Comput Mech (2013) 51:1051–1059
DOI 10.1007/s00466-012-0787-6

ORIGINAL PAPER

Conformal solid T-spline construction from boundary T-spline representations

Yongjie Zhang · Wenyan Wang · Thomas J. R. Hughes

Received: 19 July 2012 / Accepted: 28 August 2012 / Published online: 14 September 2012

Abstract To achieve a tight integration of design and analysis, conformal solid T-spline construction with the input boundary spline representation preserved is desirable. However, to the best of our knowledge, this is still an open problem. In this paper, we provide its first solution. The input boundary T-spline surface has genus-zero topology and only contains eight extraordinary nodes, with an isoparametric line connecting each pair. One cube is adopted as the parametric domain for the solid T-spline. Starting from the cube with all the nodes on the input surface as T-junctions, we adaptively subdivide the domain based on the octree structure until each face or edge contains at most one face T-junction or one edge T-junction. Next, we insert two boundary layers between the input T-spline surface and the boundary of the subdivision result. Finally, knot intervals are calculated from the T-mesh and the solid T-spline is constructed. The obtained T-spline is conformal to the input T-spline surface with exactly the same boundary representation and continuity. For the interior region, the continuity is C^2 everywhere except for the local region surrounding irregular nodes. Several examples are presented to demonstrate the performance of the algorithm.

Keywords Solid T-spline · Conformal boundary · Genus-zero topology · Octree · Boundary layer

Y. Zhang (✉) · W. Wang
Department of Mechanical Engineering, Carnegie Mellon University Pittsburgh, Pittsburgh, PA 15213, USA
e-mail: jessicaz@andrew.cmu.edu

T. J. R. Hughes
Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA

1 Introduction

Spline boundary representation is the most popular technology for solid modeling in computer aided design (CAD). To integrate design and analysis, the root idea of isogeometric analysis [2,7], one challenge is to automatically create a conformal solid NURBS/T-spline model with the given spline boundary representation preserved exactly. To the best of our knowledge, this paper presents the first attempt to construct conformal solid T-splines from boundary spline representations.

The advent of T-splines [10,11] provides not only more flexibility for modeling, but also an ideal discretization for isogeometric analysis [9]. Several methods have been developed for solid spline modeling. Hua et al. [6] developed a trivariate simplex spline model which is defined over a tetrahedral decomposition of any 3D domain with complicated geometry and arbitrary topology. In [14], a skeleton-based solid NURBS construction method for patient-specific vascular geometric models was presented. In [1], a swept volume parametrization was built for isogeometric analysis. A framework was developed in [8] to model a single trivariate B-spline from input boundary triangle meshes with genus-zero topology. Escobar et al. [5] proposed a solid T-spline modeling algorithm based on optimization from the surface triangulation. In our earlier work, we developed rational T-splines whose basis functions satisfy partition of unity by definition [13]. In addition, we developed solid T-spline construction methods for both genus-zero geometry [15] and high genus geometry [12] from boundary triangle meshes, using a unit cube or a polycube as the parametric domain. These developments provide a foundation upon which the conformal trivariate spline model can be built from spline boundary surface representations.

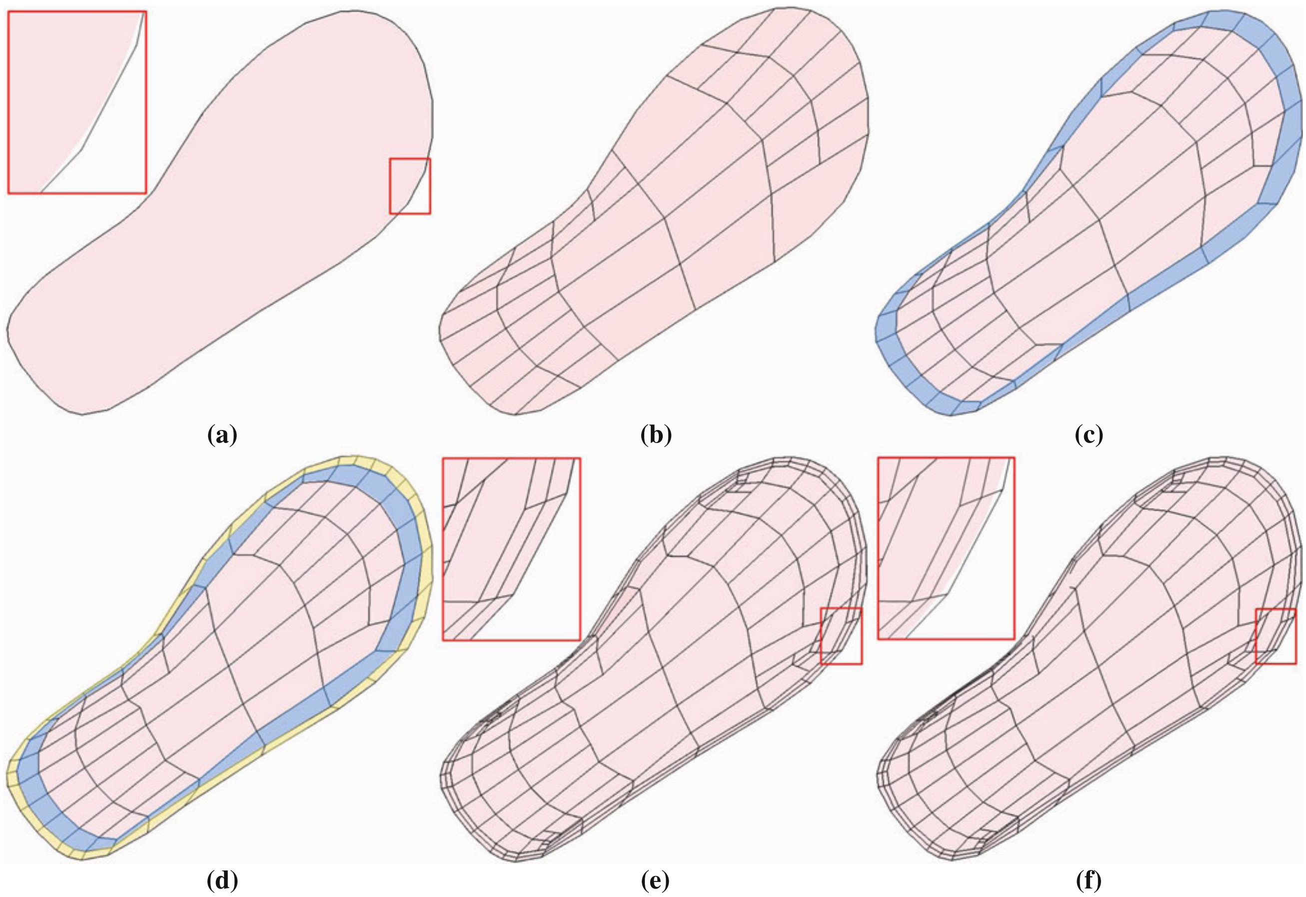

Fig. 1 *Flow chart* of conformal solid T-spline construction from given boundary NURBS/T-spline surface representations with genus-zero topology and eight extraordinary nodes. **a** The input boundary; **b** octree subdivision and projection; **c** first boundary layer construction; **d** second boundary layer construction; **e** handling irregular nodes; and **f** the constructed solid T-spline with T-mesh

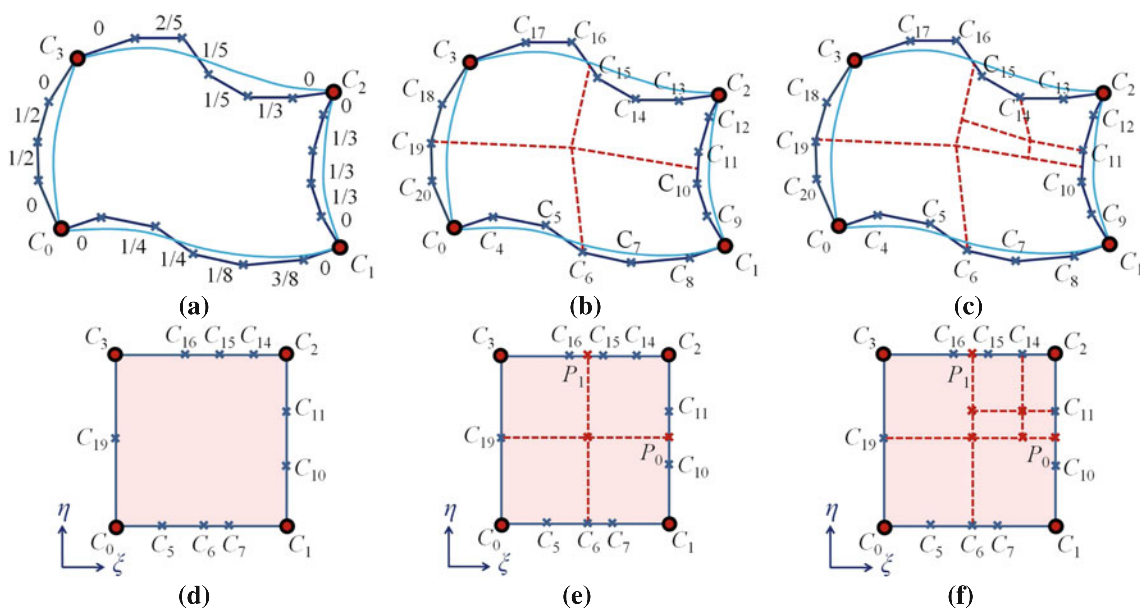

Fig. 2 The subdivision process for one 2D example. **a** The input boundary *curve* with its control *polygon* in the physical space; **b** subdivision result after one refinement; and **c** final subdivision result. **d**–**f** The corresponding result in the parametric space for **a**–**c**, respectively.

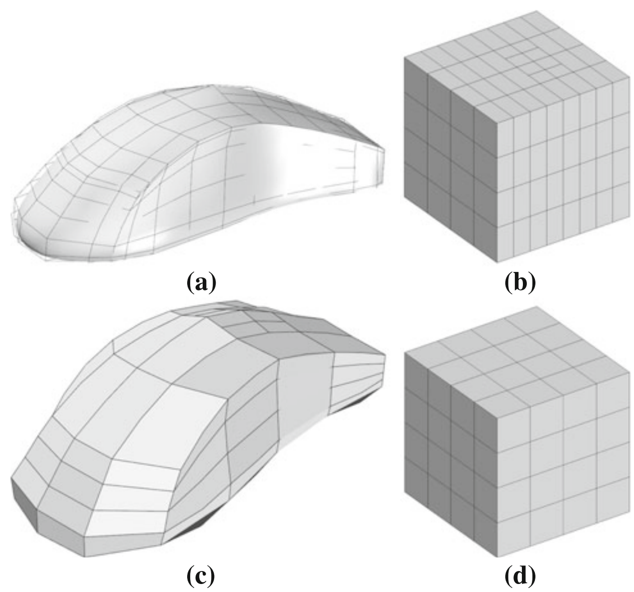

Fig. 3 The subdivision result for the Mouse model. **a** The input T-spline surface; **b** the parametric mapping result; **c** the subdivision result in the physical space; and **d** the subdivision result in the parametric space

A general methodology for constructing a conformal solid T-spline from boundary T-spline/NURBS representations is very complex due to complicated geometry and topology, various T-mesh or NURBS patch connectivity, and various parametrizations. The conformal property is hard to maintain during the process of converting a bivariate surface model to a trivariate solid model. Here, we focus on geometry with genus-zero topology. In addition, the input T-spline surface only contains eight extraordinary points.

In this paper, we propose a novel method to construct conformal solid rational T-splines for genus-zero geometry, with the input boundary T-spline representation preserved exactly. We use one cube as the parametric domain for the solid T-spline. Also, an adaptive subdivision is applied to the cube and the boundary nodes obtained in the subdivision are projected to the input T-spline surface via the surface parametrization. To conform to the input boundary, two boundary layers are generated between the input boundary surface and the boundary of the subdivision result. Templates for irregular nodes are then applied to get a gap-free T-mesh. Finally, a solid rational T-spline is built from the T-mesh and solid Bézier elements are extracted.

The remainder of this paper is organized as follows. Section 2 presents an overview of the method. The octree subdivision algorithm is described in Sect. 3 and the boundary layer method is explained in Sect. 4. Section 5 discusses the method to deal with irregular nodes and solid T-spline construction. Section 6 presents some examples and Sect. 7 draws conclusions.

2 Problem description and algorithm overview

Figure 1 shows the pipeline of conformal solid T-spline construction. The input can be a regular T-spline surface,

$$S(\xi,\ \eta) = \frac{\sum_{i=0}^{n} w_i C_i B_i(\xi,\ \eta)}{\sum_{i=0}^{n} w_i B_i(\xi,\ \eta)}, \qquad (\xi,\ \eta) \in \Omega, \tag{1}$$

or a rational T-spline surface,

$$S(\xi,\ \eta) = \frac{\sum_{i=0}^{n} w_i C_i R_i(\xi,\ \eta)}{\sum_{i=0}^{n} w_i R_i(\xi,\ \eta)}, \qquad (\xi,\ \eta) \in \Omega, \tag{2}$$

where w_i is the weight for the control point C_i, $B_i(\xi,\ \eta) = N_i^{\xi}(\xi) N_i^{\eta}(\eta)$, N_i^{ξ} and N_i^{η} are B-spline basis functions defined by two local knot vectors at node C_i, $R_i(\xi,\ \eta)$ is the rational B-spline basis function, $R_i(\xi,\ \eta) = \dfrac{N_i^{\xi}(\xi) N_i^{\eta}(\eta)}{\sum_{j=0}^{n} N_j^{\xi}(\xi) N_j^{\eta}(\eta)}$, and Ω is the domain of the T-spline surface. For our conformal solid T-spline construction, the input surface should satisfy the following requirements:

– Its topology has to be genus-zero, and we use one cube $\mathcal{C}$ as the parametric domain of the solid T-spline.

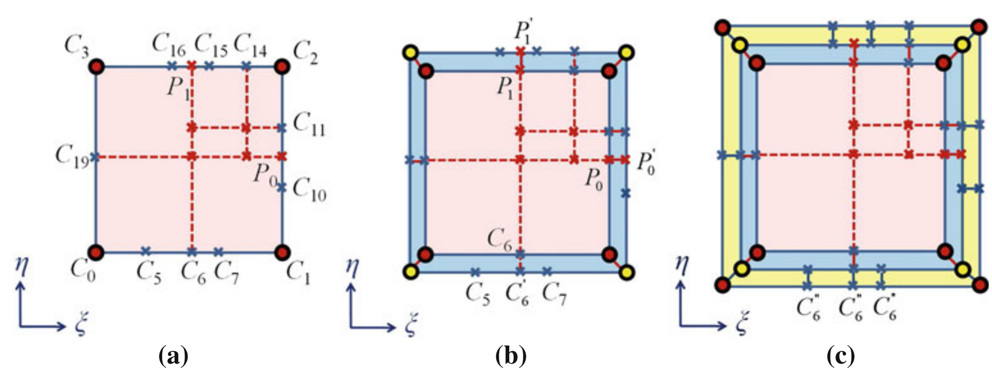

Fig. 4 The boundary layer construction operation for a 2D example. **a** The subdivision result; **b** the first boundary layer for the subdivision boundary; and **c** the second boundary layer for the input boundary surface

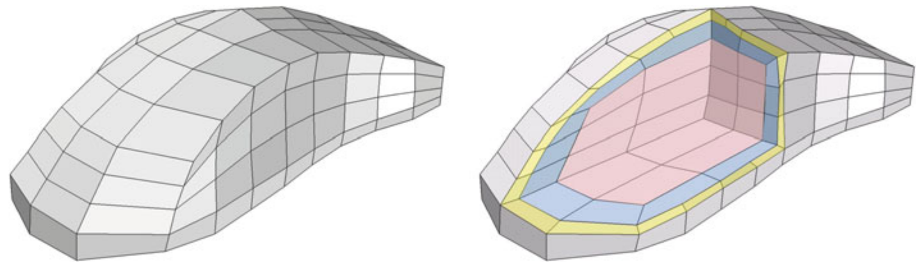

Fig. 5 The boundary layer construction result for the Mouse model. **a** The T-mesh after boundary layer construction; and **b** the interior of the T-mesh. The *pink*, *blue* and *yellow* regions are the subdivision result, the first boundary layer and the second boundary layer, respectively (Color figure online)

– The input surface must contain eight extraordinary nodes C_i $(i = 0, \ldots, 7)$ and there is an isoparametric line between each pair of extraordinary nodes $\overline{C_i C_j}$; Extraordinary nodes in 2D are irregular ones with a valence other then four. Each extraordinary node here is mapped to one corner of the cube $\mathcal{C}$.
– By splitting the input surface using the isoparametric lines $\overline{C_i C_j}$, each face of the cube $\mathcal{C}$ can serve as the parametric domain for its corresponding patch.

In summary, the boundary of the cube can serve as the parametric domain of the input boundary surface.

As shown in Fig. 1, given the input NURBS/T-spline surface shown in (a), we first use octree subdivision to obtain an initial T-mesh shown in (b). Then two boundary layers shown in (c) and (d) are built in order to preserve the input boundary representation. Then, we deal with irregular nodes and also insert zero knot interval edges to get a valid T-mesh. Finally, we calculate the knot vectors for each node and construct trivariate solid T-splines.

3 Octree subdivision and projection

We use a cube $\mathcal{C}$ as the parametric domain of the constructed solid T-spline. The eight extraordinary nodes correspond to the eight corners of the cube. The twelve isoparametric curves connecting these extraordinary nodes are associated with the edges of the cube. Based on these curves, we divide the input surface into six patches, and each one is associated with one face of $\mathcal{C}$. For the input NURBS/T-spline surface, the parametric domain of each NURBS/T-spline patch corresponds to one face of the cube $\mathcal{C}$. Then we build an initial T-mesh by applying the adaptive octree subdivision.

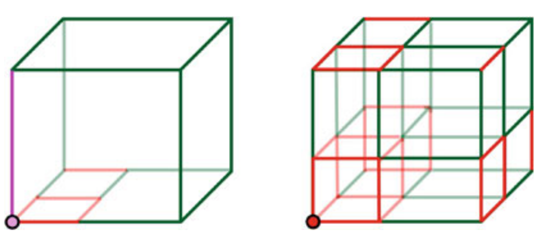

Fig. 6 Templates for a partial extraordinary node (**a**) and an extraordinary node (**b**). The *magenta* node is a partial extraordinary node, and the *red* node is an extraordinary node. The *magenta* edge is the edge with a reflection edge and the *red* edges have zero knot interval (Color figure online)

We first assign all the nodes in the input T-spline surface except the extraordinary nodes as T-junctions for the unit cube $\mathcal{C}$. Starting from $\mathcal{C}$ with T-junctions, each element is refined recursively into eight until each boundary element contains at most one face T-junction on a face, and at most one edge T-junction on an edge. Two nodes with the same parametric value are treated as one. For solid T-splines, one *edge T-junction* is a T-junction lying on one edge, and one *face T-junction* is a T-junction lying on one face. For each refinement, instead of always using the central parameter value, one parameter value of the T-junctions in this element is chosen to refine the element. If there are several T-junction parametric values in one direction, the one closest to the middle is chosen. In this way, we try to minimize the number of T-junctions. The strongly-balanced octree is adopted here. During subdivision, the newly generated boundary nodes are projected to their corresponding positions on the input T-spline surface. The interior nodes are relocated to the mass center of all neighboring elements. Then an initial T-mesh is generated.

Figure 2 demonstrates the subdivision process using a 2D example. (a) is the input B-spline boundary curve. The dark blue polygon shows the boundary control mesh and the light blue curves are the B-spline boundary, which has four sharp corners rendered in red. The values labelled around the edges show the knot intervals. (d) shows the parametric domain for the solid T-spline, and we assign nodes in the input curve as T-junctions for the parametric domain. (b) and (e) are the first refinement results in the physical and parametric space, respectively, in which the parametric values of C_6 and C_{19} are used for ξ and η directions since they are closest to the parametric center for this element. Then we find that the right upper element contains two T-junctions in the ξ direction (C_{14} and C_{15}), hence we subdivide that element again. This process continues until we obtain the result shown in (c) and (f), for which each boundary element contains only one T-junction in each direction except for the nodes which have the same parametric position with the corner nodes (C_0, C_1, C_2 and C_3).

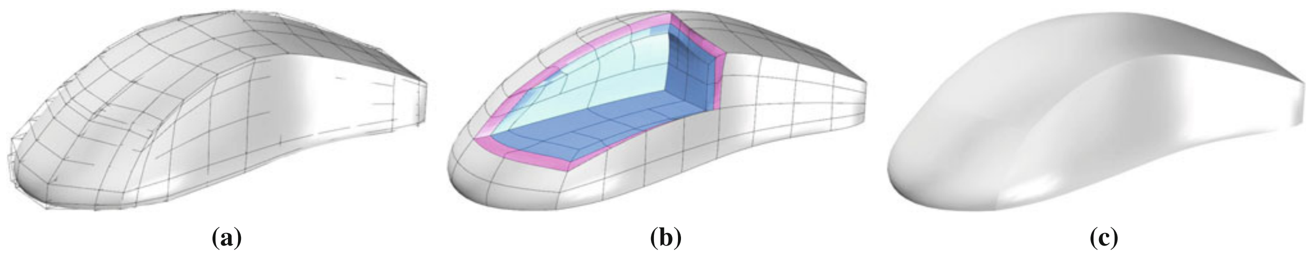

Fig. 7 Mouse model. **a** The constructed solid T-spline and T-mesh; **b** the extracted solid Bézier elements with some elements removed to show the interior mesh; and **c** the solid T-spline

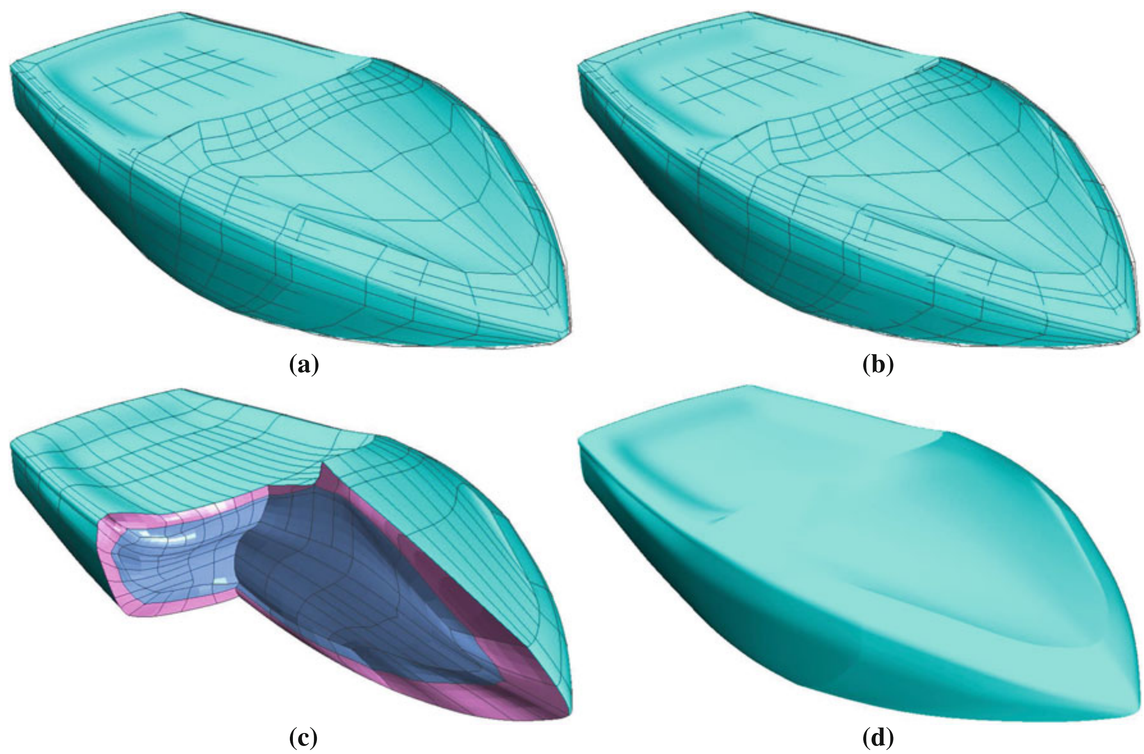

Fig. 8 Boat model. **a** The input T-spline boundary surface with the control mesh; **b** the constructed solid T-spline and T-mesh; **c** the extracted solid Bézier elements with some elements removed to show the interior mesh; and **d** the constructed solid T-spline

Figure 3 shows the subdivision result for a Mouse model. (a) and (b) show the input T-spline surface in the physical space and the parametric space, respectively. (c) and (d) are the subdivision results.

4 Boundary layer construction

We insert two boundary layers between the subdivision result and the input boundary surface in order to make the solid T-spline conformal to the input surface and also improve the quality of the T-mesh. Figure 4 illustrates the boundary layer construction using the previous 2D example, in which (a) is the subdivision result, (b) is the boundary layer result for the subdivision boundary, and (c) is the boundary layer result for the input boundary surface. First, we duplicate each boundary node lying on the element corner, insert one element for each boundary face, and obtain one boundary layer (rendered in blue in (b)). We assign the boundary T-junctions to the newly generated boundary faces. As shown in (b), the boundary T-junctions, such as C_5, C_7, C_{10}, C_{15} and C_{16}, are assigned to the new boundary after boundary layer construction. Second, we duplicate the boundary nodes which have the same parametric position with one of the input control points, such as C_5, $C_6^{'}$ and C_7. The physical position is assigned to be the same as the input. The second pillowed layer is rendered in yellow in (c). Next, we use smoothing and optimization techniques to relocate the physical positions of the interior nodes to improve the T-mesh quality.

Figure 5 shows the result after boundary layer construction for the Mouse model. In (b), the pink region is the subdivision

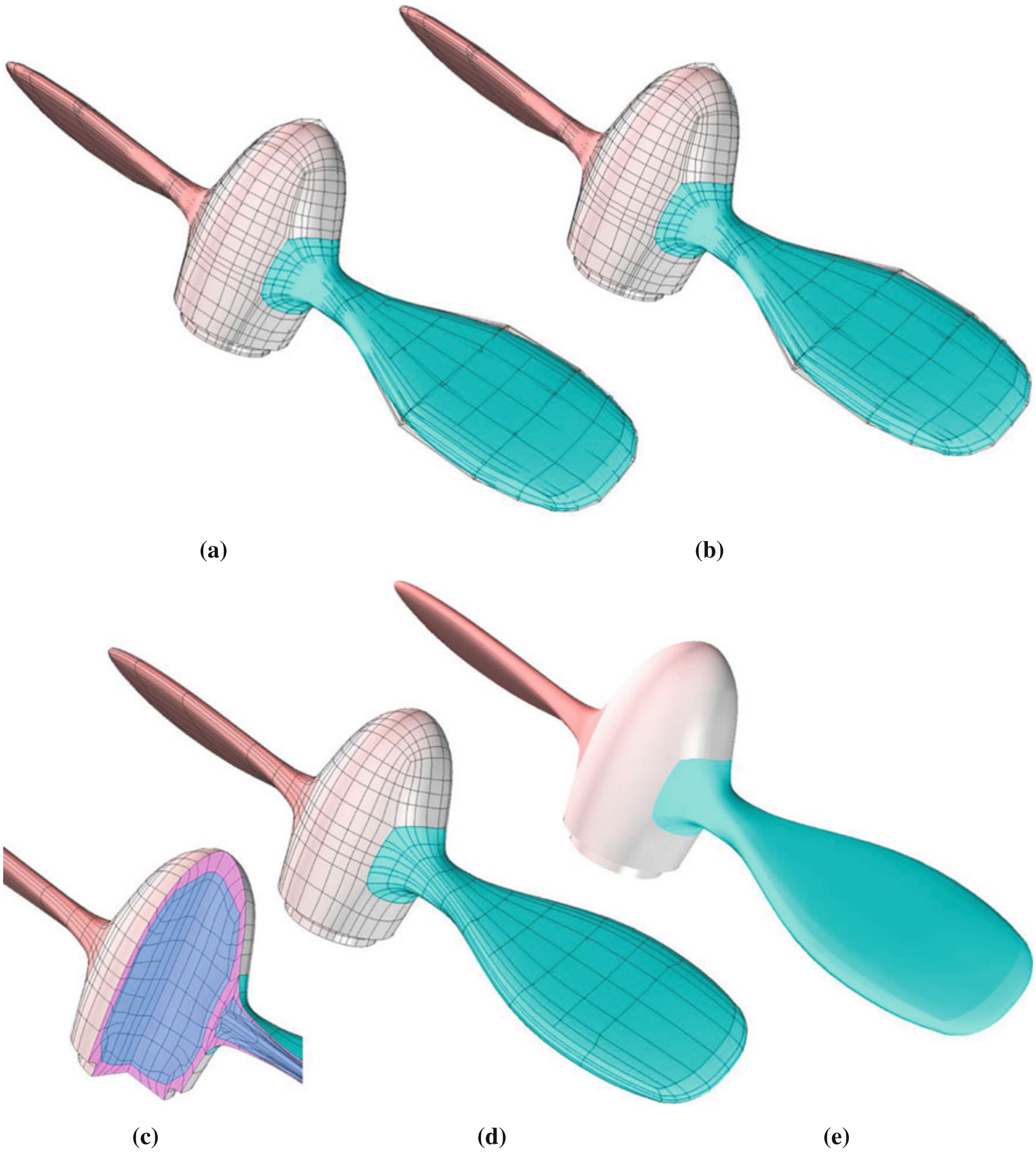

Fig. 9 Propeller model. **a** The input T-spline boundary surface with the control mesh; **b** the constructed solid T-spline and T-mesh; **c** the extracted solid Bézier elements with some elements removed to show the interior mesh; **d** the extracted solid Bézier elements; and **e** the constructed solid T-spline

part, the blue layer is the boundary layer for the subdivision result, and the yellow layer is the second boundary layer. These two layers serve as a transition region between the input boundary spline surface and the subdivision result. From this figure, we can observe that the extraordinary nodes lie in the interior and have two layers of elements away from the boundary and we generated some T-junctions between the boundary layers. We can guarantee that after this boundary layer construction step, the boundary of the solid T-mesh is exactly the same as the input. In other words, the constructed solid T-spline is conformal to the input NURBS/T-spline surface. We formalize these statements in a lemma.

Lemma 1 *Due to the boundary layer construction, the obtained solid T-spline is conformal to the input NURBS/T-spline surface. In other words, the input NURBS/T-spline surface is preserved exactly.*

Proof Due to the boundary layer construction, the second boundary layer can be treated as one hexahedral sheet. For each element in the sheet we first insert one face parallel

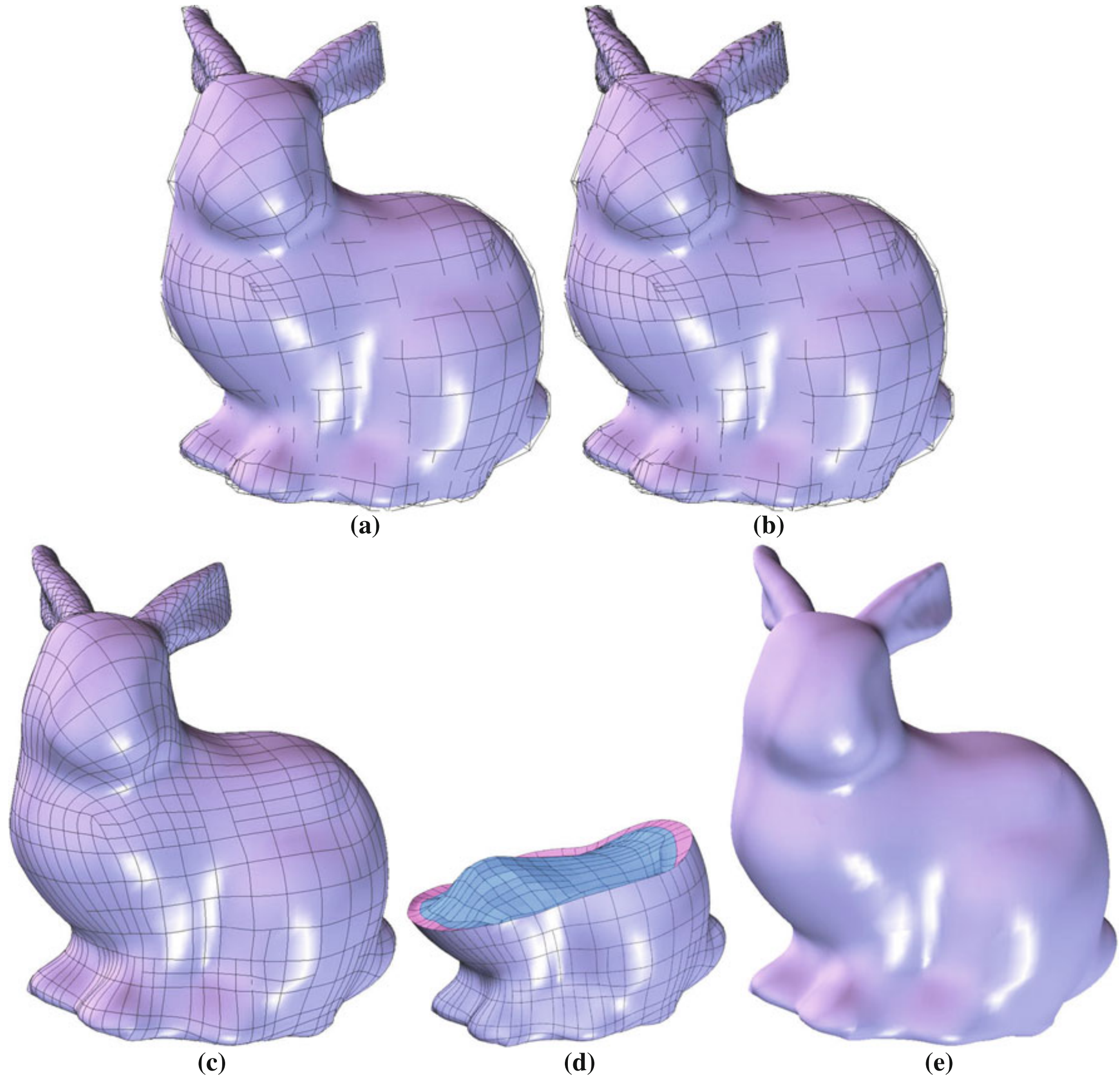

Fig. 10 Bunny model. **a** The input T-spline boundary surface with the control mesh; **b** the constructed solid T-spline and T-mesh; **c** the extracted solid Bézier elements; **d** the extracted solid Bézier elements with some elements removed to show the interior mesh; and **e** the constructed solid T-spline

Table 1 Statistics of all the tested models

Model	Input surface (nodes, elements)	T-mesh nodes	Interior extraordinary nodes	Interior partial extraordinary nodes	Bézier elements	Bézier Jacobian (worst, best)	Time (s)
Mouse	(320, 164)	1,429	8	44	454	(0.09, 1.00)	1.9
Boat	(764, 420)	2,852	8	56	1,375	(0.07, 1.00)	5.2
Propeller blade (NURBS)	(442, 256)	1,763	8	44	672	(0.02, 1.00)	3.2
Propeller blade (T-spline)	(462, 272)	2,083	8	64	1,072	(0.03, 1.00)	4.8
Propeller shaft	(1002, 512)	3,875	8	84	1,556	(0.05, 1.00)	10.3
Bunny	(1358, 1083)	7,537	8	85	7,901	(0.01, 1.00)	32.1

to the boundary face and all the edges connecting them are assigned with zero knot intervals. In this way, only the boundary nodes have non-zero basis function values on the solid T-spline boundary and the boundary surface will be defined by the boundary nodes only. In addition, since for the second boundary layer we only duplicate the nodes originated from the input surface, the solid T-mesh boundary is exactly the same as the input control polygon. In the following steps we will not modify the boundary of the T-mesh. Therefore, we can conclude that the obtained solid T-spline is conformal to the input T-spline surface, or that is, the input boundary representation is preserved. □

5 Irregular nodes and solid T-spline construction

To obtain a gap-free T-mesh, we apply templates given in [13] to the interior irregular nodes. There are two kinds of irregular nodes: extraordinary nodes and partial extraordinary nodes. Figure 6a shows the template for a partial extraordinary node, in which the magenta edge has a reflection edge. Figure 6b is a template for an extraordinary node. We insert nodes at all the intersection positions for each element containing irregular nodes.

Note that, in the obtained solid T-mesh, the layer right below the boundary only contains partial extraordinary nodes, and the edges which have reflection edges with respect to these partial extraordinary nodes are all topologically perpendicular to the T-mesh boundary. All the interior extraordinary nodes are at least two elements away from the boundary due to the two boundary layers. Hence, we will not insert new nodes on the boundary to handle extraordinary and partial extraordinary nodes, and we preserve the conformal boundary property in this step.

After dealing with the irregular nodes, the local knot vectors for each node are inferred and for each domain we detect all the nodes with non-zero basis functions and use them to build the solid T-spline element. A rational T-spline was defined in [13] whose basis functions satisfy partition of unity by definition. The entire solid T-spline model is built by looping over all the local domains. For the constructed solid T-spline, the continuity on the boundary is the same as the input, and the continuity is C^0 around the extraordinary nodes or partial extraordinary nodes, and C^2 everywhere else in the interior region. For the obtained solid T-spline, we apply Bézier extraction which provides a finite element representation of T-splines for isogeometric analysis [3,9]. To construct analysis-suitable T-splines, all the extracted Bézier elements must have a positive Jacobian [15].

6 Results and discussion

We have applied our algorithm to several models (Figs. 7, 8, 9, 10). Statistics for all the tested models are listed in Table 1. The results were computed on a PC equipped with an Intel X3470 processor and 8 GB main memory.

The input Mouse and Boat models were designed using the commercial CAD software Rhinoceros and the inputs are T-spline surfaces. Note that the constructed solid T-spline can preserve all the sharp features in the input model. The Propeller model is a NURBS/T-spline assembly that contains three parts: two blades and one shaft. The input model is watertight around the shared boundary of the three parts, and our constructed solid T-spline preserves this property exactly. The conformal solid T-spline construction can be used for assembly models without changing its mechanical fit between neighboring parts. The input of the Bunny model was generated using earlier work in [15].

For all these models the obtained solid T-spline has exactly the same spline representation as the input surface, and the obtained trivariate T-spline has all positive Bézier Jacobians. The constructed solid T-spline is tricubic and C^2-continuous except in the vicinity of partial extraordinary and extraordinary nodes in the interior. On the boundary, the solid T-spline has the same continuity as the input NURBS/T-spline surface. In addition, the presented method is not restricted to NURBS and T-splines, it can also apply to models using Locally Refined B-splines [4].

7 Conclusions

We presented a novel method to construct conformal solid T-splines for genus-zero geometry from a boundary T-spline surface. Based on the boundary layer construction technique, the solid T-spline has the same spline representation as the input spline surface and can preserve sharp features in the input model. Our method is efficient and robust. In the future, we intend to extend the algorithm to more general geometries with arbitrary topology and construct analysis-suitable solid T-splines.

Acknowledgments We would like to thank L. Liu for the help in generating the Bunny and the Propeller models. The work of Y. Zhang and W. Wang was supported by Y. Zhang's ONR-YIP Award N00014-10-1-0698 and an ONR Grant N00014-08-1-0653. The work of T. J. R. Hughes was supported by ONR Grant N00014-08-1-0992, NSF GOALI CMI-0700807/0700204, NSF CMMI-1101007 and a SINTEF grant UTA10-000374.

References

1. Aigner M, Heinrich C, Jüttler B, Pilgerstorfer E, Simeon B, Vuong AV (2009) Swept volume parameterization for isogeometric analysis. In: IMA international conference on mathematics of surfaces XIII, pp 19–44
2. Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW (2010) Isogeometric analysis using T-splines. Comput Methods Appl Mech Eng 199(5–8):229–263
3. Borden MJ, Scott MA, Evans JA, Hughes TJR (2011) Isogeometric finite element data structures based on Bézier extraction of NURBS. Int J Numer Methods Eng 87:15–47

4. Dokken T, Lyche T, Pettersen KF (2012) Polynomial splines over locally refined box-partitions. Comput Aided Geom Des (submitted)
5. Escobar JM, Cascón JM, Rodríguez E, Montenegro R (2011) A new approach to solid modeling with trivariate T-splines based on mesh optimization. Comput Methods Appl Mech Eng 200(45–46):3210–3222
6. Hua J, He Y, Qin H (2004) Multiresolution heterogeneous solid modeling and visualization using trivariate simplex splines. In: ACM symposium on solid modeling and applications, pp 47–58
7. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Comput Methods Appl Mech Eng 194:4135–4195
8. Martin T, Cohen E, Kirby RM (2009) Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Comput Aided Geom Des 26(6):648–664
9. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR (2011) Isogeometric finite element data structures based on Bézier extraction of T-splines. Int J Numer Methods Eng 88(2):126–156
10. Sederberg TW, Zheng J, Bakenov A, Nasri A (2003) T-splines and T-NURCCs. ACM Trans Graph 22(3):477–484
11. Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T (2004) T-spline simplification and local refinement. In: ACM SIGGRAPH, pp 276–283
12. Wang W, Zhang Y, Liu L, Hughes TJR (2012a) Solid T-spline construction from boundary triangulations with arbitrary genus topology. In: In ACM symposium on solid and physical modeling (accepted)
13. Wang W, Zhang Y, Xu G, Hughes TJR (2012b) Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. Comput Mech 50(1):65–84
14. Zhang Y, Bazilevs Y, Goswami S, Bajaj CL, Hughes TJR (2007) Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. Comput Methods Appl Mech Eng 196(29–30):2943–2959
15. Zhang Y, Wang W, Hughes TJR (2012) Solid T-spline construction from boundary representations for genus-zero geometry. Comput Methods Appl Mech Eng. doi:10.1016/j.cma.2012.01.014