

Constructing hexahedral shell meshes via volumetric polycube maps

Shuchu Han, Jiazhi Xia, Ying He*

School of Computer Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:

Solid modeling
Shell object
Volume parameterization
Hexahedral meshing
Harmonic map
Polycube map

ABSTRACT

Shells are three-dimensional structures. One dimension, the thickness, is much smaller than the other two dimensions. Shell structures can be widely found in many real-world objects. This paper presents a method to construct a layered hexahedral mesh for shell objects. Given a closed 2-manifold and the user-specified thickness, we construct the shell space using the distance field and then parameterize the shell space to a polycube domain. The volume parameterization induces the hexahedral tessellation in the object shell space. As a result, the constructed mesh is an all-hexahedral mesh in which most of the vertices are regular, i.e., the valence is 6 for interior vertices and 5 for boundary vertices. The mesh also has a layered structure, so that all layers have exactly the same tessellation. We prove that our parameterization is guaranteed to be bijective. As a result, the constructed hexahedral mesh is free of degeneracy, such as self-intersection, flip-over, etc. We also show that the iso-parametric line (in the thickness dimension) is orthogonal to the other two iso-parametric lines. We apply our algorithm to numerous real-world models of various geometry and topology. The promising experimental results demonstrate the efficacy of our algorithm. Although our main focus is to construct a hexahedral mesh by using volumetric polycube parameterization, the proposed framework is general that can be applied to other regular domains, such as cylinder and sphere, which is also demonstrated in the paper.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Finite element analysis is an essential tool to model various scientific and engineering phenomena, such as structural mechanics, heat flow, computational fluid dynamics, etc. An important requirement of the numerical approximation and simulation is to convert the solid model to a discrete mesh composed of smaller elements. The most common types of elements are tetrahedral and hexahedral elements. A 3D domain cannot always be meshed into hexahedral elements. However, it can be decomposed into tetrahedral elements more easily than into hexahedral elements. Thus, tetrahedral elements gain more popularity in finite element analysis. However, there are certain applications for which hexahedral elements are preferred. For example, tetrahedral meshes typically require 4–10 times more elements than a hexahedral mesh to obtain the same level of accuracy [1]. In nonlinear elastic–plastic analysis, the linear hexahedral elements may be superior even to quadratic tetrahedral elements when shear stress is dominant [2].

Constructing hexahedral meshes is usually more challenging than tetrahedral meshes [3]. In this paper, we focus on the shell objects that are three-dimensional structures wherein one dimension, the thickness, is much smaller than the other two

dimensions. These structures are widely used in manufacturing automobile bodies, sheet metal parts, etc. To construct a hexahedral shell mesh, we use the existing method, e.g., [4], computing the chordal surface by cutting the mesh of the input CAD model at its mid-plane, then constructing a quadrilateral mesh for the chordal surface. Finally, two-way mapping between the chordal surface and the boundary is used to sweep the quad elements from the chordal surface onto the boundary, resulting in a layered all-hex mesh. This method works well for shells with simple geometry and constant thickness, such that the top and bottom surfaces are similar. However, it may fail on models with complex geometry/topology and variable thicknesses, in which the chordal surface is difficult to compute. Furthermore, there is no guarantee that the resulting hexahedral mesh is free of degeneracy, such as flip-over, self-intersection, etc.

In this paper, we present a novel method to construct a layered hexahedral mesh for shell objects. Our method parameterizes the shell object into a shelled polycube in which a hexahedral tessellation can be easily constructed. The parameterization is guaranteed to be a bijection, and thus induces a hexahedral meshing on the shell object. Our method does not require computing of the medial or the chordal surface. Instead, we compute a bijective map between the outer boundary surfaces of the shell object and the polycube domain. Thus, our method works for models with complicated geometry and variable thicknesses. Fig. 1 shows the layered hexahedral mesh of the shelled Bunny model. One can see the high quality of the constructed layered mesh via the cutting view. The contributions of this paper are as follows.

* Corresponding author. Tel.: +65 65 14 1008; fax: +65 6792 6559.

E-mail addresses: schan@ntu.edu.sg (S. Han), xiaj0002@ntu.edu.sg (J. Xia), yhe@ntu.edu.sg (Y. He).

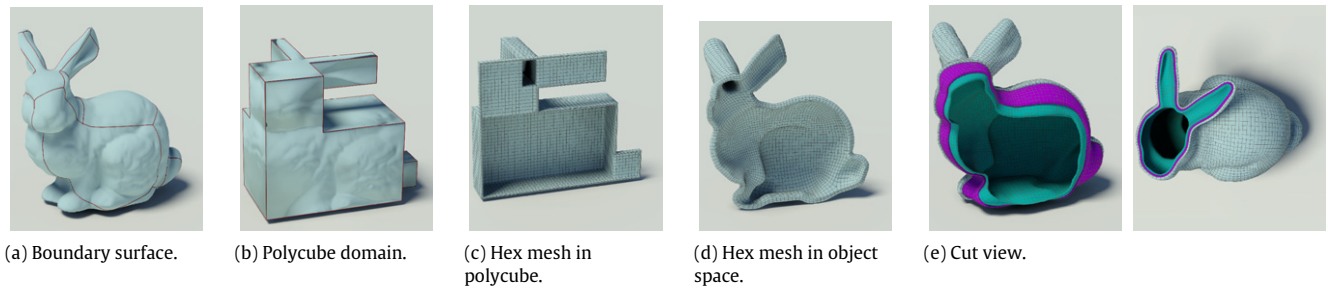


Fig. 1. Constructing a layered hexahedral mesh for the shelled Bunny model. Given a closed 2-manifold and a user-specified offset distance, we first construct the shell space using the distance field and then parameterize the shell space to a shelled polycube domain. As there is a natural hexahedral tessellation in the polycube domain, the volume parameterization induces the hexahedral mesh in the object space. The constructed mesh is an all-hexahedral mesh that most of the vertices are regular, i.e., the valence is 6 for interior vertices and 5 for boundary vertices. The mesh also has a layered structure, so that all layers have exactly the same tessellation.

- We present a volume parameterization algorithm tailored to the shelled volume and prove that the parameterization is a bijection. We also show that the isoparametric line in the thickness dimension is perpendicular to the other two isoparametric lines.
- We develop a method to construct a layered all-hex mesh for shell objects by parameterizing it to a polycube domain which has a natural hexahedral tessellation. The resulting mesh has a layered structure in which each layer has exactly the same tessellation. We demonstrate that our method works for models with complicated geometry/topology and variable thicknesses. We also show that the proposed framework is general that can be applied to other parametric domains, such as sphere and cylinder.

The rest of the paper is organized as follows. Section 2 briefly reviews the related work. The algorithm details are presented in Section 3, followed by some applications in Section 5 and experimental results in Section 6. Finally, we discuss results, benefits and limitations in Section 7.

2. Related work

Hexahedral meshing. Hexahedral meshing has been widely studied in the past two decades. Popular techniques include grid-based algorithm [5], plastering [6], whisker-weaving algorithm [7], embedded Voronoi graph [8] and chordal surface [4], just to name a few. The readers are referred to the comprehensive survey of general hexahedral and tetrahedral meshing constructions [9]. There are also a few techniques that aim to improve the quality of a hexahedral mesh, such as [10,11].

Polycube map. The polycube, a natural generalization of the cube, can serve as the parametric domain of shapes with complicated topology and geometry. Tarini et al. presented a method to construct a polycube map by projecting the vertices of the 3D model to the polycube domain [12]. Wang et al. introduced an intrinsic approach that initially maps the 3D model and the polycube to the canonical domains, and then identifies the map between them [13]. The resulting polycube map is guaranteed to be a bijection and can be used to construct manifold splines [14]. Wang et al. proposed a user-controllable polycube map wherein the users can specify the pre-images of the polycube corners [15]. Lin et al. proposed an automatic algorithm to construct a polycube map [16]. He et al. proposed a divide-and-conquer approach to construct a polycube map of arbitrary topology [17]. Xia et al. developed an editable polycube mapping framework which allows users to freely sketch the feature correspondences on the 3D model and the polycube domain [18].

Harmonic map. Harmonic maps play an important role in surface and volume parameterization. Pinkall and Polthier created a discrete harmonic map using the finite element method [19].

Observing how the harmonic function satisfies the mean value theorem, Floater presented mean value coordinates that directly discretize a harmonic map [20]. Later, the 2D mean value coordinates are generalized to 3D closed meshes [21,22]. Wang et al. generalized the discrete surface harmonic map to tetrahedral meshes [23]. Li et al. solved the harmonic volumetric mapping using the fundamental solution method [24,25]. Martin et al. parameterized topological balls using volumetric harmonic functions [26]. Xia et al. proved that Green's function on star-shape volumes has a unique critical point, and then developed a volume parameterization method that is guaranteed to be a diffeomorphism [27]. Xia et al. presented a method to parameterize the handlebodies using direct product parameterization [28]. Martin and Cohen presented an algorithm to parameterize thin solid models with higher genus or bifurcations using harmonic functions [29]. Li et al. presented an efficient algorithm to compute the harmonic volumetric mapping, which establishes a smooth correspondence between two given solid objects of the same topology [30].

Shell space. Shell structure is also widely used in computer graphics applications. Porumbescu et al. presented an algorithm to build a bijective map between shell space and texture space that can be used to generate small-scale features on surfaces [31]. Wang et al. presented a technique for rendering heterogeneous translucent materials by solving the diffusion equation in the shell space [32].

3. Hexahedral shell meshing

3.1. Overview

Our hexahedral meshing algorithm consists of three steps, constructing the shell space, parameterizing the shell space and hexahedral meshing.

In Step 1, we construct the offset surface using the user-specified thickness. Then we tessellate the shell space using a tetrahedral mesh. We also parameterize the given boundary surface to a polycube and construct the shelled polycube in a similar fashion.

In Step 2, we compute the harmonic field in the shell spaces by solving the Laplace equation with a Dirichlet boundary condition. By tracing the integral curves, we build a bijection between the shell spaces.

In Step 3, we tessellate the polycube space with a regular hexahedral mesh, then construct the layered hexahedral mesh in the object space via volumetric parameterization.

Let M and P denote the shell spaces of the given model and polycube, respectively. Let $\partial M = M_0 \cup M_1$ where M_0 and M_1 are the outer and inner boundary surfaces. Similarly, P_0 and P_1 denote the outer and inner boundary surface of P .

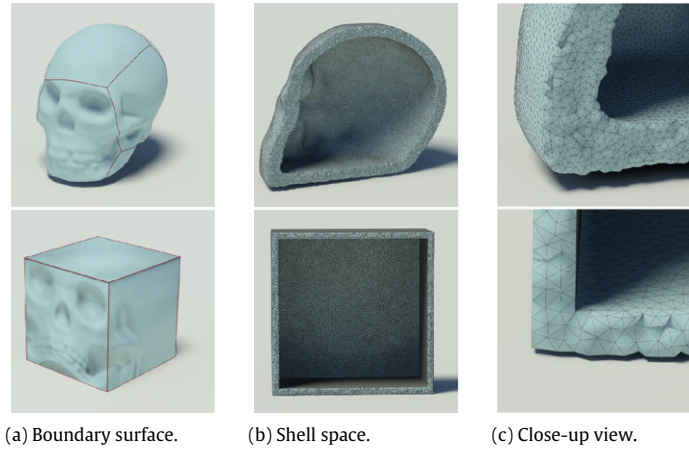


Fig. 2. Shell space construction. Row 1: we construct a distance field for the given model and extract an iso-surface with the user-specified offset distance. Then, we construct an isotropic tetrahedral mesh using a variational meshing technique [35]. Row 2: we map the outer boundary surface to a polycube and construct the tetrahedral mesh of the shell space in a similar way. Note that the sharp features (polycube edges and corners) are well preserved in the tetrahedral mesh.

The detailed algorithm is illustrated as follows.

Input: M_0 , a closed 2-manifold P_0 , the polycube with the same topology of M_0
 d : the thickness
Output: H , a quality hexahedral mesh of M

- 0.1 Construct the polycube map $\phi : M_0 \rightarrow P_0$
- 0.2 Create the offset surfaces M_1 and P_1 by the user-specified thickness d
- 0.3 Construct isotropic tetrahedral meshes for M and P
- 0.4 Parameterize M to P by volumetric harmonic field (see Alg. 2)
- 0.5 Form the hexahedral mesh of P
- 0.6 Construct the hexahedral mesh of M using the volumetric parameterization

Algorithm 1: Layered hexahedral meshing for shell objects

3.2. Constructing shell space

The shell space is enclosed by two disconnected closed surfaces. One is the given 3D model, and the other is an offset surface. The offset distance is specified by the user. We use the MPU method [33] to construct the distance field of the input model, then extract the isosurface whose isovalue is the user-specified thickness. We then construct a tetrahedral mesh to fill the shell space using Tetgen [34]. We also apply the variational meshing algorithm [35] to improve the quality of the tetrahedral mesh.

Following the divide-and-conquer approach [17], we map the boundary surface M_0 to a user-constructed polycube. The resulting polycube map has very low angle distortion and is guaranteed to be a bijection. When constructing the tetrahedral mesh for the polycube space, we must pay special attention to the sharp features (such as edges and corners) of the polycube. To preserve the features in the isotropic tetrahedral mesh, we follow the variant variational meshing technique that is tailored for mechanical models [36]. Fig. 2 shows the construction of shell space for the Skull model and its polycube domain.

3.3. Parameterizing shell space

To construct a map between $M = M_0 - M_1$ and $P = P_0 - P_1$, we first solve a volumetric harmonic map for each shell mesh. The harmonic equations $f : M \rightarrow \mathbb{R}$ and $g : P \rightarrow \mathbb{R}$ are defined as follows:

$$\Delta f = 0 \quad \text{with } f|_{\partial M_0} = 0 \text{ and } f|_{\partial M_1} = 1$$

$$\Delta g = 0 \quad \text{with } g|_{\partial P_0} = 0 \text{ and } g|_{\partial P_1} = 1.$$

Since the shells M and P are represented by tetrahedral meshes, we solve the above Laplace equations by the finite element method [23].

The tetrahedral mesh M is represented by $M = (V, E, F, T)$ where V, E, F and T are the set of vertices, edges, faces and tetrahedral, respectively. P is represented in a similar way. For every interior vertex v_i , the harmonic function $f : M \rightarrow \mathbb{R}$ satisfies the condition

$$\sum_{v_j \in Nb(v_i)} w_{ij} (f(v_j) - f(v_i)) = 0,$$

where w_{ij} is the weight assigned to edge e_{ij} . Suppose edge e_{ij} is shared by m adjacent tetrahedra, it lies against m dihedral angles $\theta_k, k = 1, \dots, m$. Then the weight w_{ij} for e_{ij} can be defined as $w_{ij} = \frac{1}{12} \sum_{k=1}^m \|e_{ij}\| \cot \theta_k$, where $\|e_{ij}\|$ is the length of edge e_{ij} . Fig. 3 shows the volume rendering of the harmonic fields in the Skull model.

Once we obtain the harmonic function, the gradient vector field is computed as follows. Suppose t is a tetrahedron with vertices (v_1, \dots, v_4) , the face on the tetrahedron against vertex v_i is f_i . We define n_i to be the vector along the normal of f_i with magnitude equaling twice the area of f_i . Then, the gradient of ∇f in t is a constant vector field

$$\nabla f = f(v_0)n_0 + f(v_1)n_1 + f(v_2)n_2 + f(v_3)n_3.$$

We then define the per-vertex gradient as the average of the per-tetrahedron gradient vectors.

Given the gradient vector field ∇f , the integral curve is a curve such that the tangent vector to the curve at any point v along the curve is precisely the vector $\nabla f(v)$. In Appendix, we show that each integral curve has unique ending points, one on the inner boundary and the other on the outer boundary. Furthermore, any two integral curves do not intersect.

We construct the volume parameterization $\phi : M \rightarrow P$ as follows: for every interior point $v \in M$, let $\gamma \in M$ be the integral curve that passes through v and follows the gradient vector field of f . The integral curve γ intersects M_0 and M_1 at v_0 and v_1 , respectively. Let $\gamma' \in P$ be the integral curve in P that starts from $v'_0 = h(v) \in P_0$, follows the gradient vector field of g , and terminates at $v'_1 \in P_1$. The image of $v, v' = \phi(v) \in \gamma'$ is a unique point such that $g(v') = f(v)$. In Appendix, we prove that the map ϕ is bijective.

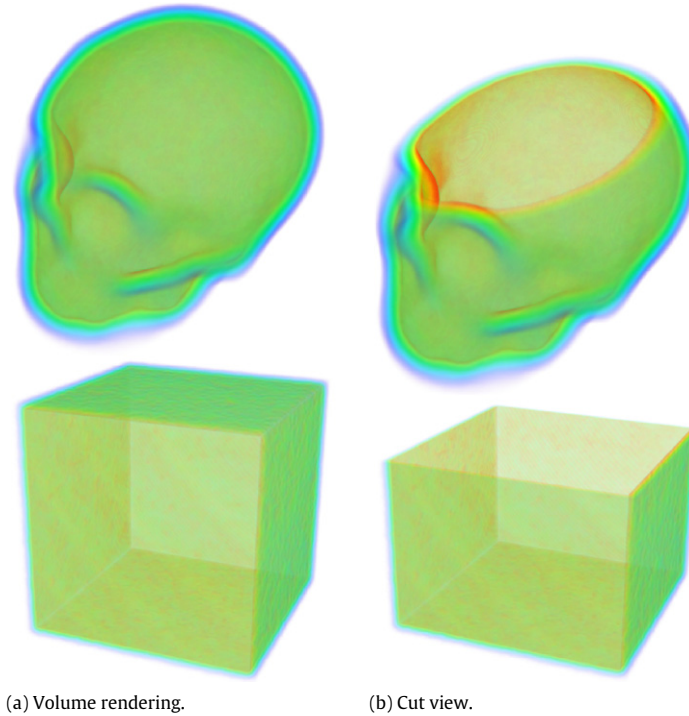


Fig. 3. Volumetric harmonic map on the shell space. We solve a Laplace equation in the shell space with Dirichlet boundary condition such that the values of the outer and inner boundary surfaces are 0 and 1, respectively. We use volume rendering to visualize the harmonic field in the shell space.

Input: An arbitrary point $v \in M$ in the shell object
 $f : M \rightarrow \mathbb{R}$: harmonic function on M
 $g : P \rightarrow \mathbb{R}$: harmonic function on P
 $h : M_0 \rightarrow P_0$ the bijection between the outer boundaries of M and P .
Output: The image $\phi(v) \in P$ in the polycube domain

- 1.1 Starting from v , trace the integral curve $\gamma \in M$ in both positive and negative directions of the gradient vector field ∇f . γ intersects the outer and inner boundaries at v_0 and v_1 respectively.
- 1.2 Compute $v'_0 = h(v_0) \in P_0$
- 1.3 Starting from v'_0 , trace the integral curve $\gamma' \in P$ following the positive direction of the gradient vector field ∇g . γ' intersects P_1 at v'_1
- 1.4 Locate the unique point $v' \in \gamma'$ such that $g(v') = f(v)$. Then $\phi(v) = v'$

Algorithm 2: Shell object parameterization

3.4. Tracing integral curves

Tracing integral curves play an important role in our parameterization framework. The conventional approach assumes the gradient as a constant vector inside each tetrahedron, then the tracing integral curves inside a tetrahedron is straightforward, i.e., a ray is emanated from the entrance point (a tetrahedron vertex or a point on one of its faces), follows the gradient direction and then hits one of its faces. Thus, the integral curve inside each tetrahedron is a line segment. However, in the continuous setting, the integral curve of the gradient of a harmonic function is a highly smooth curve. Thus, the traditional approach may lead to an inaccurate tracing result.

In our implementation, we approximate the gradient inside a tetrahedron by using the Barycentric interpolation and then trace

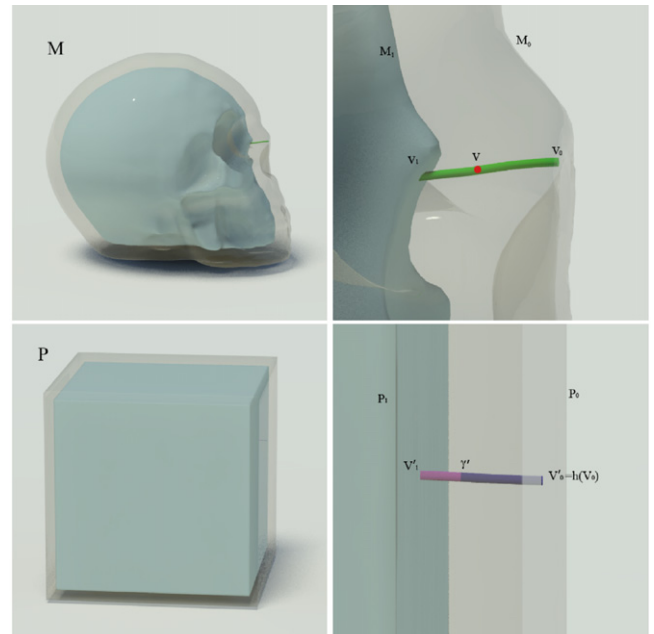


Fig. 4. Tracing integral curves. Each integral curve follows the gradient vector field of the harmonic function. Thus, it is orthogonal to the iso-surface of the harmonic function (including the two boundary surfaces). We show that each integral curve has unique ending points and all integral curves do not intersect.

the integral curves using Euler approach (see Algorithm 3). We use half-face data structure to model the tetrahedral mesh. Each interior face is shared by two tetrahedral and each boundary face is only adjacent to one tetrahedron.

Fig. 4 shows an example of tracing integral curves of the Skull model. We also compare our tracing algorithm with the traditional approach as shown in Fig. 5.

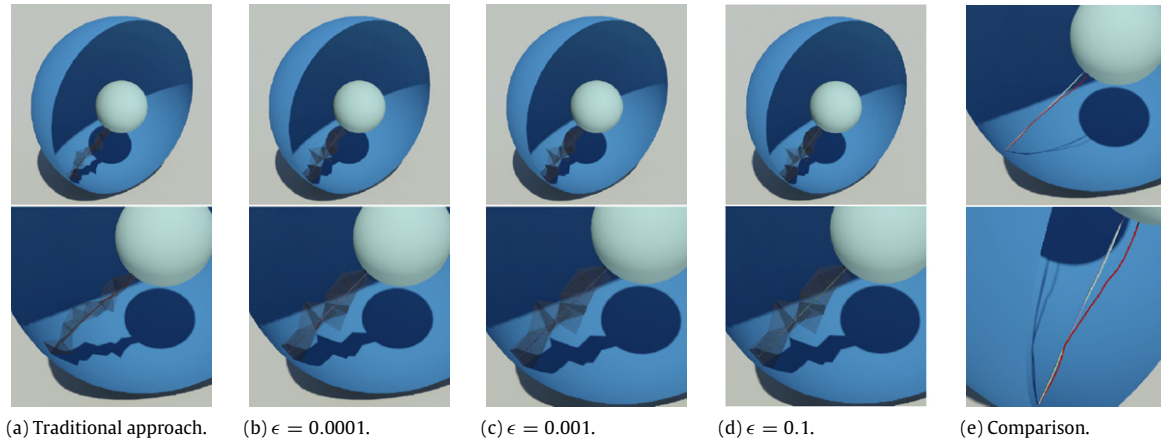


Fig. 5. Comparison of integral curve tracing algorithms. Given a unit ball with a void centered at the origin, we compute the harmonic function with Dirichlet boundary condition such that the inner (resp. outer) boundary is with boundary value 1 (resp. 0). Due to the symmetric property of the shape, each integral curve is a radial line that is perpendicular to both boundary surfaces. The traditional algorithm assumes that the gradient inside each tetrahedron is constant, thus, the resulting integral curve highly depends on the mesh tessellation and has “zig-zag” structure (see (a)). In contrast, our algorithm approximates the gradient direction by Barycentric interpolation and traces the integral curves by using Euler approach with the user-specified step length. (b)–(d) show the traced curves by our approach with various step lengths. (e) shows the close-up view of the curves by our approach (in gray) and the traditional approach (in red). Note that the three integral curves, by our approach with various step lengths, almost coincide. Clearly, our algorithm is more robust and accurate than the traditional approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Input: p , a point inside the volume
or on the boundary ϵ , step
length of tracing
 $positive$, a Boolean value indicating
the tracing direction
Output: γ , the integral curve
passing through p and
following the positive or
negative direction of the
gradient vector field

```

Find the tetrahedron  $currTet$  such that
 $p \in currTet$ ;
while
 $currTet \neq NULL$ 
do
  Compute the gradient vector  $p_v$  by
  linear interpolation of the vertex
  gradients of  $currTet$ 
  if positive then
     $p_{next} = p + \epsilon * p_v$ 
  else
     $p_{next} = p - \epsilon * p_v$ 
  end
  if  $p_{next} \in currTet$ 
  then
    update  $p_v$ 
  else
    Find the tetrahedron  $nextTet$ 
    which contains the  $p_{next}$ 
     $currTet = nextTet$ 
  end
   $p = p_{next}$ 
end
Project  $p$  to the shell boundary

```

Algorithm 3: Tracing integral curves

4. Layered hexahedral meshing

Note that the polycube has a very regular structure that can be easily tessellated into quadrilaterals. We can construct a hexahedral mesh by sweeping the quads such that each vertex

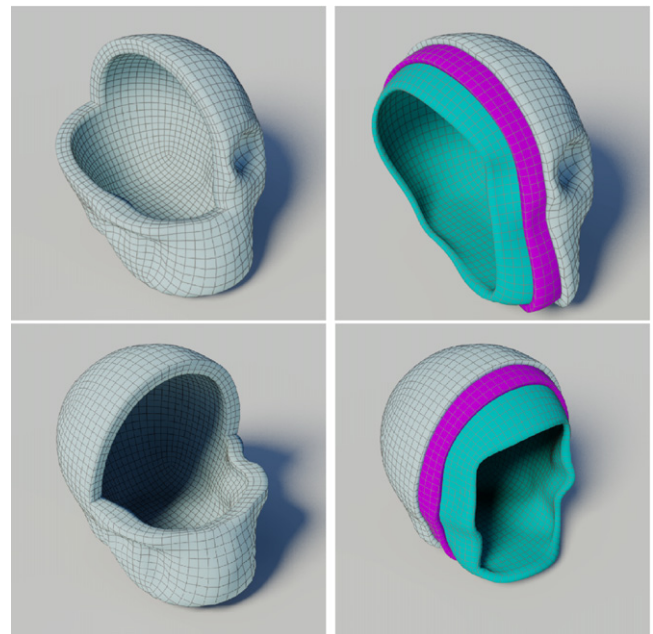


Fig. 6. Layered hexahedral meshing of the Skull model.

is moving along (or opposite to) the normal direction until they reach the other boundary surface. Then we uniformly segment the polycube domain into layers. The number of layers are specified by the user. Note that each layer has exactly the same tessellation. Since the quadrilateral mesh of the outer boundary is constructed by the polycube map, all vertices except the polycube corners, are regular, i.e., with valence 4. Therefore, after sweeping the quads to the shell space, the interior vertices are regular (i.e., with valence 6) if the corresponding vertex on the boundary mesh is regular. We should also mention that the shelled polycube domain does not need to have the same thickness as the 3D shell object, as long as the two boundary surfaces of the polycube are similar.

As we show in Appendix, the proposed shell parameterization algorithm is guaranteed to be a bijection. Thus, the constructed hexahedral mesh is free of degeneracy, i.e., self-intersection and flip over. Fig. 6 shows the layered hexahedral mesh of the Skull model.

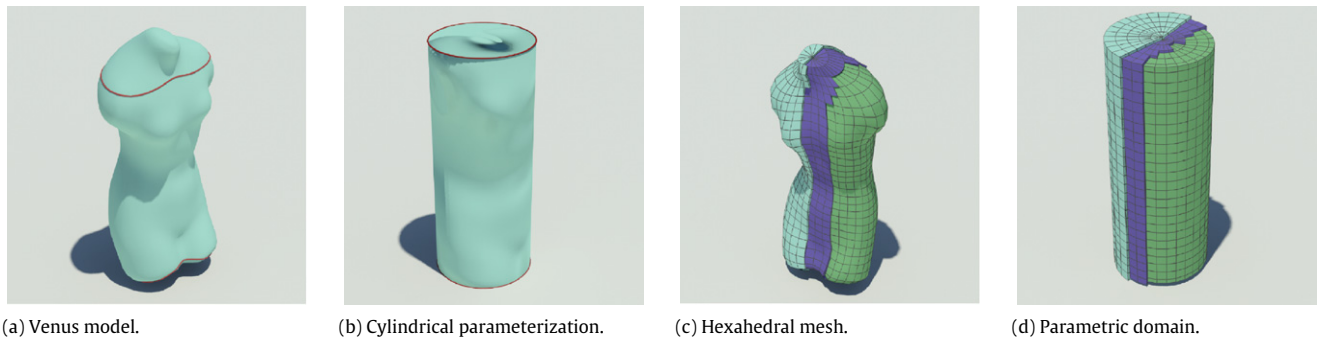


Fig. 7. Hexahedral meshing by cylindrical parameterization.

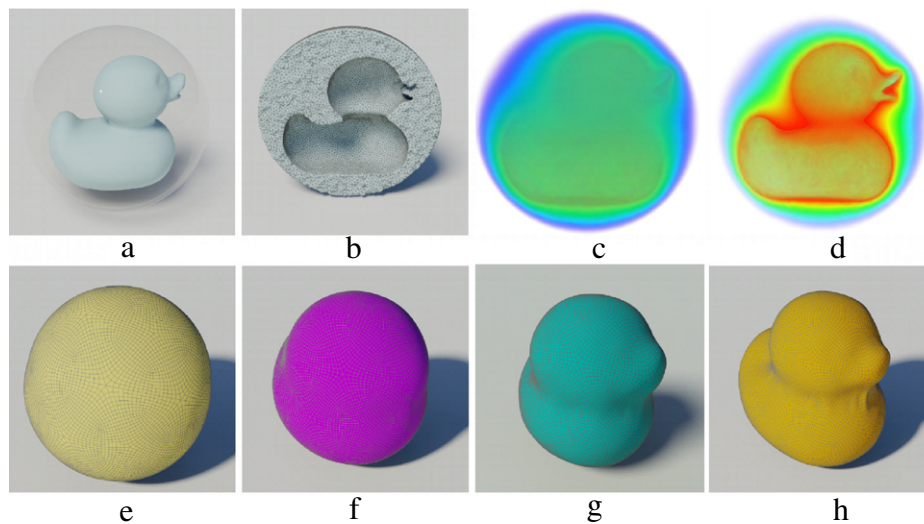


Fig. 8. Layered hexahedral meshing of the Duck model embedded in a sphere. Note the outer and inner boundaries are significantly different (see (a)). (b) shows the isotropic tetrahedral mesh of the shell space. We parameterize this shell object to a ball by a volumetric harmonic map (see (c) and (d)). The outer boundary is tessellated into truncated icosahedron and the shell space is segmented into 10 layers. (e), (f), (g) and (h) show the 3rd, 6th, 8th and 10th layers, respectively.

Although our main focus is on volumetric polycube parameterization, due to the ease of constructing hexahedral tessellation in the parametric domain, our framework is general that it can be applied to other domains as long as it has the same topology as the input model. As shown in Fig. 7, we parameterize the Venus to a shelled cylinder, and construct a hexahedral tessellation on the cylinder, which induces a layered hexahedral mesh of the Venus. In contrast to the polycube parameterization, where every polycube corner is a singularity, cylindrical parameterization has only two singularities, i.e., the centers on the top and bottom disks.

Furthermore, our approach also works for the case where the inner boundary surface M_1 and P_1 are arbitrary closed surfaces rather than the offset surfaces. The only requirement is that M_1 and P_1 are of the same topological type as M_0 and P_0 . In Fig. 8(a), we conduct an experiment by embedding the Duck model into a sphere. Thus, M_0 and M_1 are geometrically different but topologically equivalent. We construct the shelled ball as the parametric domain and tessellate the sphere using a truncated icosahedron (i.e., the soccer ball tessellation with 12 pentagons and 20 hexagons). We then increase the resolution of the sphere by Catmull–Clark subdivision and sweep the quads inwards of the ball. Fig. 8(e)–(f) show the layered hexahedral mesh of the embedded Duck model.

5. Applications

In this section, we present two applications to demonstrate the efficacy of the proposed shell parameterization technique.

5.1. Solid texture mapping

Texture mapping is one of the most essential techniques that add rich and realistic visual details to 3D models. Although 2D textures have been widely used in various graphics applications, many natural materials, such as wood and stone, may be more realistically modeled using solid textures [37,38], which provide texture information not only on surfaces, but also throughout the entire volume occupied by the object [39]. The shell models are parameterized to the polycube domain, in which solid texels can be easily constructed. By taking advantage of the bijectivity and low distortion of our parameterization algorithm, we can produce high quality, seamless solid texture mapping for 3D shell models (see Fig. 9).

5.2. Geometric texture mapping

Shell map [40] is a bijective map between shell space and texture space, which can be used to generate small scale features on surfaces using a variety of modeling techniques [41]. Our method naturally parameterizes the shell space, thus, can be used for geometric texture mapping. As shown in Fig. 10, we can generate rich visual effects by setting the texture space to contain geometric objects, scalar fields, or other shell-mapped objects.

6. Experimental results

We tested our algorithm using models of various topology. In our experiments, the user-specified offset distance (i.e., thickness)

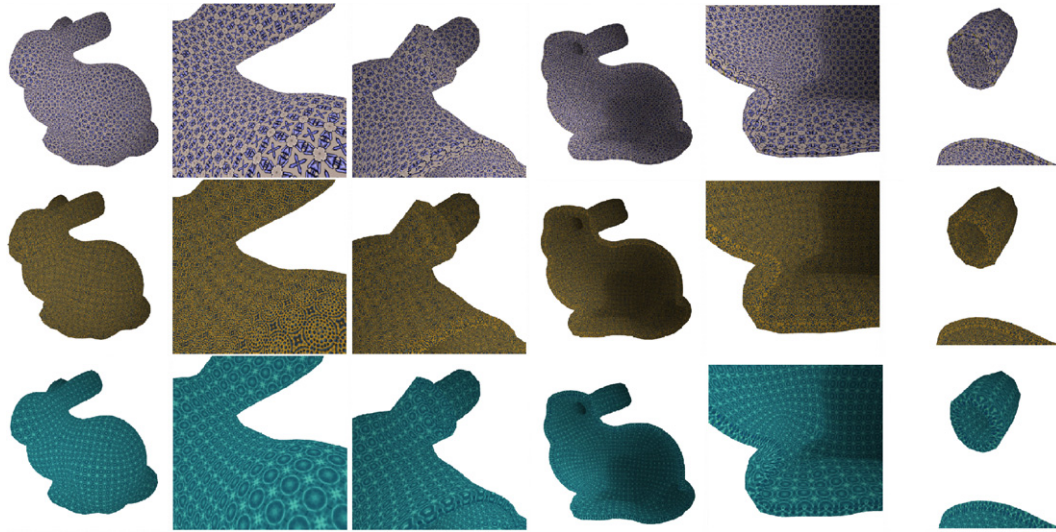


Fig. 9. Solid texture mapping of Bunny and its cut views.

can be either positive or negative. The user also specifies the number of layers in the constructed hexahedral meshes. As mentioned before, each layer has exactly the same tessellation. Figs. 11–13 show the layered hexahedral meshes. Cutaway views are used to reveal the high quality of the hexahedral inside the models. Table 1 shows the statistics of the experimental results.

To measure the quality of the hexahedral mesh, we choose the scaled Jacobian [42], the condition number of the Jacobian matrix [43] and the Oddy metric [44].

Assume $x \in \mathbb{R}^3$ is the position vector of this vertex and $x_i \in \mathbb{R}^3$ for $i = 1, 2, 3$ are its three neighbors in some fixed order. Edge vectors are defined as $e_i = x_i - x$ with $i = 1, 2, 3$ and the Jacobian matrix is $J = [e_1, e_2, e_3]$. The determinant of the Jacobian matrix is called *Jacobian*, or *scaled Jacobian* if edge vectors are normalized. An element is said to be *inverted* if one of its *Jacobians* is less than 0. We use the *Frobenius norm* as a matrix norm, $|J| = \text{tr}(J^T J)^{1/2}$. The condition number of the Jacobian matrix is defined as $k(J) = |J| |J^{-1}|$, where $J^{-1} = \frac{J^T}{\det(J)}$. Therefore, the three quality metrics for a vertex in a hexahedron are defined as follows:

$$\text{Jacobian}(x) = \det(J)$$

$$k(x) = \frac{1}{3} |J^{-1}| |J|$$

$$\text{Oddy}(x) = \frac{|J^T J|^2 - \frac{1}{3} |J|^4}{\det(J)^{\frac{4}{3}}}$$

7. Conclusion

In this paper, we developed an algorithm to parameterize the shell space. The parameterization is theoretically sound and guarantees a bijection. By parameterizing the given shell object to a shelled polycube domain, we can construct layered all-hexahedral meshes of high quality. All vertices (except the vertices on the integral curves passing polycube corners) are regular, i.e., with valence 6 for interior vertices, and valence 5 for boundary vertices. Furthermore, each layer in the constructed mesh has exactly the same tessellation. Due to the bijectivity of the proposed volume parameterization, the hexahedral mesh is free of degeneracy, such as self-intersection, flip-over, etc. We demonstrated the efficacy of our method to models of various topology.

Limitations. The proposed method has several limitations. First, we choose the polycube as the parametric domain due to its regular structure, in which one can easily construct an all-hexahedral

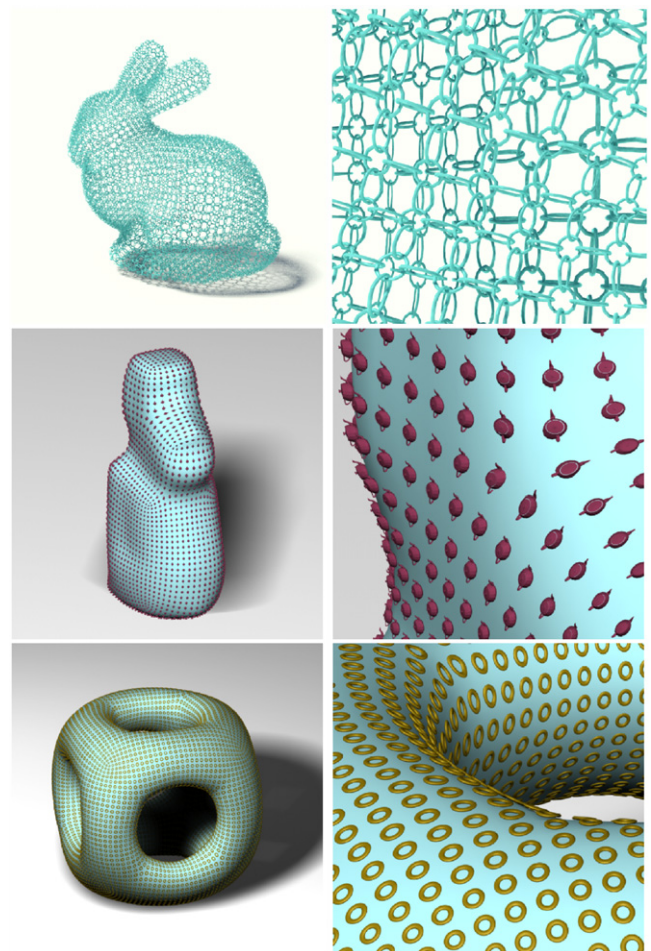


Fig. 10. Shell map based geometric textures.

mesh. On one hand, the polycube should mimic the geometry of the shell object as closely as possible to minimize the parameterization distortion. On the other hand, since each polycube corner is a singularity of the boundary surface parameterization, we should keep the polycube as simple as possible. These two requirements

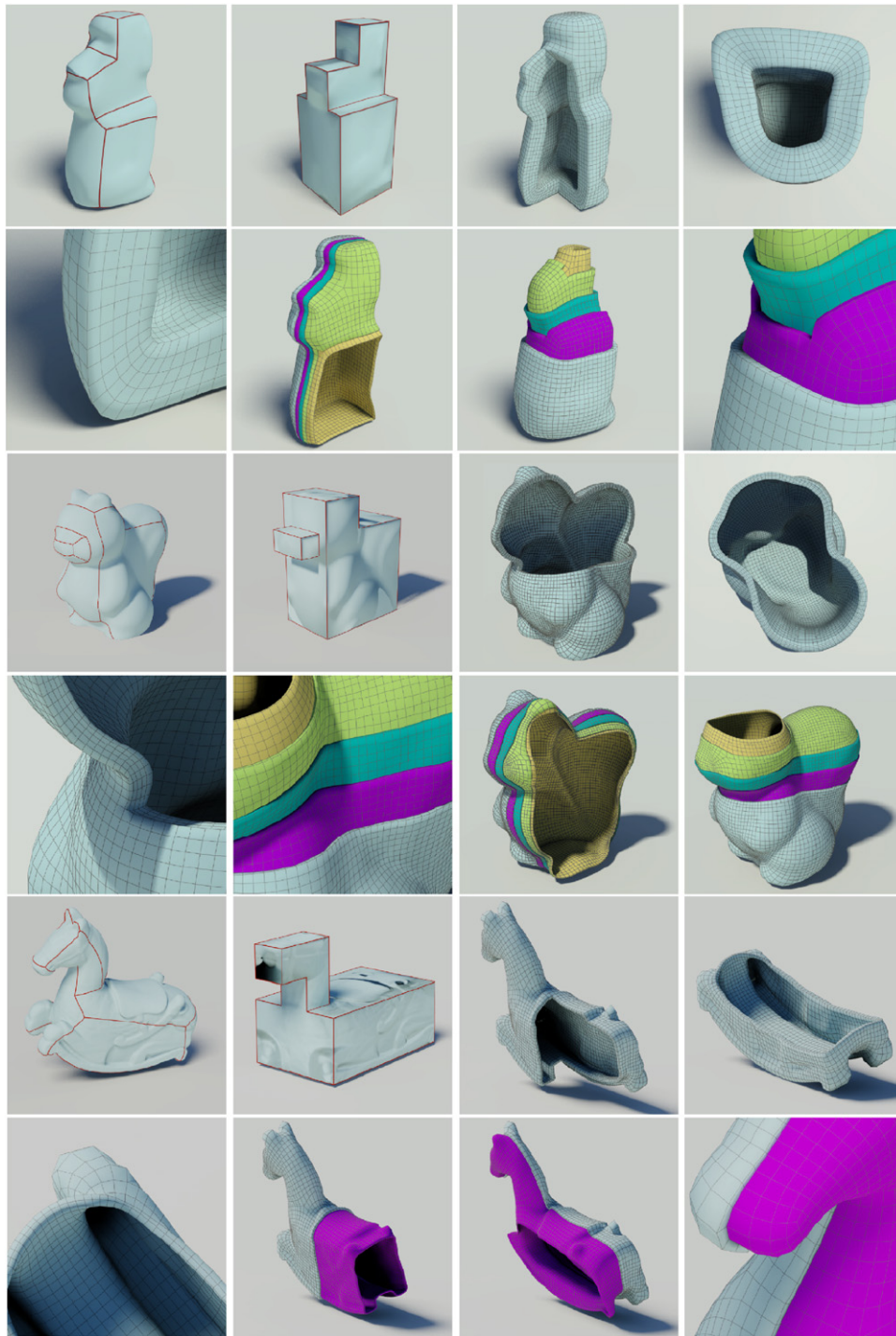


Fig. 11. Experimental results on genus-0 models.

often contradict each other. Thus, it requires the users to be very skillful in designing the parametric domain. Second, a polycube is not a good parametric domain for models with highly complex geometry and/or topology, such as trees. Thus, our method works only for a limited range of models. Third, we use the finite element method to solve the Laplace equation in the shell space and then trace the integral curves of the gradient vector field. The robustness of tracing integral curves greatly depends on the quality of the tetrahedral mesh. In our experiments, we observed that the isotropic meshes lead to good results. However, the isotropic tetrahedral meshes usually contain large numbers of vertices, which increases the computational cost.

Acknowledgments

This work was supported by NRF2008IDM-IDM004-006 and AcRF RG69/07. The models are courtesy of Stanford University, Cyberware, and Aim@Shape Shape Repository. We would like to thank the anonymous reviewers for their constructive comments and Prof. David Gu for insightful discussions.

Appendix

In this section, we prove that the shell space parameterization is a bijection and the w -isoparametric line (in the thickness dimension) is perpendicular to the other two isoparametric lines.



Fig. 12. Experimental results on high genus models.

Theorem. Given two shelled objects $M = M_0 - M_1$ and $P = P_0 - P_1$ where M_0 (P_0) and M_1 (P_1) are the outer and inner boundary surfaces of M (P), respectively. The boundary surfaces $M_i, P_i, i = 1, 2$ are of the same topological type.

Define harmonic function on $M, f : M \rightarrow \mathbb{R}, \Delta f = 0$, with Dirichlet boundary condition, $f|_{M_0} = 0$ and $f|_{M_1} = 1$. Let $C_f : M_0 \times [0, 1] \rightarrow M$ be the integral curve of the gradient field ∇f such that given an arbitrary point $v_0 \in M_0, C_f(v_0, 0) = v_0, C_f(v_0, 1) = v_1$ and $C_f(v_0, t) = v_t$, where $v_1 \in M_1$ is the other ending point and $v_t \in M$ is the point satisfying $f(v_t) = t$. Similarly, we define the harmonic function on $P, g : P \rightarrow \mathbb{R}$ and the integral curve $C_g : P_0 \times [0, 1] \rightarrow P$.

Define a homeomorphic boundary map $h : M_0 \rightarrow P_0$ and construct the volume parameterization $\phi : M \rightarrow P$ as follows:

$$\phi(C_f(v_0, t)) = C_g(h(v_0), t), \quad \forall v_0 \in M_0.$$

Then the volume parameterization ϕ has the following properties:

1. ϕ is bijective.
2. the w -isoparametric line, following the gradient of the harmonic field (i.e., in the thickness dimension), is always perpendicular to the u - and v -isoparametric lines that span the iso-surfaces of the harmonic field.

Proof of (1). First, we show that the ending points of each integral curve are on the inner and outer boundary surfaces, respectively. Note that f and g are smooth functions and their gradient vector fields are curl-free. Thus, no integral curve can form a loop inside the volume.

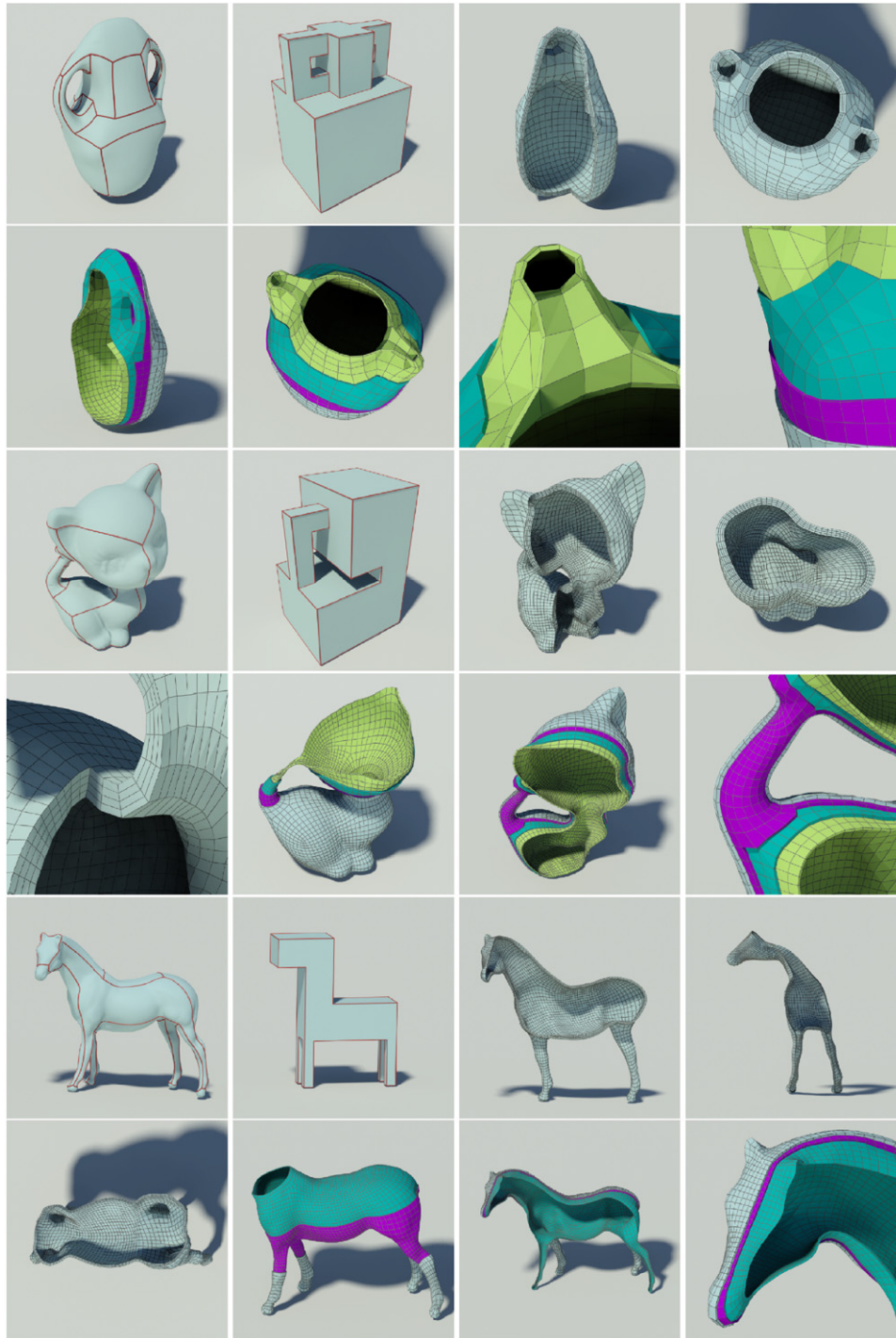


Fig. 13. More experimental results.

Second, we show that no integral curve starts and ends on the same boundary surface. The function is harmonic and there is no critical points (where the gradient vanishes) inside the volume. Thus, the function value is strictly monotonic along the integral curve. Note that all points on the same boundary surface have the same function value, so the ending points of each integral curve must be on different boundary surfaces.

Third, we show that two integral curves do not intersect. Assume that two integral curves $\gamma_1 \in M$ and $\gamma_2 \in M$ intersect at a point p . Then p is a critical point and the gradient ∇f vanishes at p . We consider two cases.

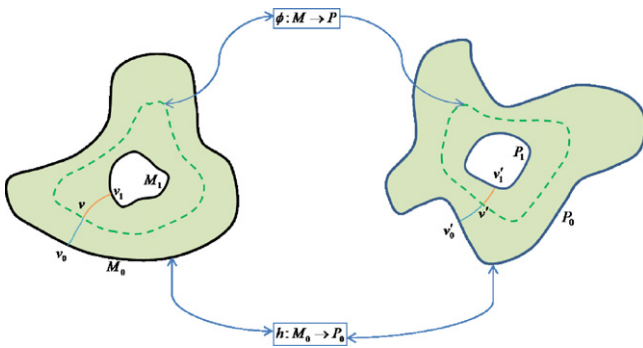
Case 1: p is an interior point. Since f is harmonic, the maximum and minimum must be on the boundaries. Therefore, the Hessian matrix at p has negative eigenvalues. Suppose $f(p) = s$, then according to Morse theory, the homotopy types of the level sets $f^{-1}(s - \epsilon)$ and $f^{-1}(s + \epsilon)$ will be different. At all the interior critical points, the Hessian matrices have negative eigenvalues and the homotopy type of the level sets will be changed. The changes of the homotopy type cannot be canceled out. Therefore, the homotopy type of M_0 is different from that of M_1 . This contradicts the given condition.

Table 1
Statistics of experimental results. g , genus of the boundary surface; $|V|$, $|T|$, # of vertices and tetrahedral of the tetrahedral mesh; L , # of layers; h , # of hexahedra in each layer; d , thickness; T_h , T_t time for computing volumetric harmonic map and tracing. Timings were measured in seconds on a workstation with 2.66 GHz CPU and 4 GB memory.

Model	g	(V , T)	L	h	d	Scaled Jacobian (best, aver., worst)	Oddy metric (best, aver., worst)	Condition number (best, aver., worst)	T_h	T_t
Bunny	0	(180 K, 839 K)	3	7232	-0.02	(1.00, 0.94, 0.05)	(7.27, 8.57, 520.78)	(2.73, 3.14, 15.15)	18.17	6.75
Skull	0	(190 K, 981 K)	3	4804	-0.04	(1.00, 0.94, 0.58)	(2.97, 5.28, 29.65)	(1.75, 1.99, 2.92)	29.58	8.98
Moai	0	(180 K, 974 K)	5	2688	-0.08	(1.00, 0.92, 0.16)	(0.01, 2.18, 307.82)	(1.00, 1.22, 10.21)	33.72	10.09
Eight	2	(180 K, 921 K)	3	12800	-0.02	(1.00, 0.94, 0.26)	(0.14, 2.85, 79.36)	(1.02, 1.48, 3.89)	26.23	23.89
Squirrel	0	(180 K, 839 K)	5	8384	+0.02	(1.00, 0.97, 0.32)	(0.24, 2.99, 62.99)	(1.04, 1.62, 6.87)	19.88	13.04
Decocube	4	(180 K, 850 K)	4	7936	+0.03	(1.00, 0.96, 0.70)	(0.41, 1.74, 13.04)	(1.07, 1.36, 3.31)	21.97	14.81
Rockerarm	1	(200 K, 943 K)	3	4672	+0.02	(1.00, 0.92, 0.25)	(0.14, 2.95, 45.77)	(1.02, 1.41, 4.34)	20.28	4.36
Isidore horse	0	(200 K, 956 K)	2	4736	-0.01	(1.00, 0.94, 0.05)	(3.38, 4.58, 280.15)	(1.89, 2.12, 15.74)	22.31	2.47
Kitty	2	(150 K, 701 K)	4	17408	-0.02	(1.00, 0.91, 0.33)	(2.45, 3.14, 150.83)	(1.22, 2.07, 11.93)	40.36	8.75
Vase	2	(150 K, 699 K)	4	4000	+0.02	(1.00, 0.82, 0.31)	(11.04, 21.48, 80.22)	(4.022, 5.63, 7.49)	15.33	9.01
Horse	0	(150 K, 593 K)	3	19776	-0.008	(1.00, 0.94, 0.35)	(13.69, 74.10, 692.76)	(2.97, 9.29, 59.01)	30.21	8.72
Duck	0	(100 K, 542 K)	10	11520	NA	(1.00, 0.98, 0.85)	(0.02, 0.97, 46.55)	(1.00, 1.09, 2.72)	19.67	23.20

Case 2: p is on the boundary. Without loss of generality, say $p \in M_0$. Then we can glue two copies of the same volume, along M_0 and reverse the gradient field of one volume. The union of the two gradient fields give us a harmonic function field. Then there is no interior critical point on the doubled volume. p becomes one interior critical point, that leads to a contradiction.

Therefore, γ_1 and γ_2 have no intersection points anywhere.



Finally, we show ϕ is bijective. From the above, we know that for an arbitrary interior point, there is a unique integral curve passing through and intersecting on the inner and outer boundaries. The two ending points are also unique. Thus, C_f and C_g are homeomorphisms. The given boundary map $h : M_0 \rightarrow P_0$ is homeomorphic, thus, it induces a homeomorphism between integral curves in M and P , $C_f(v_0, \cdot) \rightarrow C_g(h(v_0), \cdot)$, which in turns induces the bijective map ϕ .

Proof of (2). As shown in (1), ϕ bijectively maps the integral curve $\gamma \in M$ to a unique integral curve $\gamma' \in P$. Furthermore, ϕ bijectively maps every iso-surface of f to the iso-surface of g with the same iso-value. Note that the integral curve follows the gradient vector field, thus, is orthogonal to the iso-surface.

The w -isoparametric line for a fixed starting point $v_0 \in M_0$, $\phi(C_f(v_0, t))$, $t \in [0, 1]$, is in fact an integral curve. The u - and v -isoparametric lines for a fixed parameter t , $\phi(C_f(v_0, t))$, $v_0 \in M_0$, span the iso-surface (with iso-value t) of the harmonic function. Thus, the w -isoparametric line is orthogonal to u - and v -isoparametric lines. \square

References

- [1] Cifuentes AO, Kalbag A. A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis. *Finite Elements in Analysis and Design* 1992;12(3-4):313-8.
- [2] Benzley SE, Perry E, Merkley K, Clark B, Sjaardema G. A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elastoplastic analysis. In: *Proceedings of 4th international meshing roundtable*. 1995. p. 179-91.
- [3] Blacker T. Meeting the challenge for automated conformal hexahedral meshing. In: *9th international meshing roundtable*. 2000. p. 11-20.
- [4] Quadros W, Shimada K. Hex-layer: layered all-hex mesh generation on thin section solids via chordal surface transformation. In: *Proceedings of 11th international meshing roundtable*. 2002. p. 169-82.
- [5] Schneiders R, Schindler R, Weiler F. Octree-based generation of hexahedral element meshes. In: *5th International meshing roundtable*. 1996. p. 205-15.
- [6] Staten ML, Kerr RA, Owen SJ, Blacker TD. Unconstrained paving and plastering: progress update. In: *15th International meshing roundtable*. 2006. p. 469-86.
- [7] Tautges TJ, Mitchell SA. Whisker weaving: invalid connectivity resolution and primal construction algorithm. In: *4th international meshing roundtable*. 1995. p. 115-27.
- [8] Sheffer A, Etzion M, Rappoport A, Bercovier M. Hexahedral mesh generation using the embedded voronoi graph. *Engineering with Computers* 1999;15(3):248-62.
- [9] Owen SJ. A survey of unstructured mesh generation technology. In: *Proceedings of international meshing roundtable*. 1998. p. 239-67.
- [10] Zhang Y, Bajaj R, Xu G. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. In: *Proceedings of 14th international meshing roundtable*. 2005. p. 449-68.
- [11] Zhang Y, Bajaj C. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering* 2006;195(9-12):942-60.
- [12] Tarini M, Hormann K, Cignoni P, Montani C. Polycube-maps. In: *Proceedings of SIGGRAPH 2004*. 2004. p. 853-60.
- [13] Wang H, He Y, Li X, Gu X, Qin H. Polycube splines. In: *Proceedings of symposium on solid and physical modeling*. 2007. p. 241-51.
- [14] Gu X, He Y, Qin H. Manifold splines. In: *SPM'05: proceedings of the 2005 ACM symposium on solid and physical modeling*. 2005. p. 27-38.
- [15] Wang H, Jin M, He Y, Gu X, Qin H. User-controllable polycube map for manifold spline construction. In: *Proceedings of symposium on solid and physical modeling*. 2008. p. 397-404.
- [16] Lin J, Jin X, Fan Z, Wang CCL. Automatic polycube-maps. In: *Proceedings of geometric modeling and processing, GMP'08*. 2008. p. 3-16.
- [17] He Y, Wang H, Fu C-W, Qin H. A divide-and-conquer approach for automatic polycube map construction. *Computers & Graphics* 2009;33(3):369-80.
- [18] Xia J, Garcia I, He Y, Xin S-Q, Patow G. Editable polycube mapping for GPU-based subdivision surfaces. In: *Proceedings of ACM symposium on interactive 3D graphics and games, I3D'11*. 2011. p. 151-8.
- [19] Pinkall U, Polthier K. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics* 1993;2(1):15-36.
- [20] Floater MS. Mean value coordinates. *Computer Aided Geometric Design* 2003;20(1):19-27.
- [21] Floater MS, Kós G, Reimers M. Mean value coordinates in 3D. *Computer Aided Geometric Design* 2005;22(7):623-31.
- [22] Ju T, Schaefer S, Warren J. Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics* 2005;24:561-6.
- [23] Wang Y, Gu X, Yau S-T. Volumetric harmonic map. *Communications in Information and Systems* 2004;3(3):191-202.
- [24] Li X, Guo X, Wang H, He Y, Gu X, Qin H. Meshless harmonic volumetric mapping using fundamental solution methods. *IEEE Transactions on Automation Science and Engineering* 2009;6(3):409-22.
- [25] Li X, Guo X, Wang H, He Y, Gu X, Qin H. Harmonic volumetric mapping for solid modeling applications. In: *ACM symposium on solid and physical modeling*. 2007. p. 109-20.
- [26] Martin T, Cohen E, Kirby M. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In: *Symposium on solid and physical modeling*. 2008. p. 269-80.
- [27] Xia J, He Y, Han S, Fu C, Luo F, Gu X. Parameterization of star-shaped volumes using Green's functions. In: *Proceedings of geometric modeling and processing, GMP'10*. 2010. p. 219-35.
- [28] Xia J, He Y, Yin X, Han S, Gu X. Direct-product volumetric parameterization of handlebodies via harmonic fields. In: *Proceedings of the international conference on shape modeling and applications, SMI'10*. 2010. p. 3-12.
- [29] Martin T, Cohen E. Volumetric parameterization of complex objects by respecting multiple materials. *Computer Graphics* 2010;34:187-97.

- [30] Li X, Xu H, Wan S, Yin Z, Yu W. Feature-aligned harmonic volumetric mapping using MFS. *Computer Graphics* 2010;34:242–51.
- [31] Porumbescu SD, Budge B, Feng L, Joy KI. Shell maps. *ACM Transactions on Graphics* 2005;626–33.
- [32] Wang J, Zhao S, Tong X, Lin S, Lin Z, Dong Y, et al. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Transactions on Graphics* 2008;27(1):1–18.
- [33] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H-P. Multi-level partition of unity implicits. In: *SIGGRAPH'03*. 2003. p. 463–70.
- [34] Si H. Tetgen: a quality tetrahedral mesh generator and three-dimensional delaunay triangulator, <http://tetgen.berlios.de/>.
- [35] Alliez P, Cohen-Steiner D, Yvinec M, Desbrun M. Variational tetrahedral meshing. In: *ACM SIGGRAPH 2005 courses*. ACM; 2005. p. 10.
- [36] Smit MS, Bronsvort WF. Variational tetrahedral meshing of mechanical models for finite element analysis. *Computer-Aided Design and Applications* 2008;228–40.
- [37] Peachey D. Solid texturing of complex surfaces. *ACM SIGGRAPH Computer Graphics* 1985;19(3):279–86.
- [38] Perlin K. An image synthesizer. *ACM SIGGRAPH Computer Graphics* 1985; 19(3):287–96.
- [39] Kopf J, Fu C-W, Cohen-Or D, Deussen O, Lischinski D, Wong T-T. Solid texture synthesis from 2D exemplars. In: *ACM transactions on graphics. Proceedings of SIGGRAPH 2007*. vol. 26. No. 3.
- [40] Porumbescu S, Budge B, Feng L, Joy K. Shell maps. *ACM Transactions on Graphics (TOG)* 2005;24(3):626–33.
- [41] Zhou K, Huang X, Wang X, Tong Y, Desbrun M, Guo B, et al. Mesh quilting for geometric texture synthesis. *ACM Transactions on Graphics* 2006;25:690–7.
- [42] Kober C, Muller-Hannemann M. Hexahedral mesh generation for the simulation of the human mandible. In: *Proceedings of the 9th international meshing roundtable*. 2000. p. 423–34.
- [43] Knupp P. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. *International Journal for Numerical Methods in Engineering* 2000;48(3):401–20.
- [44] Oddy A, Goldak J, McDill M, Bibby M. A distortion metric for isoparametric finite elements. *Transactions of the Canadian Society for Mechanical Engineering* 1988;12:213–7.