Quadrilateral Mesh Generation III : Optimizing Singularity Configuration Based on Abel-Jacobi Theory

Xiaopeng Zheng[a,c], Yiming Zhu[a], Wei Chen[a], Na Lei[a,d,*], Zhongxuan Luo[a,c], Xianfeng Gu[b]

[a] *Dalian University of Technology, Dalian, China*
[b] *Stony Brook University, New York, US*
[c] *Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, China*
[d] *DUT-RU Co-Research Center of Advanced ICT for Active Life, Dalian, China*

Abstract

This work proposes a rigorous and practical algorithm for quad-mesh generation based the Abel-Jacobi theory of algebraic curves. We prove sufficient and necessary conditions for a flat metric with cone singularities to be compatible with a quad-mesh, in terms of the deck-transformation, then develop an algorithm based on the theorem. The algorithm has two stages: first, a meromorphic quartic differential is generated to induce a T-mesh; second, the edge lengths of the T-mesh are adjusted by solving a linear system to satisfy the deck transformation condition, which produces a quad-mesh.

In the first stage, the algorithm pipeline can be summarized as follows: calculate the homology group; compute the holomorphic differential group; construct the period matrix of the surface and Jacobi variety; calculate the Abel-Jacobi map for a given divisor; optimize the divisor to satisfy the Abel-Jacobi condition by integer programming; compute a flat Riemannian metric with cone singularities at the divisor by Ricci flow; isometrically immerse the surface punctured at the divisor onto the complex plane and pull back the canonical holomorphic differential to the surface to obtain the meromorphic quartic differential; construct a motorcycle graph to generate a T-Mesh.

*Corresponding author
Email address: nalei@dlut.edu.cn (Na Lei)

In the second stage, the deck transformation constraints are formulated as a linear equation system of the edge lengths of the T-mesh. The solution provides a flat metric with integral deck transformations, which leads to the final quad-mesh.

The proposed method is rigorous and practical. The T-mesh and quad-mesh results can be applied for constructing Splines directly. The efficiency and efficacy of the proposed algorithm are demonstrated by experimental results on surfaces with complicated topologies and geometries.

Keywords: Quadrilateral Mesh, T-mesh, Spline, Abel-Jacobi, Flat Riemannian Metric, Geodesic, Discrete Ricci flow, deck transformation, Divisor

1. Introduction

In computational mechanics, computer-aided-design, geometric design, computer graphics, medical imaging, digital geometry processing and many other engineering fields, quadrilateral mesh is a universal and crucial boundary surface representation. Although quadrilateral meshes have been broadly applied in the real industrial world, the theoretic understanding of their geometric structures remains primitive. Recently, [29] made a breakthrough from the algebraic geometric view: basically a quad-mesh induces a conformal structure and can be treated as a Riemann surface. Furthermore, a quad-mesh is equivalent to a meromorphic quartic differential with closed trajectories, and the singularities satisfy the Abel-Jacobi condition. This discovery provides a solid theoretic foundation for quad-meshing.

1.1. Abel-Jacobi Condition

Suppose a closed surface $(\Sigma, \mathbf{g})$ is embedded in Euclidean space $\mathbb{R}^3$ with the induced Euclidean Riemannian metric $\mathbf{g}$. Suppose the surface is represented as a quadrilateral mesh $\mathcal{Q}$, then $\mathcal{Q}$ induces a special combinatorial structure, a Riemannian metric structure, and a conformal structure.

Combinatorial structure: Suppose the number of vertices, edges, faces of $\mathcal{Q}$ are V, E, F, then $E = 2F$ and the Euler formula holds, $V + F - E = \chi(\Sigma)$, where $\chi(\Sigma)$ is the Euler characteristic number of Σ. The vertices with topological valence 4 are called *normal*; otherwise they are *singular*.

Riemannian metric structure: A flat metric with cone singularities $\mathbf{g}_Q$ can be induced by $\mathcal{Q}$ by treating each face as a unit planar square. A vertex with k-valence has the discrete curvature $\frac{4-k}{2}\pi$, and the total curvature satisfies the *Gauss-Bonnet condition*:

$$\sum_v \frac{4 - \mathrm{val}(v)}{2}\pi = 2\pi\chi(\Sigma), \tag{1}$$

where $\mathrm{val}(v)$ is the topological valence of v. The holonomy group induced by the metric $\mathbf{g}_Q$ on the surface $\Sigma \setminus \mathcal{S}$ with punctures at the singular vertices $\mathcal{S}$ is the rotation group

$$\mathrm{Hol}(\Sigma \setminus \mathcal{S}, \mathbf{g}_Q) = \{e^{i\frac{\pi}{2}k}, k \in \mathbb{Z}\}. \tag{2}$$

This is the so-called *holonomy condition* [11].

Furthermore, let $(\tilde{\Sigma}, \pi)$ be the universal covering space of the punctured surface $\Sigma \setminus S$, where $\pi : \tilde{\Sigma} \to \Sigma \setminus S$ is the projection map. Then π pulls back the quad-mesh metric $\mathbf{g}_Q$ to $\tilde{\mathbf{g}}_Q$. If the Deck transformation group of $\tilde{\Sigma}$ is $Deck(\tilde{\Sigma})$, then each deck transformation $\tau \in Deck(\tilde{\Sigma})$ is an isometric automorphism of $(\tilde{\Sigma}, \tilde{\mathbf{g}}_Q)$, and satisfies the relation $\pi \circ \tau = \pi$. Furthermore, it must have the form

$$\forall \tau \in Deck(\tilde{\Sigma}), \quad \tau(z) = e^{i\frac{\pi}{2}k}z + (u, v), \quad k, u, v \in \mathbb{Z}, \tag{3}$$

We call this as the *deck transformation condition*.

Conformal structure: The quad-mesh $\mathcal{Q}$ induces a conformal structure, and can be treated as a Riemann surface S_Q; furthermore, it induces a meromorphic quartic differential ω_Q, whose horizontal and vertical trajectories are finite. The vertices of $\mathcal{Q}$ with valence less than 4 are the poles of ω_Q, the vertices with valence greater than 4 are the zeros of ω_Q. The divisor of ω_Q represents the configuration of singularities of $\mathcal{Q}$, denoted as (ω_Q). Suppose φ is a holomorphic

1-form on S_Q, then φ^4 is a holomorphic quartic differential. Then (ω_Q) and $4(\varphi)$ are equivalent, and satisfy the Abel-Jacobi condition, the image of the Abel-Jacobi map, $\mu((\omega_Q) - 4(\varphi))$, is zero in the Jacobi variety $(J(S_Q))$.

1.2. T-mesh and Quad-mesh generation

The procedure to generate quadrilateral meshes can be summarized as two stages: the first stage is to construct a meromorphic quartic differential, which leads to a T-mesh; the second stage updates the edge lengths of the T-mesh to satisfy the deck transformation condition by solving a linear system, the updated metric induces the desired quad-mesh.

T-mesh generation. The algorithm for T-meshing is as follows: 1) choose an arbitrary set of points on surface as the initial singularities, which capture the geometric features of the surface, such as the extreme points of the Gaussian curvature; 2) improve the initial singularity set to satisfy the Abel-Jacobi condition; the resulting singularity set is denoted as $\mathcal{S}$; 3) construct a meromorphic quartic differential ω, whose divisor (ω) equals to $\mathcal{S}$ with multiplicity; 4) trace the horizontal and vertical critical trajectories of ω to form a motorcycle graph T, which is the T-mesh.

If the initial singularities do not satisfy the Gauss-Bonnet condition 1, we will add more poles and zeros at the critical points of Gaussian curvature. Then we compute the Abel-Jacobi map, minimize the squared norm of the image of the divisor using gradient descent method. Once the divisor satisfies the Abel-Jacobi condition, we use surface Ricci flow to compute a flat cone metric, and isometrically immerse the surface with punctures at the singularities into the plane. On the plane $\mathbb{C}$, there is a canonical holomorphic quartic differential $(dz)^4$, which can be pulled back to the surface by the immersion map to a meromorphic quartic differential ω, which is globally defined on the original surface by construction. The critical trajectories of ω form a motorcycle graph, which partitions the surface to a T-mesh.

Quad-mesh Generation. We adjust the edge lengths of the T-mesh, preserve all the faces to be rectangular, make all the translational components of deck transformations to be rational and ensure the parametric positions of singularities are rational as well. These conditions are formulated as a square linear system, the solution gives us a flat metric with cone singularities, which satisfies the Abel-Jacobi condition and the rational deck transformation condition formulated in Theorem 3.23. The metric leads to a quad-mesh directly.

1.3. Contributions

This work proves the sufficient and necessary conditions for a flat metric with cone singularities to be compatible with a quad-mesh, namely the deck transformation condition described in theorem 3.22 and 3.23. The work then proposes a novel algorithm to generate T-meshes and quad-meshes based on the Abel-Jacobi theory and the deck transformation condition theorems. The algorithm first finds a meromorphic quartic differential, which leads to a seamless parameterization and a T-mesh, then deforms the T-mesh metric to satisfy the deck transformation condition, which leads to the desired quad-mesh. To the best of our knowledge, this is the first work that constructs meromorphic quartic differentials for quad-meshing. The experimental results demonstrate the algorithm is rigorous, effective and efficient.

The work is organized as follows: Section 2 briefly reviews the most related works; Section 3 introduces the theoretic background; Section 4 explains the algorithm in details; the experimental results are reported in Section 5; finally, the work concludes in Section 6.

2. Previous Works

This section briefly reviews the most related works. We refer readers to [5] for more thorough reviews. Quad-mesh generation has a vast literature. In the following we only discuss some of the most popular approaches.

Triangle Mesh Conversion. Catmull-Clark subdivision method can be applied to convert triangular meshes to quad-meshes, and then the original vertices become singularities. Another intuitive way is to merge two triangular faces adjacent to the same edge to a quadrilateral as proposed in [38, 33, 21, 41]. These type of methods can only produce unstructured quad-meshes, without much quality control.

Patch-Based Approach. In order to generate semi-regular quad-meshes, this type of methods calculates the skeleton first, then partitions the mesh into several quadrilateral patches. Each patch is regularly tessellated into quads. There are different strategies to cluster the faces to form each patch. One way is to merge neighboring triangle faces based on the similarity among the normals [23]. The other is based on the distance among the centers of the faces [3, 10]. Poly-cube map is a normal based method to deform the surface to a poly-cube shape, such as [44, 43, 31, 22]. The Morse-Smale complex of eigenfunction of the Laplace operator naturally produces a skeleton structure, which is utilized to generate quad-meshes. The spectral surface quadrangulation method applies this method in [15, 24].

Voronoi Based Approach. This approach puts samples on the input surface, then computes a Voronoi diagram on the surface using different distances. For example, if L^p norm is applied, then the cells are similar to rectangles [28]. This method can only generate a non-structured quad-mesh.

Cross field Based Approach. This approach generates a cross field first, then by tracing the stream lines of the cross field [37] or iso-parametric lines of the parameterization induced by the field [5], the quad-mesh can be constructed. The cross fields are represented in different ways, such as the N-RoSy representation[35], the period jump technique [30] and the complex value representation [27]. Then by minimizing the discrete analogy to the harmonic energy [25], the cross field can be smoothed. The work in [42] relates the Ginzberg-Landau theory with the cross field for genus zero surface case. The theoretical foundation of [42]

is only for the topological disk case, though other proofs in the work apply for more general surfaces. This type of method is difficult to control the positions of the singularities and the global structure of the quad layout. Cross fields can be treated as the horizontal and vertical directions of a meromorphic quartic differential without magnitudes.

General Parameterization Based Approach . The parameteization method maps the surface onto a planar domain, constructs a quad-mesh on the parameter domain, and then pulls it back to the surface. There are different ways to compute the parameterization, such as using discrete harmonic forms [40], periodic global parameterization [1] and branched coverings method [26]. Several algorithms have been proposed to generate global parameterizations aligned with the cross field. The work in [34] proposed a global parameterization algorithm for the purpose of T-Spline construction, which aligns with a prescribed field robustly. Integrable polyvector fields method is developed in [13], which generates a global parameterization aligned with the fields by constructing a curl-free vector field. All these methods rely on solving elliptic partial differential equations on surfaces.

Seamless Parameterization Based Approach. The concept of *seamless parameterization* was first proposed in [36]. According to Definition 1 in that work, the transition maps between the local parameterizations have the form

$$\tau(u+\sqrt{-1}v) = e^{\sqrt{-1}\frac{k}{2}\pi}(u+\sqrt{-1}v) + (s+\sqrt{-1}t), \quad k \in \mathbb{Z}.$$

Equivalently, a seamless parameterization defines a flat metric with cone singularities, the holonomy group consists of rotations of $\frac{k}{2}\pi$, hence the deck transformations have rotations of $\frac{k}{2}\pi$. In turn, a seamless parameterization is equivalent to a meromorphic quartic differential, hence its singularities satisfy the Abel-Jacobi condition.

Later Campen and Zorin developed the concept of *seamless similarity parametrization* or *seamless similarity map* for the purpose of T-Spline construction [7].

Here the deck transformation has the form

$$\tau(u+\sqrt{-1}v) = ce^{\sqrt{-1}\frac{k}{2}\pi}(u+\sqrt{-1}v) + (s+\sqrt{-1}t), \quad k \in \mathbb{Z}, c \in \mathbb{R}_+,$$

where c represents the key scaling transformation. Therefore, the seamless similarity parameterization generalizes the seamless parameterization. There are distinctions between the two concepts: the seamless similarity parameterization only defines a connection satisfying the holonomy condition, but not a Riemannian metric with cone singularities; in contrast, the seamless parameterization defines a metric. Furthermore, the conformal structure given by the seamless parameterization covers the whole surface (including the singularities), but that of seamless similiarty parametrization doesn't cover the singularities. In our current work, we directly compute the seamless parameterization using the Abel-Jacobi condition.

Campen and Zorin [9, 8] proposed to obtain a seamless similarity parameterization by convex optimization using Newton's method. By constructing a motorcycle graph [17], a T-mesh can be obtained and further converted to a quad-mesh [7]. Note that, because a seamless similarity parameterization doesn't give a metric, (but an affine structure for the punctured surface, which is sufficient for the purpose of constructing T-Splines), their method can not define the edge lengths of the T-mesh. Then they treat the edge lengths as unknowns, and add two types of constraints: a. each edge length is a non-negative integer (or rational); b. each face of the T-mesh is a rectangle, namely the lengths of opposite sides are equal. The constraints form a linear Diophantine equation system, its solution leads to a quad-mesh. In a recent work [32], they improved the algorithm by adding more constraints to the edge lengths to reduce the angle distortions introduced by quantization.

Our current work finds a meromorphic quartic differential, gets a seamless parameterization, constructs a motorcycle graph [17] and obtains a T-mesh. Since the seamless parameterization defines a metric, the T-mesh has initial edge lengths. Then we adjust the edge lengths to satisfy the deck transforma-

tion conditions, (including singularity position constraints, deck transformation constraints and the face side constraints), and the changes of edge lengths are treated as unknowns. According to our Theorem 3.23, these constraints are sufficient and necessary conditions for quad-meshing. The constraints in Campen and Zorin's work are sufficient conditions, but not necessary. For example, an edge length connecting a singularity and a T-junction needn't to be integer for the purpose of quad-meshing. Therefore, the number of our constraints is less than that of their method, the linear system is easier to solve. (In fact, the initial metric is a solution to the same linear system.)

Integer-grid Mapping Based Approach. As a parameterization method, integer-grid mappings, similar to finite trajectories parameterization, have been widely applied to the high-quality quad mesh generation [40, 26, 6, 7, 4, 16]. The additional integer constraints are imposed on the translation part, as opposed to the seamless parameterization, the transition maps have the form

$$\tau(u+\sqrt{-1}v) = e^{\sqrt{-1}\frac{k}{2}\pi}(u+\sqrt{-1}v) + (s+\sqrt{-1}t), \quad k,s,t \in \mathbb{Z}.$$

The essence is to construct a local injective mapping from the surface to the planar integer-grid, so that the grid is pulled back to produce a quad mesh. In addition to computing integer-grid mappings directly [40], most approaches [7, 6, 4, 26] tend to obtain a seamless parameterization and then adopt some form of rounding method to adjust it to an integer-grid mapping, called quantization. Sometimes the degradation of resulting parameterization as well as the infeasibility of constrained problem are inevitable. In our work, we introduce the deck transformation condition, which has fewer constraints and the linear system is easier to solve.

Comparing to the existing approaches, our method has explicit theoretic analysis for the singularities, the dimension of solution space, the deck transformation conditions for quad-meshing. Therefore the theoretic rigor will help improve the efficiency and efficacy for quad-mesh generation.

3. Theoretic Background

This section briefly introduces the most related mathematical concepts and theorems. We refer readers to [14] for the basic concepts and theorems in Riemann surface theory, [39] for those in algebraic topology, and [29] for the detailed proof of Theorem 3.16 and Theorem 3.17.

3.1. Basic Concepts of Riemann Surface

Definition 3.1 (Riemann Surface)**.** *Suppose Σ is a two dimensional topological manifold, equipped with an atlas $\mathcal{A} = \{(U_\alpha, \varphi_\alpha)\}$, every local chart has complex coordinates $\varphi_\alpha : U_\alpha \to \mathbb{C}$, denoted as z_α, and every transition map $\varphi_{\alpha\beta}$ is biholomorphic,*

$$\varphi_{\alpha\beta} : \varphi_\alpha(U_\alpha \cap U_\beta) \to \varphi_\beta(U_\alpha \cap U_\beta), \quad z_\alpha \mapsto z_\beta,$$

then the atlas is called a conformal atlas. A topological surface with a conformal atlas is called a Riemann surface.

Suppose $(\Sigma, \mathbf{g})$ is an oriented surface with a Riemannian metric $\mathbf{g}$. For each point $p \in \Sigma$, we can find a neighborhood $U(p)$, inside $U(p)$ the *isothermal coordinates* (u, v) can be constructed, such that $\mathbf{g} = e^{2\lambda(u,v)}(du^2 + dv^2)$. The atlas formed by all the isothermal coordinates is a conformal atlas, therefore we obtain the following:

Theorem 3.2. *All oriented, metric surfaces are Riemann surfaces.*

Definition 3.3 (Meromorphic Function)**.** *A complex function on a domain $\Omega \subset \mathbb{C}$ to $\mathbb{C} \cup \{\infty\}$ is called meromorphic, if there exists a sequence of points $p_1, p_2, \cdots$ with no limit point in Ω such that*

1. *$f : \Omega \setminus \{p_1, p_2, \dots\} \to \mathbb{C}$ is holomorphic;*
2. *f has poles at $p_1, p_2, \dots$, namely $\lim_{z \to p_i} |f(z)| = \infty$.*

Definition 3.4 (Meromorphic Function on Riemann Surface)**.** *Suppose a Riemann surface $(\Sigma, \{(U_\alpha, \varphi_\alpha)\})$ is given. A complex function is defined on the*

surface $f : \Sigma \to \mathbb{C} \cup \{\infty\}$. *If on each local chart* $(U_\alpha, \varphi_\alpha)$, *the local representation of the functions* $f \circ \varphi_\alpha^{-1} : \mathbb{C} \to \mathbb{C} \cup \{\infty\}$ *is meromorphic, then* f *is called a meromorphic function defined on* Σ.

A meromorphic function can be treated as a holomorphic map from the Riemann surface to the unit sphere.

Definition 3.5 (Zeros and Poles)**.** *Given a meromorphic function* $f(z)$, *if its Laurent series has the form*

$$f(z) = \sum_{n=k}^{\infty} a_n (z-p)^n,$$

if $k > 0$, *then* p *is called a zero point of order* k; *if* $k < 0$, *then* p *is called a pole of order* k; *if* $k = 0$, *then* p *is called a regular point. We denote* $\nu_p(f) = k$.

Definition 3.6 (Meromorphic Differential)**.** *Given a Riemann surface* $(\Sigma, \{z_\alpha\})$, ω *is a meromorphic differential of order* n, *if it has the local representation,*

$$\omega = f_\alpha(z_\alpha)(dz_\alpha)^n,$$

where $f_\alpha(z_\alpha)$ *is a meromorphic function,* n *is an integer; if* $f_\alpha(z_\alpha)$ *is a holomorphic function, then* ω *is called a holomorphic differential of order* n. *If* z_α *is a pole (or a zero) of* f_α *with order* k, *then* z_α *is called a pole (or a zero) of the meromorphic differential* ω *of order* k.

A holomorphic differential of order 2 is called a *holomorphic quadratic differential*. A meromorphic differential of order 4 is called a *meromorphic quartic differential*.

Definition 3.7 (Divisor)**.** *The Abelian group freely generated by points on a Riemann surface is called the divisor group, every element is called a divisor, which has the form*

$$D = \sum_p n_p p, \quad n_p \in \mathbb{Z},$$

where only a finite number of points p*'s have non-zero coefficients* n_p. *The degree of a divisor is defined as* $deg(D) = \sum_p n_p$. *Suppose* $D_1 = \sum_p n_p p$,

$D_2 = \sum_p m_p p$, then $D_1 \pm D_2 = \sum_p (n_p \pm m_p)p$; $D_1 \leq D_2$ if and only if for all p, $n_p \leq m_p$.

Definition 3.8 (Meromorphic Differential Divisor)**.** *Suppose ω is a meromorphic differential on a Riemann surface Σ, suppose $p \in \Sigma$ is a point on Σ, we define the order of ω at p as*

$$\nu_p(\omega) = \nu_p(f_p),$$

where f_p is the local representation of ω in a neighborhood of p, $\omega = f_p(z)(dz)^n$. The divisor of ω is defined as

$$(\omega) = \sum_p \nu_p(\omega)p.$$

3.2. Abel-Jacobi Theorem

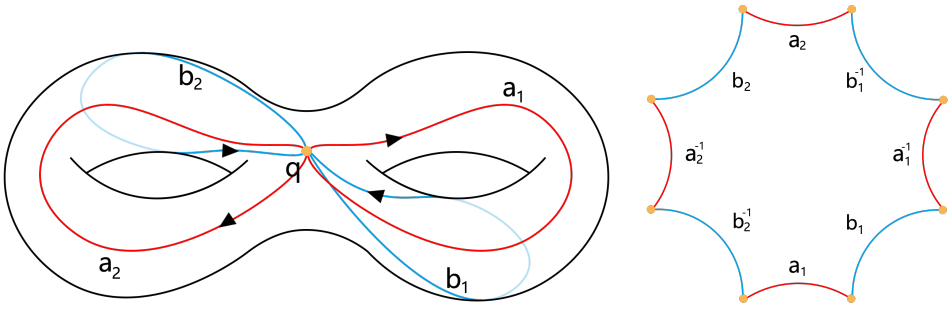


Figure 1: Canonical fundamental group basis.

Suppose $\{a_1, b_1, \ldots, a_g, b_g\}$ is a set of canonical basis for the homology group $H_1(\Sigma, \mathbb{Z})$ as shown in Fig. 1. Each a_i and b_i represent the curves around the inner and outer circumferences of the ith handle.

Let $\{\omega_1, \omega_2, \ldots, \omega_g\}$ be a normalized basis of $\Omega^1(\Sigma)$, the linear space of all holomorphic 1-forms over $\mathbb{C}$. The choice of basis is dependent on the homology basis chosen above; the normalization signifies that

$$\int_{a_i} \omega_j = \delta_{ij}, \quad i, j = 1, 2, \ldots, g.$$

For each curve γ in the homology group, we can associate a vector λ_γ in $\mathbb{C}^g$ by integrating each of the g 1-forms over γ,

$$\lambda_\gamma = \left(\int_\gamma \omega_1, \int_\gamma \omega_2, \ldots, \int_\gamma \omega_g \right)^T.$$

Definition 3.9 (Period Matrix)**.** *The matrix* (A, B) *where*

$$A = \left(\lambda_{a_1}, \lambda_{a_2}, \cdots, \lambda_{a_g} \right), \quad B = \left(\lambda_{b_1}, \lambda_{b_2}, \cdots, \lambda_{b_g} \right)$$

is called the period matrix of the Riemann surface.

We define a $2g$-real-dimensional lattice Λ in $\mathbb{C}^g$,

$$\Lambda = \left\{ \sum_{i=1}^{g} s_i \ \lambda_{a_i} + \sum_{j=1}^{g} t_j \ \lambda_{b_j}, \quad s_i, t_j \in \mathbb{Z} \right\}.$$

Definition 3.10 (Jacob Variety (Jacobian))**.** *The Jacobi variety (Jacobian) of the Riemann surface* Σ*, denoted* $J(\Sigma)$*, is the compact quotient* $\mathbb{C}^g/\Lambda$*.*

Definition 3.11 (Abel-Jacobi Map)**.** *Fix a base point* $p_0 \in \Sigma$*. The Abel-Jacobi map is a map* $\mu : \Sigma \to J(\Sigma)$*. For every point* $p \in \Sigma$*, choose a curve* γ *from* p_0 *to* p*; the Abel-Jacobi map* μ *is defined as follows:*

$$\mu(p) = \left(\int_\gamma \omega_1, \int_\gamma \omega_2, \ldots, \int_\gamma \omega_g \right) \qquad \mod \Lambda,$$

where the integrals are all along γ*.*

It can be shown that $\mu(p)$ is well-defined, that the choice of curve γ doesn't affect the value of $\mu(p)$.

Theorem 3.12 (Abel-Jacobi)**.** *Let* D *be a divisor of degree* 0 *on* Σ*, then* D *is the divisor of a meromorphic function* f *if and only if* $\mu(D) = 0$ *in the Jacobian* $J(\Sigma)$*.*

3.3. Quad-Meshes and Meromorphic Quartic Forms

We summarize the intrinsic relation between a quad-mesh and a meromorphic quartic differential.

Definition 3.13 (Quadrilateral Mesh)**.** *Suppose Σ is a topological surface, $\mathcal{Q}$ is a cell partition of Σ, if all cells of $\mathcal{Q}$ are topological quadrilaterals, then we say $(\Sigma, \mathcal{Q})$ is a quadrilateral mesh.*

On a quad-mesh, the *topological valence* of a vertex is the number of faces adjacent to the vertex.

Definition 3.14 (Singularity)**.** *Suppose $(\Sigma, \mathcal{Q})$ is a quadrilateral mesh. If the topological valence of an interior vertex is 4, then we call it a* regular vertex*, otherwise a* singularity*; if the topological valence of a boundary vertex is 2, then we call it a* regular boundary vertex*, otherwise a* boundary singularity*. The index of a singularity is defined as follows:*

$$Ind(v_i) = \begin{cases} 4 - val(v_i) & v_i \notin \partial(\Sigma, \mathcal{Q}) \\ 2 - val(v_i) & v_i \in \partial(\Sigma, \mathcal{Q}), \end{cases}$$

where $Ind(v_i)$ and $val(v_i)$ are the index and the topological valence of v_i.

The following theorems bridge quad-meshes with meromorphic quartic differentials, and the singularities with Abel-Jacobi condition. Detailed proof can be found in [29].

Theorem 3.15 (Quad-Mesh to Meromorphic Quartic Differential [29])**.** *Suppose $(\Sigma, \mathcal{Q})$ is a closed quadrilateral mesh, then*

1. *the quad-mesh $\mathcal{Q}$ induces a conformal atlas $\mathcal{A}$, such that $(\Sigma, \mathcal{A})$ form a Riemann surface, denoted as S_Q.*
2. *the quad-mesh $\mathcal{Q}$ induces a quartic differential ω_Q on S_Q. The valence-k singular vertices correspond to poles or zeros of order $k-4$. Furthermore, the trajectories of ω_Q are finite.*

Theorem 3.16 (Quartic Differential to Quad-Mesh [29])**.** *Suppose $(\Sigma, \mathcal{A})$ is a Riemann surface, ω is a meromorphic quartic differential with finite trajectories, then ω induces a quadrilateral mesh $\mathcal{Q}$, such that the poles or zeros with order k of ω corresponds to the singular vertices of $\mathcal{Q}$ with valence $k+4$.*

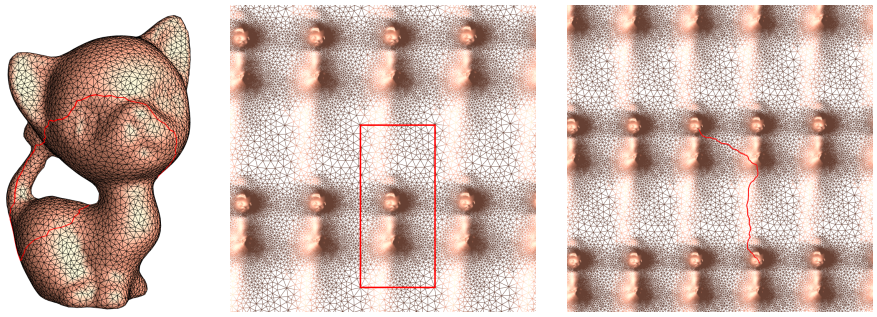


Figure 2: Left: the kitten surface Σ with a loop γ; Middle: the universal covering space $\tilde{\Sigma}$ and one fundamental domain $\bar{\Sigma}$, the red rectangle; Right: $\gamma \subset \Sigma$ is lifted to a path $\tilde{\gamma} \subset \tilde{\Sigma}$. The translations of the plane which maps a fundamental domain to another is a Deck transformation.

Theorem 3.17 (Quad-mesh singularity Abel-Jacobi condition)**.** *Suppose $\mathcal{Q}$ is a closed quadrilateral mesh, S_Q is the induced Riemann surface, ω_Q is the induced meromorphic quartic form. Assume ω_0 is an arbitrary holomorphic 1-form on S_Q, then*

$$\mu((\omega_Q) - 4(\omega_0)) = 0 \qquad \mod \Lambda \tag{4}$$

in the Jacobian $J(S_Q)$.

3.4. Deck Transformation Condition

Definition 3.18 (Covering Space)**.** *Let X be a topological space. A covering space of X is a topological space $\tilde{X}$ together with a continuous surjective map $\pi : \tilde{X} \to X$, such that for every $p \in X$, there exists an open neighborhood U of p, such that $\pi^{-1}(U)$ is a union of disjoint open sets in $\tilde{X}$, each of which is mapped homeomorphically onto U by π.*

Definition 3.19 (Universal Covering Space)**.** *Suppose X is a topological space, $(\tilde{X}, \pi)$ is called the universal covering space of X if it is a covering space of X, and it is simply connected, namely, its fundamental group $\pi_1(\tilde{X}) = \langle e \rangle$.*

Definition 3.20 (Deck Transformation)**.** *Suppose X is a topological space, $(\tilde{X}, \pi)$ is a covering space of X, an automorphism $\tau : \tilde{X} \to \tilde{X}$ is called a*

deck transformation, if $\pi \circ \tau = \pi$. *All the deck transformations form a group, which is called the deck transformation group of* $\tilde{X}$ *and denoted as* $Deck(\tilde{X})$.

Definition 3.21 (Fundamental Domain)**.** *Suppose* Σ *is a surface,* $\tilde{\Sigma}$ *is the universal covering space of* Σ. *Let* $\tilde{p} \in \tilde{\Sigma}$ *be a point, the images of* $\tilde{p}$ *under the Deck transformation group action is call an* orbit. *A* fundamental domain *of* Σ *is a simply connected subset of* $\tilde{\Sigma}$ *which contains exactly one point from each of these orbits.*

Fig. 2 illustrates these concepts, universal covering space, fundamental domain and the Deck transformation for a genus one closed surface. The left frame shows the surface Σ, the middle frame the universal covering space $\tilde{\Sigma}$, which is the plane. The red rectangle is one fundamental domain, the whole plane is tessellated to infinite many fundamental domains. Each planar translation, mapping one fundamental domain to the other is a Deck transformation. A closed loop γ on the surface is lifted to a path on the universal covering space, connecting two fundamental domains and associated with a unique deck transformation.

Given a surface Σ with a quad-mesh Q, the singularity set of Q is S, $S = \{v_0, v_1, v_2, \ldots, v_n\}$. The punctured surface $\Sigma \setminus S$ is obtained by removing S from Σ. Suppose we choose a fundamental domain $\bar{\Sigma}$ of Σ, such that the boundary $\partial\bar{\Sigma}$ doesn't go through any singularity point. Then we choose a base point p_0, and consider the generators of the fundamental group $\pi_1(\Sigma \setminus S, p_0)$,

$$\pi_1(\Sigma \setminus S, p_0) = \langle a_1, b_1, a_2, b_2, \cdots, a_g, b_g, \gamma_1, \gamma_2, \cdots, \gamma_n \rangle, \tag{5}$$

where a_i, b_i are the tunnel and handle loops corresponding to the i-th handle, the loop γ_j starts from p_0 to v_j, goes around v_j and returns to p_0. Furthermore, we require all the generators γ_j's are contained in the fundamental domain $\bar{\Sigma}$. We isometrically immerse $\bar{\Sigma}$ on the plane using $\mathbf{g}_Q$, such that v_0 is at the origin, the face edges are along horizontal and vertical directions, the mapping is denoted as $\varphi : \bar{\Sigma} \to \mathbb{C}$. All the singularities are on the integer grid, the coordinates of v_i are (m_i, n_i).

$$\varphi(v_i) = (m_i, n_i),\ m_i, n_i \in \mathbb{Z}, \quad \forall v_i \in S. \tag{6}$$

The Deck transformation group of the universal covering space $Deck(\tilde{\Sigma})$ is isomorphic to the fundamental group of the base space $\pi_1(\Sigma \setminus S, v_0)$. Suppose $v_i \in S$ with valence $\mathrm{val}(v_i)$, corresponds to the loop γ_i in $\pi(\Sigma \setminus S, v_0)$, and τ_i in the Deck transformation group of $\tilde{\Sigma}$. Since S satisfies the Abel condition, we have

$$\tau_i(z) = e^{\sqrt{-1}\frac{k_i\pi}{2}} z + (u_i + \sqrt{-1}v_i), k_i \in \mathbb{Z}, \tag{7}$$

where the valence and the rotation angle satisfy:

$$k_i + \mathrm{val}(v_i) = 0 \mod 4, \quad \forall v_i \in S \tag{8}$$

and the translation part satisfies

$$(u_i, v_i) = \begin{cases} (0,0) & k_i \equiv 0 \mod 4 \\ (m_i - n_i, m_i + n_i) & k_i \equiv 1 \mod 4 \\ 2(m_i, n_i) & k_i \equiv 2 \mod 4 \\ (m_i + n_i, n_i - m_i) & k_i \equiv 3 \mod 4. \end{cases} \tag{9}$$

Note that, for singularities whose valence are not $4k$, (u_i, v_i) and (m_i, n_i) can mutually determine each other; for singularities with valence $4k$, no matter what (m_i, n_i) are, the (u_i, v_i) are always $(0, 0)$.

Theorem 3.22 (From Quad-mesh to Deck Transformation)**.** *Suppose Σ is a closed surface with a quad-mesh $\mathcal{Q}$. The singularity set of Q is S. The quad-mesh induces a flat Riemannian metric $\mathbf{g}_Q$ with cone singularities at S. The universal covering space of the punctured surface $\Sigma \setminus S$ is $\tilde{\Sigma}$ with the pull back metric $\pi^*\mathbf{g}_Q$. Then each deck transformation $\tau \in Deck(\tilde{\Sigma})$ has the form:*

$$\tau(z) = e^{\sqrt{-1}\frac{\pi}{2}k} z + (u + \sqrt{-1}v), \quad k, u, v \in \mathbb{Z}.$$

Furthermore, if the fundamental group generators are chosen as Eqn. (5), then for each singularity $v_i \in S$, the parametric coordinates induced by the isometric immersion are integers Eqn. (6), the corresponding deck transformation is Eqn. (7), the valence and the rotation angle satisfy Eqn. (8), and the translation is given by Eqn. (9).

Proof. Suppose $\mathcal{Q}$ is a quad-mesh of Σ, then if each face is treated as a unit square, we obtain the flat metric $\mathbf{g}_Q$.

The universal covering space $\tilde{\Sigma}$ with the pull back metric $\pi^*\mathbf{g}_Q$ can be isometrically immersed in the plane, where the pre-images of the singularities are the branched points.

We fix a face $f_0 \in \mathcal{Q}$, and define a face loop,

$$\gamma := f_0, f_1, f_2, \cdots, f_{n-1}, f_n,$$

where each pair of adjecent faces f_i, f_{i+1} share a common edge, and $f_n = f_0$. Then we can lift γ to a face path $\tilde{\gamma} \subset \tilde{\Sigma}$, $\pi(\tilde{\gamma}) = \gamma$. We define an orthonormal frame on f_0, whose origin is at the center of the face, axes are parallel to the edges of f_0, and transport the frame along γ in parallel, when it reaches f_n (coinciding with f_0), the frame has a rotation angle $e^{i\frac{\pi}{2}k}$, $k \in \mathbb{Z}$. This gives the holonomy condition Eqn. 2. Furthermore, the transportation is lifted to a translation along $\tilde{\gamma}$ in the universal covering space. Since at every step, the translation from $\tilde{f}_i$ to $\tilde{f}_{i+1}$ is either horizontal or vertical unit translation, the total translation $u + iv$ is integral.

Furthermore, suppose we choose a fundamental domain $\bar{\Sigma}$ of $\Sigma \setminus S$, such that the boundary $\partial\bar{\Sigma}$ doesn't go through any singularity point. We isometrically immerse $\bar{\Sigma}$ using $\mathbf{g}_Q$. By a translation and rotation, we assume $v_0 = (0, 0)$ and the quad-edges are horizontal or vertical, the position of each singularity v_i is (m_i, n_i), $m_i, n_i \in \mathbb{Z}$, namely Eqn. (6) holds. Suppose the deck transformation corresponding to v_i is τ_i, then by direct computation, we can obtain the relation between the valence of v_i and the rotation angle of τ_i as Eqn. (8), and the translation of τ_i as Eqn. (9). Since all (m_i, n_i)'s are integers, the corresponding (u_i, v_i)'s are also integers.

Consider each handle, we choose a face path homotopic to a_i, and flatten the face path using $\mathbf{g}_Q$, then we get the deck transformation, the translation component must be integer. This holds for deck transformation corresponding to b_i as well. Since all the generators of the deck transformation group satisfy Eqn. 7, all the deck transformations have integer translation components. □

The inverse is also true.

Theorem 3.23 (From Deck Transformation to Quad-Mesh)**.** *Suppose Σ is a closed surface, a flat metric $\mathbf{g}$ with cone singularities S is defined on Σ. The universal covering space of the punctured surface $\Sigma \setminus S$ is $\tilde{\Sigma}$ with the pull back metric $\pi^*\mathbf{g}$. Suppose each $\tau \in Deck(\tilde{\Sigma})$ has the form:*

$$\tau(z) = e^{\sqrt{-1}\frac{\pi}{2}k} z + (u + \sqrt{-1}v), \quad k, u, v \in \mathbb{Z}.$$

Furthermore, suppose a fundamental domain $\bar{\Sigma}$ of $\Sigma \setminus S$ is chosen, $\partial\bar{\Sigma}$ doesn't go through any singularity, $v_0 \in S$ is selected, a set of generators of the fundamental group $\pi_1(\Sigma \setminus S, v_0)$ are chosen as Eqn. (5). If for each singularity $v_i \in S$, the parametric positions are integers as Eqn. (6), the corresponding deck transformation τ_i has the form as Eqn. (7), the valence and the rotation angle have the relation Eqn. (8), the translation is given by Eqn. (9), then there is a quad-mesh $\mathcal{Q}$ defined on Σ with the singularity set S, which induces the metric $\mathbf{g}_Q = \mathbf{g}$.

Proof. The universal covering space $(\tilde{\Sigma}, \pi^*\mathbf{g})$ can be isometrically immersed on the plane. We consider the image of one fundamental domain $\bar{\Sigma}$, by a translation, Eqn. (6) ensures the parametric positions of all singularities are on the integer lattice,

$$\Lambda := \{(m, n) \,|m, n \in \mathbb{Z}\}.$$

Since each deck transformations $\tau \in \mathrm{Deck}(\tilde{\Sigma})$ has the form Eqn. 7, the parametric positions of all the pre-images of singularities $\pi^{-1}(S)$ are on the lattice Λ. Furthermore, all the deck transformations are isometric, and preserve the integer lattice. Therefore, the projection of the lattice $\pi(\Lambda)$ defines a quad-mesh $\mathcal{Q}$ of the original surface Σ. □

Again, note that the position of singularities Eqn. (6) and the integer translation of deck transformations Eqn. 9 are not fully independent, except those for $4k$-valence singularities.

3.5. Quad-Meshing

This subsection explains how to deform a T-mesh parameterization into a quadrilateral mesh parameterization. We adjust the edge lengths of the T-mesh to satisfy a set of constraints. According to Theorem 3.23, these constraints are sufficient to guarantee a quadrilateral mesh. This reduces to solve a square linear system. Teichmüller theory is required to furter determine the rank of the linear system.

Suppose Σ is a closed surface with genus g, D is a divisor satisfying the Abel-Jacobi condition, ω is the meromorphic quartic differential, $\mathcal{T}$ is the motorcycle graph, (or equivalently the T-Mesh), induced by ω.

Lemma 3.24. *Suppose Σ is a genus g closed oriented surface, S is the singularity set, for each $v_i \in S$, its valence is k_i. Suppose the motorcycle graph $\mathcal{T} = (V, E, F)$, where V, E, F are vertex, edge and face set respectively, then*

1. *The total valence of singularities*

$$\sum_{v_i \in S} k_i = 4|S| - 8 + 8g, \tag{10}$$

2. *The number of edges*

$$|E| = 2 \sum_{v_i \in S} k_i, \tag{11}$$

3. *the number of vertices*

$$|V| = |S| + \sum_{v_i \in S} k_i, \tag{12}$$

4. *the number of faces*

$$|F| = (2 - 2g) + \sum_{v_i \in S} (k_i - 1). \tag{13}$$

Proof. 1.For Eqn (10), for each singularity $v_i \in S$, the curvature is $(4 - k_i)\frac{\pi}{2}$, according to Gauss-Bonnet,

$$\sum_{v_i \in S} (4 - k_i)\frac{\pi}{2} = 2\pi(2 - 2g),$$

hence

$$\sum_{v_i \in S} k_i = 4|S| - 8 + 8g.$$

2.For Eqn. (11), every trajectory T_i emitted from v_i will terminate when it intersects another trajectory T_j emitted from v_j orthogonally. The termination point divides a segment in T_j into two sub-segments. Therefore, initially there are $\sum_{v_i \in S} k_i$ trajectories, and they will produce the same number of termination points. Hence the total number of edges equals to $2\sum_{v_i \in S} k_i$.

3.For Eqn. (12), there are two types of vertices, the singularities and the termination points. The former has $|S|$ vertices, the later has $\sum_{v_i \in S} k_i$ vertices.

4.For Eqn. (13), according to the Euler formula

$$|V| + |F| - |E| = 2 - 2g,$$

hence

$$|F| = |E| - |V| + 2 - 2g = \sum_{v_i \in S} k_i - |S| + (2 - 2g).$$

□

For each edge $e \in E$, its length under the flat metric is denoted as d_e. We can change the length by amount x_e. Therefore, all the unknowns are in the set $\{x_e : e \in E\}$.

Singularity Position Constraint Suppose the singularities are $\{v_0, v_1, \ldots, v_n\}$. We compute a fundamental domain $\bar{\Sigma}$, the boundary $\partial\bar{\Sigma}$ doesn't go through any singularity, and construct paths $\Gamma_i \subset \bar{\Sigma}$, $i = 1, 2, \ldots, n$, each Γ_i is contained in $\bar{\Sigma}$ and connects v_0 with v_i. By isometrically embedding the faces along Γ_i, we can obtain the parametric position of v_i, denoted as $\varphi(v_i)$. We require $\varphi(v_i) \in \mathbb{Q}^2$. Each path consists of edges of $\mathcal{T}$, $\varphi(v_i)$ can be represented by the edge lengths d_e and x_e as follows: suppose Γ_i consists of edges, horizontal edges with positive (negative) orientations are H_i^+ (H_i^-), vertical edges with positive (negative) orientations are V_i^+ (V_i^-), then the parametric position of v_i is given by

$$\varphi(v_i) = \left(\sum_{e \in H_i^+} (d_e + x_e) - \sum_{e \in H_i^-} (d_e + x_e), \sum_{e \in V_i^+} (d_e + x_e) - \sum_{e \in V_i^-} (d_e + x_e) \right) \in \mathbb{Q}^2. \tag{14}$$

Hence there are $|S|-1$ constraints for x_e's.

Deck Transformation Constraint The Deck transformation group has $2g+|S|-1$ generators, corresponding to the fundamental group generators in Eqn. (5), each generator corresponds to a constraint. The singularity position constraints Eqn.(14) imply the deck transformation constraints of γ_i's in the generators of $\pi_1(\Sigma \setminus S, v_0)$. (For valence $4k$ singularities, with k a positive integer, the deck transformation conditions are automatically satisfied, independent of the edge lengths.)

Suppose the canonical handle and tunnel loops of Σ are $\{a_1, b_1, a_2, b_2, \cdots, a_g, b_g\}$, each of them is a loop with starting point v_0. We can find a homotopically equivalent curve comprised of edges in E, then flatten the faces along the loop one by one. When we return to the starting point, we can flatten the initial face again. The rigid transformation between two embedding of the initial face gives us the deck transformation. Suppose the deck transformation τ_i corresponds to a_i. The rotation part of τ_i is already $e^{i\frac{\pi}{2}k_i}$, we add the constraints to the translation part, such that both x and y components are rational numbers. τ_i can be represented by the edge lengths d_e and x_e as follows: suppose a_i consists of edges, horizontal edges with positive (negative) orientations are H_i^+ (H_i^-), vertical edges with positive (negative) orientations are V_i^+ (V_i^-), then the translation component of τ_i is given by

$$\left(\sum_{e\in H_i^+}(d_e+x_e)-\sum_{e\in H_i^-}(d_e+x_e),\sum_{e\in V_i^+}(d_e+x_e)-\sum_{e\in V_i^-}(d_e+x_e)\right)\in\mathbb{Q}^2. \tag{15}$$

Hence there are $2g$ constraints for x_e's.

Face Constraints Each face has two constraints, the total length of one side is equal to that of the opposite side. The constraint can be represented by the edge lengths d_e and x_e as follows, suppose f_i has four sides s_0, s_1, s_2, s_3 counter-clockwisely, then

$$\left(\sum_{e\in s_0}(d_e+x_e)-\sum_{e\in s_2}(d_e+x_e),\sum_{e\in s_1}(d_e+x_e)-\sum_{e\in s_3}(d_e+x_e)\right)=(0,0). \tag{16}$$

Note that, the equation for the last face linearly depends on those of other faces.

Hence there are $2(|F|-1) = 2(\sum_{v_i \in S} k_i - |S| + 1 - 2g)$ face constraints for x_e's. Furthermore, the face constraints in Eqn. (16) may not be linearly independent. For example, the face constraints for an cube with 8 corner as singularities are linearly dependent.

Linear System Here we discuss the linear system of the constraints of the singularity position, the deck transformation and the face sides. We show that the linear system is a square matrix.

Lemma 3.25. *Given a closed surface Σ of genus g with singularity set S, and a T-mesh $\mathcal{T}$, the linear system Eqn. (14), Eqn. (15) and Eqn. (16) forms a $|E| \times |E|$ square matrix.*

Proof. The number of singularity position constraints Eqn. (14) is $2(|S| - 1)$, the number of deck transformation constraints corresponding to the handle and tunnel loops is $4g$, the number of face constraints is $2(|F| - 1)$. So the total number of constraints is

$$2(|S| - 1) + 4g + 2(|F| - 1) = 2(|S| - 1) + 4g - 2 + 2|F|.$$

By Eqn. (13), $|F| = (2 - 2g) + \sum_{v_i \in S} k_i - |S|$, the above equals to

$$2|S| - 4 + 4g + 2|F| = 2|S| - 4 + 4g + (4 - 4g) + 2\sum_{v_i \in S} k_i - 2|S| = 2\sum_{v_i \in S} k_i.$$

By Eqn. (11), $|E| = 2\sum_{v_i \in S} k_i$, hence the total number of constraints equals to $|E|$. On the other hand, we know the number of unknowns equals to $|E|$. Hence the linear system is a square matrix. □

This shows the number of unknowns $\{x_e\}$ is more than the number of constraints. Note that, the initial edge lengths of the T-mesh are a solution to linear system with different constraints on the right hand side. The number of constraints is less than that of the method in [7].

The initial metric is a solution to the linear system, therefore this step deforms the conformal structure of the original surface (and the T-mesh) such that the meromorphic quartic differential (equivalently the divisor) satisfies all

the constraints. The thorough analysis involves Teichmüller theory, and will be investigated in our future work.

The proposed algorithm solves the linear system Eqn. (14), Eqn. (15) and Eqn. (16). The unknowns x_e's are real numbers, and won't be quantized. The only quantization is for the right hand sides of equations. The initial deck transformations induced by the meromorphic quartic differential give the initial values for the right hand sides, then we quantize them to $\frac{m_i}{n_i}$. The solutions will be multiplied by the least common multiple. n_i's are chosen to balance the distortion and the number of quad-faces.

The Algorithm 1 doesn't require mesh subdivision. After Algorithm 1 is completed, the Ricci flow step requires the modification of the triangulation at the new extraordinary points. If a new extraordinary point is in the interior of a triangle face, we subdivide the face into 3 smaller triangles. Because the dynamic Ricci flow alogirthm can handle triangle meshes with bad qualities, this simple subdivision is good enough for the purpose.

Surfaces with Boundaries All the theorems, lemmas discussed in this section focus on closed oriented surfaces, they can be directly generalized to compact oriented surface with boundaries using the "doubling" operator as follows: suppose Σ is a compact oriented surface with boundaries, we make a copy of Σ, denoted as Σ', then we reverse the orientation of Σ', isometrically glue Σ and Σ' along their corresponding boundaries. The obtained surface is denoted as $\bar{\Sigma}$, which is a closed oriented surface with a special symmetry. All the theoretic results hold on $\bar{\Sigma}$ and can be translated to Σ. In general, the boundaries are along the edges of the quad-mesh.

For non-oriented surface Σ, we can cover it into an oriented surface by "double covering": suppose Σ is triangulated, for each face $f \in \Sigma$, we construct two faces f^+ and f^- with opposite orientations (i.e. different ordering of vertices), then we glue the oriented faces $f_1^{\pm}$ and $f_2^{\pm}$ along a common edge, if the non-oriented faces f_1 and f_2 are adjacent in Σ, and the common edges are with opposite orientations in $f_1^{\pm}$ and $f_2^{\pm}$ respectively. The obtained double covering mesh is oriented.

4. T-Mesh and Quad-Mesh Generation Algorithms

The algorithmic pipeline can be divided into two stages, the first stage computes a meromorphic differential ω, its critical graph induces a T-mesh (motorcycle graph); the second stage further deforms the T-mesh to a quad-mesh.

4.1. T-Mesh Generation

This subsection explains the algorithm for T-mesh generation in detail. The input surface is represented as a triangle mesh Σ; the output is a meromorphic quartic differential ω, and the flat metric with cone singularities at the poles and zeros induced by ω. The pipeline of the algorithm is as follows:

1. Compute the *homology* group generators of Σ, $\{a_1, \cdots, a_g;\ b_1, \cdots, b_g\}$;
2. Compute the dual *holomorphic* 1-form basis $\{\varphi_1, \cdots, \varphi_g\}$; Construct a *holomorphic* differential φ on the Riemann surface Σ through a linear combination of basis $\{\varphi_k\}_{k=1}^{g}$, locate the zeros of φ;
3. Compute the period matrix (A, B) of surface and construct the lattice Λ, Jacobi variety $J(\Sigma)$;
4. Define the initial divisor D as the extremal points of the Gaussian curvature function, extremes with positive curvature are poles, extremes with negative curvature are zeros;
5. Compute the Abel-Jacobi map of the initial divisor D in the Jacobi variety $J(\Sigma)$;
6. Optimize the divisor D to satisfy the Abel-Jacobi condition;
7. Compute the flat metric with cone singularities at the divisor D by surface *Ricci Flow*;
8. Compute a cut-graph not going through any singularity, connect each singularity to the cut graph by the shortest path, slice the surface along the cut graph and the shortest paths to obtain a fundamental domain $\bar{\Sigma}$.
9. Isometrically immerse $\bar{\Sigma}$ into the complex plane. The immersion pulls $(dz)^4$ back to the surface and produces a meromorphic quartic differential ω.

10. Trace the critical horizontal and vertical trajectories of ω, namely isoparametric curves through singularities, to generate a motorcycle-graph, which gives the T-mesh.

In the following, we explain every single step in details. Each subsection corresponds to one step.

4.1.1. Homology group basis

In practice, we compute a special canonical homology group basis, the tunnel loops $\{a_i\}$ and handle loops $\{b_i\}$, such that each a_i and b_i intersect each other at one point. The homology group basis will determine the holomorphic 1-form basis, the cut graph. For the simplicity of presentation and the ease for debugging, we compute a canonical homology group basis. Our algorithm is mainly based on the work of Dey et al.[12], which avoids tetrahedral tessellation and modification of the original triangle mesh. The algorithm utilizes the concept of a *Reeb graph* and the linking number to produce different sets of *homology* basis.

As shown in Fig. 3 left frame, the algorithm may generate a homology basis which doesn't satisfy the intersection condition,

$$a_i \cdot b_j = \delta_{ij}, \;\; a_i \cdot a_j = 0, \;\; b_i \cdot b_j = 0, \quad i, j = 1, \ldots, g, \tag{17}$$

where $\alpha \cdot \beta$ represents the algebraic intersection number between α and β, g is the genus of the mesh. We compute the algebraic intersection between the tunnel loops and handle loops; if the intersection condition 17 is violated, we randomly reset the height function used for constructing a Reeb graph, and obtain a new set of handle loops and tunnel loops. After several iterations, we can get a canonical homology group basis, as shown in Fig. 3 right frame.

4.1.2. Holomorphic 1-form Basis

The algorithm of computing *holomorphic* 1-form is based of the work of Gu et al. [18], which is based on Hodge theory. For a genus g closed Riemann surface, the space of holomorphic one-forms has g complex dimensions, or equivalently

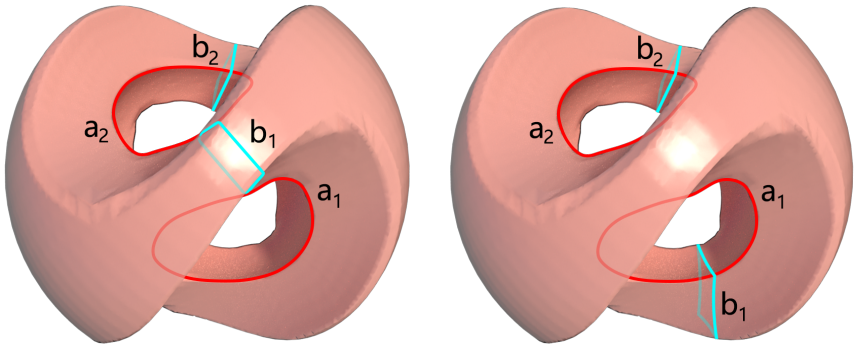


Figure 3: **Sculpt** model. **Left**: the handle loop b_1 intersects both tunnel loops a_1 and a_2, where we call the handle loop b_1 illegal. **Right**: each handle loop b_i only intersects its conjugate a_i once.

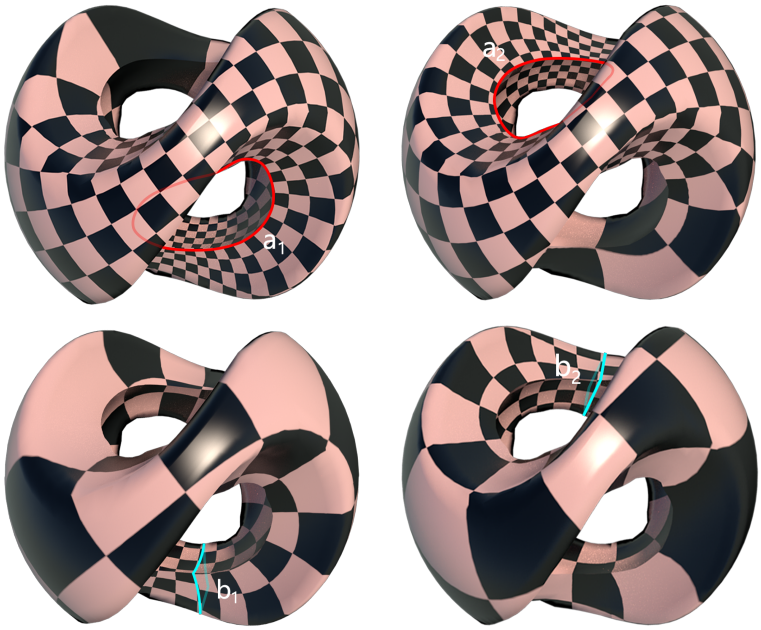


Figure 4: A *holomorphic* 1-form basis on a genus two surface, the **Sculpt** model.

$2g$ real dimensions.

Step 1. For each loop γ, we slice the mesh Σ along γ to get an open mesh $\bar{\Sigma_\gamma}$

with boundaries $\partial\Sigma_\gamma = \gamma^+ - \gamma^-$, then we construct a function

$$g_\gamma(v_i) = \begin{cases} 1 & v_i \in \gamma^+ \\ 0 & v_i \in \gamma^- \\ \text{random} & \text{otherwise.} \end{cases}$$

Then the discrete 1-form $\lambda_\gamma = dg_\gamma$ is a closed 1-form. In this way, we construct a cohomology group basis $\lambda_{a_1}, \lambda_{b_1}, \cdots, \lambda_{a_g}, \lambda_{b_g}$.

Step 2. For each closed 1-form λ, we construct a function $f : \Sigma \to \mathbb{R}$, such that $\lambda + df$ is harmonic, namely the function f satisfies the Poisson equation $\Delta f = -\delta\lambda$. In this way, we diffuse the cohomology basis to a harmonic 1-form group basis, denoted as $\omega_{a_1}, \omega_{b_1}, \cdots, \omega_{a_g}, \omega_{b_g}$.

Step 3. Each harmonic 1-form ω is equivalent to a curl-free vector field on Σ, we rotate the vector field by $\frac{\pi}{2}$ about the normal to the surface to obtain a divergence free vector field, which is equivalent to another harmonic 1-form ${}^*\omega$. The pair $\omega + \sqrt{-1}{}^*\omega$ is a holomorhic 1-form. In this way, we construct a holomorphic 1-form basis $\{\varphi_{a_1}, \varphi_{b_1}, \ldots, \varphi_{a_g}, \varphi_{b_g}\}$, where

$$\varphi_\gamma = \omega_\gamma + \sqrt{-1}{}^*\omega_\gamma, \gamma \in \{a_1, \ldots, a_g, b_1, \ldots, b_g\}.$$

According to the Riemann-Roch theory, the above set of *holomorphic* 1-forms span the linear space of all holomorphic 1-forms $\Omega^1(\Sigma)$, namely for any $\varphi \in \Omega^1(\Sigma)$,

$$\varphi = \sum_{k=1}^{g} \alpha_k \varphi_{a_k} + \sum_{l=1}^{g} \beta_l \varphi_{b_l}, \tag{18}$$

where the α_k, β_l are real linear combination coefficients.

In practice, in order to compute the zeros of φ more accurately, we choose the linear combination coefficients, such that the conformal factor function of φ is as uniform as possible. (By integrating φ, we can obtain a conformal parameterization of the surface. The ratio between the surface area element and the parametric area element is defined as the conformal factor, which measures the area distortion cased by the parameterization.) In our implementation, we assign all α_k's and β_l's to be 1. Heuristically, the resulting holomorphic 1-form meets our accuracy requirement.

4.1.3. Period matrix and Lattice

We can further construct a set of holomorphic 1-form basis $\{\varphi_1, \varphi_2, \ldots, \varphi_g\}$, such that

$$\int_{a_i} \varphi_j = \delta_{ij}, i, j = 1, 2, \ldots, g. \tag{19}$$

This can be accomplished as follows: we represent each φ_i as a linear combination of the basis $\{\varphi_{a_k}, \varphi_{b_l}\}$ as in Eqn. (18), then build a linear system by the constraints in Eqn. (19) and solve the linear combination coefficients $\{\alpha_k, \beta_l\}$. Then for each γ in the homology basis, we construct a g dimensional vector $\lambda_\gamma \in \mathbb{C}^g$,

$$\lambda_\gamma = \left(\int_\gamma \varphi_1, \int_\gamma \varphi_2, \cdots, \int_\gamma \varphi_g \right).$$

The period matrix (A, B) can be constructed as

$$A = \begin{bmatrix} \int_{a_1} \varphi_1 & \int_{a_2} \varphi_1 & \cdots & \int_{a_g} \varphi_1 \\ \int_{a_1} \varphi_2 & \int_{a_2} \varphi_2 & \cdots & \int_{a_g} \varphi_2 \\ \vdots & \vdots & \ddots & \vdots \\ \int_{a_1} \varphi_g & \int_{a_2} \varphi_g & \cdots & \int_{a_g} \varphi_g \end{bmatrix} \quad B = \begin{bmatrix} \int_{b_1} \varphi_1 & \int_{b_2} \varphi_1 & \cdots & \int_{b_g} \varphi_1 \\ \int_{b_1} \varphi_2 & \int_{b_2} \varphi_2 & \cdots & \int_{b_g} \varphi_2 \\ \vdots & \vdots & \ddots & \vdots \\ \int_{b_1} \varphi_g & \int_{b_2} \varphi_g & \cdots & \int_{b_g} \varphi_g \end{bmatrix} \tag{20}$$

by our construction, A is the $g \times g$ identity matrix, the imaginary part of B is positive definite [19]. Then we construct a lattice Λ in $\mathbb{C}^g$,

$$\Lambda = \left\{ \sum_{k=1}^{g} (s_k \lambda_{a_k} + t_k \lambda_{b_k}), \quad s_k, t_k \in \mathbb{Z} \right\}.$$

4.1.4. Abel-Jacobi Map

Given a canonical homology group basis, we slice the surface along the basis to obtain a topological disk $\bar{\Sigma}$. Fix a base point in the interior of $\bar{\Sigma}$, $p_0 \in \bar{\Sigma}$, for any point $p \in M$, we can choose arbitrarily a path $\gamma \subset \bar{\Sigma}$ connecting p and p_0, the Abel-Jacobi map $\mu : M \to J(M)$, $J(M) = \mathbb{C}^g/\Lambda$, is defined as

$$\mu(p) = \Phi(p) \mod \Lambda,$$

where

$$\Phi(p) = \left(\int_\gamma \varphi_1, \int_\gamma \varphi_2, \cdots, \int_\gamma \varphi_g \right)^T. \tag{21}$$

Similarly, given a divisor $D = \sum_{i=1}^{n} n_i p_i$, choose paths $\gamma_i \subset \bar{\Sigma}$ from p_0 to p_i,

$$\mu(D) = \sum_{i=1}^{n} n_i \mu(p_i) = \sum_{i=1}^{n} n_i \left(\int_{\gamma_i} \varphi_1, \int_{\gamma_i} \varphi_2, \cdots, \int_{\gamma_i} \varphi_g \right)^T \mod \Lambda.$$

Abel-Jacobi condition claims that if D is a principle divisor, then $\mu(D)$ is 0, namely

$$\Phi(D) - A \begin{bmatrix} s_1 \\ \vdots \\ s_g \end{bmatrix} - B \begin{bmatrix} t_1 \\ \vdots \\ t_g \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \quad s_i, t_j \in \mathbb{Z}. \tag{22}$$

By expansion, we obtain the equation

$$\begin{bmatrix} \sum_{i=1}^{n} n_i \int_{\gamma_i} \varphi_1 \\ \sum_{i=1}^{n} n_i \int_{\gamma_i} \varphi_2 \\ \vdots \\ \sum_{i=1}^{n} n_i \int_{\gamma_i} \varphi_g \end{bmatrix} - \begin{bmatrix} \sum_{k=1}^{g} (s_k \int_{a_k} \varphi_1 + t_k \int_{b_k} \varphi_1) \\ \sum_{k=1}^{g} (s_k \int_{a_k} \varphi_2 + t_k \int_{b_k} \varphi_2) \\ \vdots \\ \sum_{k=1}^{g} (s_k \int_{a_k} \varphi_g + t_k \int_{b_k} \varphi_g) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{23}$$

where s_k, t_k are integers. The base point p_0 can be arbitrarily chosen, and the same condition holds. This condition is stronger than the Gauss-Bonnet theorem.

4.1.5. Abel-Jacobi Condition Optimization

Initially, the singularities are located at the extremal points of Gaussian curvature. For extremal points with positive curvatures, we set the valence to be 3, those with negative curvature to be 5. Suppose the initial divisor D_0 doesn't satisfy the Gauss-Bonnet condition, namely $\deg(D_0) \neq 8g - 8$, we can add extra poles or zeros to obtain $D = \sum_{i=1}^{n} n_i p_i$, such that $\deg(D) = 8g - 8$. We also choose a holomorphic 1-form φ.

Our goal is to adjust the divisor D such that $\mu(D - 4(\varphi))$ is zero, where μ is the Abel-Jacobi map. We define the energy to be the squared norm, namely $\|\mu(D - 4(\varphi))\|^2$. By definition, $\mu(D - 4(\varphi))$ equals to $\Phi(D - 4(\varphi))$ modulus the lattice Λ. Each grid-point of the lattice Λ is labeled by g pairs of integers $\{(s_k, t_k), k = 1, \ldots, g\}$. Hence our optimization includes two steps: the first step

is to determine the cell inside the lattice containing $\Phi(D-4(\varphi))$. We can find the grid point $\{(s_k,t_k),k=1,\ldots,g\}$ closest to $\Phi(D-4(\varphi))$, then a cell adjacent to the closest grid-point contains $\Phi(D-4(\varphi))$. By definition, it is obvious that such kind of cell exists, but may not be unique. The second step is to further optimize the squared norm of $\mu(D-4(\varphi))$ within the fixed cell with respect to the positions of all singularities, counting their multiplicities. The singularities can be merged or split. Since there are meromorphic quartic differentials, the solution for the second step also exists. During the optimization, $\Phi(D-4(\varphi))$ may transit to adjacent cells. The algorithm dynamically trace the current cell containing it. In Algorithm 1, the current cell is dynamically determined at the step 23.

First, we determine the integer coefficients s_k and t_k, $k=1,2,\ldots,g$ in the Abel-Jacobi condition (23) by minimizing the squared norm

$$\min_{s_k,t_k\in\mathbb{Z}}\left\|\Phi(D)-\sum_{k=1}^{g}s_k\lambda_{a_k}-\sum_{k=1}^{g}t_k\lambda_{b_k}-\Phi(4(\varphi))\right\|^2. \tag{24}$$

This can be accomplished by standard integer programming method [2]. When the norms of (s_k,t_k)'s go to infinity, the energy goes to infinity. Hence we can search the solution in a compact set $\Omega\subset\mathbb{R}^{2g}$, where the number of integer grid points $\{(s_k,t_k)\}_{k=1}^g$ is finite, hence the minimizer $\{(s_k,t_k)\}_{k=1}^g$ always exists. In practice, we can simply use enumeration of integer grid points within the compact set Ω to find the optimum.

Second, once the integer coefficients $\{(s_k,t_k)\}_{k=1}^g$ are determined, we minimize the squared norm of $\mu(D)$ with respect to the positions of the poles and the zeros,

$$\min_{p_1,\ldots p_n\in\Sigma}\left\|\Phi(D)-\sum_{k=1}^{g}s_k\lambda_{a_k}-\sum_{k=1}^{g}t_k\lambda_{b_k}-\Phi(4(\varphi))\right\|^2.$$

The positions of the singularities are the variables, they may be the vertices of the mesh or the interior points of triangle faces. If a singularity p is with multiplicity n_p, then these n_p points are treated as n_p different variables and moved independently. Therefore the singularities can be split during the optimization.

They can also be merged together, even cancel out each other. Since there are meromorphic quartic differentials, the solution always exists, but may be degenerated. For example, consider a closed torus with one valence 3 singularity at q and one valence 5 singularity at p. Then the initial divisor is $D_0 = p - q$, the optimization process will merge p and q then return the final divisor $D = 0$.

Let $d \in \mathbb{C}^g$ be the constant complex vector

$$d = \sum_{k=1}^{g} s_k \lambda_{a_k} - \sum_{k=1}^{g} t_k \lambda_{b_k} - \Phi(4(\varphi)),$$

then the above energy becomes

$$E(p_1, \ldots, p_n) := \sum_{j=1}^{g} \left\| \sum_{i=1}^{n} n_i \int_{p_0}^{p_i} \varphi_j - d_j \right\|^2.$$

For each point p_i, we choose a local neighborhood Δ_i of p_i, with local parameter z_i, then the holomorphic 1-form φ_j has the local representation,

$$\varphi_j = h_j^i(z_i) dz_i,$$

where $h_j^i(z_i)$ is a holomorphic function defined on Δ_i. By direct computation, we obtain the gradient of the energy,

$$\frac{\partial E}{\partial p_i} = \sum_{j=1}^{g} \left[n_i h_j^i(p_i) \left(\sum_{i=1}^{n} n_i \int_{p_0}^{p_i} \bar{\varphi}_j - \bar{d}_j \right) + n_i \bar{h}_j^i(p_i) \left(\sum_{i=1}^{n} n_i \int_{p_0}^{p_i} \varphi_j - d_j \right) \right], \tag{25}$$

we can use gradient descent method to minimize the energy $\|\mu(D)\|^2$. In practice, we choose the triangle face containing p_i as Δ_i. We isometrically embed Δ_i onto the plane, and the planar coordinates give local parameter z_i. φ_j can be represented as a complex linear function on Δ_i, which is $h_j^i(z_i)$. We update the position of p_i along the negative direction of the gradient, until we reach the boundary of the current face Δ_i. Then we travel to the adjacent face Δ_i' along the negative gradient direction, then compute the local parameter z_i' and the local representation of φ_j. By repeating this procedure, we can minimize the squared norm of $\mu(D)$. The algorithm is presented briefly in Alg. 1.

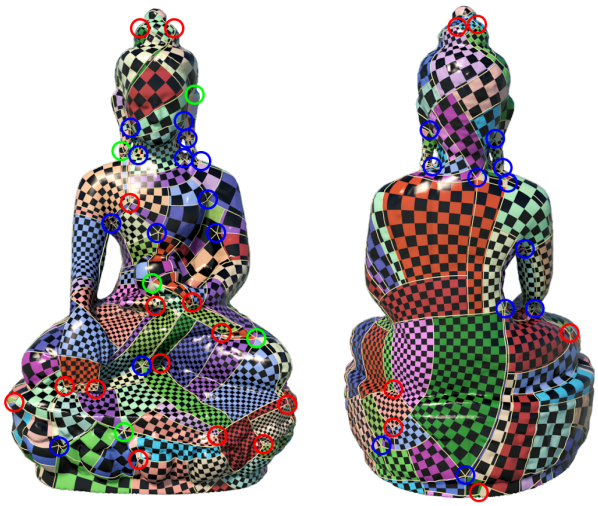


Figure 5: The singularities of the Buddha surface. The valences of the singularities are 3 (red), 5(blue) and 6 (green) respectively.

Fig. 5, Fig. 6 and Fig. 7 show an example computed using Alg. 1. The input Buddha surface is of genus 3 with complicated geometry, the number of vertices, edges and faces are $59.4k$, $178.1k$ and $118.7k$ respectively. The initial singularities are selected at the extreme points of the Gaussian curvature, then the algorithm modifies the initial divisor to satisfy the Gauss-Bonnet condition. Then, we compute the Abel-Jacobi map of the divisor, $\Phi(D)$. By minimizing the energy in Eqn. (24), we obtain the lattice point (s_k, t_k) for $k = 1, 2, 3$, and get $\mu(D)$ by quotient the lattice. Then for each singularity $p_i \in D$, we find the triangle Δ_i containing p_i, for each base holomorphic 1-form φ_j, we get the local linear representation h^i_j. We compute the gradient of the square norm with respect to p_i using Eqn. (25), and move p_i inside Δ_i along the negative gradient direction. If p_i exceeds Δ_i and transits to a neighboring face, we update the local representation of φ_j, namely h^i_j and repeat. In this way, the energy is minimized using the gradient descent method and eventually the norm of $\mu(D)$ is less than a threshold, the divisor is close to a canonical divisor and satisfies the Abel-Jacobi condition. There are 34 valence 3 singularities, 36 valence 5 singularities and 7 valence 6 singularities on the Buddha surface as shown in Fig . 5. The

divisor D uniquely determines a meromorphic quartic differential ω, the critical trajectories of ω induce the motorcycle graph as shown in Fig. 6, and the flat metric of ω induces an immersion of the surface as shown in Fig. 7, namely a seamless parameterization. We then visualize the seamless parameterization by checkerboard texture mapping. By examining these figures, we can see the right corner angles of checkers are well preserved, hence the parameterization has high conformality. By examining Fig. 7, we can see all the faces of the motorcycle graph are mapped to planar rectangles, this shows the holonomy conditions are satisfied. The threshold for this experiment is $3.0e-4$, the iterations in the optimization are 670539 and the running time is 2.021 seconds. This shows the efficiency of this algorithm.

4.1.6. Discrete Surface Ricci Flow

Suppose we have obtained the divisor $D = \sum_{i=1}^{n} n_i p_i$, if there is a non-vertex point $p_i \in D$, then we add p_i as a vertex and modify the mesh structure accordingly. After that we set the target curvature for all vertices,

$$\bar{K}(v_i) = \begin{cases} (4-n_i)\frac{\pi}{2} & v_i \in D \\ 0 & v_i \notin D \end{cases}$$

(Here we assume the surface is closed. For surfaces with boundaries, we use double covering technique to convert them to be closed ones.) For each vertex $v_i \in M$, we set the initial conformal factor as $u_i = 0$. Then the edge length is given by vertex scaling, for edge $e_{ij} = [v_i, v_j]$, its length is given by

$$l_{ij} = e^{u_i} \beta_{ij} e^{u_j},$$

where β_{ij} is the initial edge length, usually induced by the Euclidean metric of $\mathbb{R}^3$. The corner angles are calculated using Euclidean cosine law,

$$\theta_k^{ij} = \cos^{-1} \frac{l_{ik}^2 + l_{jk}^2 - l_{ij}^2}{2 l_{ik} l_{jk}},$$

the discrete Gaussian curvature is given by

$$K(v_i) = \begin{cases} 2\pi - \sum_{jk} \theta_i^{jk} & v_i \notin \partial M \\ \pi - \sum_{jk} \theta_i^{jk} & v_i \in \partial M. \end{cases}$$

The discrete Ricci energy is defined as

$$E(u_1, \ldots, u_n) = \int^{(u_1,\ldots,u_n)} \sum_{i=1}^{n} (\bar{K}_i - K_i) du_i.$$

The gradient of the energy is given by

$$\nabla E = (\bar{K}_1 - K_1, \bar{K}_2 - K_2, \cdots, \bar{K}_n - K_n)^T.$$

The Hessian matrix is given by the cotangent edge weight [20]

$$\frac{\partial^2 E}{\partial u_i \partial u_j} = \begin{cases} (\cot \theta_k^{ij} + \cot \theta_l^{ji})/2 & e_{ij} \notin \partial M \\ \cot \theta_k^{ij}/2 & e_{ij} \in \partial M. \end{cases}$$

and

$$\frac{\partial^2 E}{\partial u_i^2} = -\sum_{i \sim j} \frac{\partial^2 E}{\partial u_i \partial u_j}.$$

We can use Newton's method to optimize the Ricci energy, during the optimization, we update the triangulation to be Delaunay all the time. The work [20] proves the convergence and the finiteness of the edge flips. We can compute holonomy using the resulting Riemannian metric.

4.1.7. Isometric Immersion and Meromorphic Quartic Differential

We have obtained a canonical homology group basis $\{a_1, \ldots, a_g, b_1, \ldots, b_g\}$. The union of the basis forms a cut graph Γ of the mesh. For each pole or zero p_i in D, we find a shortest path γ_i connecting p_i to the cut graph under the original metric, furthermore, all such shortest paths γ_i's are disjoint. Then we slice Σ along the cut graph and the shortest paths, $\Gamma \bigcup \{\bigcup_{i=1}^{k} \gamma_i\}$, to obtain a fundamental domain $\tilde{\Sigma}$.

Then we flatten $\tilde{\Sigma}$ face by face using the metric obtained by the discrete surface Ricci flow. This produces an immersion $\varphi : \tilde{\Sigma} \to \mathbb{C}$. On the complex plane, there is a canonical differential dz^4, the pull back differential $\varphi^* dz^4$ is a meromorphic quartic differential ω well defined on Σ. We can use φ as a parameterization, and use checker board texture mapping to visualize the quartic differential ω.

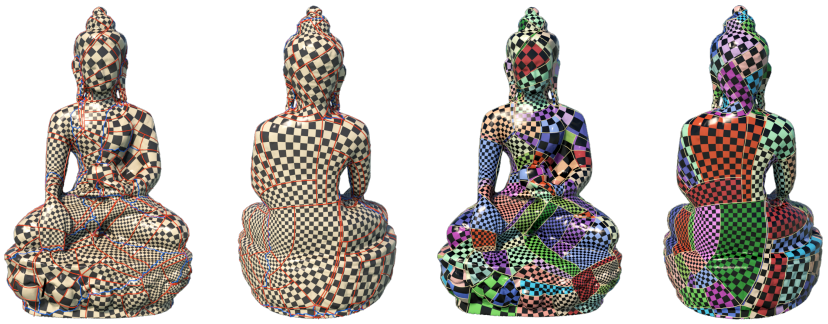


Figure 6: In the left two frames, the red curves form the motorcycle graph, the blue curves are the original cut graph. The right two frames show the T-Mesh of the Buddha surface.

4.1.8. T-Mesh Generation

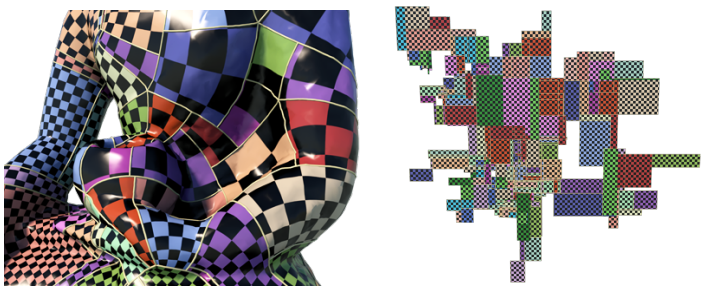
We trace the critical trajectories of the meromorphic quartic differential $\varphi^* dz^4$. Suppose the valence of v_i is k_i, then k_i trajectories are traced. Each trajectory is denoted as $\gamma_k(s_k)$, where s_k is the arc length parameter of γ_k, its image $\varphi(\gamma_k)$ is the horizontal and vertical lines through the singularity $\varphi(v_i)$ on the parameter plane. If $\gamma_i(s_i)$ intersects $\gamma_j(s_j)$ at some point q, and $s_i < s_j$ at q, then γ_j stops at q, γ_i continues. This procedure will generate the motorcycle graph [17] on the surface, as shown in the left two frames in Fig. 6.

The surface is partitioned into rectangular patches as shown in the right two frames of Fig. 6. Each surface patch is parameterized to a planar rectangle, as shown in Fig. 7. The corresponding surface patch and the planar rectangle are rendered using the same color. The motorcycle graph gives the T-mesh of the original surface Σ.

4.2. Quad-Mesh Generation

Suppose we have obtained the T-mesh $T = (V, E, F)$ from the first stage in the algorithm pipeline, where each face $f \in F$ is a planar rectangle. We denote the length of the edge $e \in E$ as d_e, and the change of the length as x_e.

In the step of isometric immersion and meromorphic quartic differential 4.1.7, the cut graph Γ is found, the fundamental domain $\tilde{\Sigma}$ is obtained, and the shortest paths γ_i's connecting p_0 and p_i are computed. We homotopically



(a) zoomed in (b) planar images

Figure 7: Each surface patch is parameterized to a planar rectangle.

deform the arcs of the cut graph Γ and the shortest paths γ_i's to be aligned with the edges in T, and cut the T-mesh T to get a fundamental domain $\tilde{T}$. We flatten face by face to isometrically immerse $\tilde{T}$ on the plane. Then we obtain the parametric positions of the singularities $\varphi(p_i) = (u_i, v_i)$, $i = 1, 2, \ldots, n$. Each position $\varphi(p_i)$ is represented by the lengths of the edges in T. Then we quantize (u_i, v_i) to be rational numbers (m_i, n_i) and construct the linear equations for singularity position constraints Eqn. (14).

In the step of homology group basis, the handle and tunnel loops have been computed $\{a_1, b_1, a_2, b_2, \cdots, a_g, b_g\}$. For each a_i, we find a *face path* in T containing a_i, and flatten the path face by face, until we return to the first face again. The rigid motion between the two images of starting face is the deck transformation τ_i corresponding to a_i. By our construction, τ_i can be represented by the edge lengths of the faces in the face path. Suppose the translation component of τ_i is (u_i, v_i), we quantize (u_i, v_i) to rational coordinates (m_i, n_i). Hence, we obtain a linear equation for the deck transformation constraints Eqn. (15).

Finally, for each face $f_i \in F$ (except the last face), we construct an equation requiring the opposite sides have the same length, namely the face constraint as Eqn. (16).

The linear system of Eqn. (14), Eqn. (15) and Eqn. (16) is solved using conventional least square method to obtain $\{x_e\}$, then each edge length is deformed to $d_e + x_e$. The flat metric of the updated T-mesh satisfies the conditions described in Theorem 3.23. We then find the least common multiple λ of all the constants on the right hand side of the linear system, and scale the T-mesh by the factor λ. Again we flatten the fundamental domain $\tilde{T}$ on the plane using the updated metric, such that $\varphi(p_0)$ is the origin, the images of the face edges are horizontal or vertical on the plane. Then we tessellate $\tilde{T}$ by the inter grid on the plane, this induces the quad-mesh of the original surface. The pipeline of the quad-mesh generation algorithm is summarized in Alg. 2.

5. Experimental Results

In this section, we report our experimental results. All the experiments were conducted on a PC with 1.60GHz Intel(R) core(TM) i5-8250U CPU, 16.0GB RAM and 64-bit Windows 10 operating system. The running time is reported in Table 1 and Table 4.

5.1. T-Mesh Generation

The singularities and the resulting T-meshes are illustrated in the figures. As shown in Fig. 8, the singularities are color-encoded, the red, blue and green circles represent the $+1$, -1 and -2 indices respectively. The white circles represents the T-junction points. Different surface patches are color-encoded differently. By carefully examining the texture patterns in Fig. 8, we can see that the adjacent patches differ by horizontal and vertical translations composed with rotations by angle $k\frac{\pi}{2}$, $k \in \mathbb{Z}$. Therefore, the parameterizations are seamless parameterizations. We can construct T-Splines on these T-meshes directly.

By examining Table 1, one can see that the running time for holomorphic 1-form is much longer than that of other steps in the algorithm pipeline. Because the computation involves many steps (homology basis, cohomology basis,

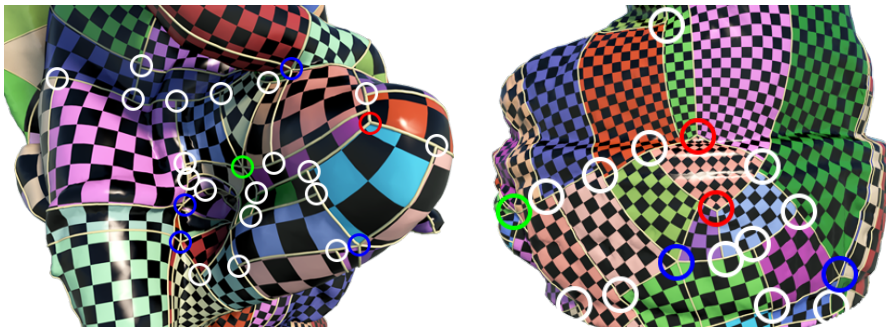


Figure 8: Singularities, white: T-junctions, blue: valence 5, green: valence 6, red :valence 3.

harmonic 1-form basis and holomorphic 1-form basis), the intermediate results are stored in mesh files, and passed by files. The file IO takes most of the time. This process can be optimized by performing all the tasks inside the memory without any file IO.

Table 1: Singularities Computation Running time

Model	Mesh Information				Holo 1-form	Holo zeros	Legalization of Singularities			Ricci Flow	Iso. Immersion
	$\#V$	$\#E$	$\#F$	$\#Genus$	Time(sec.)	Time(sec.)	Error Threshold	Iterations	Time(sec.)	Time(sec.)	Time(sec.)
Kitten	10.2k	30.7k	20.4k	1	10.247	—	3.0e-4	2132	0.002	0.013	0.006
Ornament	28.8k	86.5k	57.7k	1	47.954	—	3.0e-4	3382	0.005	6.177	0.014
Rocker arm	40.2k	120.5k	80.4k	1	39.014	—	3.0e-4	2049	0.004	12.134	0.021
Dancer	43.0k	129.1k	86.0k	1	43.913	—	3.0e-4	6069	0.005	10.0659	0.027
Bull head	75.8k	227.3k	151.5k	1	95.904	—	3.0e-4	2313	0.003	18.160	0.054
Sculpture	4.0k	12.2k	8.0k	2	4.828	0.029	1.0e-3	37601	0.052	1.485	0.002
Starcup	30.0k	90.0k	60.0k	2	51.682	0.301	3.0e-4	1167	0.005	6.654	0.013
Monk	38.5k	115.5k	77.0k	2	86.741	4.037	3.0e-4	17551	0.108	8.624	0.022
Hermanubis	39.9k	119.8k	79.9k	2	96.122	1.441	3.0e-4	17540	0.043	10.370	0.025
Amphora	82.6k	246.5k	164.3k	2	174.396	0.883	3.0e-4	5129	0.016	21.4288	0.046
Loveme	86.7k	260.2k	173.5k	2	191.663	1.156	3.0e-4	5776	0.020	25.7747	0.057
Buddha	59.4k	178.1k	118.7k	3	179.568	2.623	3.0e-4	670539	2.021	13.117	0.026
Kiss	61.7k	185.2k	123.5k	3	201.353	6.274	3.0e-4	94620	0.546	14.633	0.027
3Holes	65.0k	195.0k	130.0k	3	218.954	0.819	1.0e-3	343710	0.9333	16.756	0.032
Witch	75.0k	225.0k	150.0k	4	363.533	10.877	3.0e-4	304729	1.033	20.343	0.051

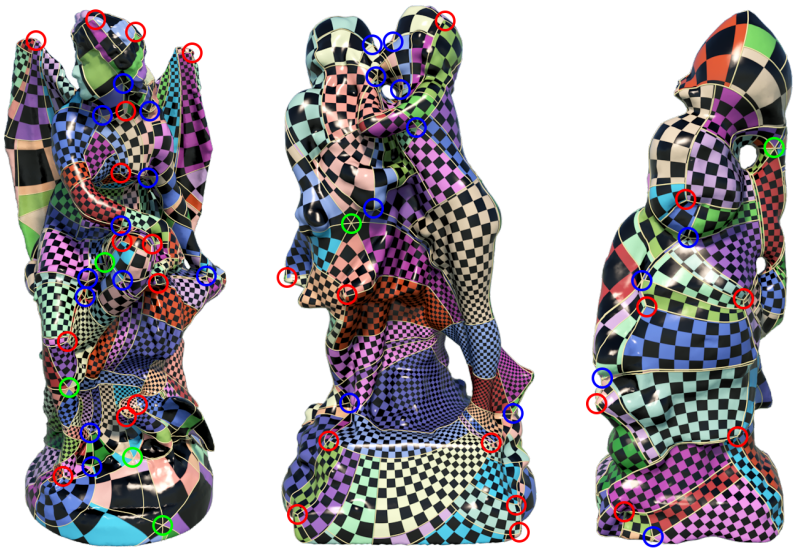
Hardware	CPU	RAM
	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz	16.0GB

5.2. Abel-Jacobi and Holonomy Condition Verification

Figure 9: Singularities and T-Mesh of the Loveme model.

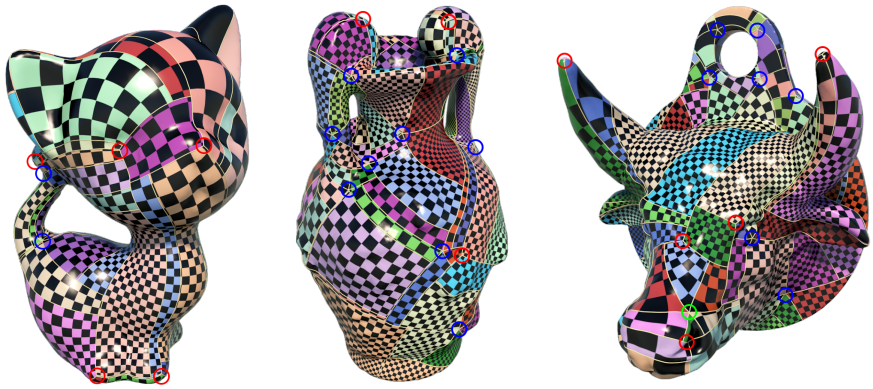
Ornament Dancer Hermanubis

Figure 10: Singularities and T-Meshes of high genus surfaces.



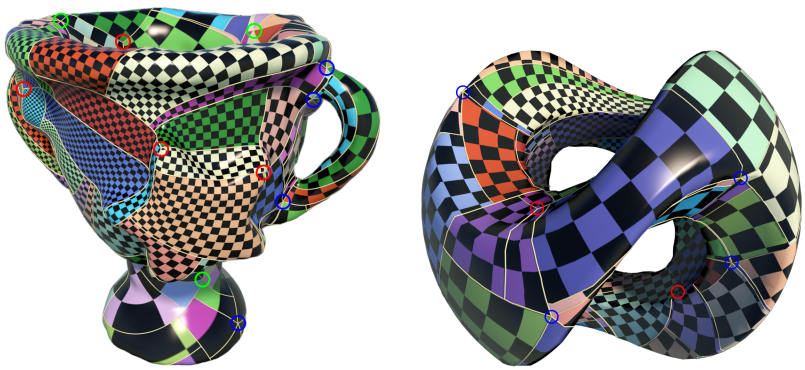
Witch model Kiss model Monk model

Figure 11: Singularities and T-Meshes of the surfaces with complicated geometries.



(a) Kitten model (b) Amphora model (c) Bull head

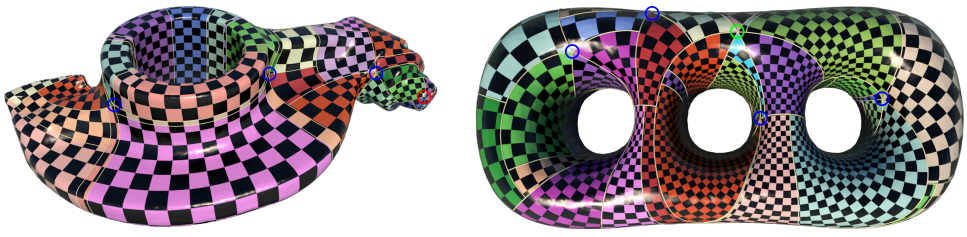
Figure 12: Singularities and T-Meshes of various surfaces.



Star cup model Sculpture model

Figure 13: Singularities and T-Meshes of high genus surfaces.

Figure 14: The motorcycle graph and T-mesh of the genus 3 kiss model.



Rocker arm 3 holes surface

Figure 15: Singularities and T-Meshes of high genus surfaces.

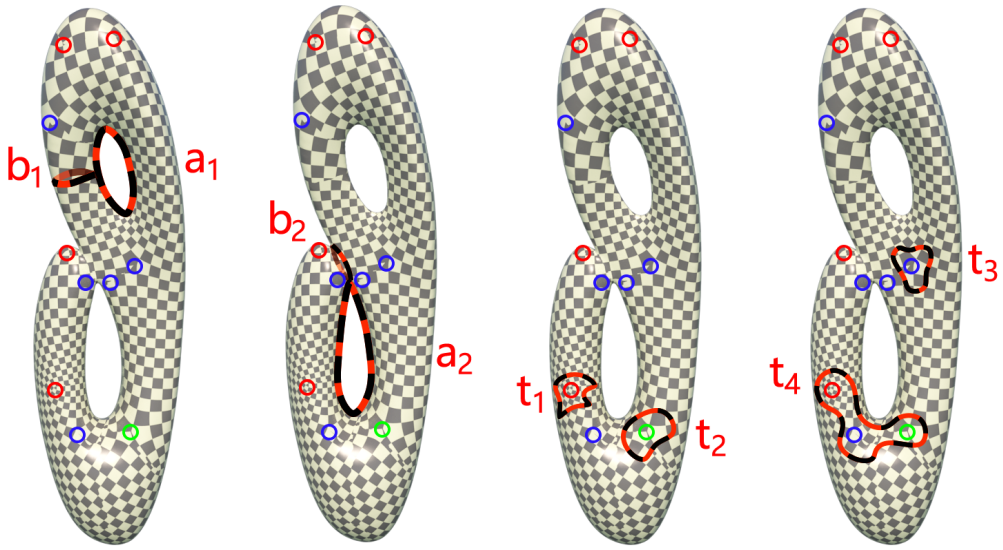


Figure 16: The loops on the genus two **Garniture** model : a_1, a_2 are the tunnel loops; b_1, b_2 are the handle loops; t_1, t_2, t_3 surround index $+1$, -2, -1 singularities respectively; t_4 encloses three singularities, with index $+1$, -1, -2 respectively.

Table 2: The holonomy of the loops in Fig. 16, rotation components.

Loops	$\mathbf{a_1}$	$\mathbf{b_1}$	$\mathbf{a_2}$	$\mathbf{b_2}$
Rotation degree(°)	90.31809	-0.12269	0.19303	89.81468
Loops	$\mathbf{t_1}$	$\mathbf{t_2}$	$\mathbf{t_3}$	$\mathbf{t_4}$
Rotation degree(°)	270.00047	540.00136	450.00192	539.99818

We verify the holonomy condition for a genus two surface as shown in Fig. 16. We compute the tunnel loops a_1, a_2 and handle loops b_1, b_2, and several loops enclosing different number of singularities. Then we compute their holonomies by parallel transportation on the flat metric computed using Ricci flow, the rotation components are reported in the Table 2. We can see that all the holonomies are very close $\frac{k}{2}\pi$, where k is an integer.

Furthermore, for every surface, we compute the image of the singularities under the Abel-Jacobi map, all the results are reported in Table 3. We can see that all the images are very close to the zero point in the Jacobian lattice, this shows the singularities satisfy the Abel-Jacobi condition. This demonstrates the

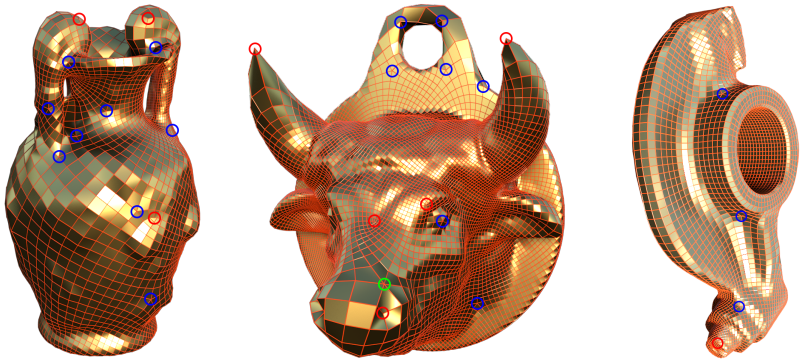
(a) Kiss model (b) Dancer model (c) Monk model (d) Witch model

Figure 17: Singularities and Quad-meshes of various surfaces.

accuracy of our proposed algorithm.

5.3. Quad-Mesh Generation

The T-meshes are converted to quad-meshes by solving the linear system of singularity position constraints Eqn. (14), deck transformation constraints Eqn. (15) and face constraints Eqn. (16). The numbers of vertices, edges, faces of T-meshes are listed in Tab. 5 and follow the predictions of Lemma 3.24.The running time for quad-mesh generation is reported in Table 4. Since the connectivity of a T-mesh is generally very simple, this step is highly efficient. The quad-meshes are directly converted from T-meshes, and illustrated in Fig. 17 and Fig. 18. The singularities of quad-meshes are exactly the same as those of T-meshes, the red, blue and green circles represent singularities with valence 3, 5 and 6 respectively. The quad-meshes are obtained by adjusting the edge lengths of T-meshes without any optimization, however we can see all the quad-faces are close to planar squares, this shows the quad-mesh is highly conformal to the original Riemannian metric, the sizes of the quad-faces vary smoothly, this shows the high regularity of conformal factors.



(a) Amphora model (b) Bull head model (c) Rocker-arm model

Figure 18: Singularities and Quad-meshes of various surfaces.

Table 3: Abel Jacobian mapping result

Model	Abel Jacobian Mapping	Model	Abel Jacobian Mapping
Kitten	$\begin{pmatrix} \text{2.13971e-04} + \text{i} * \text{7.09315e-05} \end{pmatrix}$	Ornament	$\begin{pmatrix} \text{-1.09501e-08} + \text{i} * \text{5.73307e-08} \end{pmatrix}$
Rockerarm	$\begin{pmatrix} \text{-6.05103e-05} + \text{i} * \text{6.27266e-06} \end{pmatrix}$	Dancer	$\begin{pmatrix} \text{-3.14143e-05} + \text{i} * \text{1.57991e-05} \end{pmatrix}$
Bull	$\begin{pmatrix} \text{-1.55144e-05} + \text{i} * \text{6.56513e-06} \end{pmatrix}$		
Starcup	$\begin{pmatrix} \text{4.59275e-05} - \text{i} * \text{1.27194e-04} \\ \text{8.14751e-05} - \text{i} * \text{2.32289e-04} \end{pmatrix}$	Monk	$\begin{pmatrix} \text{-1.37142e-05} - \text{i} * \text{1.84819e-04} \\ \text{4.70251e-05} + \text{i} * \text{1.90921e-04} \end{pmatrix}$
Hermanubis	$\begin{pmatrix} \text{-1.05753e-04} - \text{i} * \text{8.17228e-05} \\ \text{9.29236e-05} + \text{i} * \text{4.96067e-05} \end{pmatrix}$	Amphora	$\begin{pmatrix} \text{1.16072e-04} - \text{i} * \text{1.37645e-04} \\ \text{1.32789e-05} - \text{i} * \text{1.56983e-04} \end{pmatrix}$
Loveme	$\begin{pmatrix} \text{-9.65795e-05} + \text{i} * \text{3.60684e-05} \\ \text{-3.69644e-05} - \text{i} * \text{1.48141e-04} \end{pmatrix}$	Sculpt	$\begin{pmatrix} \text{-3.72147e-04} - \text{i} * \text{9.82485e-04} \\ \text{8.03122e-04} + \text{i} * \text{6.25321e-04} \end{pmatrix}$
Kiss	$\begin{pmatrix} \text{2.90402e-04} - \text{i} * \text{2.89651e-04} \\ \text{2.13554e-04} - \text{i} * \text{5.80312e-05} \\ \text{1.70373e-04} + \text{i} * \text{2.77541e-04} \end{pmatrix}$	3Holes	$\begin{pmatrix} \text{6.85741e-05} + \text{i} * \text{9.32962e-04} \\ \text{3.55608e-05} - \text{i} * \text{8.67721e-04} \\ \text{-1.36089e-05} + \text{i} * \text{5.60214e-04} \end{pmatrix}$
Witch	$\begin{pmatrix} \text{-1.29378e-04} - \text{i} * \text{2.40348e-04} \\ \text{-2.75192e-04} + \text{i} * \text{1.98399e-04} \\ \text{2.23835e-04} + \text{i} * \text{2.55373e-04} \\ \text{-2.64736e-04} + \text{i} * \text{2.39598e-04} \end{pmatrix}$	Buddha	$\begin{pmatrix} \text{1.16965e-04} + \text{i} * \text{2.90814e-04} \\ \text{-1.28974e-04} - \text{i} * \text{7.77251e-06} \\ \text{1.55074e-04} - \text{i} * \text{2.54977e-04} \end{pmatrix}$

6. Conclusion

This work proposes a rigorous and practical algorithm for T-mesh and quad-mesh generation based on Abel theorem. We prove the sufficient and necessary condition for a flat metric with cone singularities to be compatible with a quad-mesh 3.23, then develop the algorithm based on the Abel-Jacobi condition and

this theorem. The first stage of the algorithm is to generate a T-mesh: the initial divisor is optimized to satisfy the Abel-Jacobi condition, a meromorphic quartic differential is induced, the critical trajectories of the meromorphic quartic differential lead to the T-mesh. The second stage is to generate a quad-mesh: the edge lengths of the T-mesh is adjusted to satisfy the deck-transformation conditions by solving a linear system, a fundamental domain of the T-mesh is isometrically immersed in the plane using the updated metric, the immerse pulls back the planar integer-grid to the surface to produce the quad-mesh.

Our experimental results demonstrate that the method can handle surfaces with complicated topology and geometry. The algorithm is efficient and accurate. The resulting T-meshes can be used to construct T-Splines directly. The quad-meshes can be applied for NURBS construction.

In the future, from algorithmic point of view, we will further explore how to optimize the configurations of singularities and align the stream lines of the T-mesh/quad-mesh with the feature curves of the surfaces to improve the quality of the Spline surfaces; from theoretic point of view, we will further explore the intrinsic connection between quad-meshing and conformal geometry, especially the Teichmüller space theory and give more thorough analysis for the existence of quad-meshes with special requirements.

Table 4: Quad Mesh Generation Running time

Model	Mesh Information				*#Singularities*			Motor Cycle Graph				Finite trajectories parametrization	Quad Mesh	
	#V	*#E*	*#F*	*#Genus*	*#Valence3*	*#Valence5*	*#Valence6*	*#V*	*#E*	*#F*	Time(sec.)	Time(sec.)	*#F*	Time(sec.)
Rockerarm	40.2k	120.5k	80.4k	1	13	11	1	125	200	75	0.019	73.219	21998	0.381
Dancer	43.0k	129.1k	86.0k	1	42	40	1	415	664	249	0.043	66.269	16650	0.311
Bull	75.8k	227.3k	151.5k	1	20	18	1	195	312	117	0.035	122.526	35610	0.791
Monk	38.5k	115.5k	77.0k	2	24	22	5	263	424	159	0.033	24.980	10660	0.174
Amphora	82.6k	246.5k	164.3k	2	14	22	0	188	304	114	0.039	127.218	12242	0.362
2Kids	61.7k	185.2k	123.5k	3	21	29	4	286	464	174	0.037	41.341	16112	0.334
Witch	75.0k	225.0k	150.0k	4	46	32	19	509	824	309	0.069	134.691	84258	0.849

Hardware	CPU	RAM
	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz	16.0GB

Table 5: T-Mesh Connectivity

Model	Resulting T-Mesh Information							
	$\#V$	$\#E$	$\#F$	$\#Genus$	$\#Valence3$	$\#Valence5$	$\#Valence6$	$\sum k_i$
KITTEN	**115**	**184**	**69**	**1**	**12**	**10**	**1**	**92**
ROCKERARM	**125**	**200**	**75**	**1**	**13**	**11**	**1**	**100**
ORNAMENT	**120**	**192**	**72**	**1**	**12**	**12**	**0**	**96**
DANCER	**415**	**664**	**249**	**1**	**42**	**40**	**1**	**332**
BULL	**195**	**312**	**117**	**1**	**20**	**18**	**1**	**156**
SCULPT	**148**	**240**	**90**	**2**	**10**	**18**	**0**	**120**
STARCUP	**283**	**456**	**171**	**2**	**28**	**18**	**9**	**228**
MONK	**263**	**424**	**159**	**2**	**24**	**22**	**5**	**212**
HERMANUBIS	**228**	**368**	**138**	**2**	**23**	**11**	**10**	**184**
AMPHORA	**188**	**304**	**114**	**2**	**14**	**22**	**0**	**152**
LOVEME	**163**	**264**	**99**	**2**	**14**	**12**	**5**	**132**
BUDDHA	**401**	**648**	**243**	**3**	**34**	**36**	**7**	**324**
2KIDS	**286**	**464**	**174**	**3**	**21**	**29**	**4**	**232**
3HOLES	**111**	**184**	**69**	**3**	**3**	**13**	**3**	**92**
WITCH	**509**	**824**	**309**	**4**	**46**	**32**	**19**	**412**

Acknowledgment

The authors thank the encouragements and inspiring discussions with Dr. Tom Hughes and Dr. Kendrick Shepherd.

This work is partially supported by NSFC No. 61907005, 61720106005, 61772105 and 61936002.

References

[1] Pierre Alliez, Bruno Lévy, Alla Sheffer, and Nicolas Ray. Periodic global parameterization. *ACM Transactions on Graphics*, 25(4):1460–1485, 2006.

[2] Christian Bliek, Pierre Bonami, and Andrea Lodi. Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report. In *Proceedings of the twenty-sixth RAMP symposium*, pages 16–17, 2014.

[3] Ioana Boier-Martin, Holly Rushmeier, and Jingyi Jin. Parameterization of triangle meshes over quadrilateral domains. In *ACM International Conference Proceeding Series*, pages 193–203, 2004.

[4] David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. Integer-grid maps for reliable quad meshing. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.

[5] David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Computer Graphics Forum*, 32(6):51–76, 2013.

[6] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Transactions on Graphics*, 28(3):1–10, 2009.

[7] Marcel Campen, David Bommes, and Leif Kobbelt. Quantized global parametrization. *ACM Transactions On Graphics (tog)*, 34(6):1–12, 2015.

[8] Marcel Campen and Denis Zorin. On discrete conformal seamless similarity maps. *arXiv preprint*, page arXiv:1705.02422, 2017.

[9] Marcel Campen and Denis Zorin. Similarity maps and field-guided t-splines: a perfect couple. *ACM Transactions on Graphics*, 36(4), 2017.

[10] Nathan A Carr, Jared Hoberock, Keenan Crane, and John C Hart. Rectangular multi-chart geometry images. In *Eurographics Symposium on Geometry Processing*, pages 181–190, 2006.

[11] Wei Chen, Xiaopeng Zheng, Jingyao Ke, Na Lei, Zhongxuan Luo, and Xianfeng Gu. Quadrilateral mesh generation I : Metric based method. *Computer Methods in Applied Mechanics and Engineering*, 356:652–668, 2019.

[12] Tamal K. Dey, Fengtao Fan, and Yusu Wang. An efficient computation of handle and tunnel loops via reeb graphs. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.

[13] Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. Integrable polyvector fields. *ACM Transactions on Graphics*, 34(4):1–12, 2015.

[14] Simon Donaldson. *Riemann Surfaces*. Number 22 in Oxford Graduate Texts in Mathematics. Oxford University Press, 2011.

[15] Shen Dong, Peer Timo Bremer, Michael Garland, Valerio Pascucci, and John C Hart. Spectral surface quadrangulation. In *ACM SIGGRAPH*, pages 1057–1066, 2006.

[16] Hans-Christian Ebke, David Bommes, Marcel Campen, and Leif Kobbelt. Qex: robust quad mesh extraction. *ACM Transactions on Graphics (TOG)*, 32(6):1–10, 2013.

[17] David Eppstein, Michael T. Goodrich, Ethan Kim, and Rasmus Tamstorf. Motorcycle graphs: Canonical quad mesh partitioning. In *Computer Graphics Forum*, volume 27, pages 1477–1486, 2008.

[18] Xianfeng Gu and Shing-Tung Yau. Global conformal surface parameterization. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 127–137. Eurographics Association, 2003.

[19] Xianfeng Gu and Shing-Tung Yau. *Computational Conformal Geometry*, volume 3 of *Advanced Lectures in Mathematics*. International Press and Higher Education Press, 2007.

[20] Xianfeng David Gu, Feng Luo, Jian Sun, and Tianqi Wu. A discrete uniformization theorem for polyhedral surfaces. *Journal of Differential Geometry (JDG)*, 109(2):223–256, 2018.

[21] Topraj Gurung, Daniel Laney, Peter Lindstrom, and Jarek Rossignac. Squad: Compact representation for triangle meshes. *Computer Graphics Forum*, 30(2):355–364, 2011.

[22] Ying He, Hongyu Wang, Chi Wing Fu, and Hong Qin. A divide-and-conquer approach for automatic polycube map construction. *Computers & Graphics*, 33(3):369–380, 2009.

[23] Kangkang Hu and Yongjie Zhang. Centroidal voronoi tessellation based polycube construction for adaptive all-hexahedral mesh generation. *Computer Methods in Applied Mechanics and Engineering*, 305:405–421, 2016.

[24] Jin Huang, Muyang Zhang, Jin Ma, Xinguo Liu, Leif Kobbelt, and Hujun Bao. Spectral quadrangulation with orientation and alignment control. *Acm Transactions on Graphics*, 27(5):1–9, 2008.

[25] T. Jiang, X. Fang, J. Huang, H. Bao, Y. Tong, and M. Desbrun. Frame field generation through metric customization. *ACM Transactions on Graphics*, 34(4):1–11, 2015.

[26] Felix Kälberer, Matthias Nieser, and Konrad Polthier. Quadcover – surface parameterization using branched coverings. *Computer Graphics Forum*, 26(3):375–384, 2010.

[27] Nicolas Kowalski, Franck Ledoux, and Pascal Frey. A pde based approach to multidomain partitioning and quadrilateral meshing. In *International Meshing Roundtable*, 2013.

[28] Bruno Lévy and Yang Liu. Lp centroidal voronoi tessellation and its applications. *ACM Transactions on Graphics*, 29(4):1–11, 2010.

[29] Na Lei, Xiaopeng Zheng, Zhongxuan Luo, Feng Luo, and Xianfeng Gu. Quadrilateral mesh generation II : Meromorphic quartic differentials and Abel–Jacobi condition. *Computer Methods in Applied Mechanics and Engineering*, 366:112980, 2020.

[30] Wan-Chiu Li, Bruno Vallet, Nicolas Ray, and Bruno Lévy. Representing higher-order singularities in vector fields on piecewise linear surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1315–1322, 2006.

[31] Juncong Lin, Xiaogang Jin, Zhengwen Fan, and Charlie C. L Wang. Automatic polycube-maps. In *International Conference on Advances in Geometric Modeling and Processing*, pages 3–16, 2008.

[32] M. Lyon, M. Campen, and L. Kobbelt. Quad Layouts via Constrained T-Mesh Quantization. *Computer Graphics Forum*, 40(2):305–314, 2021.

[33] Tarini Marco, Pietroni Nico, Cignoni Paolo, Panozzo Daniele, and Puppo Enrico. Practical quad mesh simplification. *Computer Graphics Forum*, 29(2):407–418, 2010.

[34] Ashish Myles, Nico Pietroni, and Denis Zorin. Robust field-aligned global parameterization. *ACM Transactions on Graphics*, 33(4):1–14, 2014.

[35] Jonathan Palacios and Eugene Zhang. Rotational symmetry field design on surfaces. *ACM Transactions on Graphics*, 26(3):55, 2007.

[36] Budirijanto Purnomo, Jonathan D. Cohen, and Subodh Kumar. Seamless texture atlases. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 65–74, 2004.

[37] Nicolas Ray and Dmitry Sokolov. Robust polylines tracing for n-symmetry direction field on triangulated surfaces. *ACM Transactions on Graphics*, 33(3):1–11, 2014.

[38] J. F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzainet. Blossom-quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. *International Journal for Numerical Methods in Engineering*, 89(9):1102–1119, 2012.

[39] Joseph J. Rotman. *An Introduction to Algebraic Topology.* Number 19 in Graduate Texts in Mathematics. Springer, 1998.

[40] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. Designing quadrangulations with discrete harmonic forms. In *Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June*, pages 201–210, 2006.

[41] Luiz Velho and Denis Zorin. 4–8 subdivision. *Computer Aided Geometric Design*, 18(5):397–427, 2001.

[42] Ryan Viertel and Braxton Osting. An approach to quad meshing based on harmonic cross-valued maps and the ginzburg-landau theory. *arXiv*, abs/1708.02316, 2017.

[43] Hongyu Wang, Miao Jin, Ying He, Xianfeng Gu, and Hong Qin. User-controllable polycube map for manifold spline construction. In *ACM Symposium on Solid and Physical Modeling, Stony Brook, New York, Usa, June*, pages 397–404, 2008.

[44] Jiazhil Xia, Ismael Garcia, Ying He, Shi Qing Xin, and Gustavo Patow. Editable polycube map for gpu-based subdivision surfaces. In *Symposium on Interactive 3D Graphics and Games*, pages 151–158, 2011.

Algorithm 1 Optimize a Divisor to Satisfy the Abel-Jacobi Condition

Input: Closed mesh M; A group of singularities D; A *holomorphic* 1-form; Precision threshold ε.

Output: Optimized divisor D Abel-Jacobi condition.

if D doesn't satisfy Gauss-Bonnet Condition **then**
  Locate the vertices on M with local maximal Gaussian curvature as poles, or with local minimal curvature as zeros;
  Add these vertices to the divisor D, such that D satisfies the Gauss-Bonnet condition.
end if
Locate the zeros of φ to obtain the divisor (φ);
Compute $\Phi(D)$ and $\Phi(4(\varphi))$ using Eqn. 21;
Compute the Abel-Jacobi $\mu(D - 4(\varphi))$map by optimization using integer programming (Eqn.24);
for All each pole and zero p_i in D **do**
  Locate the face Δ_i containing p_i, isometrically embed Δ_i on the complex plane;
  Compute the local representation $\varphi_j(z_i) = h_j^i(z_i)dz_i$;
end for
while $\|\mu(D - 4(\varphi))\|^2 > \varepsilon$ **do**
  for All each pole and zero p_i in D **do**
    Compute the gradient of the energy (Eqn.25);
    if $p_i - \lambda\partial\nabla E/\partial p_i$ is inside Δ_i **then**
      Update the positions of the singularities $p_i \leftarrow p_i - \lambda\partial\nabla E/\partial p_i$;
    else
      Compute the intersection $q_i = \{p_i - t\partial E/\partial p_i\} \cap \partial\Delta_i$;
      Update the positions of the singularity $p_i \leftarrow q_i$;
      Update the face Δ_i to the neighboring face Δ_i' along $-\partial E/\partial p_i$;
      Isometrically embed Δ_i' and update the local representation $\varphi_j(z_i')$;
    end if
    Recompute the Abel-Jacobi map $\mu(D - 4(\varphi))$;
  end for
end while
return The divisor D.

Algorithm 2 Generate Quad-mesh from T-mesh

Input: Closed mesh M with genus g; A group of singularities D; T-mesh T;

Output: Quad-mesh Q.

Compute a fundamental domain $\bar{M}$ of the mesh M;
Compute the handle and tunnel loops $\{a_i, b_i\}_{i=1}^{g}$;
Select a singularity $v_0 \in D$, find the path $\gamma_i \subset \bar{M}$ from v_0 to $v_i \in S$;
for each loop γ in $\{a_i, b_i\}_{i=1}^{g}$ **do**
    Compute the face path homotopic to γ;
    Represent the deck transformation τ corresponding to γ by the edge lengths of the T-mesh T;
    Construct the linear equation for the deck transformation constraint Eqn. 15, namely the translation of τ is rational;
end for
for All each singularity $v_i \in D$ **do**
    Compute the face path homotopic to γ_i;
    Represent the parametric position of v_i by the edge lengths of the T-mesh T;
    Construct the linear equation for the singularity position constraint Eqn. 14, namely the position of v_i is rational;
end for
for All faces $f \in F$ except the last one **do**
    Construct the linear equation for the face side length constraint Eqn. 16
end for
Solve the linear system;
Update the edge length of T; Scale by the least common multiple;
Isometrically immerse a fundamental domain of T on the plane, such that v_0 is mapped to the origin, the edges of T are either horizontal or vertical;
The planar integer grid induces a quad-mesh Q on the fundamental domain, and then on M.
return The quad-mesh Q.