Hyperbolic Embedding of High Genus Surface with Ricci Flow Method - Conformal Geometry 2020 Final Project Report

I. Inxent

March 9, 2021

1 Introduction

According to uniformization theorem, a Riemann surface admits a Riemannian metric of constant curvature. Any compact Riemann surface with genus greater than 1 can be flatten and cover the unit disk. The report gives the implementation of hyperbolic embedding of high genus surfaces using Ricci flow method with technical details.

2 Theorems

The part provides complementary theorems of hyperbolic geometry that are useful to the algorithm implementation but not covered or well discussed in the courses. Theorems like uniformization of general surfaces or discrete entropy energy will not be elaborated, neither do trigonometric functions in hyperbolic cases which have been already covered in the courses.

The Euclidean distance is labelled as $\|\cdot\|$ and hyperbolic distance writes in d for the sake of clarity in the succeeding parts.

2.1 Isometric transform and invariant

Lemma 2.1. *Mobius transform*

$$T_p(z) = e^{i\theta}\frac{z-p}{1-\bar{p}z} \tag{1}$$

preserves metrics

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

on Poincare disk $\mathbb{D}$.

Proof. Denote image of the transform $w = \frac{z-p}{1-\bar{p}z}$ One can simply deduce inverse transform $z = \frac{p+w}{1+\bar{p}w}$ and one-form $dz = \frac{1-\|p\|^2}{(1+\bar{p}w)^2}dw$. Substitute all symbols with above into metrics ds^2

one can get

$$\begin{aligned}
ds^2 &= (\frac{2(1-\|p\|^2)}{\|1+\bar{p}w\|^2 - \|p+w\|^2})^2 \|w\|^2 \\
&= (\frac{2(1-\|p\|^2)}{(1-\|p\|^2)(1-\|w\|^2)})^2 \|w\|^2 \\
&= \frac{4\|dw\|^2}{(1-\|w\|^2)^2}
\end{aligned}$$

. □

Lemma 2.2. *The expression*

$$\delta(a,b) = \frac{\|a-b\|^2}{(1-\|a\|^2)(1-\|b\|^2)}$$

holds value under Mobius transform $T_p(z) = e^{i\theta}\frac{z-p}{1-\bar{p}z}$ within unit complex disk.

Proof. Expand $\delta(T(a), T(b))$

$$\begin{aligned}
\delta(T(a),T(b)) &= \frac{\|T(a)-T(b)\|^2}{(1-\|T(a)\|^2)(1-\|T(b)\|^2)} \\
&= \left\|\frac{(1-\|p\|^2)(a-b)}{(1-\bar{p}a)(1-\bar{p}b)}\right\|^2 \frac{\|1-\bar{p}a\|^2}{(1-\|p\|^2)(1-\|a\|^2)} \frac{\|1-\bar{p}b\|^2}{(1-\|p\|^2)(1-\|b\|^2)} \\
&= \frac{\|a-b\|^2}{(1-\|a\|^2)(1-\|b\|^2)} = \delta(a,b)
\end{aligned}$$

. □

Hence one can define an isometric invariant on the unit disk as $\delta(T(a), T(b))$ whose value holds under Mobius transform T_p. Since T_p preserves hyperbolic distance on $\mathbb{D}$, it bridges Euclidean distance and hyperbolic one through the invariant $\delta(z_1, z_2)$.

Lemma 2.3. *Hyperbolic distance d between origin and z within unit disk satisfies the relation with Euclidean distance $\|z\|$ as simply:*

$$\|z\| = \tanh\frac{d}{2}$$

.

Proof. One can integrate derivation of arch ds from 0 to $\|z\|$ to calculate hyperbolic distance d

$$\begin{aligned}
d &= \int_0^{\|z\|} ds \\
&= \int_0^{\|z\|} \frac{2dx}{1-x^2} \\
&= \ln\frac{1+\|z\|}{1-\|z\|}
\end{aligned}$$

. Inverse function gives the conclusion. □

Lemma 2.4. *Hyperbolic distance d and Euclidean distance $\|\cdot\|$ have the relation on Poincare disk $\mathbb{D}$ as:*

$$\frac{\|a-b\|^2}{(1-\|a\|^2)(1-\|b\|^2)} = \sinh^2\frac{d_{ab}}{2} \tag{2}$$

Proof. Consider two exclusive points a and b within unit disk, one can map a to 0 with Mobius transform T_a. Denote $T_a(b)=\beta$, the isometric invariant has the value

$$\delta(a,b) = \delta(0,\beta) = \frac{\|\beta\|^2}{1-\|\beta\|^2}$$

as T_a. The leftmost equation holds as T_a preserves the invariant. Notice $\|\beta\| = \tanh\frac{d_{ab}}{2}$, substitute the relation one can deduce:

$$\delta(a,b) = \frac{1}{2}(\cosh d_{ab} - 1) = \sinh^2\frac{d_{ab}}{2}$$

□

2.2 Hyperbolic Circles

In hyperbolic case circle equation has form $d(z,c)=d_r$ where c and d_r denotes the center and radius of hyperbolic circle respectively. With isometric invariant one has its form in Euclidean geometry that is

$$\frac{\|z-c\|^2}{(1-\|z\|^2)(1-\|c\|^2)} = \sinh^2\frac{d_r}{2}$$

.

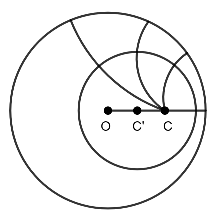


Figure 1: Hyperbolic circle where C is hyperbolic center and C' is Euclidean center. Hyperbolic diameter is portion of circle perpendicular to boundary of unit disk.

Rewrite the equation above to circle equation in Euclidean form

$$\left\| z - \frac{c}{1+\kappa} \right\|^2 = \frac{\kappa(1+\kappa-\|c\|^2)}{(1+\kappa)^2}$$

where

$$\kappa = \sinh^2\frac{d_r}{2}(1-\|c\|^2)$$

. One can see the hyperbolic circle is also an Euclidean circle whose center is closer to the origin. The relation can be used to transform a hyperbolic circle to Euclidean one for the convenience of computation.

2.3 Fuchsian Generators

Suppose one has a transform

$$T_a(z) = e^{i\theta}\frac{z-a}{1-\bar{a}z}$$

that maps a to 0 and another transform

$$T_b^{-1}(z) = e^{-i\theta}\frac{z+be^{i\theta}}{1+\overline{be^{i\theta}}z}$$

that maps 0 to b isometrically. Composite two transforms one has an isometric transform mapping a to b

$$T(z) = T_b^{-1}(T_a(z)) = \frac{(e^{i\theta}-\bar{a}b)z+(b-e^{i\theta}z)}{(e^{i\theta}\bar{b}-\bar{a})z+(1-e^{i\theta}a\bar{b})}$$

.

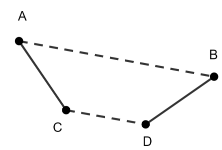


Figure 2: Mobius transform mapping two boundaries on domain

Fuchsian group generator, in the form of Mobius transform, maps two boundaries belonging to the same fundamental base, one to another and on the domain. Suppose the transform also maps c to d, one has an equation $T(c) = d$. This deduces to

$$e^{i\theta} = \frac{(d-b)(1-\bar{a}c)}{(c-a)(1-\bar{b}d)}$$

.

Combine two expressions one has a Fuchsian generator.

Lemma 2.5. *Hyperbolic distance $d_{ac} = d_{bd}$ is equivalent to*

$$\left\|\frac{(d-b)(1-\bar{a}c)}{(c-a)(1-\bar{b}d)}\right\| = 1$$

Proof. Mobius transform

$$T(z) = e^{i\theta}\frac{z-a}{1-\bar{a}z}$$

preserves hyperbolic distance between a and c i.e. d_{ac} remains unchanged after transform. Explicitly, $d_{ac} = d(0, T(c))$ and given the previous formula one has

$$\left\|e^{i\theta}\frac{c-a}{1-\bar{a}c}\right\| = \tanh\frac{d_{ac}}{2}$$

. Since the relation $d_{ac} = d_{bd}$ holds, one can derives

$$\left\|\frac{c-a}{1-\bar{a}c}\right\| = \left\|\frac{d-b}{1-\bar{b}d}\right\|$$

and conclusion is obvious. □

The lemma tells that there exist a unique Mobius transform mapping any point pairs (a, c) to (d, b) if their hyperbolic distance d are of the same. As one can see, the Mobius transform $T(z)$ has three degrees of freedom. The requirement $T(a) = b, T(c) = d$ has three restrictions, four equations along with a condition $d_{ac} = d_{bd}$.

2.4 Hyperbolic Geodesic Line

A hyperbolically geodesic line connecting two points in Poincare Disk $\mathbb{D}$ is portions of circle perpendicular to unit disk that is, the tangent directions are vertical at intersections. The conclusion is straightforward considering the fact that conformal transform $\phi(z) = \frac{z-i}{z+i}$ mapping upper half space $\mathbb{H}$ to Poincare Disk $\mathbb{D}$ preserves circle and angle.The case where two points along with origin are collinear is not considered here.

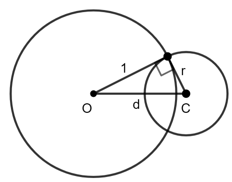


Figure 3: two perpendicular circles

The radius r of circle connecting two points p_0 and p_1 geodesically and distance from origin to center of the circle d satisfy the relation $d^2 = r^2 + 1$ which is simply given by Pythagorean theorem. By Expanding circle equation $(x - a)^2 + (y - b)^2 = r^2$ and use the relation $a^2 + b^2 = d^2$ one can retrieve the form of such circle

$$x^2 - 2ax + y^2 - 2by + 1 = 0$$

.

The center of the circle (a, b) can derived by satisfying coordinates of two points with circle equation

$$\begin{cases} a = \frac{y_2(1+x_1^2+y_1^2)-y_1(1+x_2^2+y_2^2)}{2(x_1y_2-x_2y_1)} \\ b = \frac{x_1(1+x_2^2+y_2^2)-x_2(1+x_1^2+y_1^2)}{2(x_1y_2-x_2y_1)} \end{cases}$$

, while radius $r = \sqrt{a^2 + b^2 - 1}$.

3 Algorithms

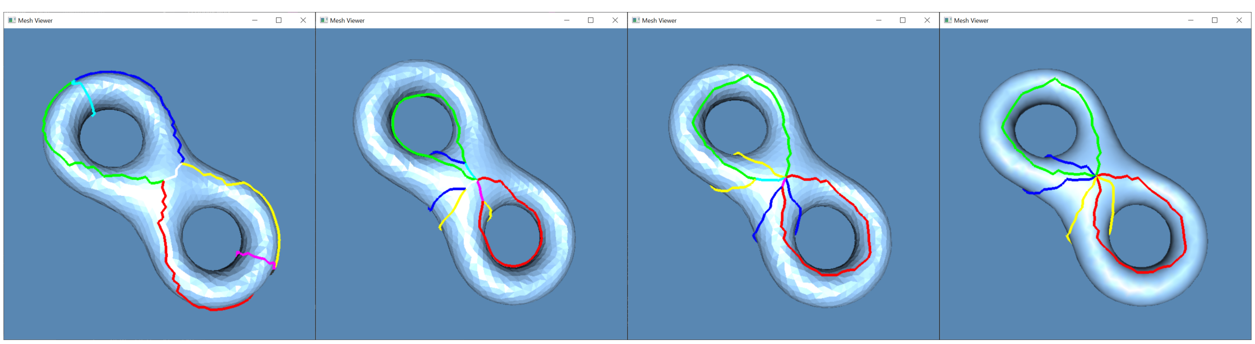
3.1 Canonical fundamental domain

In previous work, one can generate a fundamental domain via slicing the mesh along any given cut graph. Works fine, though, the original cut graph method cannot create a canonical loop system i.e. all cycles share no common edges except one vertex called base point on the mesh. Here introduce the algorithm given by Erickson and Whittlesey.

Algorithm 1: Greedy homotopy basis

Construct a spanning tree T of shortest path from selected base point v_b;
Construct a dual graph G^* on the edges of the dual graph that do not cross edges in T, prune G^* to G^{**}, remove all branches that edges do not form cycle;
Construct a maximum spanning tree T^* on the pruned dual graph G^{**}, where the weight of dual edge is length of edge;
Find edge set $\{e\}$ in which edge is not in the shortest-path tree T, corresponding dual edge is in the dual graph G^{**} but not in the maximum spanning tree T^* that is, $\{e|e \in T, e^* \in G^{**}, e^* \notin T^*\}$;
Add edge set $\{e\}$ to the shortest-path tree T and prune T out of branches not forming cycles, the edges remained form a greedy homotopy basis that passing given point;

The problem, existing in both cut graph method and greedy homotopy algorithm that loops share one or more edges on the mesh, still remains unsolved. One cannot thoroughly eliminate sharing edges when practicing the greedy homotopy algorithm, but can reduce the opportunity that loop share edges. For such a propose one can use a deformed metrics on the mesh which can be obtained through Ricci flow introduced later. Such a deformation also provides a slicing locus that is close to the geodesic line after hyperbolic embedding of the mesh. To eliminate common edges among loops, one have to edit the cut graph before slicing the mesh.



(a) cut graph (b) homotopy basis (c) deformed metrics (d) loop system

Figure 4: 4 types of cut graphs on genus-2 torus. (a) A cut graph generated by original method. (b) A greedy homotopy basis. (c) Same as (b), but using deformed metrics. (d) A loop system that forms canonical fundamental domain.

3.2 Hyperbolic Ricci flow

Algorithm 2: Hyperbolic entropy energy optimal

Target curvature $\bar{K_v} \leftarrow 0$;
Conformal factor $u_v \leftarrow 0$;
do
  Update edge length: $\sinh \frac{l_{ij}}{2} \leftarrow e^{\frac{u_i}{2}} \sinh \frac{l^{\emptyset}_{ij}}{2} e^{\frac{u_j}{2}}$;
  Update corner angle: $\theta^i_{jk} \leftarrow \theta(l_{ji}, l_{ik})$;
  Update vertex curvature: $K_i \leftarrow K(\theta^i)$;
  Compute gradient of entropy energy: $\Delta E \leftarrow \Sigma_v(\bar{K_v} - K_v)$;
  Solving linear equations: $H\delta u = \Delta E$ where H is Hessian matrix;
  Update conformal factor: $u \leftarrow u - \lambda \delta u$;
while $\|\Delta E\| < \epsilon$;

The corner angle θ can be compute using the hyperbolic cosine law

$$\cos\theta = \frac{\cosh a \cosh b - \cosh c}{\sinh a \sinh b}$$

where edges a, b are adjacent sides and c is opposite side of θ.

The discrete curvature of a vertex K_v is defined as

$$K_v = \begin{cases} \pi - \Sigma\theta^v, & \text{if } v \in \partial S \\ 2\pi - \Sigma\theta^v, & \text{otherwise} \end{cases}$$

.

One should be noticed that the edge length is updated from its original length $l^{\emptyset}$. Since one needs original length to compute, there is no need to change length on the original mesh, only to save it as a copy somewhere on the memory for computation of next several steps.

Considering a face f_{ijk} and an angle θ_i, one has

$$\begin{cases} \frac{\partial\theta_i}{\partial u_j} = \frac{\cosh l_i + \cosh l_j - \cosh l_k - 1}{A(\cosh l_k + 1)} = \frac{\partial\theta_j}{\partial u_i} \\ \frac{\partial\theta_i}{\partial u_i} = -\frac{\cosh l_i \cosh l_j - \cosh l_k}{A(\cosh l_j + 1)} - \frac{\cosh l_i \cosh l_k - \cosh l_j}{A(\cosh l_k + 1)} \end{cases}$$

where A is hyperbolic area of face f_{abc}

$$A = \frac{1}{2}\sqrt{1 - \cosh^2 l_i - \cosh^2 l_j - \cosh^2 l_k + 2\cosh l_i \cosh l_j \cosh l_k}$$

. The terms $\frac{\partial\theta_i}{\partial u_j}$ and $\frac{\partial\theta_i}{\partial u_i}$ contribute to non-diagonal and diagonal elements of Hessian matrix respectively.

3.3 Hyperbolic isometric embedding

Algorithm 3: Embed whole domain

Mark all faces $f \in M$ and vertices $v \in M$ not visited;
$f_0 \leftarrow$ any face from M;
Embed f_0 onto $\mathbb{D}$;
Queue Q.push(f_0);
Mark f_0 visited;
while *Q is not empty* **do**
    $f_0 \leftarrow$ Q.pop();
    for *face f adjacent of f_0* **do**
        if *face f is not visit* **then**
            Embed f;
            Mark f visited;
            Q.push(f);
        end
    end
end

Algorithm 4: Embed single face

Input: face f
$[v_0, v_1, v_2] \leftarrow f$;
if *v_2 is not visit* **then**
    $c_0 \leftarrow$ hyperbolic circle $c(v_0, l_{02})$;
    $c_1 \leftarrow$ hyperbolic circle $c(v_1, l_{12})$;
    $v_2 \leftarrow$ hyperbolic_intersection(c_0, c_1);
    Mark v_2 visited;
end

To embed the first face one can easily set v_0 to $(0, 0)$ and v_1 to $(0, \tanh \frac{l_{01}}{2})$. The intersection of hyperbolic circles in not direct. One can transform hyperbolic circle to Euclidean one with method discussed in last section, then apply Euclidean circles intersection to get the embedding position on the disk $\mathbb{D}$.

Algorithm 5: Euclidean circles intersection (CCW)

Input: Circles c_0, c_1
Output: Intersection of two circles v_2
$v_0, r_0 \leftarrow c_0$;
$v_1, r_1 \leftarrow c_1$;
$r \leftarrow \|v_0 - v_1\|$;
$a \leftarrow \frac{r_0^2 - r_1^2 + r^2}{2r}$;
$h \leftarrow \sqrt{r_0^2 - a^2}$;
$\boldsymbol{t} = \hat{n} \times \frac{v_1 - v_0}{r}$;
$v_2 \leftarrow (1 - \frac{a}{r})v_0 + \frac{a}{r}v_1 + h\boldsymbol{t}$;

Result of hyperbolic Ricci flow and embedding is very sensitive to numeric error from precision limit of computer. Format float32 fails the computation while float64 is enough to create approximate result.

4 Results

In the experiments I followed the streamline as:

1: Perform hyperbolic Ricci flow until curvature reduces to around 1^{-10};

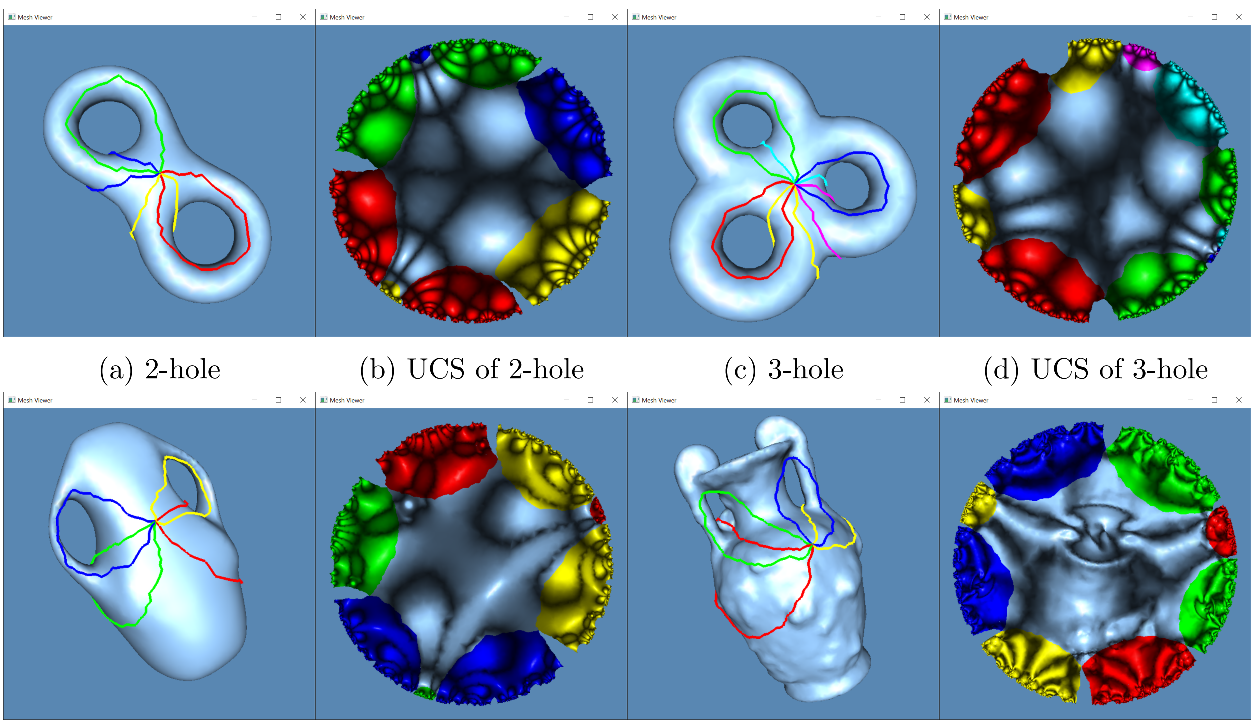
2: Generate a fundamental domain using cut graph;

3: Embed the domain onto Poincare disk $\mathbb{D}$ isometrically;

4: Apply transforms in Fuchsian group to the domain to tessellate the disk;

5: Fine boundaries with geodesic lines (not shown int the results).

The domain is tessellated up to T^3 i.e., applied to at most 3 transforms from Fuchsian group. More transforms cannot be held by memory as the number of tessellations grows exponentially, neither need it as T^3 almost cover the whole Poincare disk.



(a) 2-hole (b) UCS of 2-hole (c) 3-hole (d) UCS of 3-hole

(e) Vase (f) UCS of Vase (g) Amphora (h) UCS of Amphora

Figure 5: Original meshes and corresponding hyperbolic embedding

Mesh	Genus	Verts	Faces	Time(s)
2-hole	2	2057	4118	0.71
3-hole	3	1754	3516	0.63
Vase	2	5501	11006	2.0
Amphora	2	10003	20010	4.0

Table 1: Computational time from Ricci flow to embedding, on a laptop with 1.7 GHz CPU and 16 GB RAM. Error bound for Ricci flow is set to $1e^{-9}$.

5 Future work

5.1 Fining boundaries with slicing

The boundaries fining in this implementation casts circles to the domain mesh without really changing the mesh. To slice along a curve one needs to find the intersection of the curve and each edge on boundary. One can use TetGen or other mesh generation tools to fulfill the propose.

5.2 Homotopy detection

After canonical fundamental domain can be realized, one can apply homotopy detection, i.e., given a loop on the surface, it can be decomposed into combination of homotopic bases and got categorized.

To detect homotopy of a loop, one can trace the loop from a chosen base point on the loop, recording the boundaries it crosses. The domain where base point is reached corresponds to an element in Fuchsian group which represents the class of the loop.