Stony Brook University

Algorithms in Computational Conformal Geometry

A Dissertation Presented

by

Yinghua Li

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

August 2012

Stony Brook University
The Graduate School

Yinghua Li

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Joseph S.B. Mitchell - Dissertation Advisor
Professor, Applied Mathematics and Statistics

Xiangmin Jiao - Chairperson of Defense
Associate Professor, Applied Mathematics and Statistics

Esther M. Arkin
Professor, Applied Mathematics and Statistics

Xianfeng Gu
Associate Professor, Computer Science

This dissertation is accepted by the Graduate School.

Charles Taber
Interim Dean of the Graduate School

Abstract of the Dissertation

Algorithms in Computational Conformal geometry

by
Yinghua Li

Doctor of Philosophy
in
Applied Mathematics and Statistics

Stony Brook University
2012

Computational conformal geometry is an intersectional field combining modern geometry theories from pure mathematics with computational algorithms from computer science.

In the first part of this dissertation, we firstly review a powerful tool in computational conformal geometry, the discrete surface Ricci flow, which is used to conformally deform the given Riemannian metric of a surface to a Riemannian metric according to a user defined Gaussian curvature on interior points and geodesic curvature along the boundaries. Using the discrete Ricci flow to embed the high genus surface into the hyperbolic plane, we propose an efficient algorithm to compute the shortest words for loops given on triangulated surface meshes. The design of this algorithm is inspired and guided by the work of Dehn and Birman-Series. In support of the shortest word algorithm, we also propose efficient algorithms to compute shortest paths and shortest loops under hyperbolic metrics using a novel technique, called transient embedding, to work with the universal covering space. In addition, we employ several techniques to relieve the numerical errors. Experimental results are given to demonstrate the performance in practice.

In the second part, we introduce two Delaunay refinement algorithms which give quality meshes on two-dimensional hyperbolic Poincaré disk in computing. These two Delaunay refinement algorithms are generalizations

of Chew's second algorithm and Ruppert's refinement algorithm, both of them are based on the Planar Straight Line Graph (PSLG) in Euclidean geometry. By modifying some definitions and adding new constraints, these two algorithms can be applied to surface meshes embedded in the hyperbolic Poincaré disk. The generalizations will work on global meshes, and termination of these two algorithms will be given under constraints.

To my parents

Contents

List of Figures

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor Joseph S.B. Mitchell and my co-advisor Xianfeng Gu, who has supported me with their patience and knowledge whilst allowing me the room to work in my own way. This dissertation would not have been possible without the guidance and the help from them.

I would like to thank Xiaotan Yin, Min Zhang, Wei Zeng, Weixin Guo for their help and suggestions.

Thanks also due to many friends here in Stony Brook, especially Loy Weng, Jason Zou, Xiaojie Wang, Wentao Jiang, Huaixin Zheng and Ruirui Jiang for the happiness through these years.

At last, I would like to thank the faculty and staff in the Department of Applied Mathematics and Statistics for their help in these years.

Chapter 1

Introduction

Conformal geometry studies the conformal structure of general surfaces. The conformal structure is a structure to measure the intersection angles between two curves on the surface, it's more rigid than topological structure which gives the neighborhood information, and it's more flexible than Riemannian metric which is a structure to measure the lengths of curves on the surface, the areas of domains on the surface and the intersection angle between curves. Computational conformal geometry is an intersectional field combining modern geometry theories from pure mathematics with computational algorithms from computer science. Computational conformal geometric methods can handle most of the geometric processing tasks for 3D shapes, which include shape representation, geometric compression, surface repairing, shape de-noising and smoothing, surface stitching and merging, meshing and re-meshing, surface classification, shape comparison, surface matching

and recognition, shape manipulation, and many others. The power comes from Poincaré's uniformization theorem, which states that all closed metric surfaces can be conformally mapped to one of the three canonical spaces, the sphere $\mathbb{S}^2$, the plane $\mathbb{R}^2$ or the hyperbolic disk $\mathbb{H}^2$. Gu and Yau [14] wrote a book which introduce the theories and algorithms of computational conformal geometry.

One of the powerful tools to compute the uniformization metric of surfaces is the surface Ricci flow, which was introduce by Hamilton [15]. A circle packing algorithm was introduced by Thurston in [29]. Chow and Luo discovered their intrinsic relations and laid down the theoretic foundation for discrete Ricci flow in [6], where the existence and convergence of the discrete Ricci flow were established. The algorithm is a gradient descend algorithm, which is not efficient in practice. Jin, Kim, Luo and Gu [18] improved the algorithm in later work by Newtons method. There are also some kind of uniformizations for open metric surfaces, such as circle domain[17], where the boundary components are mapped to circles in the sense of the underlying metric. Combing surface Ricci flow and Koebe's iteration, Zhang, Li, Zeng and Gu [33] computed the uniformization metric for circle domain of open metric surfaces. There are vast applications of the discrete Ricci flow in computational conformal geometry. In this dissertation, we will give an application in computing shortest words on hyperbolic surface, it's a typical example which solves a topological problem using geometric approaches.

With the development of three dimensional scanning technologies, 3D

shapes in real life can be easily acquired. One important task in computational conformal geometry is to find an algorithm to discretize smooth surfaces, such that the Gaussian curvature and mean curvature function converge. J.M. Morvan [24] gave a theorem which states that the Gaussian curvature and mean curvature function will converge on a sequence of triangulations of a surface if the minimal angle in each triangulation is bounded below and the maximum circumradius in each triangulation converges to 0. Also a lot of shape analysis applications, such as surface matching, registration, tracking, and object recognition, use the techniques from classical differential geometry. Most differential operators are approximated by discrete operators, partial differential equations are converted to large sparse linear systems using Finite Element Methods. When using finite element methods, it is important to produce qualify meshes with minimal angle θ.

Mesh refinement have been studied since 1990s, it is a technique for generating quality meshes by adding Steiner vertices into the input meshes. Two important works are those of Chew [5] and Ruppert [27]. As Delauany triangulation maximizes the minimum angle in the plane, Ruppert [27] gave a refinement algorithm of constraint Delaunay triangulation of Planar Straight Line Graph (PSLG) with output minimal angle $\alpha \leq 20.7°$, and gave the proof of termination and size-optimal for his refinement algorithm assuming that all the input angles are no less than $\pi/2$. In practice, Ruppert's algorithm performs well when the threshold α is set to be as big as around $\pi/6$. After Ruppert's work, some papers have addressed the small input angle condition

as well as other improvements[25][28][20]. Miller, Pav, and Walkington [23] gave an improved analysis of Ruppert's Delaunay refinement algorithm, they showed that the algorithm terminates for a minimum angle threshold as high as 26.5° under mild assumptions on the input.

Chew [5] originally gave his second refinement algorithm about constraint Delaunay triangulation of a PSLG with output minimal angle $\alpha = 30°$, and a similar proof of termination was given, also Chew generalized the algorithm to curve surfaces(surface meshes embedded in R^3) with new definition of the circumcenter. Chew's second refinement algorithm does not have any guarantees on grading or number of triangles. Shewchuk [28] showed that Chew's algorithm produces meshes that are nicely graded and size-optimal if the angle bound is relaxed to less than 26.5°, and the resulting meshes have fewer vertices than Ruppert's algorithm in practice. Recently, Alexander Rand [1] extended the analysis by Miller, Pav and Walkington, gave a proof of the termination of Chew's second Delaunay refinement algorithm for any minimum angle threshold less than 28.60°, and this holds not only for circumcenters but also for off-center Steiner vertices.

Using the discrete Ricci flow, one can conformally deform a given surface to a surface embedded in one of the three canonical spaces. If we can get a quality mesh with minimal angle bounded below on the canonical spaces, it will induce a quality mesh on the original mesh. Thus we want to generalize Chew's second Delaunay refinement algorithm and Ruppert's Delaunay refinement algorithm to the surface meshes embedded in their covering spaces

with constant curvatures $\mathbb{S}^2$,$\mathbb{R}^2$ and $\mathbb{H}^2$(with curvature $+1, 0, -1$). A similar work has been done by Zeng, Shi and Gu [32], they used Ruppert's method in the uniformization spaces. We want to give a theoretic analysis about the termination of the generalized algorithms on surfaces embedded in uniformization spaces.

For $\mathbb{R}^2$ case, which are periodic surface meshes embedded in the plane, the above two refinement algorithms can be generalized directly without any difficulty. While in $\mathbb{S}^2$ and $\mathbb{H}^2$ cases, it is not so easy to define 'good' triangulations on them using angles of the geodesic triangles. We need to modify the algorithms and give new definition of qualify triangulations and add constraints to ensure the termination of refinement algorithms. In the second part of this dissertation, we will focus on the case that surfaces S are embedded in the hyperbolic Poincaré disk $\mathbb{D}$: $S \cong \mathbb{D}/G$, where G is a Fuchsian group.

The organization of material is as follows. Chapter 2 will include discrete surface Ricci flow and its applications, in which we also give a brief introduction about some basic concepts in Geometry and topology, surface Ricci flow, hyperbolic embedding and solving the shortest words vis shortest loops on the hyperbolic plane. In Chapter 3, we will give two generalized Delaunay refinement algorithms on surfaces embedding in the hyperbolic Poincaré Disk Model. A brief introduction about the hyperbolic space and the hyperbolic Poincaré disk model will be given, the hyperbolic triangles and Delauany triangulation on the Poincaré disk will be discussed. The generalized Chew's

second algorithm and Ruppert's refinement algorithm will be described in the sections in this chapter.

Chapter 2

Discrete Ricci Flow and Its Applications

2.1 Brief Introduction of Geometry and Topology

2.1.1 Riemannian Geometry

Let M be a second countable Hausdorff differentiable manifold of dimension n. A Riemannian metric on M is a family of positive definite inner products on the tangent bundle

$$g_p : T_pM \times T_pM \to \mathbb{R} \qquad p \in M \tag{2.1}$$

such that, for all differentiable vector field X, Y on M

$$p \to g_p(X(p), Y(p))$$

defines a differential function $M \to R$. The assignment of an inner product g_p to each point p of the manifold is called a metric tensor. If we use the local coordinates $\{\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\}$, then the above metric tensor can be written as

$$g_{ij}(p) := g_p((\frac{\partial}{\partial x_i})_p, (\frac{\partial}{\partial x_j})_p).$$

Equivalently, the metric tensor can be weitten in terms of the dual basis $\{dx_1, \ldots, dx_n\}$ of the cotangent bundle as

$$g = \sum_{i,j} g_{ij} dx_i \otimes dx_j. \tag{2.2}$$

Endowed with this metric, the differential manifold (M, g) is a Riemannian manifold.

If $\gamma : [a, b] \to M$ is a continuously differentiable curve in the Riemannian manifold, then the length of γ is given by

$$L(\gamma) = \int_a^b \|\gamma'(t)\| dt = \int_a^b \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt.$$

Then the distance between any two points x and y on M is defined as

$$d(x, y) = inf\{L(\gamma) : \gamma \text{ is a continuously differential curve joining x and y}\}.$$

With this definition, every connected Riemannian manifold M becomes a metric space. Geodesics in a Riemannian manifold are then the locally distance-minimizing paths.

2.1.2 Algebraic Topology

Given a topological surface S, the fundamental group $\pi_1(S, p)$ is formed by the sets of equivalence classes of all loops $\{f : [0, 1] \to S | f$ is continuous and $f(0) = f(1) = p\}$ with the base point p. Two loops are equivalent iff one of the loop can be continuously deformed to the other loop without leaving the surface S. The formal definition is given as follows.

Definition 2.1 (Homotopy equivalence)**.** *Two loops f and g are homotopy equivalent, if there is a continuous map $h : [0, 1] \times [0, 1] \to S$ with the property that, for all $0 \leq t \leq 1$*

$$h(t, 0) = f(t), h(t, 1) = g(t), \quad \textit{and} \quad h(0, t) = p = h(1, t).$$

The multiplication of two loops f and g is defined as

$$f \cdot g = \begin{cases} f(2t), & 0 \leq t \leq 1/2; \\ g(1 - 2t), & 1/2 \leq t \leq 1. \end{cases}$$

The product of two homotopy classes of loops $[f]$ and $[g]$ is then defined as $[f \cdot g]$. The homotopy classes with this operation form a group, which is

called the fundamental group at p. If S is path connected, then we have $\pi(S,p) \cong \pi(S,q)$ for all $p,q \in S$, so we can simply denote it as $\pi_1(S)$.The identity element is the constant map at the base point, and the inverse of a loop f is the loop g defined by $g(t) = f(1-t)$. That is, g follows f backwards. S is called simply connected if and only if S is path connected and $\pi_1 s$ is trivial, i.e., $\pi_1 s$ consists only of the identity element.

Suppose S is a genus g closed surface. A canonical set of generators of $\pi_1(S)$ consists of $\{a_1, b_1, a_2, b_2, \ldots, a_g, b_g\}$, such that the pair a_i and b_i has one intersection point, the pairs $\{a_i, a_j\}$, $\{b_i, b_j\}$ and $\{a_i, b_j\}$, have no intersections, where $i \neq j$. In this dissertation, we consider the fundamental group at a base point p, then we can move all the intersection points to the point p to get a canonical set of generators of $\pi_1(S,p)$.

A covering space of S is a space $\tilde{S}$ together with a continuous surjective map $h : \tilde{S} \to S$, such that for every $p \in S$ there exists an open neighborhood U of p such that $h^{-1}(U)$ is a disjoint union of open sets in $\tilde{S}$, each of which is mapped homeomorphically onto U by h. The map h is called the covering map. A connected covering space $\tilde{S}$ is a universal cover if it is simply connected. Suppose $\gamma \subset S$ is a loop through the base point p on S. Let $\tilde{p}_0 \in \tilde{S}$ be a preimage of the base point $\tilde{p}_0 \in h^{-1}(p)$, then there exists a unique path $\tilde{\gamma} \subset \tilde{S}$ lying over γ (i.e. $h(\tilde{\gamma}) = \gamma$) and $\tilde{\gamma}(0) = \tilde{p}_0$. $\tilde{\gamma}$ is a lift of γ.

A deck transformation of a cover $h : \tilde{S} \to S$ is a homeomorphism

$$h : \tilde{S} \to \tilde{S},$$

such that $h \circ f = h$.

All deck transformations form a group, the so-called deck transformation group. A fundamental domain of S is a simply connected domain, which intersects each orbit of the deck transformation group only once. A fundamental domain can be obtained by slicing a surface S along canonical fundamental group generators. Deck transformations map fundamental domains to fundamental domains. The deck transformation group $Deck(S)$ is isomorphic to the fundamental group $\pi_1(S, p)$. Let $\tilde{p}_0 \in h^{-1}(p)$, $\phi \in Deck(S)$, γ is a path in the universal cover connecting $\tilde{p}_0$ and $\phi(\tilde{p}_0)$, then the projection of $\tilde{\gamma}$ is a loop on S, ϕ corresponds to the homotopy class of the loop,

$$\phi \to [h(\tilde{\gamma})].$$

This gives the isomorphism between $Deck(S)$ and $\pi_1(S, p)$.

The universal covering space for a high genus surface is the hyperbolic space $\mathbb{H}^2$. The Möbius transformation group $PSL(2, R)$ is regarded as a group of isometries of the hyperbolic plane, or conformal transformations of the unit disc, or conformal transformations of the upper half plane. A discrete subgroup of $PSL(2, R)$ is called a Fuchsian group.

2.2 Surface Ricci Flow

2.2.1 Riemann surface

Definition 2.2. *A function* $f : \mathbb{C} \to \mathbb{C}, z = x + yi \to \omega = u + vi$ *is holomorphic, if it satisfies the following Cauchy-Riemann equation:*

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \tag{2.3}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \tag{2.4}$$

If a holomorphic function f is bijective and f^{-1} is holomorphic, then f is said to be biholomorphic or a conformal mapping.

Definition 2.3. *A Riemann Surface represents a two-dimensional manifold* M *with an atlas* $\{(U_\alpha, z_\alpha\})$*, such that* $\{U_\alpha\}$ *is an open covering,* $M \subset \cup U_\alpha$*;* $z_\alpha : U_\alpha \to \mathbb{C}$ *is a homeomorphism from* $U_\alpha \to z_\alpha(U_\alpha)$*. If* $U_\alpha \cap U_\beta \neq \emptyset$*, then*

$$z_\beta \circ z_\alpha^{-1} : z_\alpha(U_\alpha \cap U_\beta) \to z_\beta(U_\alpha \cap U_\beta)$$

is biholomorphic.

The atlas $\{(U_\alpha, z_\alpha)\}$ is called the complex atlas or conformal atlas of the Riemann surface. Given two conformal atlases $\{(U_\alpha, z_\alpha)\}$ and $\{(U_\beta, z_\beta)\}$, if their union is still a conformal atlas, then we say they are equivalent. Each equivalence class of conformal atlases is called a conformal structure or complex structure.

Suppose M has a Riemannian metric g, then we require the conformal structure to be compatible with the Riemannian metric, namely, on each local chart of $\{(U_\alpha, z_\alpha)\}$,

$$g = e^{2\lambda(z_\alpha)} dz_\alpha d\bar{z}_\alpha,$$

which means that the local parameters are isothermal coordinates.

Given a metric surface with a differential atlas $\{(U_\alpha, z_\alpha)\}$, if all local coordinates are isothermal coordinates, the $\{(U_\alpha, z_\alpha)\}$ is a conformal structure. According to the existence theorem of the isothermal coordinates for arbitrary metric surface, any metric surface has a conformal structure compatible with its Riemannian metric, therefor any metric surface is Riemann surface.

Suppose M and $\tilde{M}$ are two Riemann surfaces. A mapping $f : M \to \tilde{M}$ is called a conformal mapping, if the mapping $\tilde{\phi} \circ f \circ \phi^{-1}$ between any local parameter charts (U, ϕ) and $(\tilde{U}, \tilde{\phi})$. Furthermore, if f is one to one and onto, and f^{-1} is also holomorphic, then M and $\tilde{M}$ are conformally equivalent.

Suppose S is a surface embedded in $\mathbb{R}^3$, then it has an induced Euclidean metric $\mathbf{g}$. A *conformal metric deformation* refers to changing the Riemannian metric to $e^{2\lambda}\mathbf{g}$, where $\lambda : S \to \mathbb{R}$ is a function defined on the surface. A conformal deformation preserves angles. Klein−Poincaré uniformization theorem ([13] page 206) states that there exits a conformal deformation, such that the Gaussian curvature under the new metric is one of the three constants $\{+1, 0, -1\}$, depending on the topology of the surface. Such kind of

Riemannian metric is called the *canonical metric* or *uniformization metric* of the surface. The topology of a surface is represented by a tuple (g, b), where g is the genus of the surface, and b is the number of boundary components.

Suppose S is a closed surface (topology type $(g, 0)$) with canonical metric.

- $g = 0$: S is the unit sphere.
- $g = 1$: S can be represented as $\mathbb{R}^2/G$, where G is a translation group, generated by $z \to z+1$, and $z \to z+\tau$, where $|\tau| \geqslant 1$, and $|Re(\tau)| \leqslant \frac{1}{2}$. The projection of the Euclidean metric to S is the *canonical metric* on S.
- $g = 2$: There is an essentially unique hyperbolic rigid motion group G, acting on the hyperbolic space $\mathbb{H}^2$, so that S equals to $\mathbb{H}^2/g$. Here $\mathbb{H}^2$ is the Poincaré model of hyperbolic space, which is the unit disk with canonical metric $ds^2 = \frac{4dzd\bar{z}}{(1-z\bar{z})^2}$. All the hyperbolic rigid motions on $\mathbb{H}^2$ are *Möbius transformations*, which has the form $e^{i\theta}\frac{z-z_0}{1-\bar{z}_0 z}$, $|z_0| \leq 1$.

A surface S with boundary components (topology type $(g > 1, b > 0)$) can be topologically embedded in a closed surface $\bar{S}$, where $\bar{S}-S$ has finitely many components, each component is a topological disk. Then S is *topologically finite*.

Suppose $\bar{S}$ is a closed Riemann surface of genus $g > 0$, and suppose S is a subsurface of $\bar{S}$ where $\bar{S} - S$ has finitely many components, each of which is either a point of a closed circular disc in the canonical metric on $\bar{S}$; then

we say S is a *circle domain* on$\bar{S}$. The uniformization theorem for surfaces with boundaries is as follows,

Theorem 2.1. *Let S be a topologically finite Riemann surface of genus $g > 0$. Then there is a closed Riemann surface $\bar{S}$ of genus g, and there is a conformal embedding $f : S \to \bar{S}$, so that $f(S)$ is a circle domain on $\bar{S}$. This representation is unique.*

The proof can be found in [22]. In fact, He and Schramm[17] gave a more general result: a Riemann surface with finite genus and at most countably many boundaries can be conformally mapped to a circle domain. It is still an open problem to show that a Riemann surface with finite genus and uncountably many boundaries can be conformally mapped to a circle domain.

2.2.2 Surface Ricci Flow

The surface Ricci flow is a powerful tool to computer the uniformization metric for surfaces. Given a point in a surface with a Riemannian metric g. Choose an isothermal coordinates (u, v) at the point so that,

$$g = e^{2\lambda(u,v)}(du^2 + dv^2), \tag{2.5}$$

then the Gaussian curvature is given by

$$K(u, v) = -\frac{1}{e^{2\lambda(u,v)}}(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2})\lambda. \tag{2.6}$$

Use the notation $\triangle_g = \frac{1}{e^{2\lambda}}(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2})$-the Laplace-Beltrami operator in the metric g, then

$$K = -\triangle_g \lambda. \tag{2.7}$$

If $\bar{g} = e^{2\bar{\lambda}(u,v)}(du^2 + dv^2)$, then $\bar{K} = -\triangle_{\bar{g}}\lambda$. By direct computation, we can get

$$\bar{K} = \frac{1}{e^{2\tau}}(K - \triangle_g \tau),$$

where $\tau = \bar{\lambda} - \lambda$.

Similarly, if the surface has boundary, then the geodesic curvature $\bar{k}$ of $\bar{g}$ along the boundary can be calculated by

$$\bar{k} = \frac{1}{e^{\tau}}(k - \partial_n \tau),$$

where n is the outwards normal on the tangent planes along the boundary.

The prescribing Gaussian curvature problem asks given a function $\bar{K}$ defined on a surface (M, g) with a Riemannian metric g, if there is a metric $\bar{g}$ conformal to g so that the Gaussian curvature of $\bar{g}$ is the given function $\bar{K}$. An efficient algorithm to compute the conformal metric $\bar{g}$ with prescribed curvature $\bar{K}$ is the surface Ricci flow method.

Definition 2.4 (Ricci flow)**.** *Suppose S is a smooth surface with Riemannian metric g. The Ricci flow is the process to deform the metric $g(t)$ according*

to its Gaussian curvature $K(t)$, where t is the time parameter:

$$\frac{dg_{ij}(t)}{dt} = -K(t)g_{ij}(t). \tag{2.8}$$

The Ricci flow was introduced by Richard Hamilton [15] in 1981 in order to gain insight into the geometrization conjecture of William Thurston, which concerns the topological classification of three-dimensional smooth manifolds. Hamilton's idea was to define a kind of nonlinear diffusion equation which would tend to smooth out irregularities in the metric. Informally, the Ricci flow tends to expand negatively curved regions of the manifold, and contract positively curved regions. It plays an important role in the proof of the Poincare conjecture by Perelman.

Let $g(t) = e^{2u(t)}g(0)$, then the Ricci flow is

$$\frac{du(t)}{dt} = -2K(t). \tag{2.9}$$

To study the convergence of the Ricci flow (2.9), Hamilton introduced the following normalized Ricci flow

$$\frac{du(t)}{dt} = -2(K(t) - 2\pi\frac{\chi(S)}{A(t)}), \tag{2.10}$$

which preserve the area of the surface during the flow, here $A(t)$ is the area of the Riemannian metric $g(t)$.

Theorem 2.2 (Hamilton 1988[15]). *For a closed surface of non-positive Euler characteristic, the normalized Ricci flow will converge to a metric such that the Gaussian curvature is constant everywhere.*

Theorem 2.3 (Chow 1991[7]). *For a closed surface of positive Euler characteristic, the normalized Ricci flow will converge to metric such that the Gaussian curvature is constant everywhere.*

The above two theorems postulate that the Ricci flow defined in 2.9 is convergent and lead to a conformal uniformization metric $g(\infty)$.

The Ricci flow can be easily modified to compute a metric with a prescribed curvature $\bar{K}$:

$$\frac{du(t)}{dt} = 2(\bar{K} - K(t)). \tag{2.11}$$

2.2.3 Discrete Ricci Flow

In this subsection, we introduce the practical algorithms for simulating smooth surface Ricci flow. The goal is to construct a discrete Riemannian metric-the general form of circle packing metric with a prescribed Gaussian curvature on the triangular meshes.

The general form of circle packing involves circles intersecting in angles is firstly studied in details by Thurston[29].
Suppose the vertex set of a triangular mesh M is $V = \{v_1, v_2, \ldots, v_n\}$, then the circle radii is defined to be a funtion

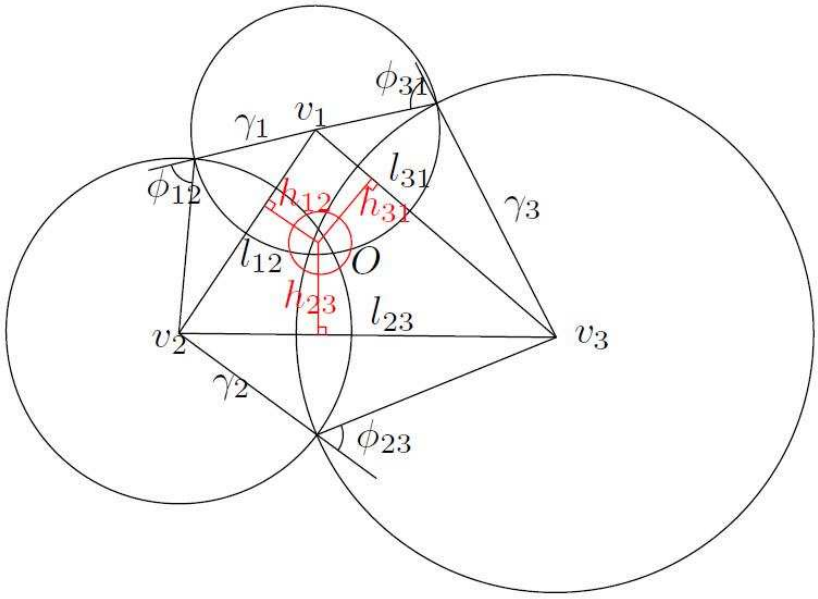

Figure 2.1: General Circle Packing

$$\Gamma : V \longrightarrow \mathbb{R}^+, \qquad \Gamma(v_i) = \gamma_i.$$

Then the general surface circle packing is given in the following way:

1. Take triangulation of a surface of arbitrary topology;
2. The intersection angle Φ is an arbitrary acute angle;
3. Each circle is generalized to a cone centered at the vertex;

4. The edge length is determined by the following cosine laws:

$$\begin{aligned} l_{ij} &= \sqrt{\gamma_i^2 + \gamma_j^2 + 2cos\phi_{ij}\gamma_i\gamma_j} && \mathbb{E}^2, \\ \cosh(l_{ij}) &= \cosh\gamma_i \cosh\gamma_j + \sinh\gamma_i \sinh\gamma_j \cos\phi_{ij} && \mathbb{H}^2, \\ \cos(l_{ij}) &= \cos\gamma_i \cos\gamma_j + \sin\gamma_i \sin\gamma_j \cos\phi_{ij} && \mathbb{S}^2. \end{aligned}$$

Definition 2.5 (Discrete metric)**.** *Suppose M is a triangular mesh, a discrete metric is a function defined on non-oriented edges of M,*

$$l : E \longrightarrow \mathbb{R}^+,$$

such that for each triangle $[v_i, v_j, v_k]$, the triangle inequality holds

$$l([v_i, v_j]) + l([v_j, v_k]) > l([v_k, v_i]).$$

Definition 2.6 (Circle packing metric)**.** *Suppose M is a triangular mesh, with a circle packing (M, Γ, Φ). Then the circle packing metric is defined as*

$$\begin{aligned} l([v_i, v_j]) = & \sqrt{\gamma_i^2 + \gamma_j^2 + 2\cos\phi_{ij}\gamma_i\gamma_j} && \mathbb{E}^2; \\ l([v_i, v_j]) = & \cosh^{-1}(\cosh\gamma_i \cosh\gamma_j + \sinh\gamma_i \sinh\gamma_j \cos\phi_{ij}) && \mathbb{H}^2; \\ l([v_i, v_j]) = & \cos^{-1}(\cos\gamma_i \cos\gamma_j + \sin\gamma_i \sin\gamma_j \cos\phi_{ij}) && \mathbb{S}^2. \end{aligned}$$

Thurston proved that if all the intersection angles are acute, then the edge length induced by a circle packing triangle satisfy the triangle inequality.

Definition 2.7 (Conformal equivalence)**.** *Two different circle packing metric* (M, Γ_1, Φ_1) *and* (M, Γ_2, Φ_2) *are called conformal equivalent if*

$$\Phi_1 = \Phi_2. \tag{2.12}$$

Given the circle packing metric, we can compute the edge length of each triangle in the triangular mesh, and the inner angles of the triangles. Using the inner angles of the triangles in the triangular mesh, we can define the discrete Gaussian curvature.

Definition 2.8 (Discrete Gaussian Curvature)**.** *Suppose all the corner angles surrounding a vertex* v *on a closed mesh* M *without boundary is* $\alpha_1, \ldots, \alpha_n$*, then the curvature at the vertex* v *is*

$$K(v) = 2\pi - \sum_{i=0}^{n} \alpha_i. \tag{2.13}$$

The Gauss-Bonnet theorem also holds for discrete meshes.

Theorem 2.4 (Discrete Gauss-Bonnet Theorem)**.** *Suppose* M *is closed triangular mesh with a discrete metric. Then*

$$\sum_{v \in M} K(v) = 2\pi\chi(M),$$

where $\chi(M) = V + F - E$ is the Euler number of the mesh.

The above results can be generalized to meshes with boundaries. If v is a boundary vertex of a mesh M, and the summation of all the corner angles surrounding v is α, then the discrete Gaussian Curvature at v is

$$K(v) = \pi - \alpha. \tag{2.14}$$

Now, we can define the discrete Ricci flow, which was introduced by Chow and Luo in 2003[6].

Suppose we are given a mesh M with a circle packing metric (M, Γ, Φ). The circle centered at the vertex v_i has radius γ_i. If the current Gaussian curvature of v_i is K_i and the prescribed Gaussian curvature is $\bar{K}_i$, then the discrete Ricci flow is defined as

$$\frac{du_i(t)}{dt} = (\bar{K}_i - K_i(t)) \tag{2.15}$$

where

$$u_i = \begin{cases} \ln \gamma_i, & \mathbb{E}^2, \\ \ln \tanh \gamma_i, & \mathbb{H}^2, \\ \ln \tan \gamma_i, & \mathbb{S}^2. \end{cases}$$

If we add one normalization step to ensure $\sum_{v_i \in M} u_i = 0$, then it is equivalent to the area preserving constraint in the smooth surface Ricci flow. Chow and Luo proved that the discrete Ricci flow is exponentially convergent in [6] for

Euclidean case.

Theorem 2.5. *Suppose (M, ϕ) is a closed weighted mesh, whose $\Phi \in [0, \pi]$. Given any initial circle packing metric based on the weighted mesh, the solution to the discrete Ricci flow in the Euclidean geometry with the given initial values exists for all time and converges exponentially fast.*

To improve the algorithm for implementation of the discrete surface Ricci flow, we use Newton's method. It can be proved that the discrete Ricci flow is exactly the negative gradient flow of the following energy function–*Riccci energy*,

$$E(u) = \int_{u=u_0}^{u} \sum K_i du_i.$$

where we choose a special initial metric $u_0 = (0, \ldots, 0) \in \{u \in \mathbb{R}^n | \sum_{i=1}^n u_i = 0\}$. The Hessian matrix of E with respect to u is positive define. So it has at most one global minimum. The Hessian matrix H of E with respect to u is given as follows:

$$\begin{aligned} H_{ij} = \frac{\partial^2 E(u)}{\partial u_i \partial u_j} &= \frac{\partial K_i}{\partial u_j} = -w_{ij}, \quad if \quad [v_i, v_j] \in M, \\ H_{ii} = \frac{\partial^2 E(u)}{\partial u_i^2} &= \frac{\partial K_i}{\partial u_i} = \sum_{[v_i, v_j] \in M} w_{ij}, \quad v_i \in M \\ H_{ij} = \frac{\partial^2 E(u)}{\partial u_i \partial u_j} &= 0, \text{otherwise} \end{aligned} \tag{2.16}$$

where $w_{ij} = \frac{h_{ij}^k + h_{ij}^l}{l_{ij}}$ if $[v_i, v_j]$ is an edge shared by two faces $[v_i, v_j, v_k]$ and $[v_i, v_j, v_l]$, and $w_{ij} = \frac{h_{ij}^k}{l_{ij}}$ for a boundary edge $[v_i, v_j]$ which attaches to one

face $[v_i, v_j, v_k]$.

In order to compute a metric with the prescribed curvature $\bar{K}$, the energy can be reformulated as

$$E(u) = \int_{u=u_0}^{u} \sum_{i=1}^{n} (\bar{K}_i - K_i) du_i$$

Algorithm 1. *Newton's method of discrete Ricci flow*

- *Input: A mesh M imbedded in $\mathbb{R}^3$, target curvature $\bar{K}$, curvature error threshold ϵ.*
- *output: a circle packing metric (M, Γ, Φ) which induces $\bar{K}$.*

Compute the initial circle packing metric (M, Γ_0, Φ);

Compute the initial curvature.

$$u = 0$$

while $max|\bar{K}_i - K_i| > \epsilon$ do

for all edge $e = [v_i, v_j] \in M$ do

Compute the edge weight $w_{ij}(u)$ to form the Hessian matrix H.

end

$$du \longleftarrow H^{-1}(\bar{K} - K)$$

$$u \longleftarrow u + du$$

$$K \longleftarrow K(u)$$

end

$$\bar{u} \longleftarrow u.$$

2.3 Applications of the Discrete Ricci Flow

Using the discrete Ricci flow, one can do many applications, such as isometric planar embedding, optimal discrete conformal parameterization, surface matching, hyperbolic embedding[21]. Also Combining with discrete surface Ricci flow and the Koebe's iteration method, the uniformization metric of the circle domain of surfaces with boundaries are firstly be computed [33]. In this section, we introduce an interesting algebraic problem, the shortest word problem, which can be solved by the discrete Ricci flow. We will give a brief introduction for hyperbolic embedding of the universal covering space, which will give us a way to compute the shortest word for the Fuchsian groups in the next section, and it is a preprocess step for the generalized Delaunay refinement algorithms on the hyperbolic plan in the next chapter.

2.3.1 Hyperbolic Embedding

Suppose M is a mesh with a circle packing metric in hyperbolic background geometry. Let $(\bar{M}, \pi)$ be the universal covering space of M, where π is the projection. Suppose (Γ, Φ) is a circle packing metric on M, then the pull back metric on $\bar{M}$ is also a hyperbolic circle packing metric, denoted

as $(\bar{M}, \pi^*\Gamma, \pi^*\Phi)$. If all the discrete vertex curvatures of (M, Γ, Φ) are zeros, then all the vertex curvature $(\bar{M}, \pi^*\Gamma, \pi^*\Phi)$ are also zeros, therefore $(\bar{M}, \pi^*\Gamma, \pi^*\Phi)$ can be isometrically embedded onto the hyperbolic space $\mathbb{H}^2$. So we need to compute the hyperbolic metric on the triangular mesh using the discrete hyperbolic Ricci flow.

After we have the uniformizaion metric, we can slice the mesh along the canonical generators of the fundamental group to get a fundamental domain and embed it isometrically into the hyperbolic disk. Here we use one of the hyperbolic space model-*Poincaré* disk to be the universal covering space, which is the unit disk $|z| < 1$ on the complex plane with the metric

$$ds^2 = \frac{4dzd\bar{z}}{(1 - |z|^2)^2}. \tag{2.17}$$

We will give a brief introduction about the Poncaré disk model for the hyperbolic plane in the beginning of next chapter.

The detailed algorithms for hyperbolic embedding of the universal covering space is described in [19] and [21]. The algorithm pipeline is as follows:
1) Embed a canonical fundamental domain using the hyperbolic cosine law.
2) Compute the deck transformation group generators.
3) Tile the whole canonical domain $\mathbb{H}^2$.

Once the universal covering space is embedded onto the *Poincaré* disk, we can use hyperbolic lines to separate the fundamental domains. Each fundamental domain becomes a hyperbolic polygon, which is called the funda-

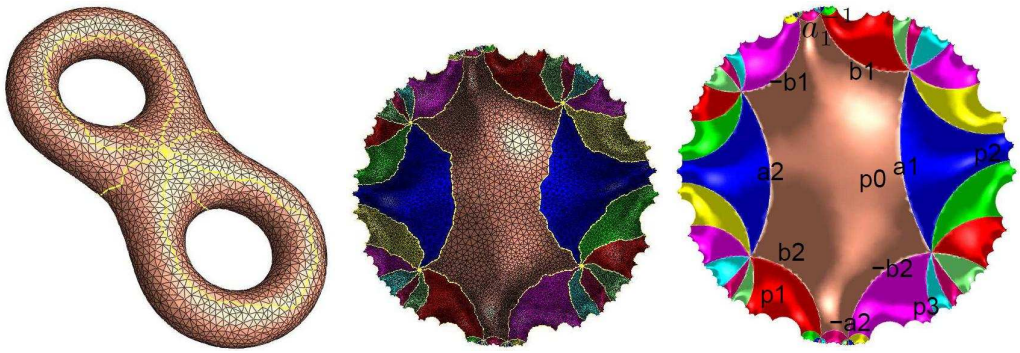

Figure 2.2: (1): a 2-torus. (2)embedding of 2-torus in th universal covering space.(3) Fundamental domain bounded by geodesic arcs.

mental polygon. This is a key setting in the shortest word problem described in the next section. The follow pictures show an USC of a discrete 2-torus mesh.

2.3.2 Shortest Word Problem for High Genus Surfaces

Shortest Words vis Shortest Loops

The shortest words problem has been studied in several different settings by different approaches. It was essentially solved by Dehn [9][8] using geometric and topological methods. Later Birman and Series [4] revisited Dehns algorithm and used it to solve the recognition of simple loops. Regarding the computational complexity, Parry proved in [26] that for an arbitrary group, checking whether a word is equivalent to the identity is polynomial, but finding the shortest words is NP-hard by a reduction to the Traveling Salesman Problem (TSP). In [11] Epstein et al. proved that it takes polynomial time

to find the length of the shortest word for free groups and hyperbolic groups (even though both grow exponentially); However, they did not provide any result on finding the exact shortest words.

Two loops on a surface are homotopic, if one can be deformed to the other continuously on the surface. Homotopic classes of loops form a group, the so-called fundamental group of the surface, which is one of the most important topological invariants of the surface. Given a loop, finding the representation of its homotopy class in the fundamental group has profound importance in computational topology.

Unfortunately, the word representation for a given loop is not unique, since the generators of the fundamental group are in general with non-trivial relations. Such an ambiguity gives rise to a very challenging problem, that is to find the shortest word representation for a homotopy class in the fundamental group with given generators.

By the arguments in [4], finding the shortest words is relatively easy for open surfaces, genus zero closed surfaces (where the fundamental group is trivial) or genus one closed surfaces (where the fundamental group is Abelian). In this section, our attention is focused on a more challenging case, which is for closed surfaces with genus greater than one. The problem we are solving can be specified as:

Problem (The Shortest Word Problem). Suppose S is a genus $g > 1$ closed surface equipped with a hyperbolic metric g and a base point p. A set of base loops (i.e. the generators of the fundamental group $\pi_1(S, p)$) through point p

is given. For an arbitrary loop $\gamma \subset S$, find its shortest word representation in $\pi_1(S, p)$ using the alphabet consisting of those base loops and their inverses.

There is a very simple intuition behind our approach: reducing the (combinatorially) shortest words problem to finding the (geometrically) shortest homotopic loops through the base point.

Definition 2.9 (Shortest Loop through a Base Point)**.** *Suppose S is a surface with a Riemannian metric G and a base point p. $\gamma \subset S$ is a loop on S passing through p. We say γ is a shortest loop through base point p if its geodesic curvature is zero everywhere except for at the base point p.*

The reduction is justified by the following claim, which is a direct corollary of the lemmas and theorems provided in [4]:

Theorem 2.6 (Shortest Words vis Shortest Loops)**.** *Let S be a high genus closed surface with a base point p equipped with a hyperbolic metric g. Suppose a special set of fundamental group generators are given, which are all shortest loops through p and disjoint to one another anywhere else. Then for a loop γ, its word representation with respect to the given generators is the shortest word in $\pi_1(S, p)$ if γ is a shortest loop through p.*

By the reduction from the above theorem, we need two major geometric processes to solve the problem. The first one is to deform the initial Riemannian metric to a hyperbolic metric; that is to compute the uniformization metric. This is achieved by using discrete hyperbolic Ricci flow described in

the previous section. It is used here to achieve the hyperbolic background geometry required by the above theorem.

The second geometric process is to deform a loop (either a generator or the target loop) to be the shortest one through the same base point and homotopic to the original one. This process requires the construction of the Universal Covering Space. The basic idea here is to lift the cycle on the original surface to an open path in the UCS, so that the shortest loop problem is transformed to the shortest path problem.

Given a triangular mesh M with a base vertex p, a set of fundamental group generators $a_1, b_1, a_2, b_2, \ldots, a_g, b_g$ intersecting only at p, a loop γ through p. Then the majored steps to compute the shortest word of γ are as follows:

(1): Compute the hyperbolic metric using Ricci flow;

(2): Compute the shortest loop for each base generator;

(3): Compute the shortest loop for the loop γ;

(4): Compose the shortest word.

Remark 2.1. *Regarding computing shortest loops on surfaces for step 2 and 3, there have been several remarkable works in the literature. To name a few of them, Erickson and Whittlesey [12] gave a very fast greedy algorithm to compute the shortest system of loops relaxing the homotopy condition. Dey et al. [10] proposed an algorithm to compute tunnel and handle base loops that are both topologically correct but also geometrically relevant. Yin et al. [30]*

developed an algorithm to compute the shortest loop within a given homotopy class utilizing UCS; they also provided a space efficient data structure to handle the otherwise exponentially growing UCS.

For the last step, suppose γ is a shortest loop though the base point , so are all the fundamental group generators
$\{a_1, \ldots, a_g, b_1, \ldots, b_g\}$. Then we can trace γ on the surface as the following. If γ crosses $a_i(a_i^{-1})$, then we add a letter $b_i(b_i^{-1})$ to the word, if γ crosses $b_i(b_i^{-1})$, we add $a_i(a_i^{-1})$ to the word. Some special cases can be handled:(1) the geodesic passes through a vertex which is the intersection of several generators.(2)the geodesic contains a boundary edge of the fundamental domain.

After computing the hyperbolic metric, we divide the problem into three sub-problems

- The Shortest Path Problem: Given a path on a triangulated surface S equipped with a hyperbolic metric, how to compute the shortest path that can be homotopically deformed to the original one with both end points fixed? As a special case, what if the end points of the path coincide and form a loop based at a fixed point?

- The Shortest Loop Problem: Given a loop on a triangulated surface S equipped with a hyperbolic metric, how to compute the shortest loop (under hyperbolic metric) that is homotopic to the original one?

- The Shortest Word Problem: On a triangulated surface S equipped

with a hyperbolic metric and a canonical basis of the fundamental group, how to compute the shortest word representation for an arbitrary loop on S?

Algorithms

Here we give the algorithms to solve the three sub-problems, more detail analysis can be founded in [31].

Computing the shortest path is a basic procedure that is used extensively in later computations. Here we present a simple yet efficient algorithm to shrink a path homotopically. The input to the algorithm is a path $\gamma = p_1 p_2 \cdots p_n$ on S, where p_i is either a vertex $v_j \in V$ or on an edge $e_k \in E$. The output is a new path $\gamma_s = q_1 q_2 \cdots q_m$ that is homotopic to γ and is shortest between p_1 and p_n ($q_1 = p_1$ and $q_m = p_n$).

The algorithm works on both the surface S and its universal covering space $\bar{S}$. It follows a common philosophy in topology: lift the given path $\gamma \subset S$ to $\bar{\gamma} \subset \bar{S}$ (step 1 to step 3) and shrink it in $\bar{S}$ (step 4 to step 7). But the novelty of this algorithm is that it uses a transient embedding scheme for both the lifting and shrinking process. As a consequence only the one-ring neighbors of γ and $\bar{\gamma}$ need to be embedded in $\bar{S}$. The fundamental domain and its copies are conceptually used but not actually embedded in this work.

Algorithm 2. *The Shortest Path Algorithm*

1. Embed one-ring neighbor $N(p_1)$ into $\bar{S}$;

2. Repeat the following for each remaining point p_i on the path:

   (a) Address its one-ring neighbor $N(p_i) = f_{i,1}, , f_{i,d_i}$.

   (b) For each $\bar{f_j} \in N(\bar{p}_i)$, ignore it if it is embedded in the previous iteration or embed it otherwise.

3. Record the coordinates of $\bar{p}_1$ and $\bar{p}_n$.

4. Apply a rigid motion to $\bar{S}$ to align $\bar{p}_1$ with the origin and $\bar{p}_n$ into the X-axis; rename them as $\bar{q}_1$ and $\bar{q}_n$.

5. Re-embed $N(\bar{q}_1)$.

6. Trace out a ray $\bar{q}_1 \cdots \bar{q}_m$ along the X-axis from $\bar{q}_1$ (i.e. the origin) towards $\bar{q}_n$; do the following for each $\bar{q}_j$:

   (a) Embed the un-embedded part of the one-ring neighbor $N(\bar{q}_j)$ around $\bar{q}_j$;

   (b) Advance the ray from $\bar{q}_j$ along X-axis until it hits a point $\bar{q}_{j+1}$ on the boundary of $N(\bar{q}_j)$;

7. Denote the trace ray as $\bar{\gamma}_s = \bar{q}_1 \cdots \bar{q}_m$. Project it back onto S as $\gamma_s = q_1 q_2 \cdots q_m$.

There is a fundamental difference between our algorithm and other existing ones. In a previously published algorithm, the "embedded" marker is kept permanently and globally in a fundamental domain; once a face is marked, it will remain marked until the whole fundamental domain is embedded. On the other hand, our algorithm only keeps this marker transiently and locally. It means that an "embedded" marker is only kept for two consecutive iterations and only within a neighborhood of up to two points. Once the iteration for p_i is done, the markers for faces in $N(p_{i-1}) - N(p_i)$ (i.e. those exclusively embedded in the iteration for p_{i-1}) will be cleared, and these faces will be treated as "un-embedded" and therefore become immediately available to be re-embedded.

We name such a scheme **transient embedding**. When a path is lifted from S into $\bar{S}$, only its one-ring neighbor needs to be embedded in $\bar{S}$. Therefore the algorithm is linear in both time and space complexity with respect to the length of the given path. It avoids embedding one or more copies of the fundamental domain that are traversed by the path being lifted.

The above shortest path algorithm can be used to shrink a loop, but has to fix one point in the loop. In many cases, such as the shortest word computation, we need a shortest loop without any point fixed. Here we propose a shortest loop algorithm that uses the shortest path algorithm iteratively. The input to this algorithm is a general loop $\gamma = p_1 p_2 \cdots p_n p_1$ on the original surface. The output is a loop $\gamma_S = q_1 q_2 \cdots q_n q_1$ that is homotopic to γ and is shortest under the hyperbolic metric associated with S.

Algorithm 3. *The Shortest Loop Algorithm*

1. Let $p = p_1$ be the initial fixed point.
2. Repeat the following on γ until its length becomes stable:
   (a) Shrink γ keeping p fixed using the shortest path algorithm;
   (b) Check the length of γ (under hyperbolic metric), and stop the iteration if its difference to the previous iteration falls below a threshold;
   (c) Set p to a new fixed point on γ and go to the next iteration;
3. Set $\gamma_S = \gamma$ and exit.

In this algorithm, we construct a sequence of shortest loops with fixed points to approach the shortest loop without fixed points. In each iteration, we shrink γ with fixed point p; pick a new point $p' \neq p$ to serve as the fixed point and let p be free to deform. This process is actually a variation of the midpoint shortening process in [3], which has been proved in [16] to converge to the unique shortest loop under hyperbolic metrics and will not get stuck in any local minimum.

Now we can use the following algorithm to find the shortest word representation for the given loop under this basis. The input to the algorithm

is a a set of shortest loops $\mathbf{B} = \{a_1, b_1, \cdots, a_g, b_g\}$ that only intersect at a base point p, and another shortest loop $\gamma_S = p_1 p_2 \cdots p_n p_1$. The output is a shortest word representation of γ. Note that all the loops are directed ones on S, and the algorithm works completely on S and does not need $\bar{S}$ any more.

Algorithm 4. *The Shortest Word Algorithm*

1. Let $\mathbf{W}$ be the word representation of γ_S, initialized as empty.
2. Go through γ_S to find all its intersections with loops in $\mathbf{B}$; save these intersection points in order as $\mathbf{S} = \{s_1, s_2, \cdots, s_k\}$.
3. For each intersection $s_i \in \mathbf{S}$ do the following:
   (a) If $s_i \neq p$, append a corresponding letter to $\mathbf{W}$ (explained in the remark 2.1).
   (b) Otherwise, perturb γ_S to go around p from the side with less (or equal) base loops, and append corresponding letters $\mathbf{W}$ in the order of intersections.
4. Output word $\mathbf{W}$.

The following is one example of the shortest word[31]. Our algorithms

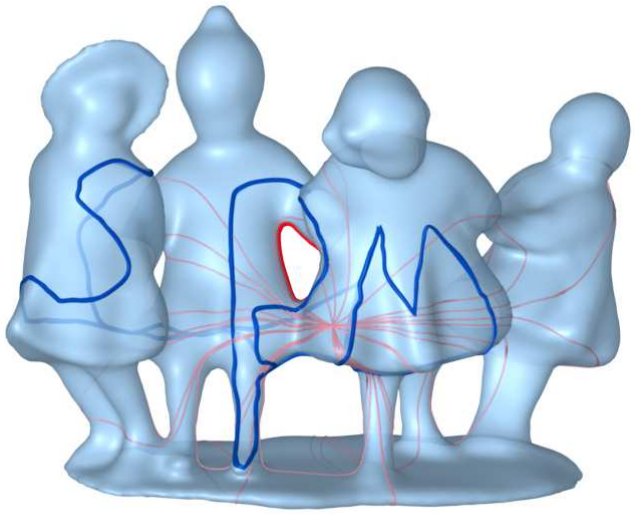

Figure 2.3: Shortest word (red) for a general loop (blue) on a genus-8 hyperbolic surface with a canonical basis of 16 shortest loops (pink).

work with edge lengths of triangulated surfaces and are therefore subject to numerical errors, which is unavoidable for any numerical method of the same nature. For example, the given hyperbolic metric may not be a hundred percent accurate, the floating point representations with limited bits will cause accumulated truncation errors, etc. However, it can be shown that our shortest word computation is robust under numerical inaccuracy. This is partly due to the fact that although we use geometric information in the algorithm, the problem itself is a topological one. On the other hand, in the computation of shortest paths and shortest words, we employ a couple of techniques to relieve the numerical errors.

Although only high genus closed surfaces are considered in this work, the

algorithms proposed here can be potentially extended to cover other surfaces (e.g. with boundaries) that also admit hyperbolic metrics, which will be an interesting topic to explore in the future.

Chapter 3

Delaunay Refinement Algorithms on the Hyperbolic Plane

Delaunay refinement algorithms take a planar straight-line graph and returns a conforming Delaunay triangulation of only quality triangles by inserting Stiener points. In this chapter, we will introduce two generalized Delaunay refinement algorithms on surfaces embedded in the hyperbolic Poincaré disk model.

3.1 Hyperbolic Space

In this section, we will given a brief introduction of the hyperbolic Poincaré disk model for the hyperbolic plane. One can refer to the book written by Luo, Gu and Dai [21] for more details about hyperbolic space models and hyperbolic triangles.

Hyperbolic n-space H_n is the maximally symmetric, simply connected, n-dimensional Riemannian manifold with constant sectional curvature -1. There are several models which realize H_n: the hyperboloid model, the Klein model, the Poincaré disk model and the Poincaré half-plane model. Here we introduce the Poincaré disk model in two dimension, which is the unit disk $\mathbb{D} = \{z = x + yi | x^2 + y^2 < 1\}$ with the metric

$$ds^2 = \frac{4dz \wedge d\bar{z}}{(1 - |z|^2)^2}.$$

The geodesics in the Poincaré disk are the diameters or the arcs of circles which are perpendicular to the unit circle $\mathbb{S}^1 = \partial\mathbb{D}$.

Lemma 3.1. *The geodesic distance between any two points z and w in the Poincaré disk is given by*

$$d(z, w) = 2arctanh\frac{|z - w|}{|z^* w - 1|}. \tag{3.1}$$

Lemma 3.2. *The rigid motions of the Poincaré disk model are Möbius trans-*

formations

$$\phi(z) = e^{i\theta}\frac{z-a}{1-a^*z}, \qquad |a|<1,$$

and all the Möbius transformations are smooth angle preserving mappings.

Each Möbius transformation is composed of rotation($z \to e^{i\theta}z$), translation ($z \to z+a$) and inversion ($z \to 1/z$). Using Möbius transformations, we can always map two crossing geodesics to two diameters of the unit circle, and the intersection angle does not change.

Definition 3.1. *A hyperbolic circle $C(z_0, r)$ with center z_0 and radius r is defined to be*

$$C(z_0, r) \triangleq \{z | d(z, z_0) = r\}.$$

A hyperbolic circle in the Poincaré disk is also a Euclidean circle with different center and radius.

Lemma 3.3. *(1)A hyperbolic circle $C(z_0, r)$ is also a Euclidean circle $\tilde{C}(\tilde{z}_0, \tilde{r})$ with*

$$\tilde{z}_0 = \frac{1-\mu^2}{1-\mu^2\bar{z}_0 z_0}z_0, \quad \tilde{r}^2 = |\tilde{z}_0|^2 - \frac{z_0\bar{z}_0-\mu^2}{1-\mu^2\bar{z}_0 z_0}, \tag{3.2}$$

where $\mu = \tanh\frac{r}{2}$.

(2)Conversely, for a Euclidean circle $\tilde{C}(\tilde{z}_0, \tilde{r})$ contained in the unit circle, it is also a hyperbolic circle $C(z_0, r)$ with

$$r = \frac{1}{2}\ln\frac{(1+|\tilde{z}_0|+r)(1-|\tilde{z}_0|+r)}{(1-|\tilde{z}_0|-r)(1+|\tilde{z}_0|-r)}, \quad z_0 = \frac{1-\sqrt{K}}{1+\sqrt{K}}\frac{\tilde{z}_0}{|\tilde{z}_0|}, \tag{3.3}$$

where

$$K = \ln \frac{(1+|\tilde{z}_0|+r)(1+|\tilde{z}_0|-r)}{(1-|\tilde{z}_0|-r)(1-|\tilde{z}_0|+r)}.$$

Proof. (1)Using Möbius transformation $\omega = \frac{z-z_0}{1-\bar{z}_0 z}$, one can map the hyperbolic circle $C(z_0, r)$ to hyperbolic circle $C(0, r)$ isometrically. $C(0, r)$ is also a Euclidean circle $\tilde{C}(0, \tanh\frac{r}{2})$, then we have the following equation.

$$\frac{z-z_0}{1-\bar{z}_0 z}\frac{\bar{z}-\bar{z}_0}{1-z_0\bar{z}} = \omega\bar{\omega} = \tanh^2(r/2) = \mu^2$$
$$\Leftrightarrow |z - \frac{1-\mu^2}{1-\mu^2\bar{z}_0 z_0}z_0|^2 = |\frac{1-\mu^2}{1-\mu^2\bar{z}_0 z_0}z_0|^2 - \frac{z_0\bar{z}_0-\mu^2}{1-\mu^2\bar{z}_0 z_0},$$

which is an equation of a Euclidean circle.

(2)consider the reflection about the diameter L of the unit circle, which passes through the origin and $\tilde{z}_0$, it is a hyperbolic isometry and preserves the circle $C(z_0, r)$, so the center z_0 should be located on L. The intersections of L and $\tilde{C}(\tilde{z}_0, \tilde{r})$ are $(|\tilde{z}_0| \pm \tilde{r})\frac{\tilde{z}_0}{|\tilde{z}_0|}$, so we have

$$r = \frac{1}{2}d((|\tilde{z}_0|+\tilde{r})\frac{\tilde{z}_0}{|\tilde{z}_0|}, (|\tilde{z}_0|-\tilde{r})\frac{\tilde{z}_0}{|\tilde{z}_0|}),$$

which gives the formula for r in (3.3). Now assume $z_0 = \alpha\frac{\tilde{z}_0}{|\tilde{z}_0|}$, slove the equation

$$d(z_0, (|\tilde{z}_0|+\tilde{r})\frac{\tilde{z}_0}{|\tilde{z}_0|}) = d(z_0, (|\tilde{z}_0|-\tilde{r})\frac{\tilde{z}_0}{|\tilde{z}_0|}),$$

we have $\alpha = \frac{1-\sqrt{K}}{1-\sqrt{K}}$. □

3.2 Delaunay Triangulation on the Poincaré Disk

3.2.1 Hyperbolic Triangle

In the following, we will introduce some facts and lemmas about hyperbolic triangles which are useful in the following sections.

A Hyperbolic triangle is a triangle with geodesic arcs between any two points in the Poincaré disk model $\mathbb{D} = \{z = x + yi | x^2 + y^2 < 1\}$ of 2-dimensional hyperbolic space.

For Euclidean triangles, the cosine law and sine law give relations between the edge lengths and the angles. There are also similar relations for hyperbolic triangles.

Lemma 3.4. *For a hyperbolic triangle with the edge length a, b, c, and α ,β, γ are the angles which are opposite to the corresponding edges, we have:*

(1)Hyperbolic cosine law

$$\cosh c = \cosh a \cosh b - \sinh a \sinh b \cos \gamma, \tag{3.4}$$

and its dual

$$\cos \gamma = \cos \alpha \cos \beta - \sin \alpha \sin \beta \cosh c. \tag{3.5}$$

(2)Hyperbolic sine law

$$\frac{Sinh(a)}{Sin(\alpha)} = \frac{Sinh(b)}{Sin(\beta)} = \frac{Sinh(c)}{Sin(\gamma)}. \tag{3.6}$$

(3)Formula of area

$$Area(T) = \alpha + \beta + \gamma. \tag{3.7}$$

The area of a hyperbolic triangle is no great than π*, and* $\delta = \pi - (\alpha + \beta + \gamma) > 0$ *is called the defect of a hyperbolic triangle.*

Using above hyperbolic cosine law, we can easily verify the following facts.

Fact 3.1. *For any hyperbolic triangle* $\triangle ABC$*, if* D *a is point located on the geodesic edge* BC*, then either* $d(A, B)$ *or* $d(A, C)$ *is greater than* $d(A, D)$*. (Figure 3.2.1)*

Fact 3.2. *As shown in Figure 3.2, the central angle* α^* *is always no less than* 2α*, where* α *is the inscribed angle. Fix the angle* α^**, the minimal of* α *is obtained when* ΔABC *is an isosceles triangle, and* 2α *will go to* α^* *when the point* A *approaches point* B *or point* C*.*

Through above lemmas and facts, the minimal angle of a bad shape hyperbolic triangle ΔABC is not easy to be defined as in Chew's paper. The minimal angle is correlated with the user inputs, and the position of A with edge length a fixed. In a hyperbolic circle with diameter d and with a geodesic

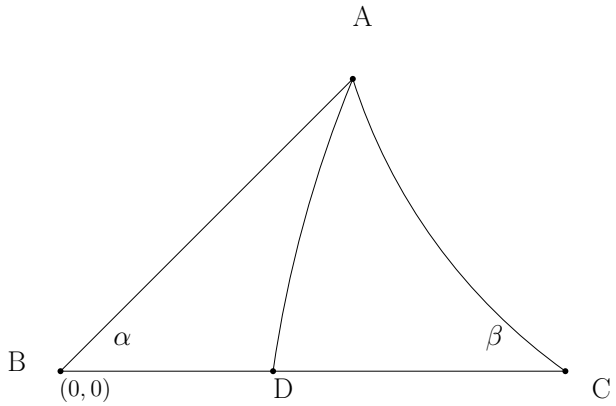

Figure 3.1: Hyperbolic Triangle

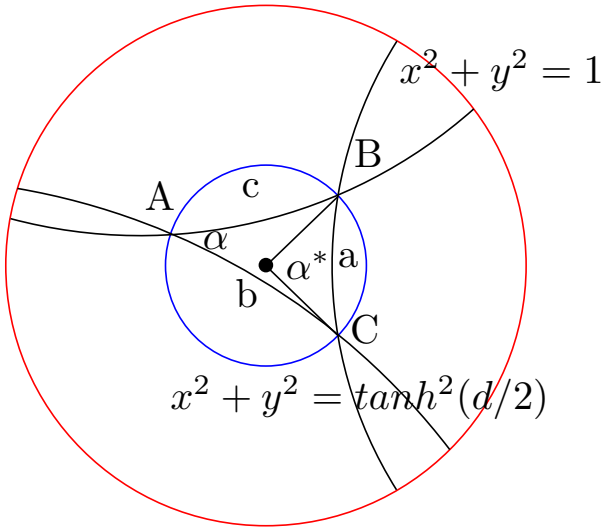

Figure 3.2: Central angle and inscribed angle in a hyperbolic circle.

chord whose edge length a is fixed, although the inscribed angle varies when the position of A changes, but the central angle α^* is always fixed. Then we can define what is a well shape hyperbolic triangle.

Definition 3.2 (Well shape triangle)**.** *A triangle T with circumradius h in a triangulation is said to be well shape if*

(1)the central angle corresponding to the minimal edge is more than $\theta > 0$;

(2) and circumradius h of T is less than $\varepsilon > 0$ which is defined by the user.

We have two approaches to define the well shape hyperbolic triangle, one is to give fixed minimal central angle for all triangles, the other is to let the minimal central angle correlated to the triangle's circumradius. We use the first one for generalized Ruppert's algorithm, and the later one in generalized Chew's algorithm by defining the minimal central angle to be $\theta^*(h)$, which is the angle such that the opposite chord have the edge length h in a h-circle in the Poincaré disk. The formula for $\theta^*(h)$ is given by

$$\theta^*(h) \triangleq \arccos \frac{\cosh^2(h) - \cosh(h)}{\sinh^2(h)}. \tag{3.8}$$

Remark 3.1. *(1) When h goes to 0, the angle $\theta^*(h)$ will go to $\pi/3$, one will discover the minimal angle of the planar case of Chew's algorithm. When h increases to ∞, the angle $\theta^*(h)$ will decrease to 0(see Figure 3.3).*

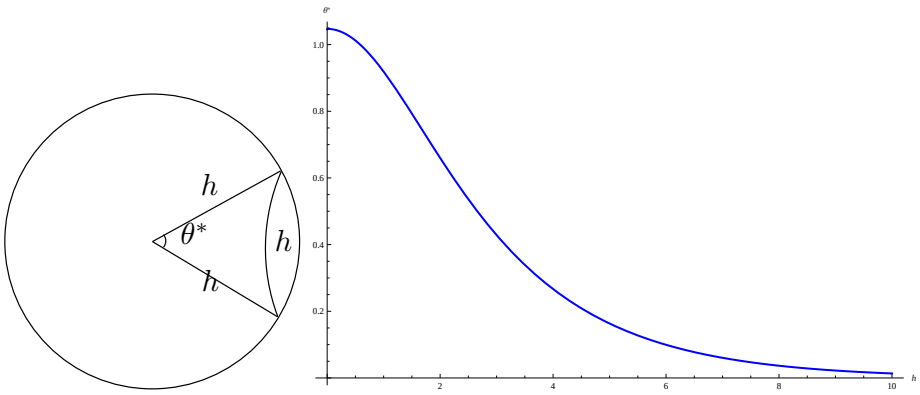

Figure 3.3: Left: An equilateral hyperbolic triangle. Right: The relation between edge length of the equilateral triangle and its angles.

From the cosine law, we have

$$\cos\theta^* = \frac{\cosh^2 h - \cosh h}{\sinh^2 h}.$$

By direct computation, the derivative of θ^* *with respect to* h *is*

$$\frac{d\theta^*}{dh} = \frac{-(\cosh h - 1)^2}{\sinh^3 h \sin\theta^*},$$

which is strictly less than 0 for any $h > 0$.

(2) On a hyperbolic circle with circumradius h *and central angle* $\theta^*(h)$*, if* AB *is the geodesic chord opposite to central angle* $2\theta^*(h)$*, we define the ratio* $\rho(h)$ *as follows,*

$$\rho(h) = \frac{d(A,B)}{h}.$$

$\rho(h)$ *decreases from* $\sqrt{3}$ *to 1 as* h *increases from* 0 *to* ∞*. When* $h \approx 2.3101$*,*

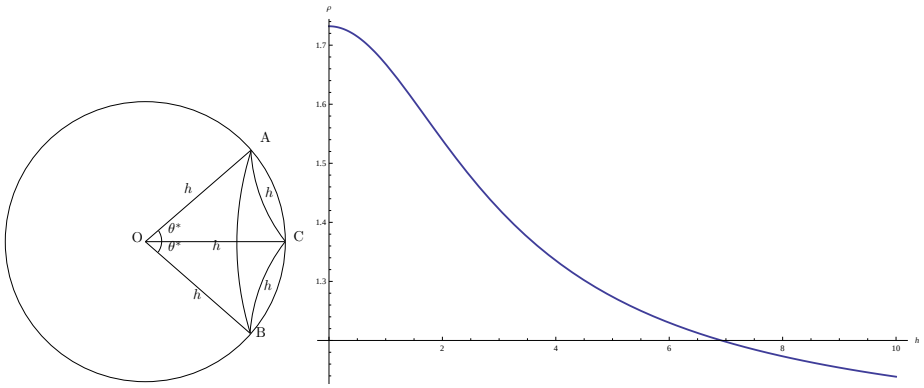

Figure 3.4: Left: An isosceles hyperbolic triangle with top angle $2\theta^*$. Right: the graph of the function $\rho(h)$.

the ratio is about 1.5 *(see Figure 3.4).*

By the hyperbolic sine law, we can get the formula of $d(A, B)$*:*

$$d(A, B) = 2 * \sinh^{-1}(\sin\theta^*(h)\sinh(h)) \tag{3.9}$$

(3)For a hyperbolic triangle T *which can be enclosed by a h-disk in the Poincaré disk, the maximal area of* T *is attained when* T *is an equilateral triangle. When h goes to 0, the area of the equilateral triangle can be estimated as:*

$$A \approx \frac{3\sqrt{3}}{4}h^2 + O(h^4). \tag{3.10}$$

Similar to the definition in Euclidean plane, a hyperbolic Delaunay triangulation for a set P of points on hyperbolic plane is a triangulation DT such that no point in P is inside the hyperbolic circumcicle of any geodesic triangle in DT. A constrained Delaunay triangulation is a generalization

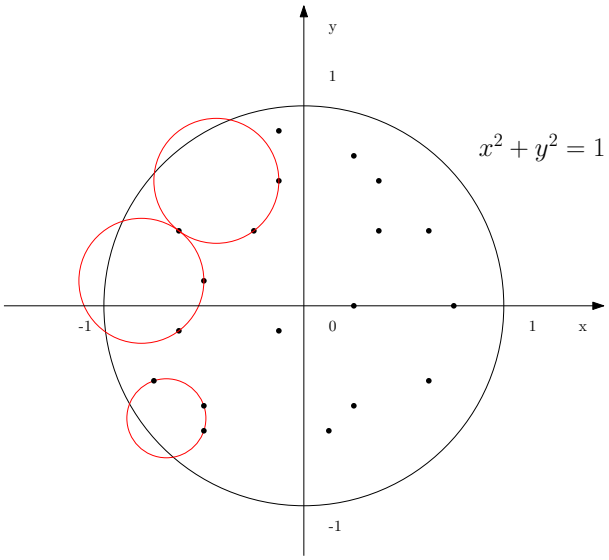

Figure 3.5: An example in which some of the Delaunay circles have infinite radii.

of the Delaunay triangulation that force certain required segments into the triangulation.

In the Poincaré disk model, a hyperbolic circle is also a Euclidean circle but has different circumcenter and radius. So we can use a Euclidean circle to check the Delaunay edges of a point set in the Poincaré disk. Generally, for a point set on the Poincaré disk, some of the Delaunay circles maybe touch the boundary of the Poincaré disk, which means that the circumcircles have infinite radii under the hyperbolic metric.(see Figure 3.5)

While for any point set on a surface mesh embedded in the Poincaré disk, the above case can not happen, because there are infinite many copies of the surface mesh on the disk, if there exists a Delaunay circle which touch

the boundary of the unit disk, then the Delaunay circle should contain some copies of the surface mesh, which contradicts the property of a Delaunay circle. But for a surface mesh embedded in the Poincaré disk, two vertices in the same equivalence class $[p]$ may be on a Delaunay circle, which means there will be a closed Delaunay edge on the surface mesh at p. To prevent a closed Delaunay edge at a vertex, we assume that all the Delaunay circles can not contain more than one point in the same equivalent class. This can be attained by careful selection of sample points on the surface.

3.2.2 Preliminary

In the following two sections, we will give two generalized Delaunay refinement algorithms on surfaces embedded in the hyperbolic Poincaré disk. For those two generalized algorithms on Poincaré disk, the input will be $G(V, E)$, which is a geodesic graph on a surface embedded in the hyperbolic Poincaré disk, and the geodesic edges in E only intersect at the vertices in V. If the surface has been embedded in the Poincaré disk, $G(V, E)$ is a kind of 'Planar' geodesic graph. For close surfaces, $G(V, E)$ may be a point cloud on the surface. While for the surfaces with boundaries, the boundary components may be divided into piece-wise geodesic edges in E. There will be infinitely many copies of the surface in the Poincaré, any two copies only differ by a Möbius transformation.

We are also assuming that the inputs have all the initial angles no less than $\frac{\pi}{2}$. For a angle less than $\frac{\pi}{2}$, we can use the same lopping-off technic to

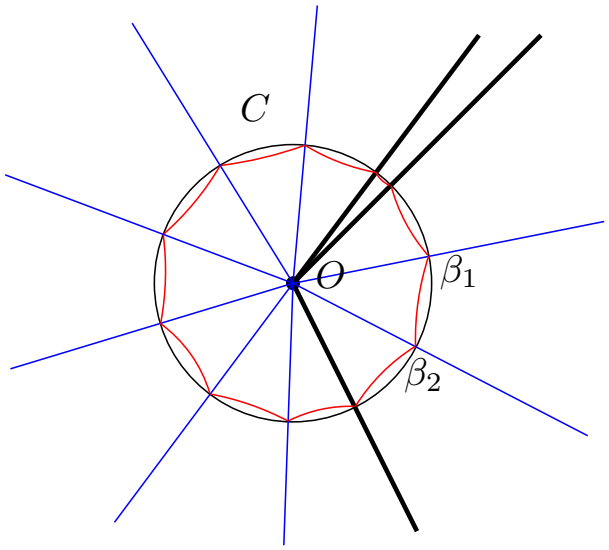

Figure 3.6: Corner Lopping around a vertex with small input angle

enclosed the small input angles. Because when we use geodesic edges, the angles and the geodesic distance are preserving under Möbius transformation, once we can do the corner lopping at origin, the other points are the same. In the Figure 3.6, the radius of the circle C is equal to one third of the local feature size of origin O, and all the new corner angles like β_1, β_2 are larger than $\pi/2$. One can do special surgery inside the circle C[27]. We will give the definition of local feature size function in section 3.4.

Since we are working on a surface embedded in the Poincaré disk with infinite copies, when do deletion, splitting and insertion, we actually do the operations in the whole equivalent class. Before giving the algorithms, we firstly give the following lemma.

Lemma 3.5. *Suppose $T = \triangle ABC$ is a hyperbolic triangle in a Delaunay*

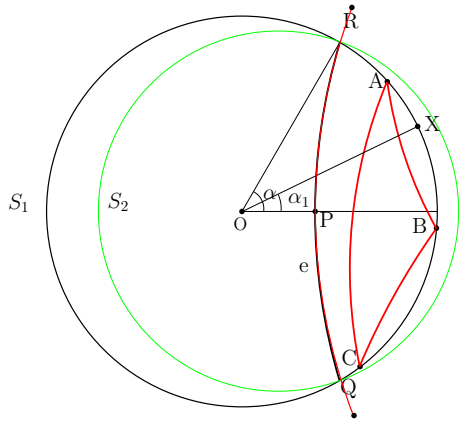

Figure 3.7: An example about encroachment

Triangulation on the Poincaré disk, and T is on one side of an edge e, while its circumcenter O is on the other side of e, then T is totally enclosed in the hyperbolic circle with diameter e and we call the circumcenter O encroaches the geodesic edge e.

Proof. Suppose the circumcircle of T is $S_1(O, h)$ with center at the origin O. By the property of the Delaunay triangulation, the endpoints of e should be outside or on the circle S_1. Suppose e intersects S_1 at R and Q, and $S_2(P, r)$ is the hyperbolic circle with diameter QR and center at point P, we want to show that S_2 encloses the triangle T, in other words, the hyperbolic distance between P and any point $X \in \overset{\frown}{RAQ}$ is less than $d(P, R)$.

Using hyperbolic cosine law, we can compute the hyperbolic distance:

$$r = d(P, R) = arccosh(cosh(h)cosh(r) - sinh(h)sinh(r)cos(\alpha)),$$

$$d(P,X) = arccosh(cosh(h)cosh(r) - sinh(h)sinh(r)cos(\alpha_1)),$$

where $cos(\alpha_1) > cos(\alpha)$. Because the function $y = arccosh(x)$ is an increasing function when $x > 0$, then $d(P,X) < r$. So, the triangle T is enclosed in the hyperbolic diametric circle of RQ, and hence it is enclosed in the hyperbolic diametric circle of e. □

Remark 3.2. *In the above proof, we see that $d(P,X)$ obtains minimum when $\alpha_1 = 0$, which means X is located on the geodesic passing through OP, and $d(P,X)$ obtains maximum when $X = R$ or $X = Q$.*

3.3 Generalized Chew's Second Algorithm

Chew's second algorithm starts with a constrained Delaunay triangulation of the input vertices on the plane. A poor-quality triangle is defined to be a triangle which has angle less than 30° or has circumradius bigger than user-defined size. At each step, the circumcenter of a poor-quality triangle is inserted into the triangulation except that the circumcenter lies on the opposite side of an input segment as the poor-quality triangle, in which case the midpoint of the segment will be inserted and any previously inserted circumcenters inside the diametral circle of the segment will be removed. Chew prove his algorithm terminates and can be generalized to curve surfaces.

We will give the generalized Chew's second algorithm on the surfaces embedded in the Poincaré disk and give a proof about termination in this section. Before giving the detail algorithm, we define a quantity h^* by the

minimal of the following[Chew]:(1)the smallest line-of-sight distance between two nonintersecting sources(vertices or edges) in the initial set of sources, (2)one half the length of the smallest source edge that appears in the initial set of sources, and (3)the value $\varepsilon > 0$ associated with the user-defined size grading function.

Algorithm 5. *Algorithm: Generalized of Chew's second Delaunay refinement algorithm on surfaces embedded in the Poincaré disk:*

- *Input: 1. A geodesic graph $G(V, E)$ on a surface embedded in the Poincaré disk; 2. User-defined size grading $\varepsilon > 0$.*
- *Output: A well shape constrained Delaunay triangulation of G with minimal central angle $\theta^*(\varepsilon)$.*

1. *Compute constrained Delaunay triangulation DT of G.*
2. *While DT contains poor triangle, select the one T with the largest circumradius do*
3. *if the circumcenter of T encroaches an input edge S*
    - *if the edge length is between $2\rho(h^*) * h^*$ and $4h^*$,*
      split S by one-third split by adding their endpoints into G
      remove circumcenters within the distance one-third of the edge length of S from a new vertex

- *else split S by adding its midpoint to G*

  remove circumcenters within the diametric circle of S

update the CDT of G

4. *else*

   Insert the circumcenter of T into G and update the CDT of G

5. *end if*

6. *end while*

Remark 3.3. *In Chew's original algorithm, the criteria of the minimal angle are fixed to be $\pi/3$ for Euclidean Triangles, while in the above generalized algorithm for hyperbolic triangles, the criteria of the minimal angle are not fixed, it depends on the circumradius h of the hyperbolic triangle which contains the angle.*

We will give the proof of termination of the algorithm in the following, the idea is to use induction which is similar to the original proof given by Chew. The proof is more complicated on the Poincaré disk than the Euclidean case. Before giving the main theorem, we need the following lemma.

Lemma 3.6. *Suppose C_1 is a hyperbolic circle centered at origin and with radius $\rho(h)h/2$, and C_2 and C_3 are two hyperbolic circles centered at the points $(\pm\rho(h)h/2, 0)$ with radius h, then we have the following:*

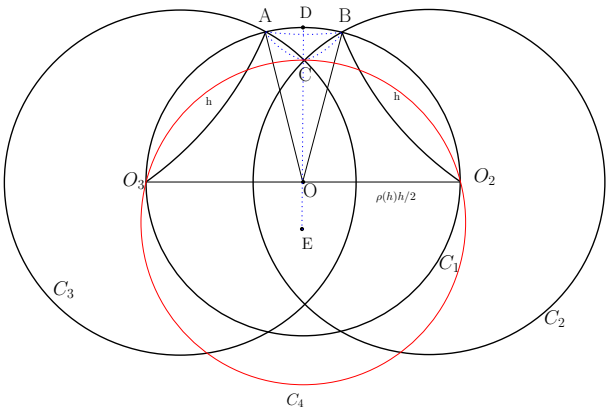

Figure 3.8: Hyperbolic circles described in Lemma 3.6

1. *The diameter of the connected components of* $D_1 \setminus \{D_2 \cup D_3\}$ *is no greater than* h*, where* D_i *is the hyperbolic disk enclosed by circle* $C_i (i = 1, 2, 3)$*.*

2. *If* T *is a hyperbolic triangle with two vertices to be the endpoints of* e*, and the other vertex located in the region* $D_1 \setminus \{D_2 \cup D_3\}$*, then the circumradius of* T *must be no greater than* h*.*

Proof. 1. At first we want to show that the geodesic distance between A and B is less than h when $h > 0$. It is equivalent to show that the angles $\angle BOO_2 = \angle BOO_3$ are great than $\pi/3$ alternatively.

By hyperbolic Cosine law, we have

$$\begin{aligned}\cos\angle AOO_3 &= \frac{\cosh^2(\rho(h)h/2)-\cosh(h)}{\sinh^2(\rho(h)h/2)}\\ &= \frac{1+\sinh^2(\rho(h)h/2)-\cosh(h)}{\sinh^2(\rho(h)h/2)},\end{aligned}$$

then

$$\begin{aligned}&\cos\angle AOO_3 < \frac{1}{2}\\ &\Longleftrightarrow 1+\sinh^2(\rho(h)h/2)-\cosh(h) < \frac{1}{2}\sinh^2(\rho(h)h/2)\\ &\Longleftrightarrow 2(\cosh(h)-1) > \sin^2\theta^*\sinh^2(h) \qquad \text{(by 3.9)}\\ &\Longleftrightarrow 2(\cosh(h)-1) > (1-(\frac{\cosh(h)(\cosh(h)-1)}{\sinh^2(h)})^2)\sinh^2(h)\\ &\Longleftrightarrow 2 > \cosh(h)+1-\frac{\cosh^2(h)}{\sinh^2(h)}(\cosh(h)-1)\\ &\Longleftrightarrow \sinh^2(h) > \cosh(h)\sinh^2(h)-\cosh^2(h)(\cosh(h)-1)\\ &\Longleftrightarrow \cosh^2(h)-1 > \cosh(h)(\cosh^2(h)-1)-\cosh^2(h)(\cosh(h)-1)\\ &\Longleftrightarrow \cosh(h) > 1.\end{aligned}$$

So the angle $\angle AOO_3$ is greater than $\pi/3$, which means $d(A,B)\leq h$.

It's obviously that the following inequalities holds:

$$d(A,B)/2 < d(A,C) = d(B,C) < \rho(h)h/2 < h;$$

$$d(C, D) < d(A, C) = d(B, C).$$

Then for any two points S, T, where S is on the circular arc $\widehat{AB}$ and T is on the circular arc $\widehat{BC}$, extend the geodesic edge ST to SR, R is the point located on the geodesic passing through B and C. By fact 1, either $d(S,T) < d(S,R) < d(S,C) < d(A,C)$ or $d(S,T) < d(S,R) < d(S,B) < d(A,B)$. Then the diameter of the region bounded by the circular arcs $\widehat{AB}$, $\widehat{BC}$ and $\widehat{CA}$ is less than $h > 0$.

2. Now we need to show that the circumradius of T is no greater than h. Observe that, the circumcenter c of T must be on the geodesic passing through O and C, also c is below the geodesic O_2O_3. As a circle passing through O_2 and O_3 shifts up from C_4 to C_1 along the geodesic OD, the whole upper component of $D_1 \setminus \{D_2 \cup D_3\}$ will be swept by the circle. Then the circumcenter c is located on the geodesic edge OE, where E is the circumcenter of the circle C_4 which had circumradius h, then we have $\rho(h) * h/2 < d(c, O_2) < h$.

□

Now we give the main theorem about the termination of the algorithm.

Theorem 3.1. *The algorithm terminates when* $h^* < 2.3101$.

Proof. we want to use induction to show that the algorithm produce a hyperbolic CDT such that any two vertices in the mesh have hyperbolic distance no less than h^*.

Initially, the above statement holds. We assume the statement holds at cycle i, then at cycle $i+1$, new vertices are added if the following happens:

- the circumcenter of a failing triangle T_1 is added into the CDT. Then the nearest neighbor of the circumcenter must be the vertices of T_1. T_1 failed for the following reasons:

  1. T_1 have an edge with the central angle less than $\theta^*(r)$, where r is the circumradius of T_1. By assumption of induction, all the edge lengths of T_1 are no less than h^*, then r must be no less than h^* .
  2. T_1 is not well-sized. To produce an edge of length less than h^*, the circumradius r must be less than h^*, thus T_1 will be well-sized triangle.

- new vertices are added during the splitting of a source edge into either 2 or 3 pieces. Suppose that the edge to be split is e and have edge length d, and we denote T_2 as the triangle which causes the splitting and has circumcenter on the other side of geodesic edge e. To produce an edge with length less than h^*, d should be between h^* and $2h^*$.

  1. $h^* \leq d < \rho(h^*)h^*$

     In this case, no splitting can occur.

     To see this, by lemma 3.5, the triangle T_2 should be enclosed by the hyperbolic diametric circle of edge e; and also by induction assumption, the vertices of T_2 other than the endpoints of e must

be outside the hyperbolic circles center at the endpoints of e and with radius h^*. But from lemma 3.6, the region is to small to fit two vertices. Then T_2 should be a triangle with e to be one of its edges, and the remaining vertex located in the region, but this kind of triangle T_2 is a well-shape triangle with circumradius less than h^* by the lemma, such a triangle will not cause edge splitting.

2. $\rho(h^*)h^* \leq d \leq 2h^*$

   Similar to Chew's argument, this kind of bad source-edges can never occur. Initially, all the edges have edge length at least $2h^*$ by the definition of h^*. Then a bad edge appears only after an edge splitting, there are two kind of edge splitting:

   - half splitting

     then the edge length of the edge to be split should be between $2\rho(h^*)h^*$ and $4h^*$, but edges with edge length in this range only has one-third splitting, and will produce edges with edge lengths between h^* and $\frac{4}{3}h^*$ when $h > 2.3101$ from the remark in last section.

   - one-third splitting

     for this case, the edge length should be between $3\rho(h^*)h^*$ and $6h^*$, but this kind of edges are never split into thirds.

   so the bad edges never occur.

From above, we know that new source-edges can never have edge length less than h^*, and for all the new vertex produced in splitting, the step 3 of removing the too close vertices ensure that all the other vertices distantiate the new vertex at least h^*. □

For the above generalized Chew's second algorithm on surfaces embedded in the Poincaré disk, the termination is guaranteed when $h^* < 2.3101$ for input mesh. When $h = 2.3101$, the central angle θ^* is about $33.3°$. If h goes to 0, θ^* will go to $60°$.

If originally $h^* \geq 2.3101$, the algorithm may not terminate. While the algorithm does not terminate, the insertion of the cirumcenters or midpoints will occur infinitely times, one may re-define the quantity h^* in the algorithm in some suitable step to let the algorithm terminate.

3.4 Generalized Ruppert's Refinement Algorithm

In this section, we give the generalized Ruppert's Delaunay refinement algorithm on surface S which is embedded in the Poincaré disk. Ruppert's algorithm is an algorithms takes a planar straight-line graph and returns a conforming Delaunay triangulation of only quality triangles. The algorithm begins with a Delaunay triangulation of the input vertices. Then insert the circumcenter of a poor-quality triangle into the triangulation, unless this

circumcenter encroaches some input segment, in this case, the encroached segment is split into half. A triangle is defined to be poor-quality if it has an angle less than some prescribed threshold.

On the surface which is embedded in the hyperbolic Poincaré disk, we are actually dealing with 'Planar' geodesic graph (PGG) in Poincaré disk. Unlike the definition of a skinny triangle in generalized Chew's algorithm, we need another kind of definition of a skinny triangle.

Definition 3.3 (Skinny Triangle)**.** *A Triangle T is said to be skinny if the central angle opposite to the shortest edge of T is less than α in the circumcircle of T*

Also, we maybe still have a user input h, which is the upper bound of circumradius of triangles.

Algorithm 6. *Algorithm: Generalization of Ruppert's Delaunay refinement algorithm on the Poincaré Disk:*

- *Input: 1. A planar geodesic graph $G = (V, E)$ on the surface embedded in the uniformization space($\mathbb{H}^2$), Minimum central angle bound α which is related.*
- *Output: A Delaunay triangulation with minimum central angle α.*

1. *Compute constrained Delaunay triangulation DT of G.*
2. *While DT contains any geodesic edge e which is encroached upon by a vertex, split e into half by adding the midpoint into the vertex set*

3. while DT has a skinny triangle T

- *split the geodesic edges which are encroached upon by the circumcenter of T into half;*
- *otherwise add the circumcenter of T into the vertices list of G*

4. repeat 1 until no edges are encroached upon, and no skinny triangle.

Before showing that the above algorithm terminates under some constraints, we introduce some notations.

Definition 3.4.

$$\rho(r,\theta) \triangleq \frac{\cosh^{-1}(\cosh^2 r - \sinh^2 r \cos\theta)}{r} \qquad (r > 0, \theta \in [0, \pi/2]) \qquad (3.11)$$

Fixed r or θ, the function $\rho(r,\theta)$ is an increasing function of the other variable, this can be verified directly by computing the partial derivatives. When $\theta = \pi/2$, the range of $\rho(r,\theta)$ is $(\sqrt{2}, 2)$.

We define the local feature size in PGG.

Definition 3.5. *Given a PGG S, the local feature size at a point, $lfs_S(p)$(or just $lfs(p)$), is the radius of the smallest geodesic disk centered at p that intersects two non-incident vertices or segments of S.*

Remark 3.4. *Notice that, the surface mesh is embedded on the Poincaré disk, the definition of local feature size could be not well defined. For example, S is a 2-torus, the inputs are v which is a point on S and 8 geodesic edges*

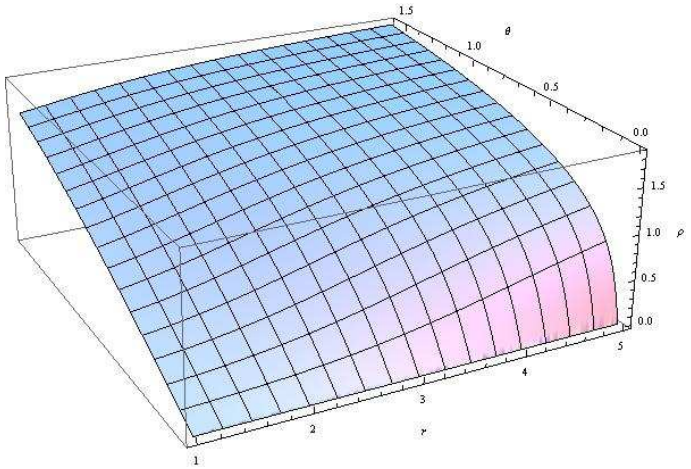

Figure 3.9: The graph of the function $\rho(r, \theta)$

which form a canonical basis of $\pi_1(S, v)$, then there are no disjoint features in this kind of input. So we must require that the input must be complicated enough such that the local feature size is well defined. So we give the following assumptions.

Assumption

- Any features can not connect to itself by another feature;
- Any features can not appear more than once on the boundary of a face.

When the local feature size is well defined, it has the following property.

Lemma 3.7. *Given any PGG S, and any two points p and q in the Poincaré disk,*

$$lfs_S(q) \leq lfs_S(p) + d(p, q).$$

In the following, we denote the distance between p and its nearest vertex by $n(p)$.

Lemma 3.8. *There exists two constants C_1 and C_2 which are both no less than 1, such that the following statements hold:*
(1) If a point p is the circumcenter of a skinny triangle, then $n(p) \geq lfs(p)/C_1$;
(2) If a vertex p is added as the midpoint of a split segment, then $n(p) \geq lfs(p)/C_2$.

Proof. The proof is to use induction. For any initial input vertex p, by the definition of local feature size function, $n(p) \geq lfs(p)$. Assuming the lemma is true for all previous vertices, we consider the vertices added later.

1. p is the circumcenter of skinny triangle $\triangle ABC$, with circumradius r and the smallest central angle opposite to the geodesic edge AB. and WLOG, we assume A was added after B.

    - A was a vertex of the input, then so was B; we have $lfs(a) \leq d(A, B)$.
    - A was added as circumcenter of some triangle with circumradius r', then $r' \leq d(A, B)$. By assumption, we have $lfs(A) \leq r'C_1 \leq d(A, B)C_1$.
    - A was a midpoint of a segment that was split. Also by assumption, $lfs(A) \leq d(A, B)C_2$

Then we have $lfs(A) \leq d(A,B)C_2$ for all subcases above, now we assume the following condition holds:

$$C_2 \geq C_1 \geq 1. \tag{3.12}$$

By the hyperbolic cosine law and previous lemma, we have

$$lfs(p) \leq lfs(A) + d(p,A) \leq d(A,B)C_2 + r = (\rho(r,\theta)C_2 + 1)r.$$

For $\rho(r,\theta)$ is an increasing function fixed either r or θ, then we have

$$r \geq \frac{lfs(p)}{1 + \rho(\bar{r},\alpha)C_2},$$

where $\bar{r}$ is the maximum circumradius in the surface mesh. Then we get the desired bound on r if the following condition holds:

$$C_1 \geq 1 + \rho(\bar{r},\alpha)C_2. \tag{3.13}$$

2. p is added to split a segment s with length $2 * r$. Segment s is split because some vertex or circumcenter A is inside s's diametral circle, which had radius r. There are two cases for A:

   - A lies on some segment t, which can not be incident to s, since we are assuming that all the input angles are at least $\pi/2$. Then by the definition of local feature size, $lfs(p) \leq r$. Assumed the

condition $C_2 \geq 1$, this case is done.

- A is a circumcenter to be added into DT, but rejected because it lay inside the diametral circle of s. Suppose it was the center of circumcenter C with radius r'. By applying this lemma to A, we have $r' \geq lfs(A)/C_1$. Also the endpoints of s must be outside the Delaunay circle C', thus $r' \leq \rho(r, \pi/2)r$. So we have

$$\begin{aligned} lfs(p) \leq lfs(A) + r &\leq r'C_1 + r \\ &\leq (1 + \rho(r, \pi/2)C_1)r \\ &\leq (1 + \rho(\bar{r}, \pi/2)C_1)r. \end{aligned}$$

  Then we get the desired bound on r if the following condition holds:

$$C_2 \geq 1 + \rho(\bar{r}, \pi/2)C_1. \tag{3.14}$$

□

Giving above three inequalities, the feasible region is

$$1 - \rho(r, \theta) * \rho(r, \pi/2) > 0. \tag{3.15}$$

When $\theta \geq 2 * 20.7°$, the above inequality never holds. For any angle in $(0, 2 * 20.7°]$, there exits an upper bound for r such that the above inequality holds, which means when one require bigger minimum angle, the constraint

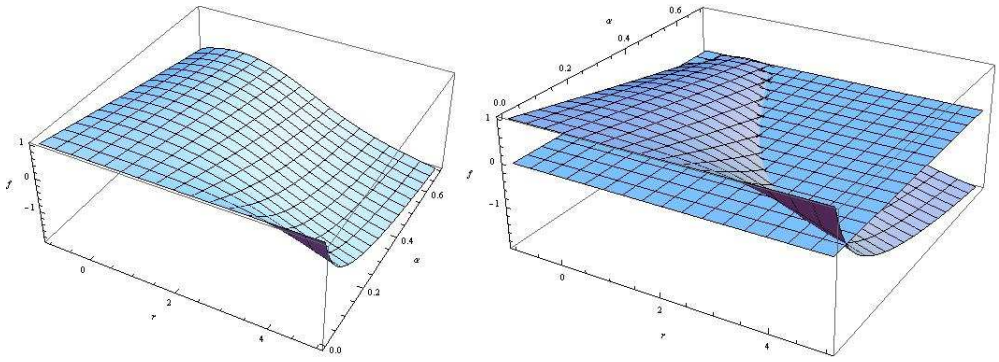

Figure 3.10: Left: the graph of the function $1 - \rho(r, \theta) * \rho(r, \pi/2)$. Right: the feasible region for constraint 3.15.

for maximum circumradius of the triangles in the Delaunay triangulation will be smaller.

Theorem 3.2. *Given a vertex p of the output mesh, $n(p) \geq lfs(p)/(C_2+1)$.*

Proof. Suppose the nearest neighbor of p is q, follow the proof of above lemma, we only need to show the case when q was added after p, then applying the lemma to q, we have

$$d(p,q) \geq \frac{lfs(q)}{C_2} \geq \frac{lfs(p) - d(p,q)}{C_2}$$
$$\Longrightarrow n(p) = d(p,q) \geq lfs(p)/(C_2+1).$$

□

This theorem shows that if the constraint 3.15 is satisfied, then the distance between a inserted vertex and any other vertex will be least $lfs(p)/(C_2+$

1), which means the generalized Ruppert's refinement algorithm on surfaces embedded in Poincaré disk will terminate.

For Ruppert's refinement algorithm, it has quadratic worst-case running time [2]. Suppose we select a portion of the universal covering space which covers the central fundamental domain, and all its direct neighbors, then there are $4g(4g-2)$ fundamental domain. If we choose n samples on the surface, then the complexity of the generalized Ruppert's refinement algorithm still has quadratic worst-case running time.

Chapter 4

conclusion

In this dissertation, we firstly review the discrete surface Ricci flow, which is a powerful tool in computational conformal geometry. As one of the applications of the discrete Ricci flow, we propose a numerical solution to finding the shortest words in the fundamental group of triangulated hyperbolic surfaces that are closed and oriented. The basic philosophy is to use geometric approaches to solve a topological problem. In particular, we develop a transient embedding scheme to compute shortest paths on surfaces, which only requires embedding the one-ring neighbor of the given path in the universal covering space and is therefore linear (with a small coefficient) in the length of the path. Then a variation of the Birkhoff midpoint shortening process is proposed to deform a piecewise geodesic loop (with only one piece actually) iteratively, which is guaranteed to converge to a shortest loop that is the unique global minimum. Finally a shortest word representation of a given

loop is induced from its shortest loop image using a simplified version of the revised Dehn's algorithm. Numerical accuracy is taken into the consideration throughout the whole process. Several techniques are used to relieve numerical errors and increase the robustness. Although only high genus closed surfaces are considered in this work, the algorithms proposed in the shortest words problem can be potentially extended to cover other surfaces (e.g. with boundaries) that also admit hyperbolic metrics, which will be an interesting topic to explore in the future.

In the second part, we give two Delaunay refinement algorithms on surfaces embedded in the hyperbolic Poincaré disk. The proofs of termination of these two algorithms are given under some constraints. The input is on the surface embedded in the Poincaré disk in this dissertation, the proofs also works for general 'Planar' geodesic graphs in the Poincaré disk if the Delaunay circles have finite radii. Since the Poincaré disk model and the upper half plane model for hyperbolic plane are conformal and have similar properties, these two generalized Delaunay refinement algorithms in the Poincaré can be generalized to the upper half plane model, and the proofs are the same. Also there are several interesting directions which may improve the generalized refinement algorithms.

- Comparing to Chew's second algorithm, Ruppert's refinement algorithm is much more easy to implemented, but Chew's second algorithm usually give fewer vertices than Ruppert's method in Euclidean case, this should also be true for generalization algorithms on Poincaré disk.

- In the generalized Chew's second algorithm, the minimal central angle of a 'bad' triangle depends on the circumradius of the triangle itself, we can define in a simpler way: the minimal central angle of a 'bad' triangle is less than $\pi/3$. If we change to this setting, the lemma 3.6 will be trivial, but the proof of the termination will be failed. Although the idea in the proof will failed, when the mesh size are small, it is also a kind of generalization of Chew's second algorithm. There should be similar ways to show the termination of this generalized algorithm on Poincaré disk like [28].

- Also, we use corner lopping for small input angle, there are some other analysis ideas which allow small input angles and have bigger minimal output angle for Chew's second algorithm [1] and Ruppert's refinement algorithm [23]. Chew's second algorithm produces meshes (of domains without small input angles) that are nicely graded and size-optimal if the angle bound is relaxed from 30^0 to less than 26.5^0 [28], the similar approaches should be achieved in the hyperbolic case similarly.

All the above questions may be explored in the future.

Bibliography

[1] R. Alexander. Where and how Chew's second Delaunay refinement algorithm works. In *Proceedings of the 23rd Canadian Conference on Computational Geometry*, pages 157–162, 2011.

[2] Jernej Barbic. Quadratic example for Delaunay refinement. 2002.

[3] George D. Birkhoff. Dynamical systems with two degrees of freedom. *Transaction of American Mathematical Society*, 18:199–300, 1917.

[4] J.S. Birman and C. Series. Dehns algorithm revisited with applications to simple curves on surfaces. *Combinatorial Group Theory and Topology, Alta, Utah*, 1987.

[5] L.P. Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proceedings of the ninth annual symposium on Computational geometry*, SCG '93, pages 274–280, New York, NY, USA, 1993. ACM.

[6] B. Chow and F. Luo. Combinatorial Ricci flows on surfaces. *J. Differential Geom*, 63(1), 2003.

[7] Bennett Chow. The Ricci flow on the 2-sphere. *J. Differential Geom*, 33 no.2:325–334, 1991.

[8] M. Dehn. Transformation der kurven auf zweiseitigen flächen. *Mathematische Annalen*, 72.

[9] M. Dehn. Über unendliche diskontinuierliche gruppen. *Mathematische Annalen*, 71.

[10] T.K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. *ACM Transactions on Graphics*, 27.

[11] D.B.A. Epstein, J.W. Cannon, D.F. Holt, S.V.F. Levy, M.S. Patterson, and W.P. Thurston. Word processing in groups. *AK Peters*, 1992.

[12] J. Erickon and K. Whittlesey. Greedy optimal homotopy and homology generators. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms(SODA)*, pages 1038–1046.

[13] H. M. Farkas and I. kra. *Riemann Surfaces*. Springer, 1991.

[14] X. Gu and S.T. Yau. *Computational Conformal Geometry*, volume 3 of *Advanced Lectures in Mathematics*. International Press, 2008.

[15] R. S. Hamilton. The Ricci flow on surfaces. *Mathematics and General relativity (Santa Cruz, CA, 1986), Contemporary Math., American Math. Society, Providence, RI*, 71:237–262, 1988.

[16] J. Hass and P. Scott. Shortening curves on surfaces. *Topology*, 33(1):25–43, 1994.

[17] Z.X. He and Schramm O. Fixed points, Koebe uniformization and circle packings. *The Annals of Mathematics*, 137(2):369–406, 1993.

[18] M. Jin, J. Kim, F. Luo, and X. Gu. Discrete surface Ricci flow. *IEEE Transaction on Visualization and Computer Graphics*, 14(5):1030–1043, 2008.

[19] M. Jin, F. Luo, and X. Gu. Computing surface hyperbolic structure and real projective structure. In *SPM '06: Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling*, pages 105–116, 2006.

[20] C. Kadow. A fully incremental Delaunay refinement algorithm. In *Posterpresentation 10th International Meshing Roundtable*, 2001.

[21] F. Luo, X. Gu, and J. Dai. *Variational Principles for Discrete Surfaces: Theories and Algorithms*. International Press, 2008.

[22] B. Maskit. Canonical domain on riemann surface. *Proceeding of the American Mathematical Society.*, 106(3):713–721, July 1989.

[23] G.L. Miller, S.E. Pav, and N.J. Walkington. When and why Ruppert's algorithm works. In *Sandia National Laboratory*, pages 91–102, 2003.

[24] J. M. Morvan. *Generalized Curvatures*, volume 2 of *Geometry and Computing*. Springer, 2008.

[25] C. Ollivier-gooch and C. Boivin. Guaranteed-quality simplicial mesh generation with cell size and grading control. *Engineering with Computers*, 17(3):269–286, 2001.

[26] W. Parry. Growth series of some wreath products. *Transactions of the American Mathematical Society*, 331.

[27] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18(3):549–585, 1995.

[28] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1-3), 2002.

[29] W.P. Thurston. *Three-dimensional geometry and topology*. Princeton University Press, 1997.

[30] X. Yin, M. Jin, and X. Gu. Computing shortest cycles using universal covering space. *The Visual Computer: International Journal of Computer Graphics*, 23.

[31] X. Yin, Y. Li, W. Han, F. Luo, X. Gu, and S.T. Yau. Computing shortest words via shortest loops on hyperbolic surfaces. *Computer-Aided Design (CAD)*, 43.

[32] W. Zeng, R. Shi, and X. Gu. Global surface remeshing using symmetric Delaunay triangulation in uniformization spaces. *2011 Eighth Inter-*

national Symposium on Voronoi Diagrams in Science and Engineering, 2011.

[33] M. Zhang, Y. Li, W. Zeng, and X. Gu. Canonical conformal mapping for high genus surfaces with boundaries. *Computers and Graphics*, 36(5):417–426, 2012.