C¹ Bicubic Splines over General T-meshes

Xin Li, Jiansong Deng, Falai Chen
Department of Mathematics, University of Science and Technology of China, Hefei, Anhui, China, 230026
Email: lixustc, dengjs, chenfl@ustc.edu.cn

Abstract

The present authors have introduced polynomial splines over T-meshes (PHT-splines) and provided the theories and applications for PHT-splines over hierarchical T-meshes. This paper generalizes PHT-splines to arbitrary topology over general T-meshes with any structures. The general PHT-spline surfaces can be constructed through an unified scheme to interpolate the local geometric information at the basis vertices of the T-mesh. We also discuss the edge insertion and removal algorithms for PHT-splines over general T-meshes. As applications, we present algorithms to construct a spline surface over a T-mesh from a quadrilateral mesh.

Key words: Splines, PHT-splines, T-meshes, Basis Function

1. Introduction

Sederberg et al. [3], [4] invented the notions of T-splines to eliminate most superfluous control points for NURBS surfaces and permit real local refinement. T-splines support many valuable operations within a consistent framework. The present authors introduced polynomial spline spaces over T-meshes(PHT-spline) in [1], where the function on each cell is a bi-degree polynomial, and achieves specified smoothness across the common edges. And we further constructed basis functions for spline spaces over hierarchical T-meshes and discussed their applications in surface fitting and stitching in [2].

The present paper generalizes the PHT-splines to arbitrary topology over general T-meshes with any structures which have more flexibility in surface modeling and finite element analysis. The main contributions of the present paper include the following issues.

- Construct the basis functions for PHT-splines over any type of T-meshes which must be nest structures in former papers;
- Provide a general edge insertion and edge deletion algorithm. In the former papers, we can only insert and remove edges of the highest level directly;
- Define an unify interpolation scheme for arbitrary topology T-mesh.

The remainder of the paper is organized as follows. We construct the basis functions for PHT-splines over general T-meshes and describe the edge insertion and edge deletion operators in Section 2. Section 3 discusses an unify interpolation scheme for arbitrary topology PHT-splines. In the end, section 4 concludes the paper with some summaries and future work.

2. Basis Functions Construction

The definition for polynomial splines over T-meshes are exact the same as those in [1], [2]. In [2], the authors constructed the basis functions for the coarse tensor-product mesh first and modify the basis functions level by level according to the mesh’s nested structures. In this section, we construct a set of basis functions for PHT-spline over general T-mesh with any type of structure. The algorithm has several advantages compared with that in [2].

- The method suits for any type of T-mesh which must has nested structure in [2];
- The basis functions are represented in B-spline form which are represented in Bézier forms in [2]. We only have almost 1/4 terms in this representation for each basis function.

For any function $b(u,v)$ defined on the T-mesh, geometric information at some parametric position $(u_0,v_0)$ is the function value, the first order partial derivatives and the mixed partial derivative of $b(u,v)$ at $(u_0,v_0)$. If all these values are zero, we call the function $b(u,v)$ has vanish geometric information at $(u_0,v_0)$.

A T-vertex is called a share T-vertex of two adjacent face if the T-vertex belongs to both faces and lies in the interior of an edge of one face. For example, in Figure 1, $v_3$ is a share T-vertex of face $F_1$ and $F_3$. It is also a share T-vertex of face $F_2$ and $F_3$ but it not a share T-vertex of face $F_1$ and $F_2$. Share-vertex is very important for construction basis functions.

Suppose we are given a T-mesh such as that in Figure 1.a, we can get a new T-mesh by extending all the T-vertices to the opposite edge (the red dashed lines in Figure 1.a). In the new T-mesh, we assume all the knots with multiplicity two and, at the vertex $V_i$, we can define four B-spline functions $N_i(u,v) = (N_{i1}(u,v),N_{i2}(u,v),N_{i3}(u,v),N_{i4}(u,v))$ by the knots for the four nearest regular vertices in four directions. For example, the four B-spline functions associated with $v_2$ are the four B-spline basis functions defined...
by the knot sequences $[x_3, x_3, x_3, x_3, x_3] 	imes [t_2, t_2, t_3, t_3, t_4]$ and those for $v_{10}$ are the B-spline basis functions defined by the knot sequences $[x_1, x_1, x_3, x_3, x_3, x_5] 	imes [t_0, t_0, t_1, t_1, t_2, t_2]$. We also define the four basis functions for the new cross vertices by the knot sequences $[x_1, x_1, x_3, x_3, x_3, x_3] 	imes [x_0, x_0, x_0, x_2, x_2, x_3]$. We compute $b_j(u,v)$ for the four b-basis functions associated with basis $N_i(u,v)$.

We compute $b_j(u,v)$ in the following four steps:

1) Forming set $L(j)$ for the $j$-th vertex;
2) Computing the geometric information at the $j$-th vertex;
3) Computing the geometric information at other vertices;
4) Computing $c_j^i$.

Forming $L(j)$ (Step 1): We illustrate the algorithm for computing $L(j)$ with an example in Figure 1. The algorithm will check the vertex itself and its neighbor vertices recursively. Figure 1 shows a T-mesh from which we wish to compute the $L(0)$ for basis function associated with basis vertex $v_0$. First, we add $v_0$ into the set and check the four neighbor vertices $v_1, v_2, v_3$ and $v_4$. As $v_i, i = 1, 2, 3$ are share T-vertices of two adjacent faces of edge $v_0v_1$, so we add $v_1, v_2, v_3$ into the set. The iterative of checking $v_1, v_2, v_3$, we will add $v_3$ and $v_{10}$ into the set according to the same fashion. In the next iterative, we will add $v_0$ into the set when we check vertex $v_3$. The algorithm will terminate since no new vertices added into the set when we check the vertices added in the last iterative. All vertices in $L(0)$ are marked with black in Figure 1.b.

Computing the geometric information at the $j$-th vertex (Step 2): In this step, we first find a rectangle containing inside all the vertices in $L(j)$. Suppose the position of the left-down and right-up point of the rectangle are $(u_1, v_1)$ and $(u_2, v_2)$, the position for $j$-th basis vertex is $(u_0, v_0)$, then the geometric information for $b_j(u,v)$ at $(u_0, v_0)$ can be represented in the form

$$b_j(u,v) = \sum_{i \in L(j)} N_i(u,v)c_j^i. \hspace{1cm} (1)$$

Here $b_j(u,v), N_i(u,v)$ are both $1 \times 4$ vector and $c_j^i$ is a $4 \times 4$ matrix. $L(j)$ is a set of indices of the T-vertices which have non-zero $c_j^i$.

We compute $b_j(u,v)$ in the following four steps:

1) Forming set $L(j)$ for the $j$-th vertex;
2) Computing the geometric information at the $j$-th vertex;
3) Computing the geometric information at other vertices;
4) Computing $c_j^i$.

Computing $c_j^i$ (Step 4): $c_j^i$ can be computed by interpolation the geometric information at each vertex according to the geometric information.

The above algorithm is very efficient since this matrix of the linear system is almost diagonal in general. Figure 2 illustrates an example for two basis functions over a T-mesh without nested structure. So we cannot apply the method in [2] to construct the basis functions. The color lines are the images of mapping the T-mesh to the surface with moving a little direction in the normal direction.

2.1. Edge Insertion and Edge Deletion

In this section, we will discuss the general edge insertion and deletion algorithms for spline surface $S(u,v)$ which is the most important operator for splines.

The first step of the operation is to construct the basis functions for the new T-mesh. Without loss of generality, suppose the edge insertion leads one new basis function $v_{\text{new}}$. First we can construct the basis functions construction for the new basis vertex according to section 2. Suppose the four new basis functions are $n_i(u,v), i = 0, \ldots, 3$. Then for any existing basis function $b_j^i(u,v)$, it will replaced by $\hat{b}_j^i(u,v) = b_j^i(u,v) - \sum_{i=0}^3 \lambda_i n_i(u,v)$. Here $\lambda_i$'s satisfy that $b_j^i(u,v)$ and $\sum_{i=0}^3 \lambda_i n_i(u,v)$ have the same geometric information at $v_{\text{new}}$. So $\hat{b}_j^i(u,v)$ will have vanish geometric information at $v_{\text{new}}$. We show a modification of one basis function with some edges insertion in Figure 3.
Figure 3. The modification of one basis function with general edge insertion operator.

The second step is to modify the control points. Here, we just compute the new control points and keep the existing control points unchanged. First, we compute the geometric information of \( S(u,v) \) at the new basis vertices and then interpolate these geometric information to get the new control points.

Edge deletion is the inverse operator of the edge insertion. Since edge deletion is not an exact operator, there are many opinions to define the surface after edge removed. The geometric information of \( S(u,v) \) at the new basis vertices unchanged. For any basis function \( B(u,v) \), suppose the modification of \( B(u,v) \) in the new spline space is \( R(u,v) \).

Suppose the eight basis functions associated with the two basis vertices are \( b_k(u,v), k = 0,\ldots,3 \) and \( c_k(u,v), k = 0,\ldots,3 \), respectively if we remove two basis vertices. The other cases are similar. Then there exist constants \( \lambda^k, k = 0,\ldots,3 \) and \( \mu^k, k = 0,\ldots,3 \) such that

\[
R(u,v) = B(u,v) + \sum_{k=0}^{3} \lambda^k b_k(u,v) + \sum_{k=0}^{3} \mu^k c_k(u,v).
\]  

Similarly, Equation (2) is equivalent to linear systems with \( \lambda^k \) and \( \mu^k \) being unknowns, respectively. According to the dimension formula, the solutions of these linear systems are also exist and unique.

Figure 4. The modification of one basis function with general edge removal operator

In both cases, we can compute the Bézier ordinates for \( R(u,v) \) in \( T^1 \). So it is easy to compute the Bézier ordinates for \( R(u,v) \) in \( T^2 \) according to Bézier subdivision algorithm.

We show a modification of one basis function with some edges removal in Figure 4.

3. Arbitrary Topology PHT-spline

However the PHT-splines defined in the previous section can only model objects with simple topology. In this section, we will define an arbitrary topology \( G^1 \) PHT-spline by generalization the interpolation algorithm to extraordinary vertices.

Let \( g_k \) and \( h_k \) be two independent unit orthogonal vectors lying in the tangent plane of \( \Phi \) at \( v_k, k = 0,1,\ldots,N \), then the following interpolation problem can be considered for PHT-splines. Given a set of values or vectors \( f^j, f^j_0, f^j_1, f^j_{uv} \) for extraordinary vertices \( v_j \) and \( f^j, f^j_0, f^j_1, f^j_{uv} \) for other basis vertices \( v_i \), find \( \Phi_{|\gamma} = \phi \) such that

1. \( \Phi(v_i) = f^i, D_g \Phi(v_i) = f^i_g, D_h \Phi(v_i) = f^i_h, D_{gh} \Phi(v_i) = f^i_{uv} \) for regular vertices \( v_i \),
2. \( \Phi(v_j) = f^j, D_g \Phi(v_j) = f^j_g, D_h \Phi(v_j) = f^j_h \) for extraordinary vertices \( v_j \).

In the next section, we will express the maps \( \Phi \) in Bézier form according to the geometric information at all the vertices.

3.1. Bézier Forms

In this section, we will express the map \( \Phi \) in Bézier form according to the geometric information at all the vertices. The construction mainly contains two steps.

Step 1: Define a bi-cubic map for each face

The bi-cubic Bézier form for \( \phi \) which corresponds to the face without extraordinary vertices is very easy to compute since we have known the local geometric information for each corner and the basis functions for each regular vertices. However, we should have different rules for the maps corresponding to an extraordinary faces because we don’t have mixed partial derivatives. We will illustrated the equations for the initial bi-cubic maps according to Figure 5. In the figure, \( V \) is a valence \( n \) vertex with \( n \) neighbor vertices \( V_i \) which is valence \( n_i \). The knot interval for edge \( VV_i \) is \( a_i \), respectively.

According to the geometric information at the vertices, we can determine the Bézier control points \( C, A_i \) and \( D_i \). The only undermined Bézier control points for the initial bi-cubic maps are those associated with the twist vector at the extraordinary vertices such as \( B_i \) in Figure 5.

Figure 5. The notations for \( G^1 \) smooth of map corresponding to the extraordinary face.

Denote \( b_i(t) = \begin{cases} 2 \cos \frac{2\pi}{N}(1-t) - 2 \cos \frac{2\pi}{N}t, & \text{if } n_i \neq 4; \\ 2 \cos \frac{2\pi}{N}(1-t)^2, & \text{else.} \end{cases} \)
and \( U_i = b_i'(0)\frac{d_0 - c}{a_i} + b_i(0)\frac{d_2 - 2d_0 + c}{a_i^2}. \)

Here \( b_i'(t) \) is the first derivative of \( b_i(t) \). Then we can compute \( B_i \) by the following equation.

\[
B_i = A_i + A_{i+1} - C + a_i a_{i+1} \left( \frac{U_i}{4} + \frac{U_{i+1}}{4} \right). \tag{3}
\]

### Step 2: \( G^1 \) smooth for the extraordinary vertex

Suppose the next two rows of \( b_1 \) and \( b_5 \) are labeled in Figure 5b. The black control points are determined by the local geometric information which will kept unchanged. The main process of \( G^1 \) smooth are listed below.

1) Compute \( P_2 \) and \( P_3 \) according to the following equations.

\[
P_2 = 2P_1 - C + \frac{a_i^2}{b_i(0)} \left( \frac{U_{i-1} + 2U_i + U_{i+1}}{4} - \frac{b_i'(0)(P_1 - P_0)}{a_i} \right),
\]

If \( n_i = 4 \),

\[
P_3 = P_2 + \frac{P_0 - P_1}{2} - \frac{P_0 - P_5}{10}.
\]

Otherwise, \( P_3 \) is computed by the similar equations as \( P_2 \).

2) Denote \( P(t) \), \( Q(t) \) and \( R(t) \) are three quintic Bézier curves with control points \( P_i, Q_i \) and \( R_i \) respectively. Let \( V(t) \) be a cubic Bézier curve with control points \( Q_0 = Q_{-0}, 5(Q_0 - R_0), 5(Q_0 - R_5), 3(Q_0 - R_5) \) and \( Q_5 = R_5 \). Then \( Q(t) \) and \( R(t) \) are linear combination of \( P(t) \) and \( V(t) \).

\[
Q(t) = P(t) + \frac{a_{i+1}a_{i-1}b_i'(t)P'(t)}{5(a_{i+1} + a_{i-1})a_i} + \frac{a_{i+1}}{a_{i+1} + a_{i-1}} V(t),
\]

\[
R(t) = P(t) + \frac{a_{i+1}a_{i-1}b_i'(t)P'(t)}{5(a_{i+1} + a_{i-1})a_i} - \frac{a_{i-1}}{a_{i+1} + a_{i-1}} V(t).
\]

A direct application of arbitrary topology PHT-spline is to construct a smooth spline surface from a given 2-manifold mesh in \( \mathbb{R}^3 \) with quadrilateral faces of arbitrary topological genus. This can be achieved by interpolation the local geometric information at the basis vertices which can be estimated from the mesh. For a regular vertex, we can estimate the local geometric information by traditional B-spline theory. For an valence \( n \) extraordinary point, we can define the information by the corresponding information of Catmull-Clark subdivision. Figure 6 illustrated several examples for constructing spline over T-mesh form the given quadrilateral meshes.

### 4. Conclusions and Future Work

The paper generalizes the PHT-splines to arbitrary topology over general T-meshes with any structures. PHT-spline surfaces can be constructed through a unify scheme to interpolate the local geometric information at the basis vertices. We also discuss the general edge insertion and removal algorithms for PHT-splines. It is evident that all the process can be generalized to PHT-spline with higher degrees.

There exist a number of problems for future research. For example, how to compute the dimension of the spline space \( \mathcal{S}(3,3,2,2,\ldots) \) over general T-meshes or hierarchical T-meshes \( \mathcal{T} \), and how to construct the basis functions for the spline space? And what is the relationship between T-splines and PHT-splines?

### Acknowledgement

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

### References


