Hexahedral Shell Mesh Construction Via Volumetric Polycube Map

Shuchu Han Jiazhi Xia Ying He
School of Computer Engineering, Nanyang Technological University
Singapore, 639798
{schan|xiaj0002|yhe}@ntu.edu.sg

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 that all layers have exactly the same tessellation. We prove 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 demonstrate the efficacy of our method upon models of various topology.

Keywords

Solid modeling, shell object, volume parameterization, hexahedral meshing, polycube map

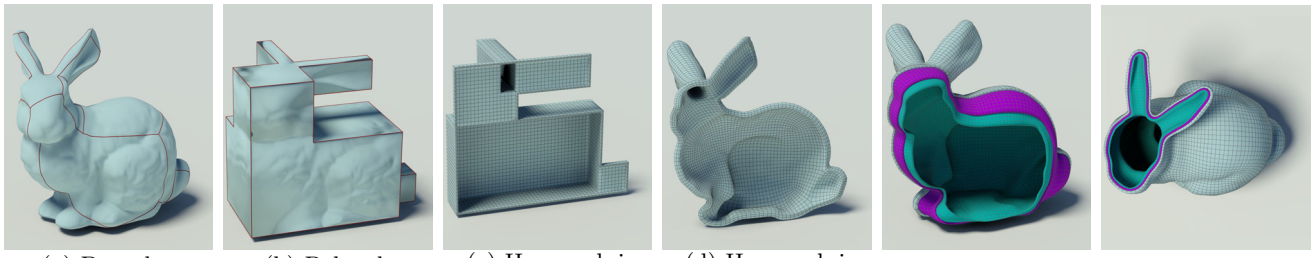
1. INTRODUCTION

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 [4]. In nonlinear elastic-plastic analysis, the linear hexahedral elements may be superior even to quadratic tetrahedral elements when shear stress in 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, such as for automobile bodies, sheet metal parts, etc. To construct a hexahedral shell mesh, we use the existing method, e.g., [21], 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 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. Figure 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 as follows:

- We present a volume parameterization algorithm tailored to the shelled volume and prove that the parameterization is a bijection. We also show that the



(a) Boundary surface (b) Polycube domain (c) Hex mesh in polycube space (d) Hex mesh in object space (e) Cut view

Figure 1: Constructing layered hexahedral mesh for 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 that all layers have exactly the same tessellation.

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.

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 experimental results in Section 4. Finally, we discuss results, benefits and limitations in Section 5.

2. RELATED WORK

Hexahedral meshing. Hexahedral meshing has been widely studied in the past two decades. Popular techniques include grid-based algorithm [22], platstering [26], whisker-weaving algorithm [28], embedded Voronoi graph [23] and chordal surface [21], just to name a few. The readers are referred to the comprehensive survey of general hexahedral and tetrahedral meshing construction [18]. There are also a few techniques that aim to improve the quality of hexahedral mesh, such as [36] [35].

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 [27]. 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 [29]. The resulting polycube map is guaranteed to be a bijection and can be used to construct manifold splines [7]. Wang et al. proposed a user-controllable polycube map wherein the users can specify the pre-images of the polycube corners[30]. Lin et al proposed an automatic algorithm to construct polycube map [14]. He et al. proposed a divide-and-conquer approach to construct a polycube map of arbitrary topology [8].

Harmonic map. Harmonic maps play an important role in surface and volume parameterization. Pinkall and Polthier created 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 discretizes a harmonic map[5]. Later, the 2D mean value coordinates are generalized to 3D closed meshes [6],[9]. Wang et al. generalized the discrete surface harmonic map to tetrahedral meshes [32]. Li et al. solved the harmonic volumetric mapping using fundamental solution method [13][12]. Martin et al. parameterized topological balls using volumetric harmonic functions[15]. 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 [33]. Xia et al. presented a method to parameterize the handlebodies using direct product parameterization [34].

Shell space. Shell structure is also widely used in computer graphics applications. Porumbescu et al. [20] 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. Wang et al. [31] presented a technique for rendering heterogenous translucent materials by solving the diffusion equation in the shell space.

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.

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* $\phi : M_0 \to P_0$
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 [17] 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 [24]. We also apply the variational meshing algorithm [1] to improve the quality of the tetrahedral mesh.

Following the divide-and-conquer approach [8], 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 [25]. Figure 2 shows the construction of shell space for the Skull model and its polycube domain.

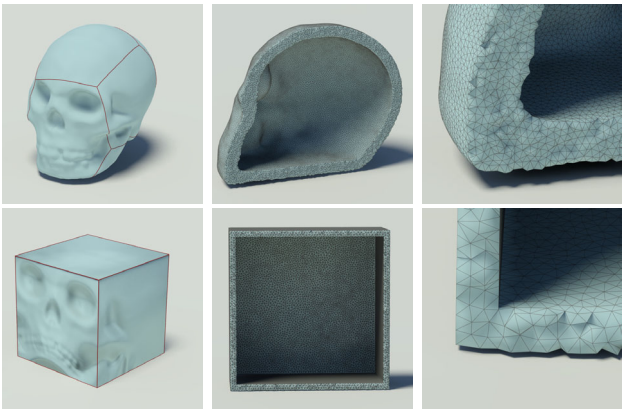
3.3 Parameterizing Shell Space

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

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

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

Since the shells M and P are represented by tetrahedral meshes, we solve the above Laplace equations by finite element method [32].



(a) Boundary surface (b) Shell space (c) Close-up view

Figure 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 variational meshing technique [1]. 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 the sharp features (polycube edges and corners) are well preserved in the tetrahedral mesh.

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 the similar way. For every interior vertex v_i, the harmonic function $f : M \to \mathbb{R}$ satisfies the condition

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

where w_{ij} is weight assigned to edge e_{ij}. Suppose edge e_{ij} is shared by m adjacent tetrahedra, it lies against m dihedral angles θ_k, $k = 1, \cdots, m$. 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}. Figure 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 $(v_1, \cdots, v_4)$, 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 equalling 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.

Input: An arbitrary point $v \in M$ in the shell object
$f : M \to \mathbb{R}$: harmonic function on M
$g : P \to \mathbb{R}$: harmonic funciton on P
$h : M_0 \to P_0$ 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.*

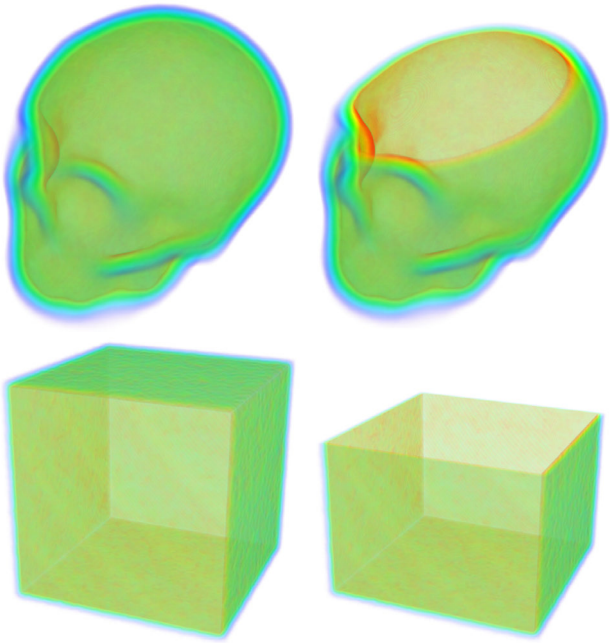
1.2 *Compute $v_0^{'} = h(v_0) \in P_0$*

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

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 the Appendix, we show that each integral curve has unique ending points, one on the inner boundary, the other on the outer boundary. Furthermore, any two integral curves do not intersect.



(a) Volume rendering (b) Cut view

Figure 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.

We construct the volume parameterization $\phi : M \to P$ 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 the Appendix, we prove that the map ϕ is bijective.

The key component in our volume parameterization is to trace the integral curves in the volumes. 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. We demonstrate the integral curve tracing in Fig. 4. The detailed tracing algorithm is shown as follows:

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* $!=$ *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: Integral Curve Tracing

3.4 Layered Hexahedral Meshing

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

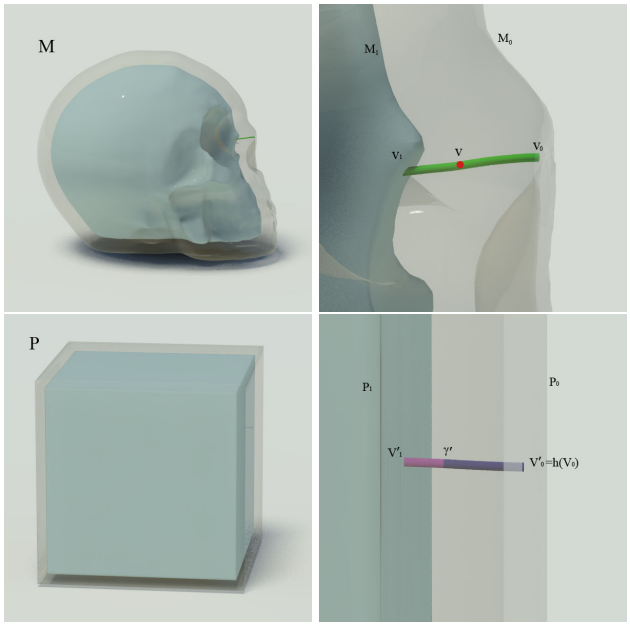


Figure 4: Tracing integral curves. Each integral curve follows the gradient vector field of the harmonic function. Thus, it is orthogonal to the iso-surface of the harmonic function (including the two boundary surfaces). We show that each integral curve has unique ending points and all integral curves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

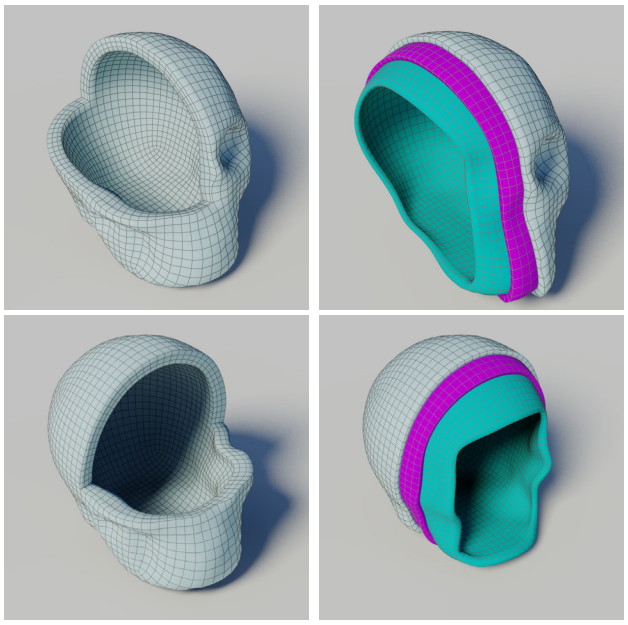


Figure 5: Layered hexahedral meshing of the Skull model.

As we show in the 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. Figure 5 shows the layered hexahedral mesh of the Skull model.

4. EXPERIMENTAL RESULTS

We tested our algorithm using models of various topology. In our experiments, the user-specified offset distance (i.e., thickness) 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. Figure 7 and 8 show the layered hexahedral meshes. Cutaway views are used to show 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 metric [11], the condition number of the Jacobian matrix [10] and the Oddy metric [16].

For a vertex of a hexahedron the metrics are formed as follows. Assume $x \in R^3$ is the position vector of this vertex and $x_i \in R^3$ 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 ≤ 0. We use the *Frobenius norm* as a matrix norm, $|J| = tr(J^T J)^{1/2}$. The condition number of the Jacobian matrix is defined as $k(J) = |J||J^{-1}|$, where $|J^{-1}| = \frac{|J|}{det(J)}$. The Oddy metric is $Oddy(x) = \frac{|J^T J|^2 - \frac{1}{3}|J|^4}{det(J)^{\frac{4}{3}}}$. As shown in Table 1 and Fig. 5, 7 and 8, the constructed meshes are visually pleasing and of high quality.

In this paper, we parameterize the shell space to a polycube domain mainly because it is relatively easy to construct the hexahedral tessellation. In fact, the proposed shell space parameterization works for arbitrary parametric domains as long as it has the same topology as the input model. Furthermore, the inner boundary surface M_1 and P_1 could be an arbitrary closed surface rather than the offset surface. The only requirement is that M_1 and P_1 are of the same topological type as M_0 and P_0. In Fig. 6 (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. Figure 6 (e)-(f) shows the layered hexahedral mesh of the embedded Duck model.

5. 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

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	180K	839K	3	7232	-0.02	(1.00,0.94,0.05)	(7.27,8.57,520.78)	(2.73,3.14,15.15)	18.1	6.7
Skull	0	190K	981K	3	4804	-0.04	(1.00,0.94,0.58)	(2.97,5.28,29.65)	(1.75,1.99,2.92)	29.5	8.9
Moai	0	180K	974K	5	2688	-0.08	(1.00,0.92,0.16)	(0.01,2.18,307.82)	(1.00,1.22,10.21)	33.7	10.0
Eight	2	180K	921K	3	12800	-0.02	(1.00,0.94,0.26)	(0.14,2.85,79.36)	(1.02,1.48,3.89)	26.2	23.8
Squirrel	0	180K	839K	5	8384	+0.02	(1.00,0.97,0.32)	(0.24,2.99,62.99)	(1.04,1.62,6.87)	19.8	13.0
Decocube	4	180K	850K	4	7936	+0.03	(1.00,0.96,0.70)	(0.41,1.74,13.04)	1.07,1.36,3.31)	21.9	14.8
Rockerarm	1	200K	943K	3	4672	+0.02	(1.00,0.92,0.25)	(0.14,2.95,45.77)	(1.02,1.41,4.34)	20.2	4.3
Isidore horse	0	200K	956K	2	4736	-0.01	(1.00,0.94,0.05)	(3.38,4.58,280.15)	(1.89,2.12,15.74)	22.3	2.4
Duck	0	100K	542K	10	11520	NA	(1.00,0.98,0.85)	(0.02,0.97,46.55)	(1.00,1.09,2.72)	19.6	23.2

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 2.66HGz CPU and 4GB memory.

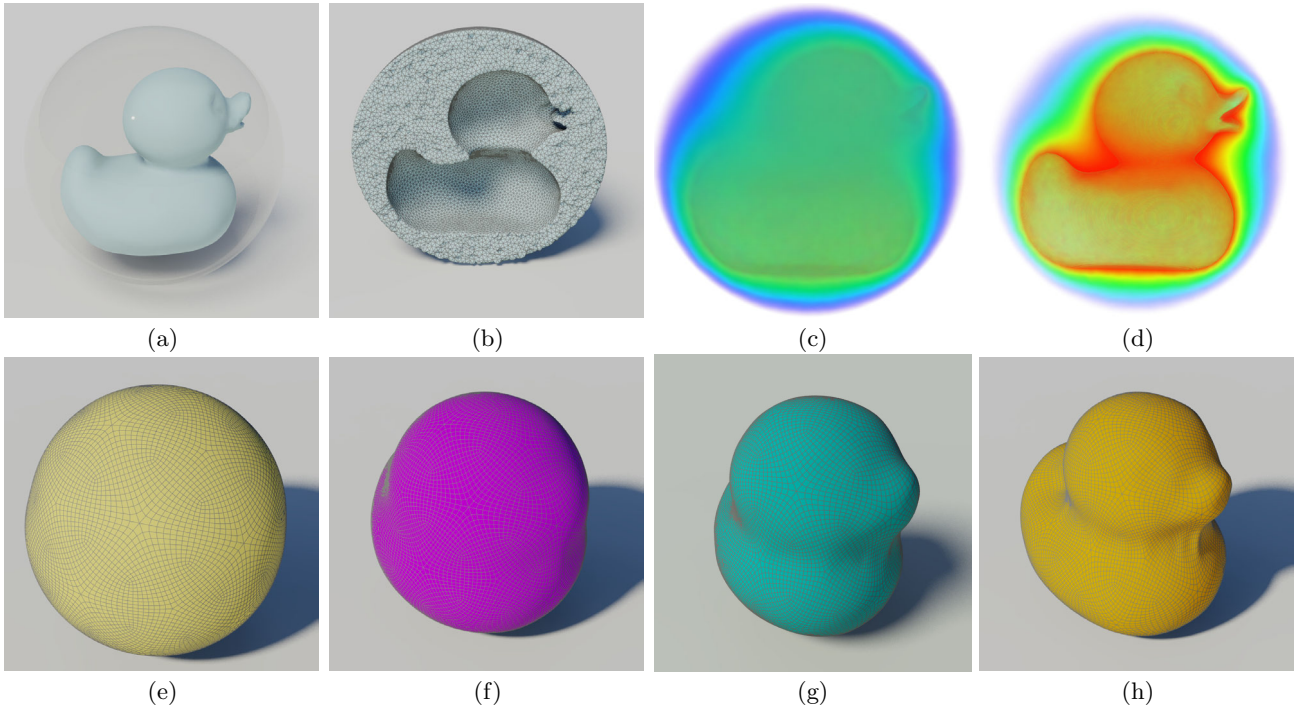


Figure 6: Layered hexahedral meshing of the Duck model embedded in a sphere. Note the outer and inner boundaries are significantly different (see (a)). (b) shows the isotropic tetrahedral mesh of the shell space. We parameterize this shell object to a ball by volumetric harmonic map (see (c) and (d)).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

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 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 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 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 Uni-

versity, Cyberware, and Aim@Shape Shape Repository. We would like to thank the anonymous reviewers for their constructive comments and David Gu for insightful discussions.

6. REFERENCES

[1] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. In *SIGGRAPH '05*, pages 617–625, 2005.

[2] S. E. Benzley, E. Perry, K. Merkley, B. Clark, and G. Sjaardema. A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis. In *Proceedings of 4th IMR*, pages 179–191, 1995.

[3] T. Blacker. Meeting the challenge for automated conformal hexahedral meshing. In *Proceedings of 9th IMR*, pages 11–20, 2000.

[4] A. O. Cifuentes and A. Kalbag. A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis. *Finite Elem. Anal. Des.*, 12(3-4):313–318, 1992.

[5] M. S. Floater. Mean value coordinates. *CAGD*, 20(1):19–27, 2003.

[6] M. S. Floater, G. Kós, and M. Reimers. Mean value coordinates in 3D. *CAGD*, 22(7):623–631, 2005.

[7] X. Gu, Y. He, and H. Qin. Manifold splines. *Graph. Models*, 68(3):237–254, 2006.

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

[9] T. Ju and S. Schaefer. Mean value coordinates for closed triangular meshes. In *ACM Trans. Graph*, volume 24, pages 561–566, 2005.

[10] P. Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part i - a framework for surface mesh optimization. 48:401–420, 2000.

[11] C. Kober and M. Muller-Hannemann. Hexahedral mesh generation for the simulation of the human mandible. In *Proceedings of the 9th IMR*, pages 423–434, 2000.

[12] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin. Harmonic volumetric mapping for solid modeling applications. In *Proceedings of ACM Symposium on Solid and Physical Modeling*, pages 109–120, 2007.

[13] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin. Meshless harmonic volumetric mapping using fundamental solution methods. *IEEE Trans. on Automation Science and Engineering*, 6(3):409–422, 2009.

[14] J. Lin, X. Jin, Z. Fan, and C. C. L. Wang. Automatic polycube-maps. In *GMP*, pages 3–16, 2008.

[15] T. Martin, E. Cohen, and M. Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In *Proceeding of ACM Symposium on Solid and Physical Modeling*, 2008.

[16] G. J. M. M. Oddy, A. and M. Bibby. Distortion metric for isoparametric finite elements. 12:213–217, 1988.

[17] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. In *SIGGRAPH '03*, pages 463–470, 2003.

[18] S. J. Owen. A survey of unstructured mesh generation technology. In *Proceedings of 7th IMR*, pages 239–267, 1998.

[19] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experiment. Math.*, 2(1):15–36, 1993.

[20] S. D. Porumbescu, B. Budge, L. Feng, and K. I. Joy. Shell maps. *ACM Transactions on Graphics*, pages 626–633, 2005.

[21] W. Quadros and K. Shimada. Hex-layer: Layered all-hex mesh generation on thin section solids via chordal surface transformation. In *Proceedings of 11th IMR*, pages 169–182, 2002.

[22] R. Schneiders, R. Schindler, and F. a. Weiler. Octree-based generation of hexahedral element meshes. In *Proceedings of 5th IMR*, pages 205–215, 1996.

[23] A. Sheffer, M. Etzion1, A. Rappoport, and M. Bercovier. Hexahedral mesh generation using the embedded voronoi graph. 15:248–262, 1999.

[24] H. Si. Tetgen: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. http://tetgen.berlios.de/.

[25] M. S. Smit and W. F. Bronsvoort. Variational tetrahedral meshing of mechanical models for finite element analysis. In *Computer-Aided Design and Applications*, pages 228–240, 2008.

[26] M. L. Staten, R. A. Kerr, S. J. Owen, and T. D. Blacker. Unconstrained paving and plastering: Progress update. In *Proceedings of 15th IMR*, pages 469–486, 2006.

[27] M. Tarini, K. Hormann, P. Cignoni, and C. Montani. Polycube-maps. In *Proceedings of SIGGRAPH 2004*, pages 853–860, 2004.

[28] T. J. Tautges and S. A. a. Mitchell. Whisker weaving: Invalid connectivity resolution and primal construction algorithm. In *Proceedings of 4th IMR*, pages 115–127, 1995.

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

[30] H. Wang, M. Jin, Y. He, X. Gu, and H. Qin. User-controllable polycube map for manifold spline construction. In *Proceedings of ACM Symposium on Solid and Physical Modeling*, pages 397–404, 2008.

[31] J. Wang, S. Zhao, X. Tong, S. Lin, Z. Lin, Y. Dong, B. Guo, and H.-Y. Shum. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Trans. Graph.*, 27(1):1–18, 2008.

[32] Y. Wang, X. Gu, and S.-T. Yau. Volumetric harmonic map. *Communications in Information and Systems*, 3(3):191–202, 2004.

[33] J. Xia, Y. He, S. Han, C.-W. Fu, F. Luo, and X. Gu. Parameterizing star shaped volumes using Green's function. In *Proceedings of Geometric Modeling and Processing (GMP '10)*, 2010.

[34] J. Xia, Y. He, X. Yin, S. Han, and X. Gu. Direct-product volumetric parameterization of handlebodies via harmonic fields. In *Proceedings of the International Conference on Shape Modeling and Applications 2010*, 2010.

[35] Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*, 195(9-12):942 – 960, 2006.

[36] Y. Zhang, R. Bajaj, and G. Xu. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. In *Proceedings of 14th IMR*, pages 449–468, 2005.

Appendix

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

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, $f : M \to \mathbb{R}$, $\triangle f = 0$, with Dirichlet boundary condition, $f|_{M_0} = 0$ and $f|_{M_1} = 1$. Let $C_f : M_0 \times [0, 1] \to M$ 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, $g : P \to \mathbb{R}$ and the integral curve $C_g : P_0 \times [0, 1] \to P$.

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

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

Then the volume parametrization 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 the ending points of each integral curve are on the inner and outer boundary surfaces respectively. Note f and g are smooth functions, their gradient vector fields are curl-free. Thus, no integral curve can form a loop inside the volume.

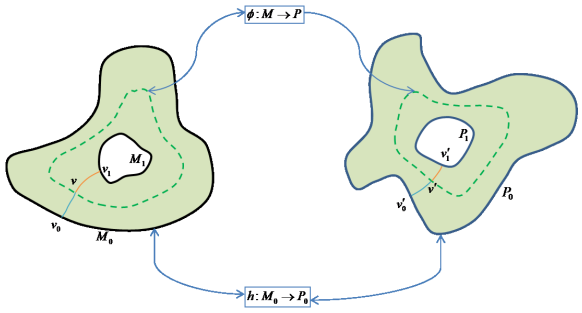
Second, we show that no integral curve that starts and ends on the same boundary surface. The function is harmonic and there is no critical points (where the gradient vanishes) inside the volume. Thus, the function value is strictly monotonic along the integral curve. Note that all points on the same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

Third, we show that two integral curves do not intersect. Assume 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 eigenvalue values. 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, the homotopy type of the level sets will be changed. The changes of the homotopy type can not be canceled out. Therefore, the homotopy type of M_0 is different from that of M_1. This contradicts the given condition.

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.



Last, 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 $h : M_0 \to P_0$ is homeomorphic, thus, it induces a homeomorphism between integral curves in M and P, $C_f(v_0, \cdot) \to C_g(h(v_0), \cdot)$, 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. QED.

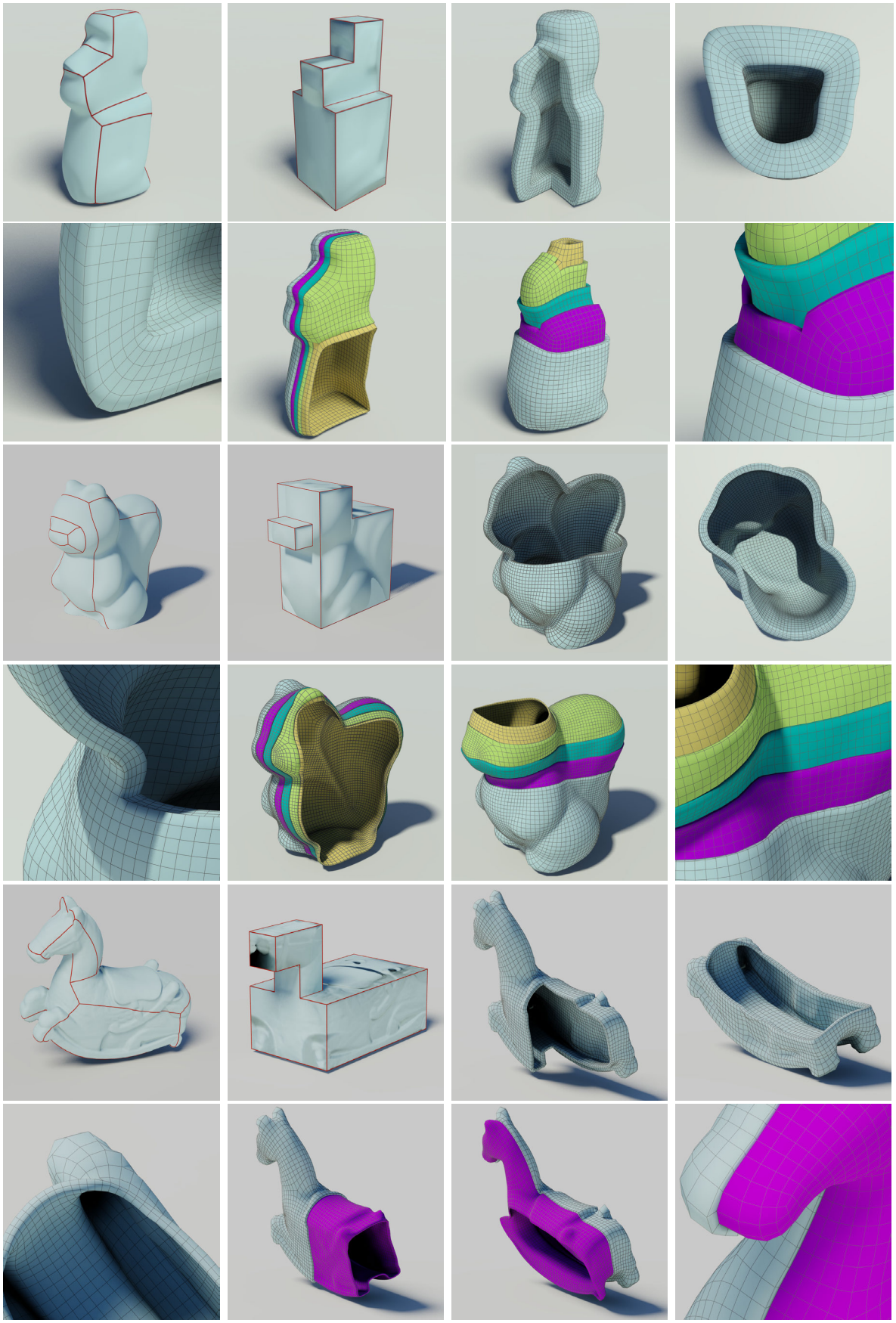


Figure 7: Experimental results on genus-0 models

Figure 8: Experimental results on high genus models