Multiresolution Computation of Conformal Structures of Surfaces

Xianfeng Gu[†] Yalin Wang[‡] Shing-Tung Yau[*]

[†]CISE, University of Florida, Gainesville, FL 32611

[‡] Mathematics Department, UCLA, Los Angeles, CA 90095

[*]Mathematics Department, Harvard University, Cambridge, MA 02138

{[†]gu@cise.ufl.edu [‡]ylwang@math.ucla.edu [*]yau@math.harvard.edu}

Abstract

An efficient multiresolution method to compute global conformal structures of nonzero genus triangle meshes is introduced. The homology, cohomology groups of meshes are computed explicitly, then a basis of harmonic one forms and a basis of holomorphic one forms are constructed. A progressive mesh is generated to represent the original surface at different resolutions. The conformal structure is computed for the coarse level first, then used as the estimation for that of the finer level, by using conjugate gradient method it can be refined to the conformal structure of the finer level.

1 Introduction

Geometric surfaces are represented as triangle meshes in computer aided geometry design and computer graphics. We treat the surfaces as complex manifolds and compute their holomorphic differentials (conformal structures). The obtained conformal structures and conformal invariants have broad applications, such as geometric classification by conformal transformation groups, geometric pattern recognition, global surface parameterization, texture mapping, and geometric processing etc. In the biomedical fields, global conformal parameterization can be applied to cortical surface matching problems.

The computation of conformal structures for meshes is based on theories from Riemann geometry. In our previous works, we have established practical algorithms to compute conformal structures. To the best of our knowledge, we are the first group to develop an algorithm to compute conformal structures for arbitrary surfaces represented as meshes. In this paper, we address the efficiency problem of the algorithms by introducing a multiresolution computation method.

The conformal structures are only determined by the geometry of the mesh, independent of triangulation and insensitive to resolution. Based on this fact, we are able to use the multiresolution method to improve the efficiency of the algorithm. For each mesh, we construct a progressive mesh first. Because the holomorphic differentials defined on the coarse level mesh are good approximations for those on the fine level mesh, we can compute them on the coarse level mesh, then refine them along with the mesh refinement. Our numerical experiments demonstrate that the multiresolution method improves the efficiency a great deal.

2 Previous work

Conformal parameterization method for genus zero surfaces have been studied and developed for the purpose of texture mapping, remeshing, but they can not discover the conformal structure of the surfaces.

Most works in conformal parametrization only deal with genus zero surfaces. There are several basic approaches, such as variational method [13, 12, 11, 5], approximation of Riemann-Cauchy equation [1], linearization of Laplacian operator [6].

The problem of computing global conformal structures for general closed meshes is first solved by Gu and Yau in [5] and [4]. The proposed method approximates De Rham cohomology by simplicial cohomology, and compute a basis of holomorphic one-forms. The method has solid theoretic bases. Gu and Yau generalize the method for surfaces with boundaries in [4]. Also the method is simplified, such that there is no restriction of the geometric realization for homology basis.

The conformal structure can be directly computed using global conformal parameterization method. The method introduced in this paper is based on those in [5] and [4] and improved by multiresolution method. This method is much more efficient and automatic.

The progressive mesh has been introduced by Hoppe et al in [7, 9], and widely used for mesh optimization [3, 14], efficient rendering applications [15, 8].

Global surface parameterization is also studied by Khodakovsky et al. in [10]. In [2], a novel method to solve sparse linear system using hardware with conjugate gradient method combined with multigrid method.

3 Progressive Mesh

Hoppe et al. [7] introduce a multiresolution representation for meshes - Progressive Mesh, which transform a mesh by edge collapse transformations, and recover it by vertex split transformations. An edge collapse transformation $ecol\{v_s, v_t\}$ unifies 2 adjacent vertices v_s and v_t into a single vertex v_s. The vertex v_t and the two adjacent faces $\{v_s, v_t, v_l\}$ and $\{v_t, v_s, v_r\}$ vanish in the process. The edge collapse transformations are invertible. The inverse transformation is $vertexsplit$. A vertex split transformation $vsplit\{s, l, r, t, A\}$ adds near vertex v_s a new vertex v_t and two new faces $\{v_s, v_t, v_l\}$ and $\{v_t, v_s, v_r\}$. Because edge collapse transformation is invertible, we can therefore represent an arbitrary triangle mesh M as a simple mesh M^0 together with a sequence of n $vsplit$ records. The progressive mesh representation of a mesh M is $(M^0, \{vsplit_0, vsplit_1, \cdots, vsplit_{n-1}\})$.

As an example, the mesh M of figure 3 was simplified down to the coarse mesh M^0 of figure using edges collapse transformations. The original mesh is with 50k faces, the base mesh M^0 is as simple as 4 faces.

4 Computing Homology

Given a triangle mesh $M = \{F, E, V\}$, we use combinatorial method to compute the homology group $H_1(M, Z)$ generators. Our method is similar to the classical retraction method in algebraic topology. The basic process is to remove a topological disk D as large as possible from M, then $H_1(M, Z)$ is equivalent to $H_1(M/D, Z)$. If D includes all the faces of M, then $G = M/D$ is a graph formed by some edges and vertices of M. The computation for $H_1(G, Z)$ is relatively easier. D is called a fundamental domain of M. The following is the detailed algorithm.

In the following discussion, we assume all faces and edges are oriented. We use $[v_1, v_2, \cdots, v_k]$ to represent the simplex spanned by $v_1, v_2, \cdots, v_k$, and use ∂ to represent the boundary operator. For examples, suppose a face $f = [v_0, v_1, v_2]$, where v_i are counter clock wisely ordered, then $\partial f = [v_0, v_1] + [v_1, v_2] + [v_2, v_0]$, $\partial [v_0, v_1] = v_1 - v_0$.

4.1 Fundamental Domain

In the following algorithm, the given mesh is denoted as M, the fundamental domain is denoted as D, its boundary ∂D is an ordered list of oriented edges. Q is a queue to store all non removed faces attaching to ∂D.

1. Choose an arbitrary face $f_0 \in M$, let $D = f_0$, $\partial D = \partial f_0$, put all the neighboring faces of f_0 which share an edge with f_0 to a queue Q.
2. while Q is not empty
   (a) remove the first face f in Q, suppose $\partial f = e_0 + e_1 + e_2$.
   (b) $D = D \cup f$.
   (c) find the first $e_i \in \partial f$, such that $-e_i \in \partial D$, replace $-e_i$ in ∂D by $\{e_{i+1}, e_{i+2}\}$ (keeping the order).
   (d) put all the neighboring faces which share an edge with f and not in D or Q to Q.
3. Remove all adjacent oriented edges in ∂D, which are opposite to each other, i.e. remove all pairs $\{e_k, -e_k\}$ from ∂D.

The resulting D includes all faces of M, which are sorted according to their enqueuing order. Define the graph $G = \{e, v | e \in \partial D, v \in \partial D\}$. The edges and vertices of the final boundary of D form the graph G. We will compute the homology basis of G, namely $H_1(G, Z)$.

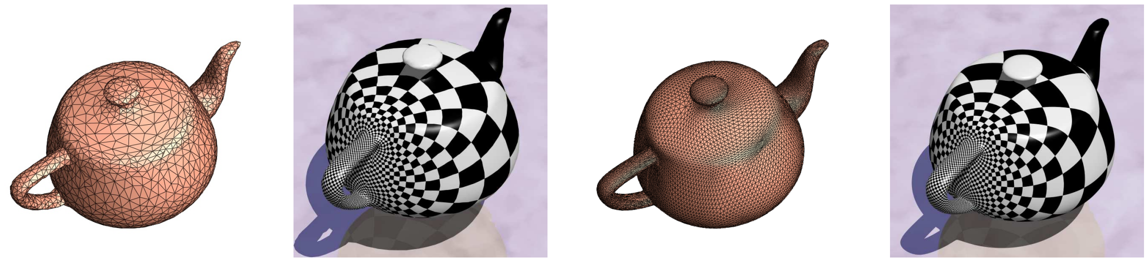
4.2 Homology Generators

Suppose T is a spanning tree of G, then $G/T = \{e_1, e_2, \cdots, e_{2g}\}$, e_i are disjoint edges. Suppose $\partial e_i = t_i - s_i$, t_i and s_i are two ending vertices of edge e_i, also two leaves of T.

We choose one vertex r as the root of T. By using depth first traversing T, we can find the shortest path from r to any leave. Suppose $[r, s_i]$ is the shortest path from r to s_i, $[r, t_i]$ is the shortest path from r to t_i, then $\zeta_i = [r, s_i] \cup e_i \cup -[r, t_i]$ is a closed loop, where $-[r, t_i]$ means the reversed path of $[r, t_i]$.

Then $\{\zeta_1, \zeta_2, \cdots, \zeta_{2g}\}$ is a set of basis of $H_1(G, Z)$, also $H_1(M, Z)$.

2



(a) 4k faces resolution (b) Result of (a) (c) 34K faces resolution (d) result of (c)

Figure 1. Surfaces are represented as triangle meshes. (a) and (c) are such representations with different resolutions. (b) and (d) are holomorphic differentials of (a) and (c), visualized by the texture-mapping of a checker board image. The conformality is illustrated by this texture mapping.

5 Computing Cohomology

We want to construct explicitly a basis for the cohomology group of M, $H^1(M,Z)$. A one form is a function defined on the edges of M, $\omega : E \to R$. We will find a set of one forms $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$, such that ω_i is closed,

$$\int_{\partial f} \omega_j = \omega_j(e_0) + \omega_j(e_1) + \omega_j(e_2) = 0 \qquad (1)$$

where $\partial f = e_0 + e_1 + e_2$, f is an arbitrary face of M. Also ω_i is dual to homology base ζ_i,

$$\int_{\zeta_i} \omega_j = \delta_i^j. \qquad (2)$$

where δ_i^j is the Kronecker symbol.

The following is the algorithm for constructing ω_i.

1. let $\omega_i(e_i) = 1$ and $\omega(e) = 0$, for any edge $e \in G$ and $e \neq e_i$.

2. Suppose D is ordered in the way that $D = \{f_1, f_2, \ldots, f_n\}$, reverse the order of D to $\{f_n, f_{n-1}, \cdots, f_1\}$.

3. while D is not empty

   (a) get the first face f of D, remove f from D,$\partial f = e_0 + e_1 + e_2$.

   (b) divide $\{e_k\}$ to two sets, $\Gamma = \{e \in \partial f | - e \in \partial D\}$, $\Pi = \{e \in \partial f | - e \notin \partial D\}$.

   (c) Choose the value of $\omega_i(e_k), e_k \in \Pi$ arbitrarily, such that $\sum_{e \in \Pi} \omega_i(e) = -\sum_{e\in\Gamma} \omega_i(e)$, if Π is empty, then the right hand side is zero.

   (d) Update the boundary of D, let $\partial D = \partial D + \partial f$.

6 Computing Harmonic one forms

In this step, we would like to diffuse the one forms computed in the last step to be harmonic. A harmonic one form is defined as the one minimizing harmonic energy. First we define discrete harmonic energy, given an edge $[u,v] \in M$, the harmonic energy string coefficient is defined as

$$k_{u,v} = \frac{1}{2}(ctan\alpha + ctan\beta), \qquad (3)$$

where α and β are two angles opposite to edge $[u,v]$. The harmonic energy for a one form ω on M is given by

$$E(\omega) = \frac{1}{2} \sum_{[u,v]\in M} k_{u,v}\omega([u,v])^2. \qquad (4)$$

Then the discrete Laplacian is a function defined on all the vertices on M. Suppose $u \in M$ is a vertex,

$$\Delta\omega(u) = \sum_{[u,v]\in M} k_{u,v}\omega([u,v]), \qquad (5)$$

$\Delta\omega$ is the discrete Laplacian of ω. Harmonic one form satisfies the following condition, for any vertex $u \in M$

$$\Delta\omega(u) = 0. \qquad (6)$$

Given a closed one form ω, we would like to find a function $f : V \to R$, such that $\Delta(\omega + df) = 0$, where df is defined as

$$df([u,v]) = f(v) - f(u) \qquad (7)$$

df is called an exact one-form. Hence we can add an exact one form to a closed one form, such that

$$\Delta(\omega+df)(u) = \sum_{[u,v]\in M} k_{u,v}(\omega([u,v])+f(v)-f(u)) = 0 \qquad (8)$$

The above equation is a sparse linear system, and can be solved using conjugate gradient method directly. This way we can convert the closed one forms computed in the last step to harmonic one forms.

7 Computing Holomorphic one forms

Suppose a set of harmonic one form basis $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$ have been found, we can define the discrete hodge star operator on them as follows. Given a face f, $\partial f = e_0 + e_1 + e_2$, we embed f in R^2, and build a local coordinate system (x, y) on f. Then all closed one form ω can be represented as $\phi dx + \tau dy$, such that

$$\int_{e_i} \phi dx + \tau dy = \omega(e_i) \tag{9}$$

where ϕ and τ are piecewise constant functions defined on faces. Then Hodge star operator is defined as

$$*(\phi dx + \tau dy) = (\phi dy - \tau dx). \tag{10}$$

We denote the Hodge star result of ω as ${}^*\omega$, then ${}^*\omega$ is well defined on each face, we call it the conjugate one form of ω. Given an edge e, there are two faces f_0, f_1 associated with it, we define ${}^*\omega(e) = \frac{1}{2}({}^*\omega_{f_0}(e) + {}^*\omega_{f_1}(e))$. The the holomorphic one form basis is given by $\{\omega_1 + \sqrt{-1}\,{}^*\omega_1, \omega_2 + \sqrt{-1}\,{}^*\omega_2, \cdots, \omega_{2g} + \sqrt{-1}\,{}^*\omega_{2g}\}$.

8 Surface with boundaries

For surface with boundaries, we use double covering techniques to convert it to a symmetric closed surface.

Suppose surface M has boundaries, we construct a copy of M denoted as M', then reverse the orientation of M' by changing the order of vertices of each face from $[u, v, w]$ to $[v, u, w]$. We then glue M and M' together along their boundaries. The resulting mesh is denoted as $\overline{M}$, and called the *double covering* of M. The double covering is closed so we can apply the method discussed in the previous sections.

For each interior vertex $v \in M$, there are two copies of v in $\overline{M}$, we denote them as v_1 and v_2, and say they are *dual* to each other, denoted as $\overline{v_1} = v_2$ and $\overline{v_2} = v_1$. For each boundary vertex $v \in M$, there is only one copy in $\overline{M}$, denoted as v, we say v is dual to itself, i.e. $\overline{v} = v$.

We now compute harmonic one forms of $\overline{M}$. According to Riemann surface theories [16], all symmetric harmonic one forms of $\overline{M}$ restricted on M are also harmonic one forms of M. A symmetric harmonic one form has the following property:

$$\omega[u, v] = \omega[\overline{u}, \overline{v}]. \tag{11}$$

Given a harmonic one form ω on $\overline{M}$, we can define a symmetric harmonic one form $\overline{\omega}$ as the following

$$\overline{\omega}([u, v]) = \frac{1}{2}(\omega([u, v]) + \omega([\overline{u}, \overline{v}])). \tag{12}$$

Assume $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$ is a set of harmonic one form basis of $\overline{M}$, then $\{\overline{\omega_1}, \overline{\omega_2}, \cdots, \overline{\omega_{2g}}\}$ is a basis of harmonic one forms of M. Then we can proceed to compute the holomorphic one form basis of M.

9 Multiresolution

The complexity of computing homology basis, harmonic one-form basis and holomorphic one-form basis are linear respectively. But for large scale geometric models, the computing process is still very time consuming. In order to improve the efficiency, we apply multi-resolution method to compute them.

Progressive mesh is used for this purpose, because edge collapse won't change the topology of the original surface, we can compute the homology basis in the coarse level. Also we compute harmonic one form in the coarse level. When we refine the mesh by vertex split transformation, we can use the coarse level result as the initial estimation for the harmonic one form of the finer level, and apply conjugate gradient algorithm to refine it. The following is the detailed algorithm:

1. Compute the progressive mesh of M, the base mesh is M_0.
2. Compute homology basis for the base mesh M_0.
3. Compute Cohomology basis for M_0.
4. Compute harmonic one form basis for M_0.
5. Refine M_0 by a sequence vertex split transformations, and refine the harmonic one form:
   (a) perform a vertex split, $\{v_s, v_t, v_l, v_r, A\}$
   (b) set $\omega([v_t, v_l]) = \omega([v_s, v_l])$,$\omega([v_t, v_r]) = \omega([v_s, v_r])$, $\omega([v_s, v_t]) = 0$.
   (c) if the number of vertex split transformation reaches a threshold, using conjugate gradient method to find a function f, such that $\omega + df$ is harmonic. Let $\omega = \omega + df$.
6. Use conjugate gradient method to find a function f, such that $\omega + df$ is a harmonic one form.

The conformal structure of surfaces is defined as the period matrix, which can be computed as the following:

$$P = (p_{ij}) = \int_{\zeta_i} \omega_j + \sqrt{-1}\,{}^*\omega_j. \tag{13}$$

From P, it can be verified whether two surfaces can be mapped to each other through conformal mappings.

4

10 Implementation and Results

We implement our algorithm using $C++$ on windows platform. We test the method for several real surface models. All meshes are constructed by laser scanners. Figure 3 illustrates a teapot model with different resolutions. The holomorphic one form is demonstrated by texture mapping. Figure 4 shows a result using multiresolution method. The bunny mesh is represented as a progressive mesh, and the holomorphic one forms are computed for different levels of resolution. Figure 4 and 3 demonstrate that the conformal structure is intrinsic to the geometry, and insensitive to the resolution. Figure 5 illustrates a surface with boundary case. We punch small holes at the tips of each finger, and double cover the mesh with boundaries. The five holomorphic one form bases are illustrated in the figure. The mesh has 60k faces.

We compare the speed for computing conformal structure for the same model with and without using multiresolution method. The speed is improved to two to ten times.

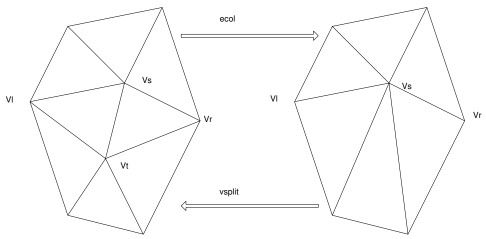


Figure 2. Edge collapse and vertex split transformations. v_l, v_r, v_s, v_t **are shown in the figure.**

11 Summary and Discussion

We introduce an efficient method to compute global conformal surface parameterization using multiresolution method. The computing process is as follows: we first compute a homology basis, construct a cohomology basis, then diffuse the cohomology basis to be harmonic 1-forms, then apply Hodge star operator on the harmonic 1-forms to get holomorphic 1-forms. Because global conformal structure is intrinsic to the surface geometry, so the lower resolution result can be used as a good estimation for that of higher resolution. We use conjugate gradient method to solve the large sparse linear system and use the lower resolution result as the initial estimation. The algorithm speed is improved up to ten times faster. The method introduced here can be generalized for zero genus surfaces, which is non-linear. The method can be generalized for other surface parameterization methods also.

References

[1] N. Ray B. Levy, S. Petitjean and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *Computer Graphics (Proceedings of SIGGRAPH 02)*. Addison Wesley, 2002.

[2] J. Bolz, L. Farmer, E. Grinsppun, and P. Schroder. Sparse matrix solvers on the gpu: Conjugate gradients and multigrid. In *ACM SIGGRAPH 2003*.

[3] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Computer Graphics (Proceedings of SIGGRAPH 95)*, pages 173–182.

[4] X. Gu, Y. Wang, T. Chan, P. Thompson, and S-T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. In *Information Processing Medical Imaging 2003*.

[5] Xianfeng Gu and Shing-Tung Yau. Computing conformal structures of surafces. *Communication of Informtion and Systems*, December 2002.

[6] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE TVCG*, 6(2):181–189, 2000.

[7] H. Hoppe. Progressive meshes. In *Computer Graphics (Proceedings of SIGGRAPH 96)*, pages 99–108.

[8] H. Hoppe. View-dependent refinement of progressive meshes. In *ACM SIGGRAPH 1997*, pages 189–198.

[9] H. Hoppe, T.Derose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *ACM SIGGRAPH 1993*, pages 19–26.

[10] A. Khodakovsky, N. Litke, and P. Schroder. View-dependent refinement of progressive meshes. In *ACM SIGGRAPH 2003*.

[11] M. Meyer M. Desbrun and P.Alliez. Intrinsic parametrizations of surface meshes. In *Proceedings of Eurographics*, 2002.

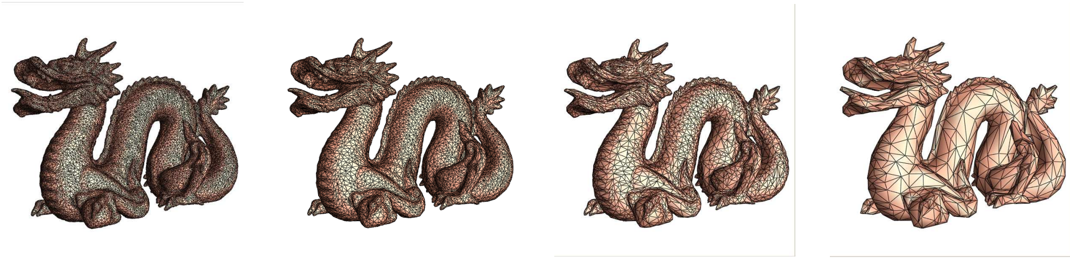
[12] M. Meyer P. Alliez and M. Desbrun. Interactive geomety remeshing. In *Computer Graphics (Proceedings of SIGGRAPH 02)*, pages 347–354.

[13] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugate. In *Experimental Mathematics 2(1)*, pages 15–36, 1993.

[14] J. Popovic and H. Hoppe. Progressive simplicial complexes. In *ACM SIGGRAPH 1997*, pages 217–224.

[15] P. Sander, J. Snyder, S. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *Computer Graphics (Proceedings of SIGGRAPH 01)*, pages 409–416, 2001.

[16] R. Schoen and S.T. Yau. *Lectures on Harmonic Maps.* International Press, Harvard University, Cambridge MA, 1997.



(a) 50K faces (b) 25K faces (c) 12 K faces (d) 6K faces

Figure 3. Progressive mesh for the dragon model. (a) through (d) are the mesh at different level of details.

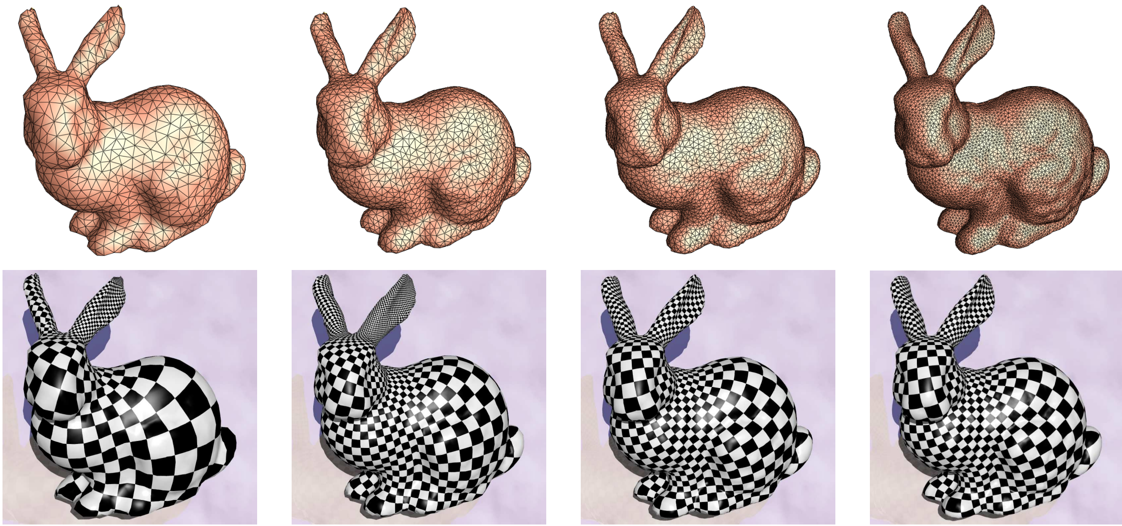


Figure 4. Multi-resolution for the Stanford bunny model. The resolutions are 5k faces, 10k faces, 18k faces and 40k faces respectively. The holomorphic one forms are visualized by texture mapping a checker board image. It is shown that the holomorphic one form is intrinsic to the geometry, and insensitive to the resolution.

Figure 5. Double covering surface of the hand model. A hole is punched at each finger tip, and the bottom of the wrist is removed. There are five holomorphic one form bases, illustrated by texture mapping.

6