biquadratic submesh spline spaces over hierarchical T-meshes. We also give a method to find valid biquadratic submeshes.

3.1. Submeshes

Definition 3.1: Given a T-mesh $\mathcal{T}$, suppose there are two knot vectors $S^k = [s_0^k, s_1^k, \cdots, s_{m+1}^k], T^k = [t_0^k, t_1^k, \cdots, t_{n+1}^k]$ that satisfy the following conditions:

1) S^k and T^k are in increasing order;
2) For all $0 \leqslant i \leqslant m+1, 0 \leqslant j \leqslant n+1,\ \ s_i^k, [t_0^k, t_{n+1}^k])$ and $[s_0^k, s_{m+1}^k], t_j^k)$ are edges of $\mathcal{T}$.

Then we say that S^k and T^k define a bi-degree $\ m, n)$ **submesh** in $\mathcal{T}$, denoted as

$$\mathcal{M}_k := S^k \times T^k = [s_0^k, s_1^k, \cdots, s_{m+1}^k] \times [t_0^k, t_1^k, \cdots, t_{n+1}^k].$$

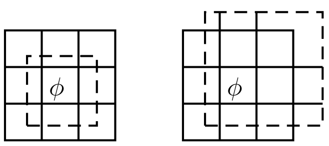
In this paper, we assume $m = n$ for simplicity. If a rectangle with four vertices $\ s_0, t_0),\ \ s_1, t_0),\ \ s_0, t_1)$, and $\ s_1, t_1)$ is a cell of the given T-mesh $\mathcal{T}$, then $[s_0, s_1] \times [t_0, t_1]$ is a bi-degree $\ 0, 0)$ submesh of $\mathcal{T}$. According to the definition, a bi-degree (2,2)(called ***biquadratic***) submesh is a tensor-product grid with knot vectors $[s_0, s_1, s_2, s_3]$ and $[t_0, t_1, t_2, t_3]$, and it has a center cell $\ [s_1, s_2], [t_1, t_2])$. Generally, two different biquadratic submeshes may have the same center cell. We will only select one as the valid biquadratic submesh for a certain center cell.

Definition 3.2: Given a T-mesh $\mathcal{T}$, suppose that there are two different biquadratic submeshes $\mathcal{M}_1 : [s_0^1, s_1^1, s_2^1, s_3^1] \times [t_0^1, t_1^1, t_2^1, t_3^1]$ and $\mathcal{M}_2 : [s_0^2, s_1^2, s_2^2, s_3^2] \times [t_0^2, t_1^2, t_2^2, t_3^2]$ in $\mathcal{T}$. If $s_1^1 = s_1^2, s_2^1 = s_2^2, t_1^1 = t_1^2, t_2^1 = t_2^2$, then we call $\mathcal{M}_1$ and $\mathcal{M}_2$ are two **co-cell submeshes**. There are two relations between two co-cell submeshes:

1) If $s_0^1 \leqslant s_0^2, s_3^1 \geqslant s_3^2;\ t_0^1 \leqslant t_0^2, t_3^1 \geqslant t_3^2$, then we call $\mathcal{M}_1$ includes $\mathcal{M}_2$;
2) If $s_0^1 \leqslant s_0^2, s_3^1 \leqslant s_3^2$ or $s_0^1 \geqslant s_0^2, s_3^1 \geqslant s_3^2$ or $t_0^1 \leqslant t_0^2, t_3^1 \leqslant t_3^2$ or $t_0^1 \geqslant t_0^2, t_3^1 \geqslant t_3^2$, then we call $\mathcal{M}_1$ intersects $\mathcal{M}_2$.

For two co-cell submeshes $\mathcal{M}_1$ and $\mathcal{M}_2$, if $\mathcal{M}_1$ intersects $\mathcal{M}_2$, then there must be another co-cell submesh $\mathcal{M}_3$, such that both $\mathcal{M}_1$ and $\mathcal{M}_2$ include $\mathcal{M}_3$.

For an arbitrary biquadratic submesh $\mathcal{M}_k$, if there is no co-cell submesh which includes it, we call $\mathcal{M}_k$ a **valid biquadratic submesh**.



a b

Figure 4. Two biquadratic submeshes with the same center cell ϕ

Figure 4 show the relations of two co-cell submeshes of the cell ϕ, $\mathcal{M}_1$ is with solid line boundary edges and $\mathcal{M}_2$ is with dashed line boundary edges. In Figure 4.a, $\mathcal{M}_1$ includes $\mathcal{M}_2$, while in Figure 4.b $\mathcal{M}_1$ intersects $\mathcal{M}_2$.

3.2. Biquadratic Submesh Spline Spaces over Hierarchical T-meshes

Now we introduce the concept of the biquadratic submesh spline spaces over hierarchical T-meshes.

Definition 3.3: Given a hierarchical T-mesh $\mathcal{T}$, suppose that the set of all valid biquadratic submeshes in $\mathcal{T}$ is $\{\mathcal{M}_k\}_{k=1}^K$. Define a biquadratic tensor-product B-spline basis function $N_k\ s, t)$ for each $\mathcal{M}_k$. All these $N_k\ s, t)$ expand a linear space, which is defined to be the **biquadratic submesh spline space over hierarchical T-meshes**. A **submesh spline** is defined as:

$$\mathbf{S}\ s, t) = \sum_{k=1}^{K} \mathbf{d}_k B_k\ s, t),\ B_k\ s, t) = \frac{N_k\ s, t)}{\sum\limits_{i=1}^{K} N_i\ s, t)},\ \ s, t) \in \Omega$$

where $\mathbf{d}_k\ k = 1, 2, \ldots, K)$ are control points, Ω is the region occupied by all the cells in $\mathcal{T}$. For each $k \in \{1, 2, \ldots, K\}$, $B_k\ s, t)$ is called a **submesh function** of $\mathcal{M}_k$.

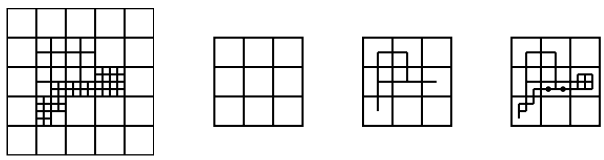
According to the definition above, it is easy to show that all the submesh functions $B_k\ s, t)$ satisfy the following properties:

1) $B_k\ s, t) \geqslant 0$;
2) For any k, $B_k\ s, t)$ has compact support;
3) The submesh functions form a partition of unity.

3.3. A Method to Find Valid Submeshes

In [3], we have proved the dimension formula of biquadratic spline spaces with smoothness of order one over hierarchical T-meshes. Given a hierarchical T-mesh $\mathcal{T}$, suppose that F is the number of cells in the the crossing-vertex relationship graph of $\mathcal{T}$. Then $\dim\mathbf{S}\ 2, 2, 1, 1, \mathcal{T}) = F$. Based on this result, we can find all valid biquadratic submeshes and define biquadratic submesh functions.

The main challenge is how to find all valid biquadratic submeshes in a hierarchical T-mesh. Here we propose a method to find all the valid biquadratic submeshes. We construct the crossing-vertex relationship graph $\mathcal{G}$ from the given hierarchical T-mesh $\mathcal{T}$. And then we use $\mathcal{G}$ to find all valid biquadratic submeshes in $\mathcal{T}$.



$\mathcal{T}$ $\mathcal{G}^0$ $\mathcal{G}^1$ $\mathcal{G} = \mathcal{G}^2$

Figure 5. Construct the crossing-vertex relationship graph $\mathcal{G}$ from a hierarchical T-mesh $\mathcal{T}$

Given a hierarchical T-mesh $\mathcal{T}$, we can construct the crossing-vertex relationship graph $\mathcal{G}$ from $\mathcal{T}$ level by level.

For example, as shown in Figure 5, the hierarchical T-mesh $\mathcal{T}$ have 3 levels. For the level k ($k = 0, 1, 2$), the corresponding crossing-vertex relationship graph is $\mathcal{G}^k$. Then the crossing-vertex relationship graph of $\mathcal{T}$ is $\mathcal{G}^2$.

Suppose that $\{g_i\}_{i=1}^F$ are all the cells of $\mathcal{G}$. For each cell g_i, denote its rectangular bounding box as $\tilde{g}_i$. We use $\{\tilde{g}_i\}_{i=1}^F$ to find valid biquadratic submeshes in $\mathcal{T}$. For each $\tilde{g}_i$, we can find a corresponding rectangle T_i in $\mathcal{T}$, where T_i consists of one or several cells of $\mathcal{T}$. Apparently, the four corner vertices of T_i are all crossing vertices of $\mathcal{T}$. Suppose that the knot coordinates of T_i are $[s_1^i, s_2^i], [t_1^i, t_2^i])$, the left, right, bottom and top l-edges of T_i are l, r, b and t.

In the following, we need four sets $V_i, i = 0, 1, 2, 3$. Their initial values are assumed empty. For l, r, b and t, we delete their interior T-vertices, and reserve the crossing vertices and two endpoints. Then for every vertex v_j in l, if the horizontal l-edge through v_j contains a vertex in r, we reserve the vertex v_j; otherwise we delete the vertex v_j. We denote the set of remainder vertices in l as V_{lr}. Analogously, we can get a set V_{bt} from b. For each t coordinate t_j of element v_j in V_{lr}, if $t_j > t_2^i$, push t_j back into V_1; if $t_j < t_1^i$, push t_j back into V_3. For each s coordinate s_j of element v_j in V_{bt}, if $s_j > s_2^i$, push s_j back into V_2; if $s_j < s_1^i$, push s_j back into V_0. Here V_0 and V_3 is sorted in descending order for s and t coordinates, V_2 and V_1 is sorted in ascending order for s and t coordinates.

With V_0, V_1, V_2 and V_3 in hand, we can ascertain the valid biquadratic submesh for T_i. Find an element of each V_i, that is s_0^i in V_0, s_3^i in V_2, t_0^i in V_3 and t_3^i in V_1. If they form a submesh, then the valid biquadratic submesh for T_i is $[s_0^i, s_1^i, s_2^i, s_3^i] \times [t_0^i, t_1^i, t_2^i, t_3^i]$. Otherwise, there is no valid biquadratic submesh for T_i. The pseudo-code of the process is shown as in Algorithm 1.

Apparently, it is not always true that for every $\tilde{g}_i$ in $\mathcal{G}$, one can find a valid biquadratic submesh in $\mathcal{T}$. By the dimension formula $\dim \mathbf{S}\ 2, 2, 1, 1, \mathcal{T}) = F$, the number of the valid biquadratic submeshes is no bigger than F. According to Definition 3.2, each center cell has at most one corresponding valid biquadratic submesh. Therefore the method of finding all valid biquadratic submeshes is feasible and reasonable. Then we can define the biquadratic submesh function $B_k\ s, t)$ and the biquadratic submesh spline according to Definition 3.3.

4. Local Refinement Algorithm

Suppose that the submesh spline space over a T-mesh is S_1, when a cell of the T-mesh is divided by cross insertion, we can get a submesh spline space S_2 over the new T-mesh. S_1 is said to be a subspace of S_2 if each submesh function of S_1 can be written as a linear combination of the submesh functions of S_2, denoted by $S_1 \subset S_2$.

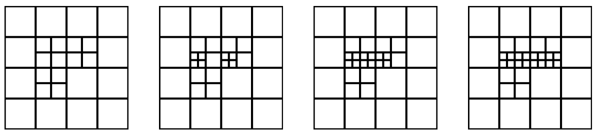
Unfortunately, $S_1 \subset S_2$ is not always true for an arbitrary T-mesh, that is, sometimes $S_1 \not\subset S_2$. For example, in Figure 6, $S_0 \subset S_1, S_1 \not\subset S_2, S_2 \subset S_3$. So we should find other cells for cross insertion to guarantee that S_1 is the subspace of the new submesh spline space. This process is called the local refinement algorithm.

The local refinement algorithm has two phases: the topology phase and the geometry phase. The topology phase identifies which cells should be divided by cross insertion in addition to the ones requested. The control points can be computed using the linear transformation in the geometry phase after all required cells are cross insertion.

We first introduce the topology phase of the algorithm.

Algorithm 1 Find Valid Submesh for T_i in $\mathcal{T}$

if V_0 or V_1 or V_2 or V_3 is empty **then**
  There is no valid biquadratic submesh for T_i, return
else
  $n_k \leftarrow 0 \quad k = 0, 1, 2, 3)$
  $W \leftarrow \{\{0,1\}, \{1,2\}, \{2,3\}, \{3,0\}\}$
  while do
    $j_0 \leftarrow W[0][0] \quad j_1 \leftarrow W[0][1]$
    Estimate whether the vertex $(V_{j_0}[n_{j_0}], V_{j_1}[n_{j_1}])$ is a valid vertex of a biquadratic submesh for T_i
    if It is true **then**
      $W \leftarrow W \backslash W[0]$
      if W is empty **then**
        $s_0^i \leftarrow V_0[n_{j_0}] \quad s_3^i \leftarrow V_2[n_{j_2}]$
        $t_0^i \leftarrow V_3[n_{j_3}] \quad t_3^i \leftarrow V_1[n_{j_1}]$
        Return submesh $[s_0^i, s_1^i, s_2^i, s_3^i] \times [t_0^i, t_1^i, t_2^i, t_3^i]$
      end if
    else {It is false}
      if $V_{j_0}[n_{j_0}]$ does not satisfy the qualification **then**
        $n_{j_0} \leftarrow n_{j_0} + 1$
        $W \leftarrow W \cup \{\{j_0 - 1, j_0\}, \{j_0, j_0 + 1\}\}$
      end if
      if $V_{j_1}[n_{j_1}]$ does not satisfy the qualification **then**
        $n_{j_1} \leftarrow n_{j_1} + 1$
        $W \leftarrow W \cup \{\{j_1 - 1, j_1\}, \{j_1, j_1 + 1\}\}$
      end if
      if (n_{j_0} is overflow) or (n_{j_1} is overflow) **then**
        There is no valid biquadratic submesh for T_i, return
      end if
    end if
  end while
end if



a. S_0 b. S_1 c. S_2 d. S_3

Figure 6. Nested sequence of submesh spline spaces

Given a hierarchial T-mesh $\mathcal{T}$, its crossing-vertex relationship graph is $\mathcal{G}$. Expand all cells of $\mathcal{G}$ to rectangular cells, denoted as $\tilde{\mathcal{G}} = \{\tilde{g}_i | i = 1, 2, \cdots, F\}$. For each $\tilde{g}_i$, its corresponding rectangle in $\mathcal{T}$ is T_i. We regard T_i as a center cell $\ [s_1^i, s_2^i], [t_1^i, t_2^i])$, then find a valid biquadratic submesh in $\mathcal{T}$. If there is no biquadratic submesh for the cell $\tilde{g}_i$, we should divide other cells in $\mathcal{T}$ by cross insertion. Denoted the left, right, bottom and top l-edges of T_i as l, r, b and t. T_i is contained in a cell ϕ_j of $\mathcal{T}$ at level k. Here there are four possible violations and at least one happens to T_i:

- **Violation 1** On the left of T_i, we can not find the s coordinates s_0^i to compose a valid submesh of T_i;
- **Violation 2** On the right of T_i, we can not find the s coordinates s_3^i to compose a valid submesh of T_i;
- **Violation 3** On the bottom of T_i, we can not find the t coordinates t_0^i to compose a valid submesh of T_i;
- **Violation 4** On the top of T_i, we can not find the t coordinates t_3^i to compose a valid submesh of T_i.

If no violation exists, the submesh spline is valid. If violations do exist, we resolve them one by one as following:

- **Rule 1** Select the nearest non-crossing cell at level k on the left of ϕ_j for cross insertion;
- **Rule 2** Select the nearest non-crossing cell at level k on the right of ϕ_j for cross insertion;
- **Rule 3** Select the nearest non-crossing cell at level k on the bottom of ϕ_j for cross insertion;
- **Rule 4** Select the nearest non-crossing cell at level k on the top of ϕ_j for cross insertion.

In conclusion, the topology phase of the local refinement algorithm consists of the following steps:

1) Divide all the desired cells by cross insertion in $\mathcal{T}$, and construct the crossing-vertex relationship graph $\mathcal{G}$;
2) Expand all the cells of $\mathcal{G}$ to rectangular cells to get $\bar{\mathcal{G}}$. If any cell of $\bar{\mathcal{G}}$ have no corresponding valid biquadratic submesh in $\mathcal{T}$, apply Rule 1, 2, 3, and 4 for the cell. Then reconstruct the crossing-vertex relationship graph $\mathcal{G}$;
3) Repeat Step 2 until all cells of $\bar{\mathcal{G}}$ have their corresponding biquadratic valid submeshes in $\mathcal{T}$.

Now we explain the geometry phase of the local refinement algorithm. Given a submesh spline $\mathbf{P}\ s,t) \in S_1$, its column vector of control points is $\mathbf{P}$. Given another submesh spline $\mathbf{Q}\ s,t) \in S_2$, its column vector of control points is $\mathbf{Q}$. Suppose that $\mathbf{P}\ s,t) \equiv \mathbf{Q}\ s,t)$ and

$$\mathbf{P}\ s,t) = \sum_{i=1}^{K} \mathbf{p}_i B_i\ s,t), \quad \mathbf{Q}\ s,t) = \sum_{j=1}^{\tilde{K}} \mathbf{q}_j \tilde{B}_j\ s,t).$$

Since $S_1 \subset S_2$, each $B_i\ s,t)$ can be written as a linear combination of the $\tilde{B}_j\ s,t)$: $B_i\ s,t) = \sum_{j=1}^{\tilde{K}} c_i^j \tilde{B}_j\ s,t)$.

So, there is a linear transformation that maps $\mathbf{P}$ into $\mathbf{Q}$: $H_{1,2}\mathbf{P} = \mathbf{Q}$, where the element at row j and column i of $H_{1,2}$ is c_i^j.

We illustrate the refinement algorithm with an example. Figure 6.b shows an initial T-mesh, its submesh spline space is S_1. When we divide a cell by cross insertion as shown in Figure 6.c, the new submesh spline space is S_2, $S_1 \not\subset S_2$. According to the algorithm, we must cross insert other cells, and get the new T-mesh as shows in Figure 6.d whose submesh spline space is S_3. Then we have $S_1 \subset S_3$.

5. Surface Fitting

Surface fitting is one of the most important research topics in computer graphics and geometric modeling. This section presents an adaptive scheme to fit open mesh models with disk topology with biquadratic submesh splines over hierarchical T-meshes.

Given an open mesh model in 3D space with disk topology, suppose the vertices are $\mathbf{P}_i\ i = 1, 2, \ldots, N)$. Using some parametrization method [6], we can obtain their corresponding parameter values $\ s_i, t_i), i = 1, 2, \ldots, N$ in a rectangular region. For simplicity, the parameter region is assumed to be $[0,1] \times [0,1]$. The surface fitting scheme can be described as follows:

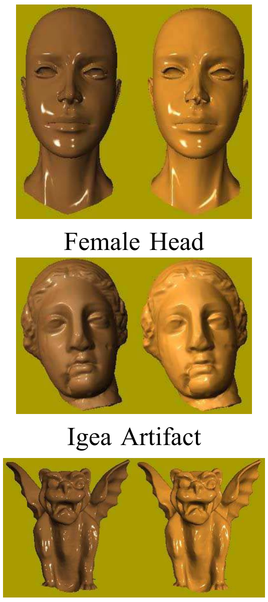
1) Let the initial T-mesh $\mathcal{T}^0$ be a tensor-product mesh, and the initial submesh spline surface $\mathbf{S}^0$ be the tensor-product B-spline surface defined over $\mathcal{T}^0$. Suppose that the fitting tolerance is $\epsilon > 0$, and set $k = 0$;
2) Compute the fitting error in each cell at level k, and mark the cells whose fitting errors are larger than ϵ;
3) If no cell is marked, $\mathbf{S}^k$ is the final submesh spline surface, and return $\mathbf{S}^k$; else, subdivide all marked cells, and obtain a new T-mesh of level $k+1$, denoted as $\widetilde{\mathcal{T}}^k$. Then according to the local refinement algorithm, subdivide some necessary cells of level k in $\widetilde{\mathcal{T}}^k$ to obtain the level $k+1$ T-mesh $\mathcal{T}^{k+1}$;
4) Find out all valid biquadratic submeshes in $\mathcal{T}^{k+1}$, define a biquadratic tensor-product B-spline basis function for each valid submesh, and compute the submesh functions of level $k+1$;
5) Use the least-squares method to compute the control points at level $k+1$ to get the submesh spline surface $\mathbf{S}^{k+1}$. Set $k = k+1$, return to Step 2.

Here the fitting error in a cell θ is ideally defined to be

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

where $\mathbf{P}\ u,v)$ is the 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.

We provide three examples to illustrate the surface fitting scheme in Figure 7. In all these examples, the initial T-meshes are square $[0,1] \times [0,1]$, and the parameterizations are obtained with the discrete harmonic mappings proposed by [4]. The tolerance of the fitting error is $\epsilon = 1\%$, which refers to the size of the bounding box of the corresponding



Female Head

Igea Artifact

Gargoyle

Original surfaces Result surfaces

Figure 7. Three examples of fitting closed meshes

model. The computation is performed on a PC with Intel Pentium 4 CPU 3.20GHZ and 1.0GB RAM. Table 1 shows the computational time and other information for the three examples, where CP stands for control points. The last column shows the number of control points when fitting with PHT-splines.

Table 1. Computation time for fitting open meshes

Mesh	#points	#levels	#time(sec.)	#CP	#CP/PHT
Female head	19231	10	8.93	2163	4432
Igea artifact	46313	6	40.71	5580	17744
Gargoyle	74721	13	132.45	8309	35764

6. Conclusions and Future Work

This paper introduces a new type of splines — submesh splines over hierarchical T-meshes, and specifically we studied biquadratic submesh splines. First we define valid submeshes and introduce a method to find valid submeshes. Then the local refinement algorithm is proposed. With the submesh splines over hierarchical T-meshes, an adaptively surface fitting scheme is presented as well.

There are still some interesting research problems about the submesh splines over hierarchical T-meshes. The natural problems are whether the set of submesh functions is linear independence and how to construct the basis functions of the spline space $\mathcal{S}$ $2, 2, 1, 1, \mathcal{T}$)? Besides, how to handle surfaces with general topologies? These are problems worthy of further research.

Acknowledgements

The authors are supported by the National Key Basic Research Project of China (No. 2004CB318000), the NSF of China (No. 60533060, 60873109, 10671192 and 10701069), One Hundred Talent Project supported by CAS, Program for New Century Excellent Talents in University, the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20060358055), and the 111 Project (No. b07033).

References

[1] J. Deng, F. Chen, and Y. Feng, "Dimensions of spline spaces over T-meshes", *Journal of Computational and Applied Mathematics*, 194, 2006, pp. 267–283.

[2] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng, "Polynomial splines over hierarchical T-meshes", *Graphical Models*, 70, 2008, pp. 76–86.

[3] J. Deng, F. Chen, and L. Jin, "Dimensions of biquadratic spline spaces over T-meshes", Preprint, arXiv: math. NA/0804.2533, 29pages.

[4] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes", in SIGGRAPH 95 Proceedings, New York: ACM Press, 1995, pp. 173–182.

[5] G. Farin, *Curves and Surfaces for CAGD — A Practical Guide*, 5th ed. Morgan Kaufmann Publishers, 2002.

[6] M. S. Floater and K. Hormann, "Surface parameterization: A tutorial and survey", in Advances in Multiresolution for Geometric Modelling, N. A. Dodgson, M. S. Floater , and M. A. Sabin, Eds. Springer-Verlag, 2004, pp. 157–186.

[7] D. R. Forsey and R. H. Bartels, "Hierarchical B-spline refinement", *Computer Graphics*, 22(4), 1988, pp. 205–212.

[8] Z. Huang, J. Deng, Y. Feng, and F. Chen, "New proof of dimension formula of spline spaces over T-meshes via smoothing cofactors", *Journal of Computational Mathematics*, 24(4), 2006, pp. 501–514.

[9] R. Kraft, "Adaptive and linearly independent multilevel B-splines", in Surface Fitting and Multiresolution Methods, A. L. Mehaute, C. Rabut, and L. L. Schumaker, Eds. Nashville: Vanderbilt University Press, 1998, pp. 209–218.

[10] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. New York: Springer-Verlag, 1997.

[11] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, "T-splines and T-NURCCs", *ACM Transactions on Graphics*, 22(3), 2003, pp. 161–172.

[12] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche, "T-spline simplification and local refinement", *ACM Transactions on Graphics*, 23(3), 2004, pp. 276–283.