ICES REPORT 12-13

April 2012

Solid T-spline Construction from Boundary Triangulations with Arbitrary Genus Topology

by

Wenyan Wang, Yongjie Zhang, Lei Liu and Thomas J.R. Hughes

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: Wenyan Wang, Yongjie Zhang, Lei Liu and Thomas J.R. Hughes, Solid T-spline Construction from Boundary Triangulations with Arbitrary Genus Topology, ICES REPORT 12-13, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, April 2012.

Solid T-spline Construction from Boundary Triangulations with Arbitrary Genus Topology

Wenyan Wang [a], Yongjie Zhang [a,*], Lei Liu [a], and Thomas J.R. Hughes [b]

[a]*Department of Mechanical Engineering, Carnegie Mellon University*
Pittsburgh, PA 15213, USA

[b]*Institute for Computational Engineering and Sciences, The University of Texas at Austin*
Austin, TX 78712, USA

Abstract

A comprehensive scheme is described to construct rational solid T-splines from boundary triangulations with arbitrary topology. To extract the topology of the input geometry, we first compute a smooth harmonic scalar field defined over the mesh and saddle points are extracted to determine the topology. By dealing with the saddle points, a polycube whose topology is equivalent to the input geometry is built and it serves as the parametric domain for the solid T-spline. A polycube mapping is then used to build a one-to-one correspondence between the input triangulation and the polycube boundary. After that, we choose the deformed octree subdivision of the polycube as the initial T-mesh, and make it valid through pillowing, quality improvement and applying templates to handle extraordinary nodes and partial extraordinary nodes. The obtained T-spline is C^2-continuous everywhere over the boundary surface except for the local region surrounding polycube corner nodes. The efficiency and robustness of the presented technique are demonstrated with several applications in isogeometric analysis.

Key words: Solid T-spline Construction, Arbitrary Genus Topology, Polycube, Rational T-spline, Isogeometric Analysis

1. Introduction

For tight integration of Design-Through-Analysis, isogeometric analysis [11] was proposed which utilizes spline functions as a basis. The current state-of-the-art in engineering design and analysis is built on disparate geometric foundations. Spline representation is popular in design while polygonal mesh representation is generally used in analysis. This leads to many translational difficulties which affect the efficiency and accuracy of the entire process. Isogeometric analysis utilizes the same basis functions as the geometry representation, and consequently, analysis can be carried out over an exact spline representation of the geometry. Similar to other physically-based analysis, solid models, which can represent both boundary shape and interior volume, are required for many applications in isogeometric analysis. A fundamental step for the unified Design-Through-Analysis technologies is to automatically construct solid trivariate spline models from boundary representations.

The advent of T-splines [16] gives more flexibility for geometric modeling, allowing local refinement, non-rectangular domains in 2D and non-cubic domains in 3D. T-splines can represent a complicated design with complex topology as a single watertight geometry, avoiding splitting the model into several patches. T-splines are a superior alternative to and also are compatible with NURBS, which is the current geometry standard in CAD systems. The flexibility of T-splines makes them an ideal discretization technology for isogeometric analysis.

Several methods have been developed recently to construct solid T-splines. A solid T-spline generation method was described for genus-zero geometries [4]. In [13], trivariate T-splines were defined based on the generalized polycube parametrization. Another spline scheme based on polycubes, called restricted trivariate polycube splines, was developed in [21]. This algorithm is based on semi-standard T-splines. It requires calculation of weights and the obtained T-spline elements are uniform. For all these methods, the polycube is generated manually for a given geometry. In addition, the constructed T-spline model may contain some negative Jacobian elements, which is unsuitable for analysis.

In our earlier work, we developed a mapping-based rational solid T-spline construction method for genus-zero geometry from the boundary surface triangulation [23]. To extend this algorithm to more general geometry, we first extract the topology of the input geometry and build a polycube which approximates the input geometry, by computing a harmonic field

[*] Corresponding author: Y. Zhang. Tel: (412) 268-5332; Fax: (412) 268-3348; E-mail address: jessicaz@andrew.cmu.edu.

and dealing with its critical points. Due to its regular structure, the polycube is suitable for serving as the parametric domain of the tensor-product spline representations. Here, a trivariate solid T-spline is built upon the generated polycube. We then build a parametric mapping between the triangulation and the boundary of the polycube. In the following steps, an octree subdivision is applied to the polycube and the initial T-mesh is a deformation of its subdivision. The subdivision continues until the surface approximation error is less than a given tolerance. After that, the valid T-mesh is obtained through pillowing, T-mesh quality improvement, and applying templates to handle extraordinary nodes and partial extraordinary nodes. Finally, Bézier elements are extracted with all positive Jacobians, and they are suitable for isogeometric analysis.

Compared with other existing methods, our solid T-spline construction scheme has several attractive properties: (1) it can be used for arbitrary topology and the polycube is created automatically; (2) the obtained trivariate T-spline has a one-piece representation and it contains very few irregular nodes where the continuity degenerates; (3) it employs rational T-spline basis, which guarantee partition of unity by definition; (4) it produces high quality analysis-suitable T-spline elements with all positive Jacobians; and (5) with an adaptive refinement scheme, the resulting T-spline model is an efficient representation for analysis.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents an overview of the construction algorithm. The polycube construction algorithm is presented in Section 4. The T-mesh construction algorithm from the polycube is described in Section 5 and solid T-spline construction is presented in Section 6. Section 7 presents examples and Section 8 draws conclusions.

2. Previous Work

Harmonic Fields. A harmonic field is the solution to the Laplace equation with given boundary conditions and it defines a scalar or vector field over the domain. For a given mesh, harmonic fields can be computed by solving a linear system of algebraic equations with imposed boundary conditions. Harmonic fields have certain desirable properties, such as smoothness, and are free of extraneous critical points. Due to these properties, harmonic fields have been shown to provide effective tool for a number of geometry processing problems. Dong *et al.* [3] traced the integral lines through the gradient and orthogonal vector fields of a harmonic field for quadrilateral remeshing of arbitrary manifolds. Based on a harmonic map, a 3D geometric metamorphosis method was developed for any two objects which are topologically equivalent to a sphere or a disk [12]. Joshi *et al.* [19] utilized harmonic coordinates, which are generalized barycentric coordinates, in volume deformation.

Surface Parameterization. A surface parameterization is a one-to-one mapping from one surface in 3D to a suitable planar domain. Parameterization is a powerful tool and necessary for many geometry processing tasks, including data fitting, texture mapping, and remeshing. Many significant advances have been made for surface parameterization [5, 6, 18]. In [8], a conformal mapping method was presented to map a genus-zero closed surface onto the unit sphere by minimizing the harmonic energy of the map. For parameterization of an arbitrary genus object to simpler surfaces of the same genus, the mesh is usually first segmented into disk-like patches and then each patch is mapped onto the corresponding plane. In [7], Gu *et al.* solved the problem of global conformal parameterization for surfaces of arbitrary topology, with or without boundaries.

Polycube Generation and Application. A polyube is a solid composed of cubes. It can be used to very roughly approximate the geometry of a 3D object while faithfully replicating its topology. Due to its highly regular structure, the polycube can be used as the parametric domain for surface parameterization and spline modeling. However, in practice due to the complexity of shapes, polycubes are usually constructed manually, entailing considerable effort. In order to produce polycubes with less user intervention, He *et al.* [9] developed a method to construct a 3D polycube by extruding the axis-aligned polygons which approximate the horizontal curved intersection contours. Based on the polycube, a global parameterization technique, polycube map [19], was first used for seamless texture mapping. Wang *et al.* developed a technique to build the polycube splines upon the polycube map for surface modeling [20]. In [13], an algorithm was developed to construct trivariate T-splines over generalized polycubes with a global "one-piece" representation for general volumetric data. In [21], a theoretical volumetric modeling framework was presented to construct restricted trivariate polycube splines, in which the blending functions are strictly bounded within the solid polycube domain.

Solid Spline Modeling. Only a few works have been devoted to solid spline modeling. In [10], a trivariate simplex spline modeling method was developed based on a tetrahedral decomposition of any 3D domain with complicated geometry and arbitrary topology. In [24], a skeleton-based method was developed to construct solid NURBS for isogeometric analysis of arterial blood flow. A method was presented in [1] to generate NURBS parameterizations of swept volumes by sweeping a closed curve, and isogeometric analysis was applied to the generated NURBS model. Based on discrete harmonic functions, a volumetric parameterization was used to construct a single trivariate B-spline [14]. By using adaptive tetrahedral meshing and a mesh untangling technique, an algorithm was developed to construct a trivariate T-spline representation of genus-zero solids [4].

It is still a challenging problem to automatically create polycubes for high genus geometry and use them in constructing analysis-suitable trivariate T-splines. In this paper, we utilize a harmonic field defined over the input mesh to build the polycube automatically, and then construct the rational solid T-spline over the polycube. We include pillowing and quality improvement techniques to guarantee that the obtained solid T-spline can be used for analysis directly.

3. Algorithm Overview

As shown in Figure 1, there are three main stages for constructing a solid T-spline from a given boundary triangle mesh with arbitrary genus topology. From the input triangle mesh, we first compute a harmonic scalar field defined over the mesh, extract the geometry topology, and then generate the polycube with the same topology. Adopting the polycube as the parametric domain, we build the valid T-mesh in the second stage and construct solid T-splines in the last stage.

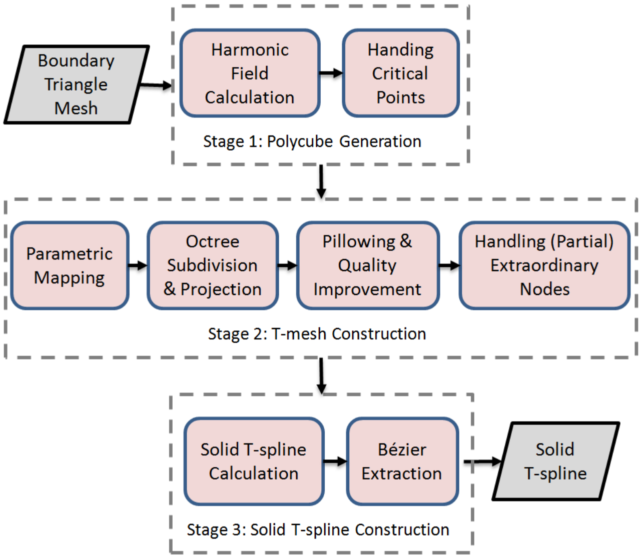
Boundary Triangle Mesh
Harmonic Field Calculation
Handing Critical Points
Stage 1: Polycube Generation
Parametric Mapping
Octree Subdivision & Projection
Pillowing & Quality Improvement
Handling (Partial) Extraordinary Nodes
Stage 2: T-mesh Construction
Solid T-spline Calculation
Bézier Extraction
Solid T-spline
Stage 3: Solid T-spline Construction

Fig. 1. An overview of the solid T-spline construction algorithm from the given boundary triangle mesh with arbitrary topology.

The polycube generation stage consists of two steps:

(i) Harmonic Field Calculation - We build a smooth harmonic scalar field defined over the input mesh. Based on this field, we compute its gradient field and an orthogonal vector field;

(ii) Handing Critical Points - From the harmonic field, we determine all the critical points where the first partial derivatives vanish. These critical points include extreme points and saddle points. We design different methods to deal with each type of point in order to build the polycube. The polycube edges are traced along the gradient and isocontour directions.

Based on the polycube, the T-mesh is constructed. There are four different kinds of nodes in the solid T-mesh: regular nodes, partial extraordinary nodes, extraordinary nodes and T-junctions. A ***regular node*** is a node around which the local T-mesh is a structured mesh, like node A in Figure 2(a). Both partial extraordinary nodes and extraordinary nodes are irregular, and they can be distinguished using reflection edges. ***Reflection edges*** are a pair of adjacent edges with one common node, and the set formed by all the elements sharing one edge is topologically symmetric with the set of elements sharing the other. For example, AB and AC in Figure 2(b) are a pair of reflection edges. A ***partial extraordinary node*** is an irregular node about which some but not all of its adjacent edges have reflection edges, like node A in Figure 2(b). An ***extraordinary node*** is an irregular node about which none of its adjacent edges has a reflection edge, such as node A in Figure 2(c). A ***T-junction*** terminates a row of control points in one or more parametric directions, which may lie on an edge or a face. In solid T-splines, an ***edge T-junction*** is a T-junction which lies on one edge, such as node M in Figure 2(d). A ***face T-junction*** is a T-junction lying on one face, such as node P in Figure 2(e).

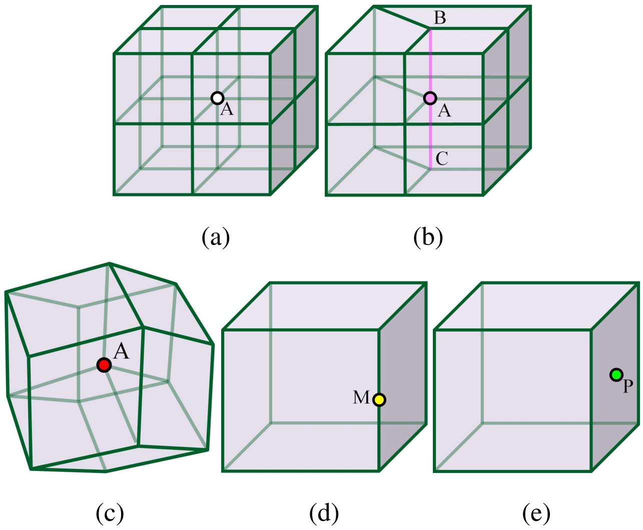

Fig. 2. Four types of nodes in solid T-meshes. (a) Regular node; (b) partial extraordinary node; (c) extraordinary node; (d) edge T-junction; and (e) face T-junction.

The T-mesh construction stage consists of four steps:

(i) Parametric Mapping - We build a parametric mapping between the input triangle mesh and the boundary of the generated polycube;

(ii) Octree Subdivision and Projection - An initial T-mesh is obtained by an octree subdivision of the polycube and each node on the polycube boundary is projected onto the boundary surface based on the mapping;

(iii) Pillowing and Quality Improvement - We insert one pillowed layer on the boundary and improve T-mesh quality by smoothing and optimization;

(iv) Handling Extraordinary Nodes and Partial Extraordinary Nodes - In order to obtain a gap-free T-mesh, we apply templates to each extraordinary node and partial extraordinary node in the initial T-mesh.

In this stage, we use the polycube as the parametric domain for the solid T-spline construction. In the octree subdivision step, we choose the existing T-junction parametric values to subdivide each octree cell as much as possible, instead of always using the central parametric value. Based on the valid T-mesh, the knot vectors for each node are determined by traversing T-mesh faces and edges [17], and the solid T-spline is constructed based on the rational T-spline definition [22]. Bézier elements are extracted to serve as the primary computational objects in isogeometric analysis. See [2, 15] for elaboration.

4. Polycube Generation

In this section, we discuss the detailed algorithm of the polycube generation from the input boundary triangulation. The polycube must be constructed in a geometrically approximate and topologically equivalent way. To achieve this goal, we first compute a harmonic function defined over the input mesh $\mathcal{T}$, derive two orthogonal fields from the harmonic function, and extract the topology structure of $\mathcal{T}$ by examining critical points.

4.1. *Harmonic Field Calculation*

To extract the topological structure of the input mesh $\mathcal{T} \subset \mathbb{R}^3$, we construct a harmonic function $f : \mathcal{T} \to \mathbb{R}$, such that

$$\triangle f = 0 \tag{1}$$

subject to the boundary condition that vertices in the predefined set S_{min} and S_{max} have the minimum and maximum values. $\triangle$ is the Laplace operator. S_{min} and S_{max} are either given by the user, or they can also be determined by selecting the bottom-most and top-most points on $\mathcal{T}$. Basically, computing such a harmonic function is to assign a scalar value to each vertex in $\mathcal{T}$. For a triangle mesh, we use the discretization of the Laplace operator

$$\triangle f_i = \sum_{j \in N_i} w_{ij}(f(V_j) - f(V_i)) = 0, \tag{2}$$

where $V_i, V_j \in \mathcal{T}$, w_{ij} is the weight and N_i is the number of vertices adjacent to V_i. Here, we choose the weights, $w_{ij} = \cot\alpha_{ij} + \cot\beta_{ij}$, where α_{ij} and β_{ij} are the opposite angles of the edge $V_i - V_j$. From the discretization of the Laplace operator, the scalar function f can be obtained by solving a linear system. Figure 3(a) shows the scalar function defined over the "Eight" model. The red region has the maximal scalar value and the blue region has the minimal scalar value.

From the scalar field f, we compute both the gradient $g_1 = \nabla f$ and one orthogonal vector field g_2 (the isocontour field). Due to the properties of a harmonic scalar field, the obtained direction fields are guaranteed to be smooth and free of extraneous critical points. For one triangle (V_i, V_j, V_k), suppose $\vec{n}$ is the unit normal vector. The gradient vector $g_1 = \nabla f$ is obtained by solving the following linear system [3]:

$$\begin{bmatrix} V_j - V_i \\ V_k - V_j \\ \vec{n} \end{bmatrix} [g_1] = \begin{bmatrix} f_j - f_i \\ f_k - f_j \\ 0 \end{bmatrix}. \tag{3}$$

The field g_2 is along the isocontour directions of f. Hence, for one scalar value, we simply find out its isocontour to obtain the vector field g_2 for the triangle mesh. Once we obtain the two orthogonal vector fields g_1 and g_2, we can trace along the flow lines. A flow line is a piecewise-linear curve defined over the mesh whose edges are along one of the vector fields. There are two cases for tracing the gradient flow, the regular case and the edge case. As shown in Figures 3(b-c), vertex A is the starting point, the green arrows represent the gradient direction for each incident triangle, and we wish to trace the flow line from A by walking across the incident triangles along the gradient direction. For the regular case in (b), starting from A we extend the gradient line by crossing one of its incident triangles. In this case, we add one new vertex B to advance the flow line. For the edge case in (c), the flow field converges on an edge AB and we simply follow this edge. In Figures 3(b-c), the red edges are the newly obtained flow line segments. The next step is to consider vertex B and trace the flow line from it using the same procedure.

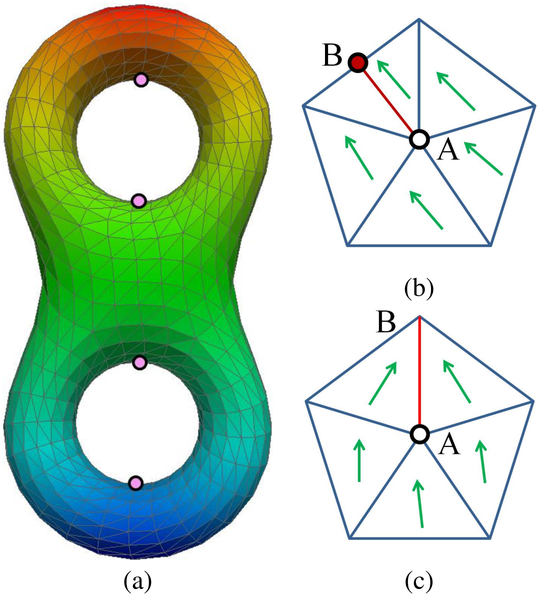

Fig. 3. (a) Harmonic scalar field with saddle points rendered in pink for the "Eight" model. (b-c) Two cases for tracing the gradient flow line from a given vertex A. (b) Regular case; and (c) edge case.

In addition, based on the scalar field, we can then determine all the critical points of f, that is, those points whose partial derivatives vanish. These points include:

- Minimal point - Points in set S_{min};
- Maximal point - Points in set S_{max};
- Splitting saddle point - Points where the geometry splits;
- Merging saddle point - Points where the geometry merges.

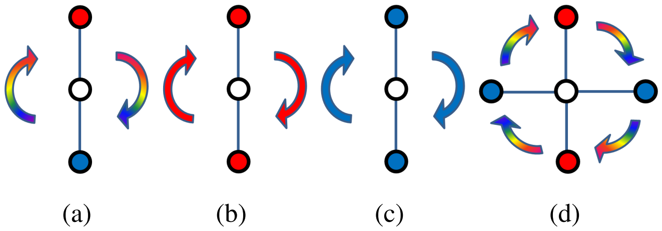

Fig. 4. Configurations around a regular (a), minimal (b), maximal (c), and saddle point (d).

Figure 4 shows the configurations of a regular, minimal, maximal and saddle point. The red points denote the vertices with a larger scalar value compared with the center point and the blue points denote the vertices with a smaller scalar value. As shown in Figure 4(a), around one regular point, vertices on one side all have a larger scalar value and vertices on the other side all have a smaller scalar value. For a minimal/maximal point, all the vertices surrounding it have a larger or smaller scalar value. For a saddle point, the sign changes alternately along its circumferential direction.

Discussion: The harmonic field is controlled by the user-defined minima and maxima constraints, it then affects the polycube generation and alignment of the solid T-spline. To ensure

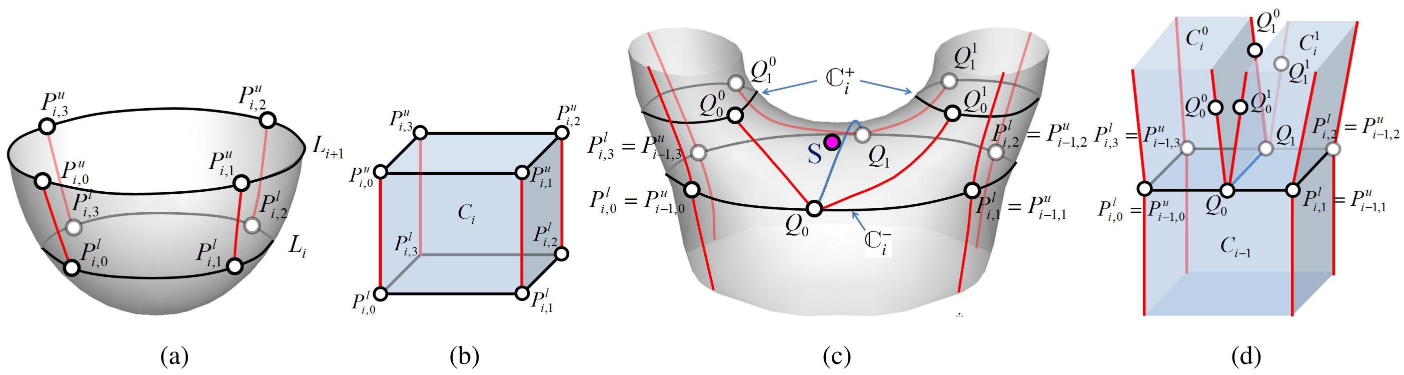

Fig. 5. (a-b) Handling the minimal level; and (c-d) handling one splitting saddle point S.

the harmonic field follows the geometric shape of the input surface, generally it is better to place these constraints on the tips of the geometry and also consider symmetry. For example, in the "Eight" model (Figure 3(a)), we assign the constraints at the top and the bottom.

4.2. *Handing Critical Points*

We need to handle various critical points: minimal, maximal, splitting saddle and merging saddle points. We first compute all the scalar levels or isocontours at which there are critical points. Let f_i denote the isovalue of L_i and $\mathbb{C}_{\mathbb{i}}$ denote the corresponding isocontour. For level L_i with saddle points, two sets of isocontours, $\mathbb{C}^-_i$ and $\mathbb{C}^+_i$, are computed by using the isovalue f_i with a small perturbation δ. For the minimal or maximal level, only one isocontour is computed.

Suppose level L_i contains one minimal point. At level L_i, four seed points ($P^l_{i,j}, j = 0, \dots, 3$) are chosen for each closed curve as shown in Figures 5(a-b), which correspond to the four lower corners of the cube C_i with the unit parametric length. The parametric value of the other vertices lying on this isocontour will be computed using the chord-length parameterization. From these seed points $P^l_{i,j}$ we trace the gradient flow line until it intersects with the isocontour $\mathbb{C}^-_{i+1}$ of the next level L_{i+1} at $P^u_{i,j}$ (L_{i+1} may contain a saddle or maximal point). The red curves in Figure 5 are traced gradient lines, and the black ones are isocontours. The four vertices $P^u_{i,j}$ ($j = 0, \dots, 3$) correspond to the four upper corners of C_i. The traced four curves are then mapped onto the four vertical edges of the cube, and $P^l_{i,j}$ and $P^u_{i,j}$ serve as the eight corners. Then the polycube construction process advances to the next level.

For level L_i with a splitting saddle point S as shown in Figures 5(c-d), we

(i) Parameterize the lower isocontour $\mathbb{C}^-_i$ using $P^l_{i,j} = P^u_{i-1,j}$ as four corners (assuming the associated cube before splitting is C_{i-1});

(ii) Find the shortest path from the splitting saddle point S to $\mathbb{C}^-_i$, get the intersection node Q_0 on $\mathbb{C}^-_i$, and calculate point Q_1 on the opposite edge with the same parametric value;

(iii) Compute the shortest path between Q_0 and Q_1 (see the blue curve in Figure 5(c), the path can not contain any edge on this isocontour);

(iv) Determine two sets of points on $\mathbb{C}^+_i$, $Q^0_0 - Q^0_1$ and $Q^1_0 - Q^1_1$, which have the shortest distance from $Q_0 - Q_1$ to $\mathbb{C}^+_i$;

(v) Construct two cubes C^0_i and C^1_i by using $P^l_{i,0}$-Q_0-Q_1-$P^l_{i,3}$ and Q_0-$P^l_{i,1}$-$P^l_{i,2}$-Q_1 as the lower corners, respectively;

(vi) continue tracing the gradient flow until the flow line intersects the isocontour at the next level.

Basically, for one splitting saddle point, we aim to find one isoparametric line to split one cube into two. Here, the isoparametric line connecting $Q_0 - Q_1$ is used to split the upper face of the cube C_{i-1} as shown in Figures 5(c-d). Similarly, for each merging saddle point, we use the same procedure to find one isoparametric line to merge two neighboring cubes and ensure they match with each other seamlessly.

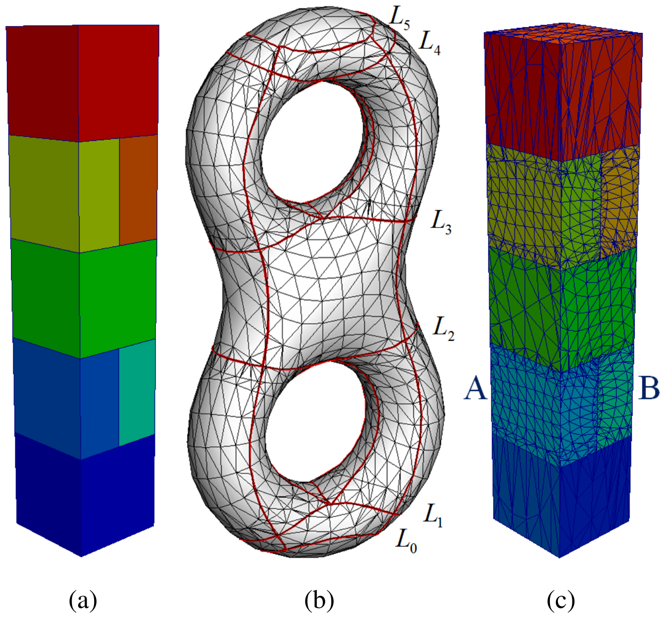

Fig. 6. Polycube construction and mapping results for the "Eight" model. (a) The constructed polycube in the parametric space; (b) the polycube in the physical space; and (c) the polycube mapping result.

By dealing with all the critical points, the polycube is constructed level by level. We always map the isocontour onto the horizontal isoparametric edges of the polycube and map the gradient flow lines onto the vertical edges. Figure 6 shows one polycube construction result for the "Eight" model. The red lines in Figure 6(b) denote the curves corresponding to the edges of the polycube. This polycube construction algorithm does not consider the symmetry property of the input geometry. However, if S_{min} and S_{max} are given symmetrically, the obtained polycube and the constructed solid T-spline can be symmetric for a symmetric input geometry.

Discussion: Here, we only consider Morse saddle points whose multiplicity is 1. Morse saddle points are handled by splitting one cube into two, or merge two cubes into one. In general, for saddle points of any multiplicity, one cube may be split into an arbitrary number of cubes. If there are two or more saddle points on one level, multiple sets of Q_0-Q_1 need to be computed.

5. T-mesh Construction

The T-mesh construction stage aims to build one valid T-mesh from the given boundary triangulation and the constructed polycube. There are four main steps in this stage: parametric mapping, adaptive octree subdivision and projection, pillowing and quality improvement, handling extraordinary and partial extraordinary nodes.

5.1. *Parametric Mapping*

This step aims to build a one-to-one parametric mapping between the input triangle mesh $\mathcal{T}$ and the boundary of the obtained polycube, which serves as the parametric domain of the solid T-spline. From the constructed polycube, we have the correspondence between the traced gradient or isocontour lines and the edges of the polycube. Based on the traced lines on the triangle mesh, we can divide the input mesh into N sub-meshes, $\mathcal{T}^i$ $(i = 0, \ldots, N)$, where N is the number of boundary faces of the polycube. Each sub-mesh is associated with one face of the polycube, $\mathcal{F}_C{}^i (i = 0, \ldots, N)$. We then use the surface parameterization to map each sub-mesh $\mathcal{T}^i$ to its corresponding polycube face $\mathcal{F}_C{}^i$.

For each sub-mesh, we first map its boundary to the boundary of $\mathcal{F}_C{}^i$ by a chord length parameterization. The parameterization for the interior vertices is calculated by solving a linear system formed by the harmonic equation $\sum\limits_{j\in N_i} w_{ij}(f(V_j) - f(V_i)) = 0$, where $w_{ij} = \cot\alpha_{ij} + \cot\beta_{ij}$. For the curve shared by two adjacent sub-meshes, we use the same parameterization. Figure 6(c) shows the mapping result for the "Eight" model. To guarantee a conformal boundary between two neighboring cubes, we always choose the same parameterization for all the edges shared by them. Note that the two neighboring cubes A and B do not share faces. There are two duplicated faces in the parametric space but they are separate in the physical space.

5.2. *Adaptive Octree Subdivision and Projection*

An initial T-mesh is generated by applying an adaptive octree subdivision to the polycube C and projecting to the boundary. For each cube in C, we create one hexahedral element, using the same parametric length and considering the physical length difference in three directions. Then we obtain a root T-mesh for the whole polycube and treat it as one single piece, instead of treating each cube separately. Starting from the root T-mesh, we subdivide one element into eight smaller ones recursively to get the refined initial T-mesh after projection. For each boundary element, we check the local distance from the T-mesh boundary to the input triangular mesh, and subdivide the element if the distance is greater than a given threshold ε.

Each obtained node has both parametric and physical coordinates. The parametric coordinates represent its position in C. For each boundary node, the physical coordinates are its associated position on the triangular mesh based on the polycube mapping. The physical coordinates of each interior node are calculated by a linearly interpolation. Note that the three isoparametric planes are not always in the middle. If one element contains T-junctions, the parametric values of the T-junctions are used to subdivide the element. If there are more than two T-junction parametric values in one direction, the one closest to the middle is used. For example, the purple element in Figure 7 has one T-junction on the ξ-edge and two T-junctions on the η-edge. For the ξ direction, the parametric value ξ_1 is used, while for the η direction η_2 is used during subdivision, because it is closer to the parametric middle. The dash lines are inserted to refine this element. In this way we can minimize the number of T-junctions in the initial T-mesh.

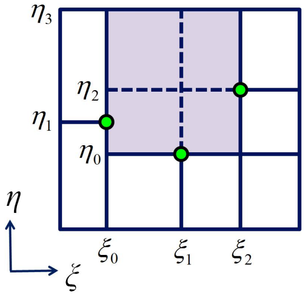

Fig. 7. T-mesh subdivision. Green nodes denote T-junctions.

The octree subdivision and projection processes continue until the local distance from each boundary element to the input mesh is less than ε and each element has at most one edge T-junction for an edge, or one face T-junction for a face. Figure 8 gives one result. (a) shows the parametric coordinates for the nodes in the T-mesh and (b) shows their physical coordinates.

5.3. *Pillowing and Quality Improvement*

To improve the T-mesh quality, we adopt the pillowing, smoothing and optimization techniques. Pillowing is a sheet-insertion technique that inserts one layer around the boundary [23]. Here we insert one pillowed layer for the initial T-mesh, which helps to improve the T-mesh quality and the T-spline surface continuity.

Figure 9 illustrates the pillowing operation for the polycube. The cubes C_i are rendered in different colors. As mentioned earlier, C_1 and C_2 do not share a face in the parametric space. In (b), we use two separate faces. The dark blue lines show the pillowed layer. In pillowing, each boundary face is duplicated to form one pillowed element, and each pillowed element has at

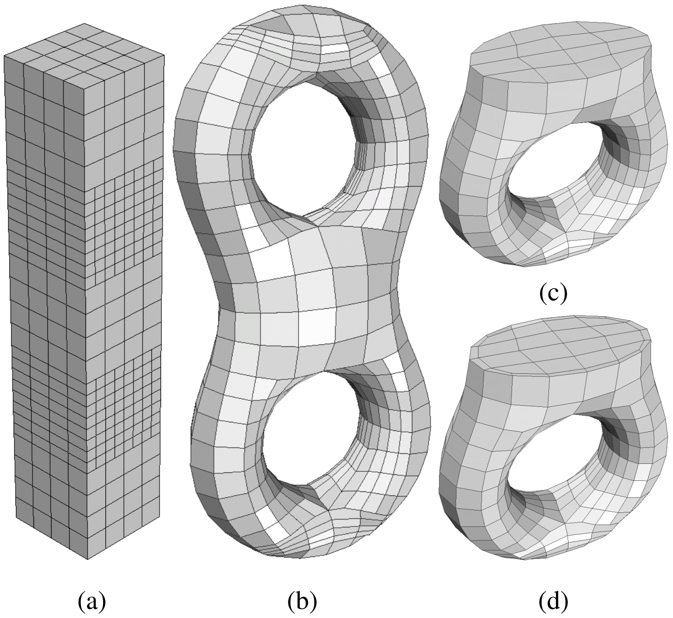

Fig. 8. The subdivision, projection, pillowing and optimization results for the "Eight" model. (a) The subdivision result in the parametric domain; (b) the initial T-mesh; (c-d) Interior of the T-mesh before and after pillowing and optimization.

most one face lying on the boundary. For the pillowed layer, the edge knot interval is a predefined constant along the pillowing direction, which stays the same for the other two directions.

As shown in Figure 9(b), after pillowing the yellow corner nodes become interior extraordinary nodes. The nodes on the blue polycube edges become interior partial extraordinary nodes. The red corner nodes, pillowed from the yellow nodes, are extraordinary nodes on the boundary surface. The constructed T-spline surface is C^0-continuous around these red boundary extraordinary nodes up to the 2-ring neighborhood, C^1-continuous from the 2-ring to 3-ring neighborhood, and C^2-continuous everywhere else. Note that after pillowing, the surface continuity across the polycube edges is improved from C^0 to C^2. The interior region is C^0-continuous around each interior extraordinary node until the 3-ring neighborhood. For the interior region across the polycube edges, the continuity is C^0 until the 2-ring neighborhood, C^1 from 2-ring to 3-ring neighborhood, and C^2 everywhere else. Since here we introduce some extraordinary and partial extraordinary nodes, we use a local parameterization for each element in the following steps.

After pillowing, smoothing and optimization are used to improve the T-mesh quality. For smoothing, each node is moved toward its mass center, and for optimization each node is moved toward an optimal position that maximizes the worst Jacobian. For one T-mesh element, the Jacobian is defined as

$$J = det(J_M) = \begin{vmatrix} \sum_{i=0}^{7} x_i \frac{\partial N_i}{\partial \xi} & \sum_{i=0}^{7} x_i \frac{\partial N_i}{\partial \eta} & \sum_{i=0}^{7} x_i \frac{\partial N_i}{\partial \zeta} \\ \sum_{i=0}^{7} y_i \frac{\partial N_i}{\partial \xi} & \sum_{i=0}^{7} y_i \frac{\partial N_i}{\partial \eta} & \sum_{i=0}^{7} y_i \frac{\partial N_i}{\partial \zeta} \\ \sum_{i=0}^{7} z_i \frac{\partial N_i}{\partial \xi} & \sum_{i=0}^{7} z_i \frac{\partial N_i}{\partial \eta} & \sum_{i=0}^{7} z_i \frac{\partial N_i}{\partial \zeta} \end{vmatrix}, \qquad (4)$$

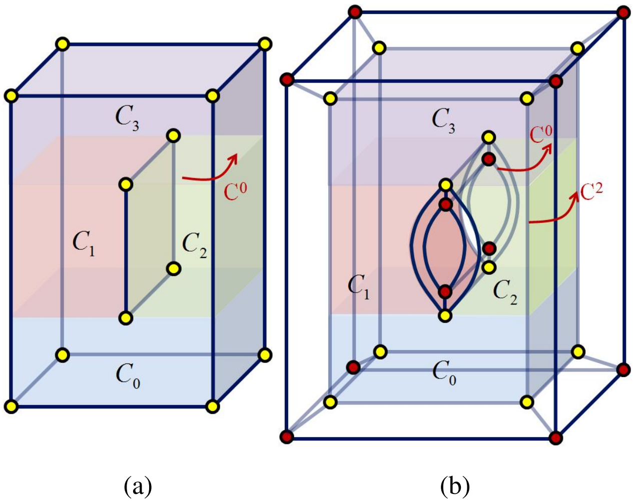

Fig. 9. The polycube before (a) and after (b) pillowing.

where N_i is a trilinear shape function. The scaled Jacobian is

$$J_s = \frac{J}{\| J_M(\cdot,0) \| \, \| J_M(\cdot,1) \| \, \| J_M(\cdot,2) \|}, \qquad (5)$$

where $J_M(\cdot,0)$, $J_M(\cdot,1)$ and $J_M(\cdot,2)$ represent the first, second and last column of the Jacobian matrix, J_M, respectively. To handle T-junctions during smoothing and optimization, we add some "virtual nodes" to refine the local region and convert the local T-mesh to a hexahedral mesh. Figure 8(d) shows the result after pillowing and optimization for the "Eight" model.

5.4. *Handling extraordinary and partial extraordinary nodes*

Unlike regular nodes, extraordinary nodes or partial extraordinary nodes may introduce gaps to the solid T-spline. In this step, we apply templates given in [22] to make the T-mesh gap-free. Figure 10(a) shows the template for a partial extraordinary node, in which the magenta edge has a reflection edge. Figure 10(b) shows one general template for an extraordinary node.

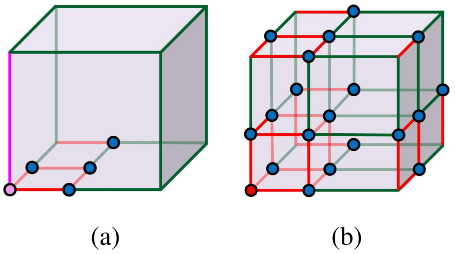

Fig. 10. The general template for a partial extraordinary node (a) and an extraordinary node (b). The magenta node is a partial extraordinary node, the red one is an extraordinary node, and the blue ones are inserted nodes. The magenta edge is the edge with a reflection edge and the red edges have zero knot interval.

6. Solid T-spline Construction and Bézier Extraction

In this stage, we aim to infer the local knot vectors for each node, build the rational solid T-spline from the T-mesh, and then extract embedded Bézier elements [2, 15]. The concept of rational T-splines was given in [22], with basis functions satisfying a partition of unity by definition. The rational solid T-spline is defined as

$$S(\xi,\eta,\zeta) = \frac{\sum_{i=0}^{n} w_i \overline{C}_i R_i(\xi,\eta,\zeta)}{\sum_{i=0}^{n} w_i R_i(\xi,\eta,\zeta)}, \quad (\xi,\eta,\zeta) \in \Omega, \tag{6}$$

where

$$R_i(\xi,\eta,\zeta) = \frac{N_i^{\xi}(\xi)N_i^{\eta}(\eta)N_i^{\zeta}(\zeta)}{\sum_{j=0}^{n} N_j^{\xi}(\xi)N_j^{\eta}(\eta)N_j^{\zeta}(\zeta)} \tag{7}$$

is the rational B-spline basis function, N_i^{ξ}, N_i^{η} and N_i^{ζ} are B-spline basis functions defined by the local knot vectors at node $\overline{C}_i$ which, for degree $d = 3$, are given by $\vec{\xi}_i = [\xi_{i0}, \xi_{i1}, \xi_{i2}, \xi_{i3}, \xi_{i4}]$, $\vec{\eta}_i = [\eta_{i0}, \eta_{i1}, \eta_{i2}, \eta_{i3}, \eta_{i4}]$ and $\vec{\zeta}_i = [\zeta_{i0}, \zeta_{i1}, \zeta_{i2}, \zeta_{i3}, \zeta_{i4}]$.

From the T-mesh, we first compute the knot intervals for each node by traversing T-mesh edges and faces. Note that we repeat knots whenever we meet an extraordinary node during the traverse. Then for each domain, we determine all the nodes with non-zero basis functions and use them to build the solid T-spline element. The entire solid T-spline model is built by looping over all the local domains.

Bézier extraction provides a finite element representation of T-splines for isogeometric analysis. Similar to [23], we first compute the "Bézier mesh", which includes all the reduced continuity lines by adding all the knot vector inference lines or isoparametric planes for T-junctions and L-junctions to the T-mesh. Then for each element in the Bézier mesh, the transformation matrix M^e between the T-spline basis functions and the Bézier basis functions is calculated satisfying:

$$B_t^e = M^e B_b^e, \tag{8}$$

where B_t^e is the vector formed by the nonzero T-spline basis functions in this element and B_b^e is the vector formed by the trivariate Bézier basis functions.

7. Results and Isogeometric Analysis

We have applied the construction algorithm to several models (Figures 11-14). The constructed solid T-spline is tricubic and C^2-continuous except in the vicinity of partial extraordinary and extraordinary nodes. Statistics for all the tested models are shown in Table 1. The Bézier Jacobian is calculated using the scaled Jacobian at the eight Gauss quadrature points for each Bézier element. The algorithm is efficient and all the results were computed on a PC equipped with an Intel X3470 processor and 8GB main memory.

The Isis model has genus zero and its polycube only contains one single cube. For the kitten model, there are three saddle points (one merging saddle point and two splitting saddle points). We only consider two of them, and the splitting saddle point close to the maximum point is skipped to get a more simplified polycube. The sculpture model has genus two and five saddle points. Again, we skipped the one near the maximal level. The time used for each model not only depends on the input mesh size, but also depends on the T-mesh size, which is determined by the given surface error tolerance and the complexity of the topology and geometry. One advantage of this harmonic function based T-spline construction is that the isoparametric lines are basically aligned with the geometric structure of the model.

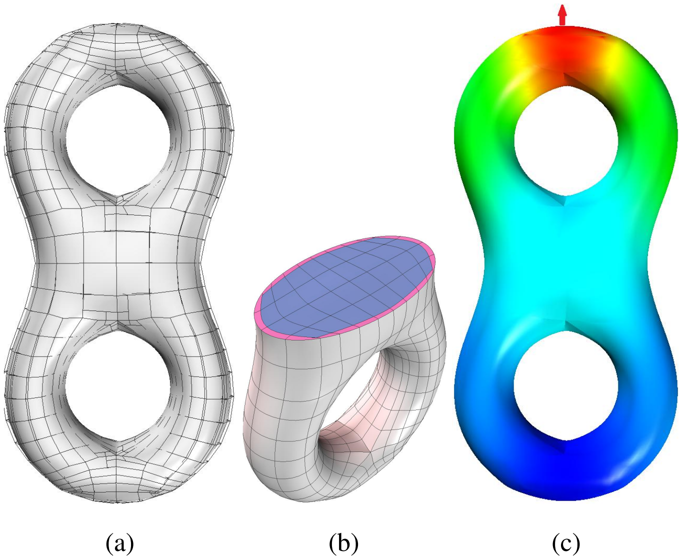

Fig. 11. The solid T-spline construction result for the "Eight" model. (a) The constructed solid T-spline and T-mesh; (b) the extracted solid Bézier mesh with some elements removed to show the interior (blue) and one pillowed layer (magenta); and (c) the isogeometric analysis result.

We have developed a 3D isogeometric analysis solver for static mechanics analysis, which uses rational T-splines as the basis, and we used it to test the obtained solid T-spline models. For all the models, we fix all the control points on the bottom and apply unit displacement on the top. The Young's modulus $E = 72.4GPa$, and the Poisson's ratio $\nu = 0.3$. The obtained displacement results are given in Figures 11(c), 12(e), 13(g) and 14(g). From these results, we can conclude that the obtained rational T-splines can be used in isogeometric analysis directly and reasonable simulation results can be obtained.

8. Conclusions

We have presented a new algorithm to construct solid T-splines for arbitrary-genus geometries from boundary triangulations. Our method is efficient and the resulting solid T-spline is analysis-suitable with C^2-continuity except for the local region around a few irregular nodes. In this paper, we only consider the geometries with Morse saddle points, and as part of our future work we will extend the algorithm to geometries with arbitrary saddle points. We will also consider engineering designs with sharp features.

Acknowledgements

Y. Zhang, W. Wang and L. Liu were supported in part by ONR Grant N00014-08-1-0653. T. J.R. Hughes was supported by ONR Grant N00014-08-1-0992, NSF GOALI CMI-

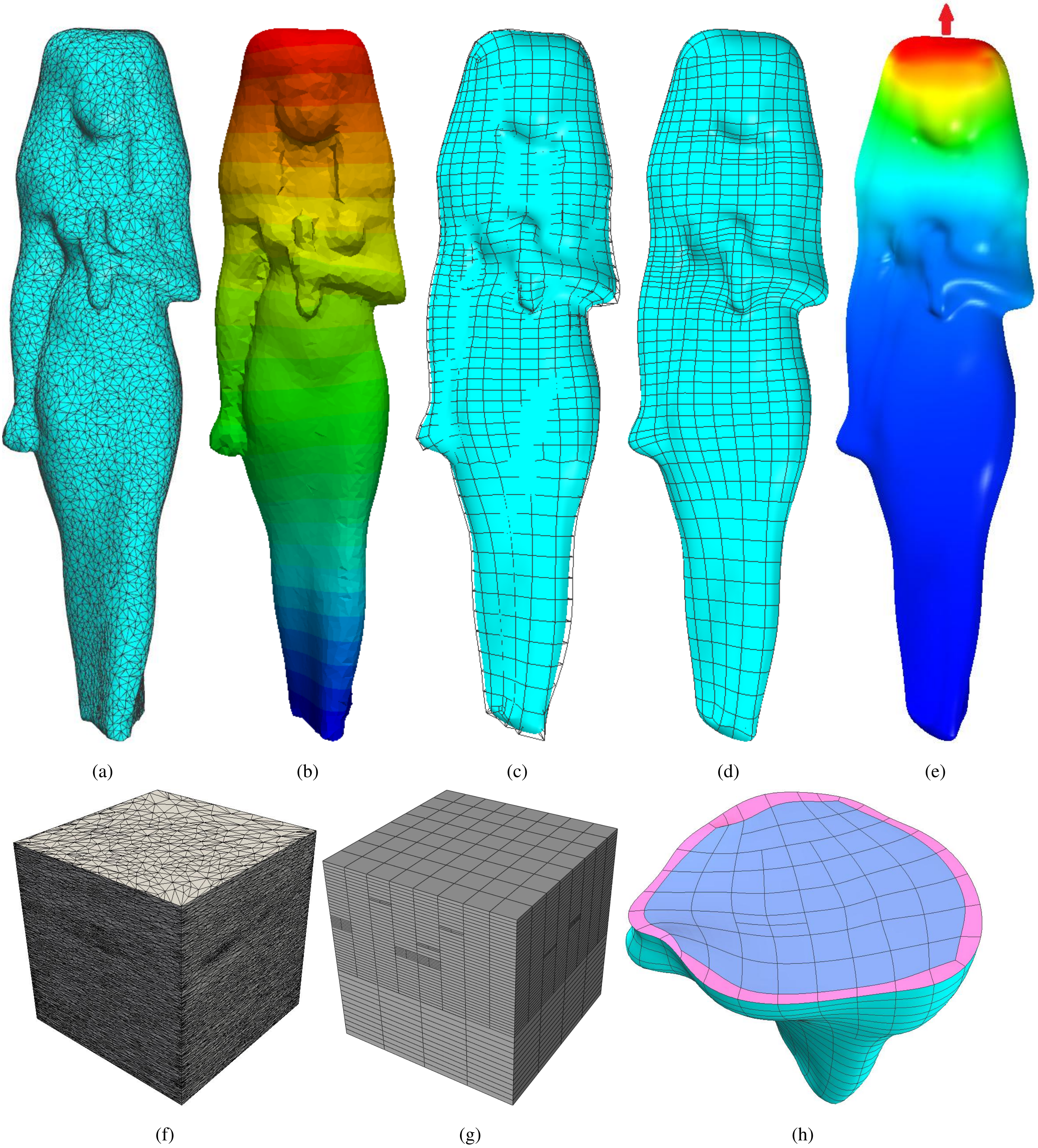

Fig. 12. Isis model with genus zero. (a) The input boundary triangle mesh; (b) the harmonic scalar field; (c) the constructed solid T-spline and T-mesh; (d) the extracted solid Bézier elements; (e) the isogeometric analysis result; (f) the constructed polycube and the mapping result; (g) the subdivision result for the parametric domain; and (h) the extracted solid Bézier mesh with some elements removed to show the interior (blue) and one pillowed layer (magenta).

Table 1. Statistics of all the tested models

Model	Genus	Input mesh	T-mesh	Extraordinary nodes	Interior partial	Bézier	Bézier Jacobian	Time
		(vertices, elements)	nodes	(Boundary, Interior)	extraordinary nodes	elements	(worst, best)	(s)
Isis	0	(5,863, 11,722)	9,310	(8, 8)	244	5,335	(0.12, 1.0)	52.0
Kitten	1	(5,377, 10,754)	4,825	(12, 12)	140	2,883	(0.08, 1.0)	14.8
Eight	2	(766, 1,536)	5,735	(16, 16)	200	1,440	(0.10, 1.00)	8.5
Sculpture	2	(8,635, 17,276)	10,549	(16, 16)	252	7,072	(0.09, 1.00)	41.5

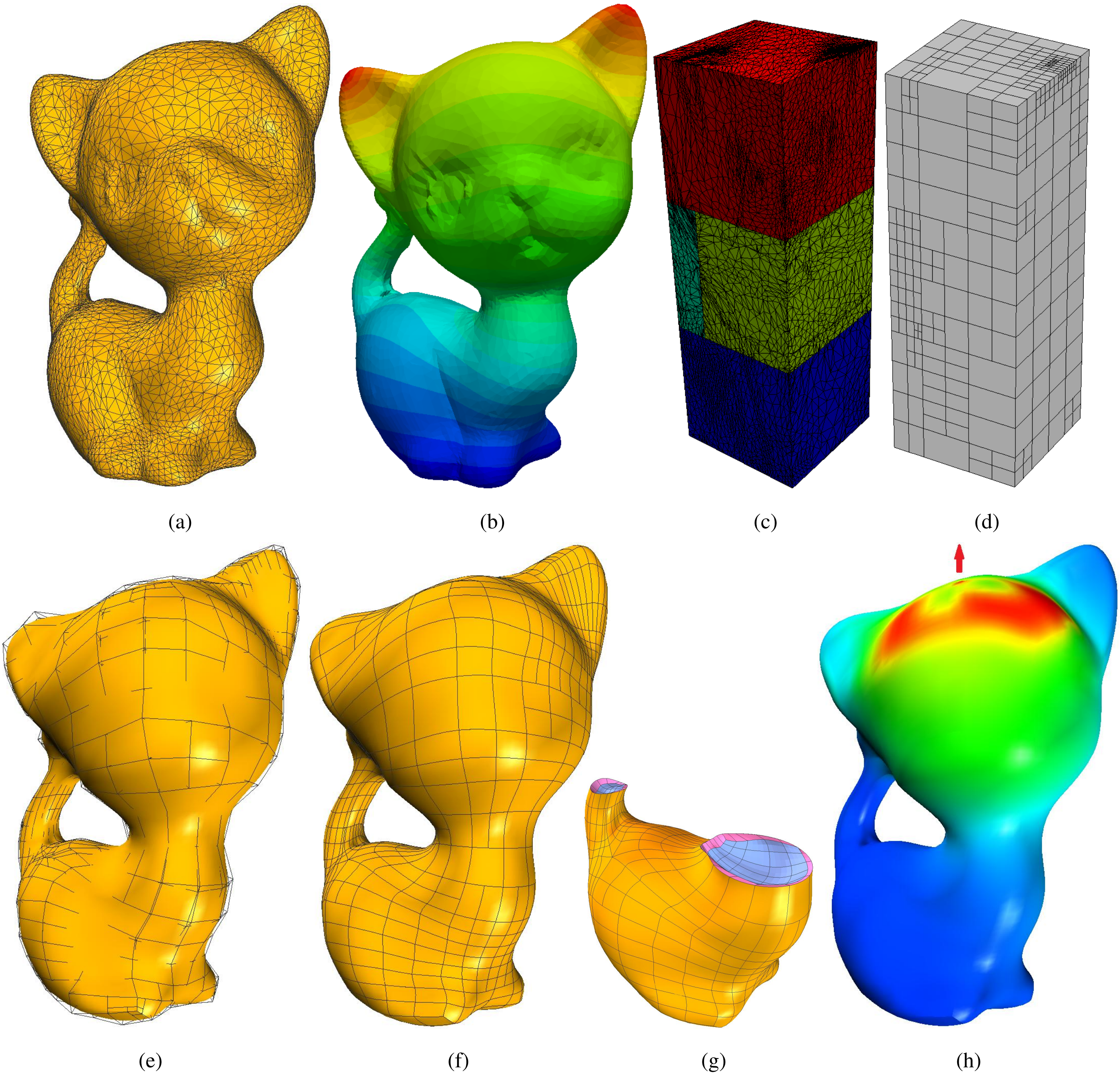

Fig. 13. Kitten model with genus one. (a) The input boundary triangle mesh; (b) the harmonic scalar field; (c) the constructed polycube and the mapping result; (d) the subdivision result for the parametric domain; (e) the constructed solid T-spline and T-mesh; (f) the extracted solid Bézier elements; (g) the extracted solid Bézier mesh with some elements removed to show the interior (blue) and one pillowed layer (magenta); and (h) the isogeometric analysis result.

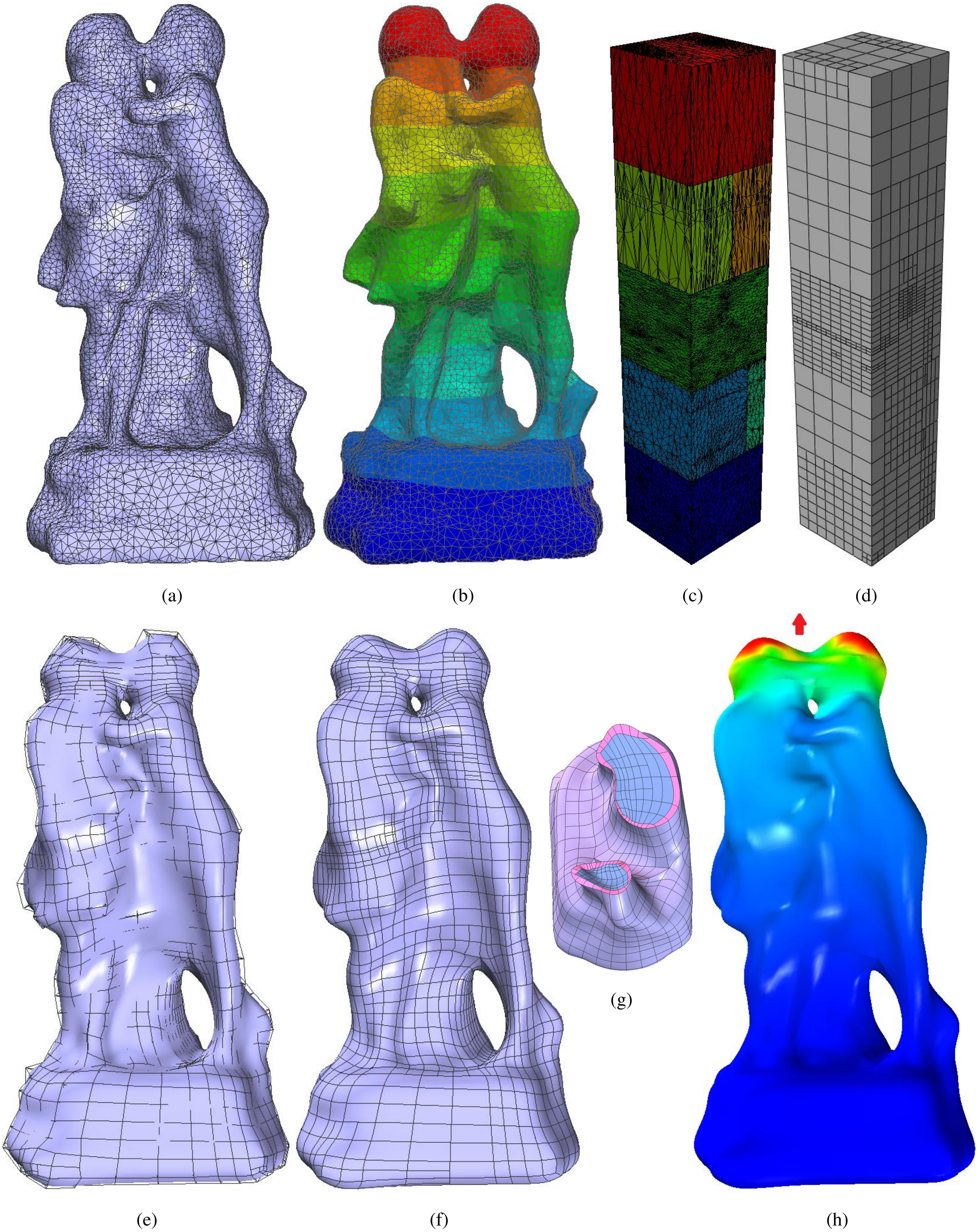

Fig. 14. Sculpture model with genus two. (a) The input boundary triangle mesh; (b) the harmonic scalar field; (c) the constructed polycube and the mapping result; (d) the subdivision result for the parametric domain; (e) the constructed solid T-spline and T-mesh; (f) the extracted solid Bézier elements; (g) the extracted solid Bézier mesh with some elements removed to show the interior (blue) and one pillowed layer (magenta); and (h) the isogeometric analysis result.

0700807/0700204, NSF CMMI-1101007 and a grant from SINTEF. The "Eight" model is from the AIM@SHAPE Shape Repository.

References

[1] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, and A. V. Vuong. Swept volume parameterization for isogeometric analysis. In *IMA International Conference on Mathematics of Surfaces XIII*, pages 19–44, 2009.

[2] M. J. Borden, M. A. Scott, J. A. Evans, and T. J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87:15–47, 2011.

[3] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aided Geom. Des.*, 22:392–423, 2005.

[4] J. M. Escobar, J. M. Cascón, E. Rodríguez, and R. Montenegro. A new approach to solid modeling with trivariate T-splines based on mesh optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(45-46): 3210–3222, 2011.

[5] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[6] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. *Advances in Multiresolution for Geometric Modelling*, pages 157–186, 2005.

[7] X. Gu and S. Yau. Global conformal surface parameterization. In *Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 127–137, 2003.

[8] X. Gu, Y. Wang, T.F. Chan, P.M. Thompson, and S.-T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transaction on Medical Imaging*, 23(8):949–958, 2004.

[9] Y. He, H. Wang, C. Fu, and H. Qin. A divide-and-conquer approach for automatic polycube map construction. *Computers & Graphics*, 33(3):369–380, 2009.

[10] J. Hua, Y. He, and H. Qin. Multiresolution heterogeneous solid modeling and visualization using trivariate simplex splines. In *ACM Symposium on Solid Modeling and Applications*, pages 47–58, 2004.

[11] T. J.R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[12] T. Kanai, H. Suzuki, and F. Kimura. 3D geometric metamorphosis based on harmonic maps. In *Pacific Graphics*, pages 97–104, 1997.

[13] B. Li, X. Li, K. Wang, and H. Qin. Generalized polycube trivariate splines. In *Shape Modeling International Conference*, pages 261–265, 2010.

[14] T. Martin, E. Cohen, and R. M. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design*, 26 (6):648–664, 2009.

[15] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88(2):126–156, 2011.

[16] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

[17] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. In *ACM SIGGRAPH*, pages 276–283, 2004.

[18] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2:105–171, 2006.

[19] M. Tarini, K. Hormann, P. Cignoni, and C. Montani. Polycube-maps. *ACM Trans. Graph.*, 23(3):853–860, 2004.

[20] H. Wang, Y. He, X. Li, X. Gu, and H. Qin. Polycube splines. In *Symposium on Solid and Physical Modeling*, pages 241–251, 2007.

[21] K. Wang, X. Li, B. Li, H. Xu, and H. Qin. Restricted trivariate polycube splines for volumetric data modeling. *IEEE Transactions on Visualization and Computer Graphics*, 18:703–716, 2012.

[22] W. Wang, Y. Zhang, G. Xu, and T. J.R. Hughes. Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. *Computational Mechanics, (2012) DOI: 10.1007/s00466-011-0674-6*.

[23] Y. Zhang, W. Wang, and T. J.R. Hughes. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering, (2012) DOI: 10.1016/j.cma.2012.01.014*.

[24] Y. Zhang, Y. Bazilevs, S. Goswami, C. L. Bajaj, and T. J.R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(29-30):2943–2959, 2007.