

Centroidal Voronoi Tessellation in Universal Covering Space of Manifold Surfaces ¹

Guodong Rong^a, Miao Jin^b, Liang Shuai^a, Xiaohu Guo^a

^a*Department of Computer Science, University of Texas at Dallas*

^b*Center for Advanced Computer Studies, University of Louisiana at Lafayette*

Abstract

The centroidal Voronoi tessellation (CVT) has found versatile applications in geometric modeling, computer graphics, and visualization, etc. In this paper, we first extend the concept of CVT from Euclidean space to spherical space and hyperbolic space, and then combine all of them into a unified framework – the CVT in universal covering space. The novel spherical and hyperbolic CVT energy functions are defined, and the relationship between minimizing the energy and the CVT is proved. We also show by our experimental results that both spherical and hyperbolic CVTs have the similar property as their Euclidean counterpart where the sites are uniformly distributed with respect to given density values. As an example of the application, we utilize the CVT in universal covering space to compute uniform partitions and high-quality remeshing results for genus-0, genus-1, and high-genus (genus>1) surfaces.

Keywords: Centroidal Voronoi Tessellation, Universal Covering Space, Spherical Space, Hyperbolic Space, Remeshing

1. Introduction

The Voronoi diagram is a well studied concept in computational geometry, and has a wide usage in different areas in geometric modeling, computer graphics, visualization, etc. [Okabe et al., 1999]. The *centroidal Voronoi tessellation* (CVT) is a special case of the Voronoi diagram, where every site coincides with the centroid of its Voronoi cell [Du et al., 1999]. The sites in

¹This paper is the extended version of our previous paper published on Symposium of Solid & Physical Modeling 2010 [Rong et al., 2010].

a CVT are uniformly distributed. This property is conjectured by Gershgorin in 1979 [Gershgorin, 1979], and has been proved in 2D convex polygons with up to six edges [Fejes Tóth, 2001].

In geometric modeling, many applications require a uniform sampling on a surface, or a partition of a surface where every region covers similar area. These tasks can be achieved simultaneously by computing a CVT on the surface where all sites are constrained on the surface. Such a CVT is usually known as the *constrained CVT* [Du et al., 2003]. It is natural to use the geodesic distance to compute the constrained CVT [Peyré and Cohen, 2004], but it is difficult to compute the geodesic distance accurately. Another alternative is to use the 3D Euclidean distance as an approximation [Liu et al., 2009; Rong et al., 2011; Yan et al., 2009], but this may lead to disconnected Voronoi cells if two regions are very close in 3D Euclidean space but are far away along the surface. A better approach is to compute the CVT in a 2D parametrization domain of the surface [Alliez et al., 2005]. By assigning appropriate density values, the computed CVT is very close to the constrained CVT computed using the geodesic distance. This method overcomes the shortages of both prior methods, and is more efficient since the computation is performed in a 2D Euclidean domain.

While to parameterize a closed surface to a 2D Euclidean domain, the original surface has to be cut into a genus-0 surface. This makes the sites unable to cross the boundaries in the parametrization domain, and leads to visible artifacts along the cutting edges. In [Alliez et al., 2005], a great deal of special care and delicate strategies, such as minimizing the total cutting edge length and matching the cut graph with the feature skeleton, are required. If the cut graph does not coincide much with a set of feature edges, the remeshing results become unacceptable as indicated in [Alliez et al., 2005].

This cutting problem can be solved by computing the CVT directly on the *universal covering space* [Klingenberg, 1982] of the surface. For closed genus-1 surfaces, their universal covering space can be embedded in 2D Euclidean space \mathbb{R}^2 , so the computation of the CVT in the universal covering space is similar as in [Alliez et al., 2005] except that sites can move freely across the cutting boundaries. The universal covering space of closed genus-0 surfaces can be embedded in 2D spherical space \mathbb{S}^2 . As proved later in Section 4, the spherical CVT is identical to the constrained CVT on the sphere. So we can compute the constrained CVT on the sphere to get the uniform samplings on these surfaces. For closed high-genus (genus>1) surfaces, their universal covering space can be embedded in 2D hyperbolic space \mathbb{H}^2 . So computing

the CVT in hyperbolic space is required and can lead to new geometric modeling techniques for high-genus surfaces.

To the best of our knowledge, the CVT in hyperbolic space has not been studied before. Furthermore, no previous work has systematically studied the CVT in universal covering space. We study the CVT in hyperbolic space in this paper, and combine it with Euclidean CVT and spherical CVT in a unified framework of the CVT in universal covering space.

One difficulty for defining the CVT is how to well define the centroid of a given region in different spaces. In this paper, we extend the *model centroid* [Galperin, 1993] to define the centroid of a Voronoi cell in 2D spherical, Euclidean, and hyperbolic spaces in a unified way. We also prove that the model centroid is in fact the central projection of the centroid in 3D Euclidean space onto the model.

Previous studies on spherical CVT all treat it as the constrained CVT on the sphere. In this paper, we directly define the CVT energy in spherical space and study its relationship with the spherical CVT. We also define the CVT energy in hyperbolic space and prove the relationship between minimizing this energy and the hyperbolic CVT. Following these conclusions, we can prove the convergence of Lloyd’s algorithm for both spherical and hyperbolic CVTs. So we can use Lloyd’s algorithm to compute them. Based on our extensive experiments, we conjecture the sites in the spherical and hyperbolic CVTs are also uniformly distributed with respect to the corresponding metrics.

We also show how to use the CVT in universal covering space to generate uniform partitions and high quality remeshing results for genus-0, genus-1, and high-genus (genus>1) surfaces. Compared with previous methods using parametrization in 2D Euclidean space such as [Alliez et al., 2005], the main advantage of using the CVT in universal covering space is that the sites can move freely anywhere on the surface.

The main contributions of this paper include:

- We formally define the CVT energy in spherical space, and prove the relationship between minimizing this energy and the spherical CVT. We also demonstrate the uniformity of the sites in the spherical CVT.
- We extend the concept of CVT into hyperbolic space. We define the CVT energy in hyperbolic space, and prove the relationship between minimizing this energy and the hyperbolic CVT. We also demonstrate the uniformity of the sites in the hyperbolic CVT.

- We prove the convergence of Lloyd’s algorithm for spherical and hyperbolic CVTs, and explain the implementation details of using Lloyd’s algorithm to compute them.
- We combine spherical, Euclidean, and hyperbolic CVTs into a unified framework – the CVT in universal covering space, and apply it on computing uniform partitions and high quality remeshing results for genus-0, genus-1, and high-genus (genus>1) surfaces.

The rest of the paper is organized as follows: Section 2 briefly reviews some related previous work. The formal definitions of the CVT in different spaces are given in Section 3, and the corresponding CVT energy functions are defined in Section 4. The relationship between minimizing this energy and the CVT is also proved in Section 4. Section 5 gives details on how to compute the CVT in difference spaces. Section 6 defines the CVT in universal covering space and applies it on geometric modeling applications. Finally, Section 7 concludes the paper with some possible future work.

2. Related Work

We briefly review some previous work on how to compute the CVT in Euclidean space. We also list applications of the CVT in geometric modeling.

2.1. Centroidal Voronoi Tessellation

The concept of the centroidal Voronoi tessellation was first introduced by Du et al. [1999], but the similar concepts have been studied in different areas long before that, e.g. optimal quantization in signal processing and k -means in pattern recognition.

One of the earliest algorithms to compute the CVT is proposed by MacQueen [1967] which is a probabilistic algorithm. Although the almost sure convergence of this algorithm is proved, its convergence is very slow. Lloyd proposed a deterministic method in 1960s which is officially published later in 1982 [Lloyd, 1982]. The convergence of Lloyd’s algorithm is later proved by Du et al. [2006]. Due to its simplicity and robustness, Lloyd’s algorithm is currently the most widely used algorithm to compute the CVT. Ju et al. [2002] combined MacQueen’s method and Lloyd’s algorithm into the probabilistic Lloyd algorithm which is suitable for parallel computation.

Du and Emelianenko [2006] proposed a Lloyd-Newton method where they first run standard Lloyd’s algorithm until the sites are almost stable and then

switch to Newton iteration. Most recently, Liu et al. [2009] proved the CVT energy function (defined in Section 4) is C^2 continuous, and thus they are able to apply a quasi-Newton method – L-BFGS algorithm – to compute the CVT. This method avoids the computation and storage of the full Hessian matrix in Lloyd-Newton method. It converges much faster than Lloyd’s algorithm and is the fastest algorithm for the CVT so far.

Previous researches on spherical CVT, such as [Du et al., 2003], [Du and Ju, 2005a], [Du and Ju, 2005b], all treat the spherical CVT as the constrained CVT on the sphere. So the above algorithms can be used to compute the spherical CVT with Euclidean metric.

2.2. Geometric Modeling Applications

Due to the uniformity of the Voronoi cells and sites, the CVT has been widely used in many applications. We only review previous work closely related to geometric modeling. For other applications, please refer to [Du et al., 1999, 2010; Rong et al., 2011] and the references therein.

Many researches on geometric modeling utilize the dual of the constrained CVT to achieve high quality remeshing results. Surazhsky et al. [2003] computed the constrained CVT by projecting the 1-ring neighbors of a site in the dual triangle mesh onto the tangential plane, and then finding the centroid of its Voronoi cell in the plane. Contrast to this local parametrization approach, Alliez et al. [2005] used a global parametrization by cutting the surface into a disk-like topology, and computing a 2D CVT in the Euclidean parametrization domain. Valette et al. [2008] directly computed an approximation of the constrained CVT as clusters of triangles. Yan et al. [2009] first computed a 3D CVT and then found the intersection between the surface and the 3D CVT.

Cohen-Steiner et al. [2004] extended the concept of centroids to planar proxies, and used a flooding scheme to compute the constrained CVT as a good shape approximation. Lu et al. [2009] computed the CVT for line segments and graphs, and used it to get meaningful segmentations of 3D models.

The CVT in all the above work is computed in 2D or 3D Euclidean space. In this paper, we study the CVT in spherical space and hyperbolic space, and combine them with the CVT in Euclidean space into a unified CVT in universal covering space. We also demonstrate the application of the CVT in universal covering space for geometric modeling.

3. Centroidal Voronoi Tessellation

In this section we first give formal definitions of the Voronoi diagram and the centroid in different spaces, and then combine them together to define the centroidal Voronoi tessellation in different spaces. For each definition, we first review its usual definition in Euclidean space, and then extend it to spherical and hyperbolic spaces. We only study the CVT in 2D spaces in this paper. The superscripts \mathbb{E} , \mathbb{S} , and \mathbb{H} are used to represent notions in Euclidean, spherical and hyperbolic spaces respectively.

3.1. Voronoi Diagram in Different Spaces

Given n points (called *sites*) $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n$ in a domain Ω , the *Voronoi cells* Ω_i is the union of all points nearer to site s_i than to other sites:

$$\Omega_i = \{\mathbf{p} \in \Omega | d(\mathbf{p}, \mathbf{s}_i) < d(\mathbf{p}, \mathbf{s}_j), i \neq j\}, \quad (1)$$

where $d(\mathbf{a}, \mathbf{b})$ is the distance defined in Ω . The union of all the Voronoi cells is the *Voronoi diagram* of the sites.

For a 2D Euclidean domain $\Omega^{\mathbb{E}} \subset \mathbb{R}^2$, the distance $d(\mathbf{a}, \mathbf{b})$ is replaced by the 2D Euclidean distance $d^{\mathbb{E}}(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\| = \sqrt{(x_{\mathbf{a}} - x_{\mathbf{b}})^2 + (y_{\mathbf{a}} - y_{\mathbf{b}})^2}$, and we have the Voronoi diagram defined in Euclidean space. Similarly, by replacing the distance $d(\cdot, \cdot)$ in Equation (1) with the spherical distance $d^{\mathbb{S}}(\cdot, \cdot)$ or the hyperbolic distance $d^{\mathbb{H}}(\cdot, \cdot)$, we can define the Voronoi diagram in a 2D spherical domain $\Omega^{\mathbb{S}} \subset \mathbb{S}^2$ or a 2D hyperbolic domain $\Omega^{\mathbb{H}} \subset \mathbb{H}^2$.

On a sphere with radius of r , the spherical distance between two points \mathbf{p} and \mathbf{q} is defined as:

$$d^{\mathbb{S}}(\mathbf{p}, \mathbf{q}) = r \cos^{-1} \left(\frac{\langle \mathbf{p}, \mathbf{q} \rangle}{r^2} \right),$$

where $\langle \mathbf{p}, \mathbf{q} \rangle = x_{\mathbf{p}}x_{\mathbf{q}} + y_{\mathbf{p}}y_{\mathbf{q}} + z_{\mathbf{p}}z_{\mathbf{q}}$ is the inner product in 3D Euclidean space. In this paper, we always consider the spherical space on a unit sphere. So the spherical distance is simplified to:

$$d^{\mathbb{S}}(\mathbf{p}, \mathbf{q}) = \cos^{-1} (\langle \mathbf{p}, \mathbf{q} \rangle).$$

Figure 1(a) shows a Voronoi diagram of 500 randomly sampled points on a sphere.

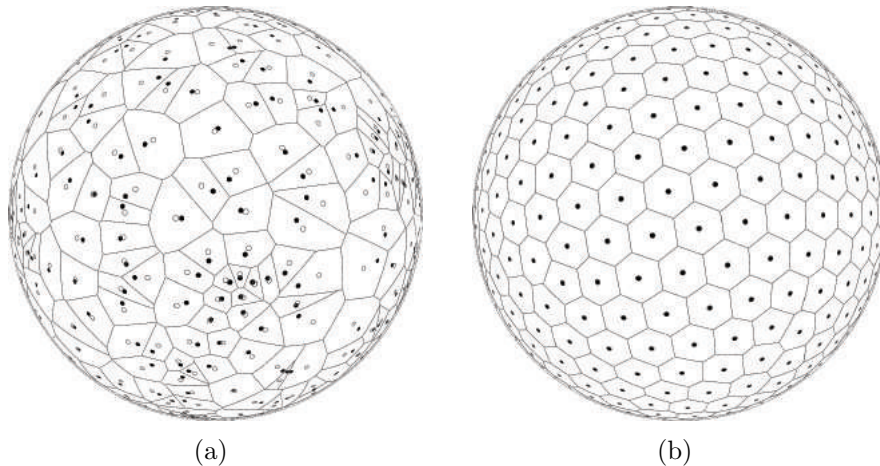


Figure 1: (a) Voronoi diagram of 500 random initial sites (solid dots) on a sphere. Hollow dots are spherical centroids of corresponding Voronoi cells. (b) Spherical CVT generated from (a). $\rho(\mathbf{p}) \equiv 1$ in this example.

Because of the relationship between the spherical distance and the 3D Euclidean distance:

$$d^{\mathbb{S}}(\mathbf{p}, \mathbf{q}) = 2 \sin^{-1} \frac{d^{\mathbb{E}}(\mathbf{p}, \mathbf{q})}{2},$$

we have $d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i) < d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_j)$ if and only if $d^{\mathbb{E}}(\mathbf{p}, \mathbf{s}_i) < d^{\mathbb{E}}(\mathbf{p}, \mathbf{s}_j)$. So the spherical Voronoi diagram is in essence same as the constrained Voronoi diagram on the sphere, i.e. the intersection between the sphere and the 3D Euclidean Voronoi diagram.

There are several different models for the hyperbolic geometry. They are all equivalent, but provide different views. Poincaré disk model and Klein disk model both use a unit disk ($x^2 + y^2 < 1$) in 2D Euclidean plane to represent the 2D hyperbolic space; the upper half-plane model uses half of the 2D Euclidean plane ($y > 0$) to represent the 2D hyperbolic space; Minkowski model embeds the 2D hyperbolic space into one sheet of a two-sheeted hyperboloid ($-x^2 - y^2 + z^2 = 1, z > 0$) in 3D Euclidean space. All these models can be easily converted from one to another. In this paper, we only use Klein disk model and Minkowski model. The details of the conversion between these two models are given later in Section 5.

Among these models, the Voronoi diagram has been studied in the upper half-plane model [Onishi and Takayama, 1996], Poincaré disk model [Nil-

foroushan and Mohades, 2006], and Klein disk model [Nielsen and Nock, 2010]. In our implementation, we compute the Voronoi diagram in Klein disk as it is easy for both computation and visualization.

The geodesic distance in Klein disk is a chord of the unit disk. The distance between two points \mathbf{p} and \mathbf{q} in Klein disk is

$$d_K^{\mathbb{H}}(\mathbf{p}, \mathbf{q}) = \cosh^{-1} \frac{1 - \langle \mathbf{p}, \mathbf{q} \rangle}{\sqrt{(1 - \|\mathbf{p}\|^2)(1 - \|\mathbf{q}\|^2)}},$$

where $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ are inner product and vector norm computed in Euclidean space as defined above. Using this distance, it can be proved that the bisector of two points in Klein disk is a straight line in Euclidean space, and the hyperbolic Voronoi diagram in Klein disk is a *power diagram* [Aurenhammer, 1987] in Euclidean space. More specifically, for every site \mathbf{s}_i in Klein disk, a corresponding weighted point $w p_i = \langle \mathbf{t}_i, w_i^2 \rangle$ can be created in Euclidean space, where $\mathbf{t}_i = \frac{\mathbf{s}_i}{2\sqrt{1-\|\mathbf{s}_i\|^2}}$ and $w_i^2 = \frac{\|\mathbf{s}_i\|^2}{4(1-\|\mathbf{s}_i\|^2)} - \frac{1}{\sqrt{1-\|\mathbf{s}_i\|^2}}$. The power diagram of weighted points $w p_i$ in Euclidean space is same as the Voronoi diagram of the sites \mathbf{s}_i in Klein disk. The derivation details can be found in [Nielsen and Nock, 2010]. Figure 2(a) shows a Voronoi diagram of 100 random sampled sites in a hexagon in Klein disk.

3.2. Centroidal Voronoi Tessellation in Different Spaces

The *centroidal Voronoi tessellation* (CVT) is a special Voronoi diagram where every site \mathbf{s}_i coincides with the centroid \mathbf{c}_i of its Voronoi cell [Du et al., 1999, 2010]. To define the CVT in Euclidean, spherical, and hyperbolic spaces, we have to first define the centroid in these spaces. Combining these definitions and the definition of the Voronoi diagram above, we can well define the CVT in these spaces.

The definition of the centroid (a.k.a. center of mass) is based on a given density field. We use $\rho(\mathbf{p})$ to denote the density value at point \mathbf{p} , and require $\rho(\mathbf{p}) \geq 0$. In Euclidean space, the centroid of a region $V_i^{\mathbb{E}} \subset \mathbb{R}^2$ is defined as:

$$\mathbf{c}_i^{\mathbb{E}} = \frac{\int_{V_i^{\mathbb{E}}} \rho(\mathbf{p}) \mathbf{p} \, d\sigma}{\int_{V_i^{\mathbb{E}}} \rho(\mathbf{p}) \, d\sigma}. \quad (2)$$

The centroid has also been studied in Riemannian manifolds [Karcher, 1977; Kobayashi and Nomizu, 1996]. In this paper, we utilize the idea of *model centroid* proposed by Galperin [1993], and extend it from discrete

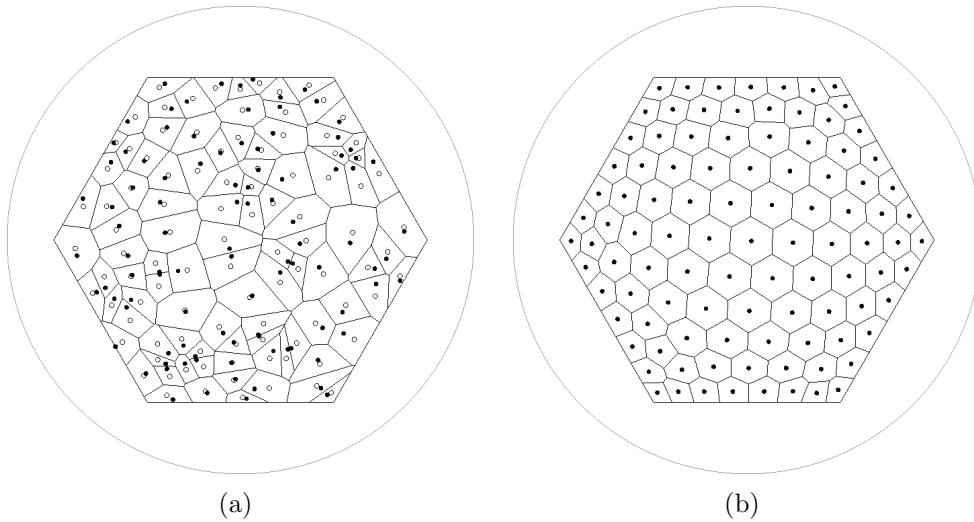


Figure 2: (a) Voronoi diagram of 100 random initial sites (solid dots) in a hexagon in Klein disk. Hollow dots are hyperbolic centroids of corresponding Voronoi cells. (b) Hyperbolic CVT generated from (a). $\rho(\mathbf{p}) \equiv 1$ in this example.

points to a continuous region. The model centroid unified the definition of the centroid in spaces with constant curvature – Euclidean space, spherical space, and hyperbolic space.

Given n discrete points \mathbf{p}_i in a k -dimensional space, and n mass values m_i corresponding to these points, the position of the centroid of these points can be located as follows: We find a “model” of the k -dimensional space in $(k + 1)$ -dimensional Euclidean space. For every point \mathbf{p}_i , a vector is built from the origin pointing to the point. The vectors are first scaled by the corresponding mass values, and then added up. The intersection between the sum vector and the model is defined as the position of the centroid of these points. This definition is proved to be well-defined for any number of discrete points and satisfies the axioms given in [Galperin, 1993].

For 2D Euclidean space, the model is the plane $z = 1$ in 3D Euclidean space. Figure 3 illustrates the side view of the computation of the centroid of two points in 2D Euclidean space. For this case, the sum vector $\mathbf{q} = m_1\mathbf{p}_1 + m_2\mathbf{p}_2$. To compute the intersection between any vector and the model (the plane $z = 1$), we can divide it by its z -component. So we have

$$\mathbf{c} = \frac{\mathbf{q}}{z_{\mathbf{q}}} = \frac{m_1\mathbf{p}_1 + m_2\mathbf{p}_2}{m_1z_{\mathbf{p}_1} + m_2z_{\mathbf{p}_2}} = \frac{m_1\mathbf{p}_1 + m_2\mathbf{p}_2}{m_1 + m_2},$$

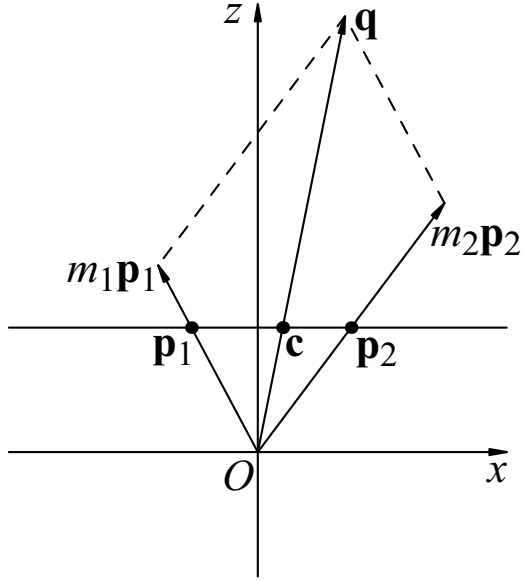


Figure 3: Side view of the computation for the centroid of two discrete points in Euclidean space.

which is same as the usual definition of the centroid of two points in Euclidean space.

When we replace the summation of the n vectors with the integral over a continuous region, we can compute the centroid of the region. Using the computation similar to above, it is easy to verify that the model centroid defined in this way is same as the centroid $\mathbf{c}_i^{\mathbb{E}}$ defined in Equation (2).

For 2D spherical space, the model is the unit sphere $x^2 + y^2 + z^2 = 1$. Figure 4 illustrates the side view of the computation of the centroid of two points in spherical space. The sum vector is same as above, i.e. $\mathbf{q} = m_1\mathbf{p}_1 + m_2\mathbf{p}_2$. To compute the intersection between any vector and the sphere, we normalize the vector using its Euclidean norm:

$$\mathbf{c} = \frac{\mathbf{q}}{\|\mathbf{q}\|} = \frac{m_1\mathbf{p}_1 + m_2\mathbf{p}_2}{\|m_1\mathbf{p}_1 + m_2\mathbf{p}_2\|}.$$

Comparing this with the Euclidean centroid of these two points

$$\mathbf{c}_E = \frac{m_1\mathbf{p}_1 + m_2\mathbf{p}_2}{m_1 + m_2},$$

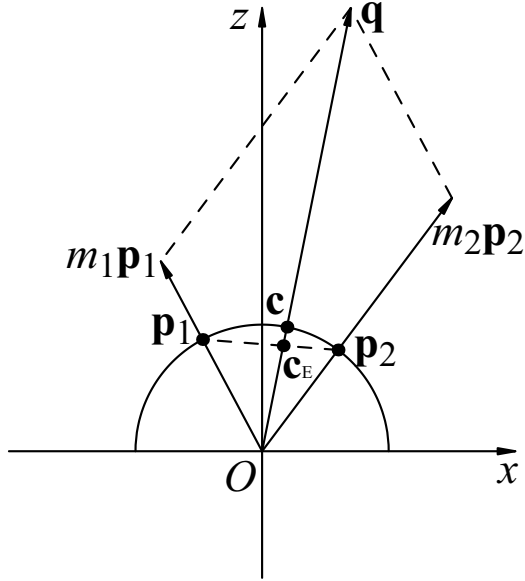


Figure 4: Side view of the computation for the centroid of two discrete points in spherical space.

we note their numerators are same, which indicates these two vectors are collinear. In other words, the spherical centroid defined by model centroid is the central projection, with respect to the origin point, of the 3D Euclidean centroid onto the sphere.

Similarly, we can extend the definition of the spherical centroid from discrete points to a continuous region $V_i^{\mathbb{S}} \subset \mathbb{S}^2$ by replacing the summation by integral:

$$\mathbf{c}_i^{\mathbb{S}} = \frac{\int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \mathbf{p} \, d\sigma}{\left\| \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \mathbf{p} \, d\sigma \right\|}, \quad (3)$$

and the projection property still holds. The hollow dots in Figure 1(a) are spherical centroids of the Voronoi cells. Figure 1(b) shows the spherical CVT generated from the 500 initial sites in Figure 1(a).

For 2D hyperbolic space, the model is Minkowski model, which is the upper sheet of a two-sheeted hyperboloid $-x^2 - y^2 + z^2 = 1$ in 3D Euclidean space. Figure 5 illustrates the side view of the computation of the centroid of two points in hyperbolic space. Again, the sum vector here is $\mathbf{q} = m_1 \mathbf{p}_1 + m_2 \mathbf{p}_2$. To compute the intersection between any vector with the

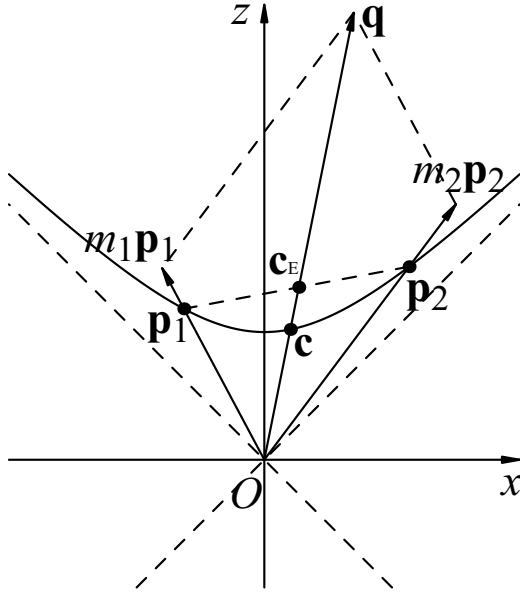


Figure 5: Side view of the computation for the centroid of two discrete points in hyperbolic space.

hyperboloid, we normalize the vector using its *Minkowski norm* defined as $\|\mathbf{p}\|_M = \sqrt{-x_{\mathbf{p}}^2 - y_{\mathbf{p}}^2 + z_{\mathbf{p}}^2}$. So, in Figure 5, we have:

$$\mathbf{c} = \frac{\mathbf{q}}{\|\mathbf{q}\|_M} = \frac{m_1\mathbf{p}_1 + m_2\mathbf{p}_2}{\|m_1\mathbf{p}_1 + m_2\mathbf{p}_2\|_M}.$$

It also has the same numerator with the 3D Euclidean centroid \mathbf{c}_E , and thus these two vectors are collinear.

When we extend this definition to continuous regions using integral, we have the hyperbolic centroid of a region $V_i^{\mathbb{H}} \subset \mathbb{H}^2$ as:

$$\mathbf{c}_i^{\mathbb{H}} = \frac{\int_{V_i^{\mathbb{H}}} \rho(\mathbf{p})\mathbf{p} \, d\sigma}{\|\int_{V_i^{\mathbb{H}}} \rho(\mathbf{p})\mathbf{p} \, d\sigma\|_M}. \quad (4)$$

It is the central projection, with respect to the origin point, of the 3D Euclidean centroid onto the hyperboloid. Note that because we use Minkowski model to define the hyperbolic centroid, the integral in Equation (4) is in fact a surface integral on the hyperboloid. Since all models of the hyperbolic geometry are equivalent, we can always compute the integral in hyperbolic

space as a surface integral on a hyperboloid by using Minkowski model. Figure 2(a) shows the hyperbolic centroids of all Voronoi cells marked as hollow dots, and the corresponding hyperbolic CVT is shown in Figure 2(b).

4. CVT Energy

In Euclidean space, for any set of sites and any partition corresponding to the sites, we can define an energy function. It is proved that the necessary condition for minimizing this energy function is that the sites and the partition form a centroidal Voronoi tessellation [Du et al., 1999]. This energy function is thus called *CVT energy*. In this section, we define the CVT energy in different spaces and study the relationship between the CVT energy and the CVT in these spaces.

4.1. CVT Energy in Different Space

In Euclidean space, suppose we have an ordered set of sites $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n)$ in a Euclidean domain $\Omega^{\mathbb{E}} \subset \mathbb{R}^2$, and a tessellation $V = (V_1^{\mathbb{E}}, V_2^{\mathbb{E}}, \dots, V_n^{\mathbb{E}})$, where part $V_i^{\mathbb{E}}$ correspond to site \mathbf{s}_i , $V_i^{\mathbb{E}} \cap V_j^{\mathbb{E}} = \emptyset$ if $i \neq j$, and $\bigcup_{i=1}^n V_i^{\mathbb{E}} = \Omega^{\mathbb{E}}$. The CVT energy of \mathbf{S} and V is defined as:

$$\begin{aligned} F^{\mathbb{E}}(\mathbf{S}, V) &= \sum_{i=1}^n \int_{V_i^{\mathbb{E}}} \rho(\mathbf{p}) (d^{\mathbb{E}}(\mathbf{p}, \mathbf{s}_i))^2 d\sigma \\ &= \sum_{i=1}^n \int_{V_i^{\mathbb{E}}} \rho(\mathbf{p}) \|\mathbf{p} - \mathbf{s}_i\|^2 d\sigma. \end{aligned} \quad (5)$$

We can also define a similar energy function in both spherical space and hyperbolic space. As proved later in this section, these functions have similar properties as their Euclidean counterpart, and thus are also appropriately called *CVT energy* in these spaces.

To define the CVT energy in spherical space, recall that the distance in spherical space is $d^{\mathbb{S}}(\mathbf{a}, \mathbf{b}) = \cos^{-1}(\langle \mathbf{a}, \mathbf{b} \rangle)$. So $\cos(d^{\mathbb{S}}(\mathbf{a}, \mathbf{b})) = \langle \mathbf{a}, \mathbf{b} \rangle$. Since $\langle \mathbf{a}, \mathbf{a} \rangle = \|\mathbf{a}\|^2$, $\cos(d^{\mathbb{S}}(\mathbf{a}, \mathbf{b}))$ has the same dimension (i.e. square of length) as $(d^{\mathbb{E}}(\mathbf{a}, \mathbf{b}))^2$. Based on this observation, we define the CVT energy in

spherical space as:

$$\begin{aligned}
F^{\mathbb{S}}(\mathbf{S}, V) &= \sum_{i=1}^n \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \cos(d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i)) \, d\sigma \\
&= \sum_{i=1}^n \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \langle \mathbf{p}, \mathbf{s}_i \rangle \, d\sigma.
\end{aligned} \tag{6}$$

We define the hyperbolic CVT energy on Minkowski model, since the hyperbolic distance on this model has a simple form. The distance between two points \mathbf{p} and \mathbf{q} on Minkowski model is:

$$d_M^{\mathbb{H}}(\mathbf{p}, \mathbf{q}) = \cosh^{-1}(\langle \mathbf{p}, \mathbf{q} \rangle_M),$$

where $\langle \mathbf{p}, \mathbf{q} \rangle_M = -x_{\mathbf{p}}x_{\mathbf{q}} - y_{\mathbf{p}}y_{\mathbf{q}} + z_{\mathbf{p}}z_{\mathbf{q}}$ is the Minkowski inner product defined in 3D Euclidean space. Similar to spherical space, we define the CVT energy in hyperbolic space as:

$$\begin{aligned}
F^{\mathbb{H}}(\mathbf{S}, V) &= \sum_{i=1}^n \int_{V_i^{\mathbb{H}}} \rho(\mathbf{p}) \cosh(d^{\mathbb{H}}(\mathbf{p}, \mathbf{s}_i)) \, d\sigma \\
&= \sum_{i=1}^n \int_{V_i^{\mathbb{H}}} \rho(\mathbf{p}) \langle \mathbf{p}, \mathbf{s}_i \rangle_M \, d\sigma.
\end{aligned} \tag{7}$$

Next, we study the properties of the CVT energy in spherical space and hyperbolic space.

4.2. CVT Energy and Voronoi Diagram

In Euclidean space, if the sites are fixed and the tessellation varies, we have the following conclusion:

Lemma 1. *When the sites are fixed, the CVT energy $F^{\mathbb{E}}(\mathbf{S}, V)$ is minimized when the tessellation V is the Voronoi diagram of the sites.*

The proof of this lemma can be found in [Du et al., 1999].

We have the similar property for spherical CVT energy as shown in the following lemma:

Lemma 2. *When the sites are fixed, the CVT energy $F^{\mathbb{S}}(\mathbf{S}, V)$ is maximized when the tessellation V is the Voronoi diagram of the sites.*

Proof. Suppose we have a tessellation V which is different to the Voronoi diagram $\Omega = (\Omega_1^{\mathbb{S}}, \Omega_2^{\mathbb{S}}, \dots, \Omega_n^{\mathbb{S}})$. For a particular point \mathbf{p} belonging to a Voronoi cell $\Omega_i^{\mathbb{S}}$, suppose in the tessellation V it belongs to a cell $V_j^{\mathbb{S}}$. Because of the definition of the Voronoi diagram, we always have

$$d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i) \leq d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_j).$$

Note $0 \leq d^{\mathbb{S}}(\mathbf{a}, \mathbf{b}) \leq \pi$ for any two points \mathbf{a} and \mathbf{b} . So,

$$\rho(\mathbf{p}) \cos(d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i)) \geq \rho(\mathbf{p}) \cos(d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_j)).$$

Since V is not a Voronoi diagram, the above formula must hold with strict inequality on some cells. Thus,

$$F^{\mathbb{S}}(\mathbf{S}, \Omega) > F^{\mathbb{S}}(\mathbf{S}, V)$$

so that $F^{\mathbb{S}}(\mathbf{S}, V)$ is maximized when the tessellation is the Voronoi diagram of the sites. \square

To be consistent with the Euclidean space and hyperbolic space, we can slightly modify the definition of the CVT energy in spherical space as:

$$\begin{aligned} F^{\mathbb{S}}(\mathbf{S}, V) &= \sum_{i=1}^n \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \cos(\pi - d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i)) \, d\sigma \\ &= - \sum_{i=1}^n \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \cos(d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i)) \, d\sigma \\ &= - \sum_{i=1}^n \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \langle \mathbf{p}, \mathbf{s}_i \rangle \, d\sigma. \end{aligned} \quad (8)$$

Obviously, for the CVT energy under this definition, we have:

Lemma 3. *When the sites are fixed, the CVT energy $F^{\mathbb{S}}(\mathbf{S}, V)$ is minimized when the tessellation V is the Voronoi diagram of the sites.*

To distinguish these two definitions, we denote the two CVT energy functions defined by Equation (6) and (8) by $F_{max}^{\mathbb{S}}(\mathbf{S}, V)$ and $F_{min}^{\mathbb{S}}(\mathbf{S}, V)$ respectively.

This relationship between the CVT energy and the Voronoi diagram also holds in hyperbolic space, and we have:

Lemma 4. *When the sites are fixed, the CVT energy $F^{\mathbb{H}}(\mathbf{S}, V)$ is minimized when the tessellation V is the Voronoi diagram of the sites.*

Proof. The proof is almost same as Lemma 2 and is omitted here. \square

4.3. CVT Energy and Centroid

Next we study the situation when the tessellation is fixed and the sites moves. For simplicity, under this situation, we ignore the tessellation and write the energy as $F(\mathbf{S})$ (with corresponding superscripts for different spaces). In Euclidean space, we have:

Lemma 5. *When the tessellation is fixed, the CVT energy $F^{\mathbb{E}}(\mathbf{S})$ is minimized when the sites locate at centroids of their corresponding parts.*

The proof of this lemma is also provided in [Du et al., 1999].

Again, this property also holds in spherical space as shown in the following lemma:

Lemma 6. *When the tessellation is fixed, the CVT energy $F_{min}^{\mathbb{S}}(\mathbf{S})$ is minimized, and the CVT energy $F_{max}^{\mathbb{S}}(\mathbf{S})$ is maximized, when the sites locate at centroids of their corresponding parts.*

Proof. We first study how to minimize the partial CVT energy on the part $V_i^{\mathbb{S}}$ corresponding to site \mathbf{s}_i :

$$\begin{aligned} F_{min}^{\mathbb{S}}(\mathbf{s}_i) &= \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \cos(\pi - d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i)) \, d\sigma \\ &= - \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \cos(d^{\mathbb{S}}(\mathbf{p}, \mathbf{s}_i)) \, d\sigma \\ &= - \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \langle \mathbf{p}, \mathbf{s}_i \rangle \, d\sigma. \end{aligned}$$

To find the minimum point of $F_{min}^{\mathbb{S}}(\mathbf{s}_i)$, we write the problem as a non-linear programming problem:

$$\begin{cases} \min_{\mathbf{s}_i \in \mathbb{R}^3} F_{min}^{\mathbb{S}}(\mathbf{s}_i) \\ g(\mathbf{s}_i) = \langle \mathbf{s}_i, \mathbf{s}_i \rangle - 1 = 0 \end{cases},$$

where $g(\mathbf{s}_i)$ is the equation of the model of spherical space (the unit sphere). The Lagrange multiplier rule states that the solution of this problem is the stationary point of

$$L(\mathbf{s}_i) = F_{min}^{\mathbb{S}}(\mathbf{s}_i) + \lambda g(\mathbf{s}_i) = - \int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \langle \mathbf{p}, \mathbf{s}_i \rangle \, d\sigma + \lambda g(\mathbf{s}_i),$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier. The necessary condition is

$$\begin{cases} \nabla_{\mathbf{s}_i} L(\mathbf{s}_i) = \nabla_{\mathbf{s}_i} F_{min}^{\mathbb{S}}(\mathbf{s}_i) + \lambda \nabla_{\mathbf{s}_i} g(\mathbf{s}_i) = 0 \\ \nabla_{\lambda} L(\mathbf{s}_i) = g(\mathbf{s}_i) = 0 \end{cases}.$$

From the first equation, we have

$$-\int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \mathbf{p} \, d\sigma + 2\lambda \mathbf{s}_i = 0.$$

So

$$\mathbf{s}_i = \frac{\int_{V_i^{\mathbb{S}}} \rho(\mathbf{p}) \mathbf{p} \, d\sigma}{2\lambda}.$$

Comparing this result with Equation (3), we can see these two vectors are collinear. Also, the second necessary condition $g(\mathbf{s}_i) = 0$ limits \mathbf{s}_i on the unit sphere. So \mathbf{s}_i is either the spherical centroid defined by Equation (3) (i.e. $\mathbf{s}_i = \mathbf{c}_i^{\mathbb{S}}$), or its anti-point on the other end of the diameter passing through the centroid (i.e. $\mathbf{s}_i = -\mathbf{c}_i^{\mathbb{S}}$). It is easy to verify that $F_{min}^{\mathbb{S}}(\mathbf{s}_i)$ has smaller value at $\mathbf{c}_i^{\mathbb{S}}$. So we proved the centroid of part $V_i^{\mathbb{S}}$ minimizes the partial CVT energy over it.

If all sites locate at centroids of their corresponding parts, all partial CVT energy functions are minimized. As a result, the sum of them, the CVT energy $F_{min}^{\mathbb{S}}(\mathbf{S})$ is minimized.

Using the same derivation, we can prove when all sites locate at centroids of their corresponding parts, the CVT energy $F_{max}^{\mathbb{S}}(\mathbf{S})$ is maximized. \square

To be consistent with Euclidean space and hyperbolic space, from now on, when we mention the CVT energy in spherical space, we always mean $F_{min}^{\mathbb{S}}(\mathbf{S})$ and thus write it as $F^{\mathbb{S}}(\mathbf{S})$.

In Section 3.1, we have proved that the Voronoi diagram in spherical space is same as the constrained Voronoi diagram on the sphere. We can also define the CVT energy for this constrained Voronoi diagram in 3D Euclidean space:

$$\begin{aligned} F_c^{\mathbb{E}}(\mathbf{S}) &= \sum_{i=1}^n \int_{V_{c,i}} \rho(\mathbf{p}) (d^{\mathbb{E}}(\mathbf{p}, \mathbf{s}_i))^2 \, d\sigma \\ &= \sum_{i=1}^n \int_{V_{c,i}} \rho(\mathbf{p}) \|\mathbf{p} - \mathbf{s}_i\|^2 \, d\sigma, \end{aligned} \quad (9)$$

where $V_{c,i}$ is the constrained Voronoi cell of site \mathbf{s}_i , i.e. the intersection between the sphere and the 3D Voronoi cell $\Omega_i^{\mathbb{E}}$. The point $\mathbf{c}_i^* \in V_{c,i}$ which minimizes the partial CVT energy $F_c^{\mathbb{E}}(\mathbf{s}_i)$ is defined as the *constrained centroid* of V_i [Du et al., 2003]. It is proved in [Du et al., 2003] that if \mathbf{c}_E is the centroid of $V_{c,i}$ in 3D Euclidean space, the vector $\mathbf{c}_E \mathbf{c}^*$ is collinear with the normal vector at \mathbf{c}^* . For sphere, this indicates that \mathbf{c}^* is the central projection of \mathbf{c}_E onto the sphere. So the constrained centroid \mathbf{c}^* is same as the spherical centroid $\mathbf{c}_i^{\mathbb{S}}$ defined by Equation (3), and the spherical CVT is thus identical to the constrained CVT on the sphere.

When the tessellation is fixed, the CVT energy in hyperbolic space also has the similar property:

Lemma 7. *When the tessellation is fixed, the CVT energy $F^{\mathbb{H}}(\mathbf{S})$ is minimized when the sites locate at centroids of their corresponding parts.*

Proof. The proof is similar to that of spherical space, and the non-linear programming problem we need to solve is:

$$\begin{cases} \min_{\mathbf{s}_i \in \mathbb{R}^3} F^{\mathbb{H}}(\mathbf{s}_i) \\ g(\mathbf{s}_i) = \langle \mathbf{s}_i, \mathbf{s}_i \rangle_M - 1 = 0 \end{cases},$$

where $g(\mathbf{s}_i)$ is the equation of the model of hyperbolic space (Minkowski model). Using the Lagrange multiplier rule, we have

$$\int_{V_i^{\mathbb{H}}} \rho(\mathbf{p}) \tilde{\mathbf{p}} \, d\sigma + 2\lambda \tilde{\mathbf{s}}_i = 0,$$

where the notation $\tilde{\mathbf{a}}$ represent a vector whose first two components are negative of those of \mathbf{a} , and the third component is same, i.e. $\tilde{\mathbf{a}} = (-x_{\mathbf{a}}, -y_{\mathbf{a}}, z_{\mathbf{a}})^T$. We can split this vector equation into three scalar equations, multiply -1 to both sides of the first two equations, and combine them again to have

$$\int_{V_i^{\mathbb{H}}} \rho(\mathbf{p}) \mathbf{p} \, d\sigma + 2\lambda \mathbf{s}_i = 0.$$

From this equation on, we can use the same argument as in Lemma 6 to finish the proof. \square

5. Computing Centroidal Voronoi Tessellations

Lloyd’s algorithm is the most widely used algorithm to compute the CVT in Euclidean space. We use Lloyd’s algorithm to compute the spherical CVT and hyperbolic CVT in this paper. Note that it is also possible to use the L-BFGS algorithm to compute them as in [Liu et al., 2009].

5.1. Lloyd’s Algorithm

Lloyd’s algorithm is an iterative algorithm to minimize the CVT energy. It starts with an arbitrary set of initial sites. In every iteration of Lloyd’s algorithm, the Voronoi diagram of current sites is first computed. Next, the centroids of Voronoi cells are computed and used as new sites for next iteration. This procedure is repeated until certain stopping condition is satisfied (e.g. the moving distance of every site is smaller than a threshold).

In Euclidean space, Lloyd’s algorithm is proved to be converged [Du et al., 2006]. The basis of the proof is Lemma 1 and 5. Since we have proved similar lemmas in both spherical space (Lemma 3 and 6) and hyperbolic space (Lemma 4 and 7), we can use the same proofs of Theorem 2.3, Corollary 2.4, Theorem 2.5, and Theorem 2.6 in [Du et al., 2006] to prove the convergence (with respect to CVT energy) of Lloyd’s algorithm in both spherical space and hyperbolic space, i.e. Lloyd’s algorithm always converges to a configuration corresponding to a local minimum of the CVT energy. Note that same as the case in Euclidean space, the stationary point found by Lloyd’s algorithm in spherical space or hyperbolic space may be either a local minimum point or a saddle point of the CVT energy.

For spherical CVT, as it is identical to the constrained CVT on the sphere, we can actually compute a constrained CVT as in [Du et al., 2003].

The case of hyperbolic CVT is more complicated. Since the hyperbolic Voronoi diagram is computed in Klein disk and the hyperbolic centroid is defined in Minkowski model, we need to convert positions between these two models. In the following of this section, we use normal letters to represent points in Minkowski model, and letters with bars for points in Klein disk. Furthermore, we use letters with primes to represent points on the plane $z = 0$.

When embedded in the plane $z = 1$ with the center on z -axis, Klein disk is the central projection of Minkowski model with respect to the origin point. The correspondence between a point $\mathbf{p}(x_{\mathbf{p}}, y_{\mathbf{p}}, z_{\mathbf{p}})$ in Minkowski model and

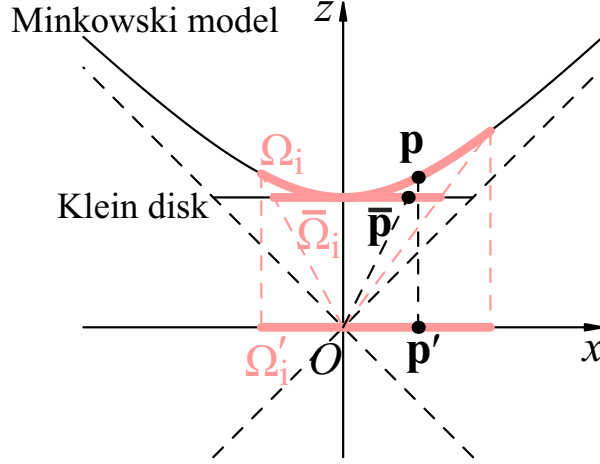


Figure 6: Illustration of mapping functions φ and ψ .

a point $\bar{\mathbf{p}}(\bar{x}_{\mathbf{p}}, \bar{y}_{\mathbf{p}})$ in Klein disk is given by the following formulas:

$$\begin{cases} \bar{x}_{\mathbf{p}} = x_{\mathbf{p}}/z_{\mathbf{p}} \\ \bar{y}_{\mathbf{p}} = y_{\mathbf{p}}/z_{\mathbf{p}} \end{cases} \quad (10)$$

$$\begin{cases} x_{\mathbf{p}} = \bar{x}_{\mathbf{p}}/\sqrt{1 - (\bar{x}_{\mathbf{p}}^2 + \bar{y}_{\mathbf{p}}^2)} \\ y_{\mathbf{p}} = \bar{y}_{\mathbf{p}}/\sqrt{1 - (\bar{x}_{\mathbf{p}}^2 + \bar{y}_{\mathbf{p}}^2)} \\ z_{\mathbf{p}} = 1/\sqrt{1 - (\bar{x}_{\mathbf{p}}^2 + \bar{y}_{\mathbf{p}}^2)} \end{cases} \quad (11)$$

Formula (11) define a mapping function φ from Klein disk to Minkowski model, where $\varphi(\bar{\mathbf{p}}) = \mathbf{p}$. We also define another mapping function ψ which orthogonally projects a point \mathbf{p} in Minkowski model to the plane $z = 0$ to get the point \mathbf{p}' , i.e. $\psi(\mathbf{p}) = \mathbf{p}'$. Note that these two mapping functions can be naturally extended to Voronoi cells:

$$\varphi(\bar{\Omega}_i^{\mathbb{H}}) = \bigcup_{\bar{\mathbf{p}} \in \bar{\Omega}_i^{\mathbb{H}}} \varphi(\bar{\mathbf{p}}) = \Omega_i^{\mathbb{H}}, \psi(\Omega_i^{\mathbb{H}}) = \bigcup_{\mathbf{p} \in \Omega_i^{\mathbb{H}}} \psi(\mathbf{p}) = \Omega_i^{\mathbb{H}}.$$

Figure 6 illustrates these two mapping functions.

Last, we would like to emphasize that the above mapping functions are only for centroid computations. The hyperbolic Voronoi diagram is directly computed as a power diagram in Euclidean space as explained in Section 3.

5.2. Implementation Details

Lloyd’s algorithm requires to compute the Voronoi diagram as the first step. For spherical space, we use STRIPACK [Renka, 1997] to compute the Voronoi diagram on a sphere. For hyperbolic space, as discussed above, computing a hyperbolic Voronoi diagram in Klein disk can be easily accomplished by computing a power diagram in Euclidean space. In our implementation, we use the CGAL library [Fabri, 2001] to compute the power diagram.

The next step of Lloyd’s algorithm is to compute the centroid for every Voronoi cell. In hyperbolic space, although the centroid of a triangle with constant density is proved to be coincident with the common point of its three medians [Stahl, 2007], for a triangle with non-constant density, it is not known how to get a close-form solution of its centroid defined by Equation (4). The same difficulty exists in spherical space. As a result, we cannot compute the centroid of a Voronoi cell by dividing it into several triangles as we did in Euclidean space. Instead, we use summations to approximate the integral to compute centroids in spherical and hyperbolic spaces.

For every spherical Voronoi cell $\Omega_i^{\mathbb{S}}$, we first apply a rigid transformation (i.e. rotation) to transform its site onto z -axis (point $(0, 0, 1)$). The transformed Voronoi cell is then orthogonally projected to the plane $z = 0$ as $\Omega_i^{\mathbb{S}}$. The plane $z = 0$ is uniformly sampled using a regular grid. We perform the summation over all samples located within $\Omega_i^{\mathbb{S}}$ to approximate the integral in Equation (3).

Similarly, for every hyperbolic Voronoi cell $\bar{\Omega}_i^{\mathbb{H}}$ in Klein disk, we first apply a Möbius transformation, the rigid motion in hyperbolic plane, to move its site \mathbf{s}_i to the origin to achieve a relatively small numerical error. Then we use the mapping functions φ and ψ to map the $\bar{\Omega}_i^{\mathbb{H}}$ from Klein disk to Minkowski model, and then to the plane $z = 0$, to get $\Omega_i^{\mathbb{H}} = \psi(\varphi(\bar{\Omega}_i^{\mathbb{H}}))$. The plane $z = 0$ is uniformly sampled using a regular grid. We perform the summation over all samples located within $\Omega_i^{\mathbb{H}}$ to approximate the integral in Equation (4).

We plot in Figure 7(a) and Figure 7(b) the spherical CVT energy $F_{max}^{\mathbb{S}}(\mathbf{S})$ and $F_{min}^{\mathbb{S}}(\mathbf{S})$ against the number of iterations starting from the 500 sites shown in Figure 1(a). The hyperbolic CVT energy $F^{\mathbb{H}}(\mathbf{S})$ corresponding to Figure 2 is shown in Figure 7(c). These results verify the lemmas proved in Section 4. Similar to in Euclidean space, the energy changes dramatically for the first several iterations, and converges gradually to a local minimum (maximum). Note although the CVT energy should monotonically decrease or increase, there is slight perturbation in the curves due to our approxima-

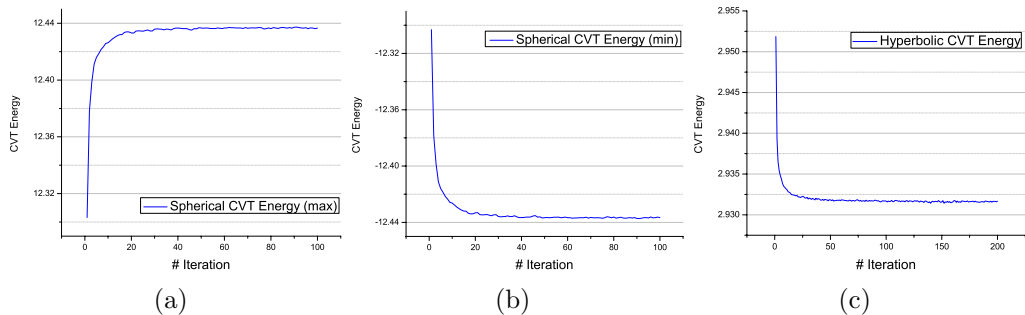


Figure 7: CVT energy against the number of iterations for Lloyd’s algorithm. (a) and (b) starts from the 500 initial sites shown in Figure 1(a). (c) starts from the 100 initial sites shown in Figure 2(a).

tion of the integral for centroid computation.

5.3. Uniformity of Centroidal Voronoi Tessellation

Gershó’s conjecture states that the sites in a Euclidean CVT are uniformly distributed in the space [Gershó, 1979]. We conjecture that it is also true for the spherical CVT and the hyperbolic CVT. Since it is difficult to visually tell the uniformity of the sites in these CVT results, we measure the geometrical uniformity of the sites and Voronoi cells. For every site \mathbf{s}_i , we define the radius r_i of its Voronoi cell, the distance d_i to its nearest neighbors, and the area a_i of its Voronoi cell as follows:

$$r_i = \max_{\mathbf{p} \in \Omega_i} d(\mathbf{p}, \mathbf{s}_i), \quad d_i = \min_{j \neq i} d(\mathbf{s}_i, \mathbf{s}_j), \quad a_i = Area(\Omega_i),$$

where $d(\cdot, \cdot)$ and $Area(\cdot)$ denote the distance between two points and the area of a Voronoi cell computed in the corresponding space respectively. For each quality measure, smaller variance means better uniformity of the sites. Figure 8 compares the three measures computed for the Voronoi diagram of 500 initial sites (Figure 1(a)) in spherical space and the spherical CVT (Figure 1(b)). Figure 9 compares the three measures computed for the Voronoi diagram of 100 initial sites in hyperbolic space (Figure 2(a)) and the hyperbolic CVT (Figure 2(b)). It is clear that the sites in both the spherical CVT and the hyperbolic CVT are very uniformly distributed.

In 2D Euclidean space, it has been proved that most of Voronoi cells in a CVT are hexagons [Newman, 1982]. We conjecture that this is also true for both spherical and hyperbolic CVTs. We have also experimentally

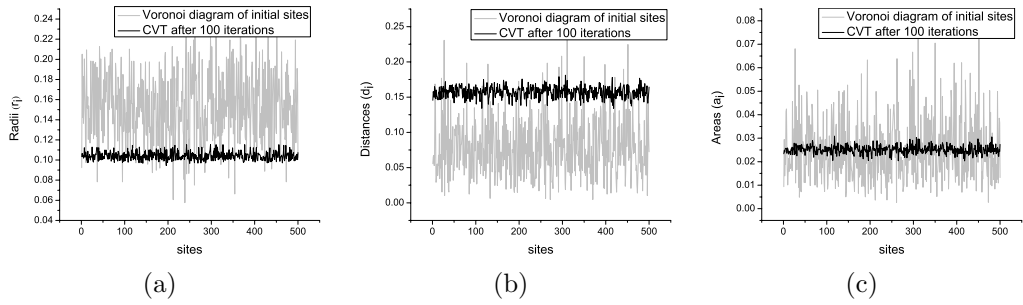


Figure 8: Comparison of (a) radius of Voronoi cell, (b) distance to nearest neighbor, and (c) area of Voronoi cell of the Voronoi diagram of 500 initial sites shown in Figure 1(a) (gray curve) and the spherical CVT shown in Figure 1(b) (black curve).

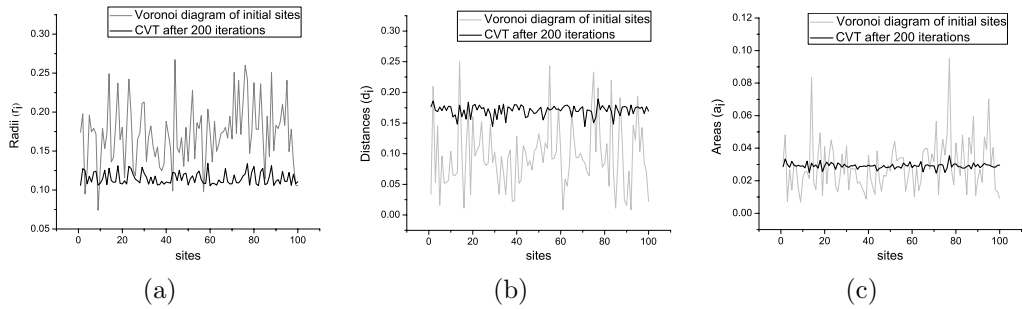


Figure 9: Comparison of (a) radius of Voronoi cell, (b) distance to nearest neighbor, and (c) area of Voronoi cell of the Voronoi diagram of 100 initial sites shown in Figure 2(a) (gray curve) and the hyperbolic CVT shown in Figure 2(b) (black curve).

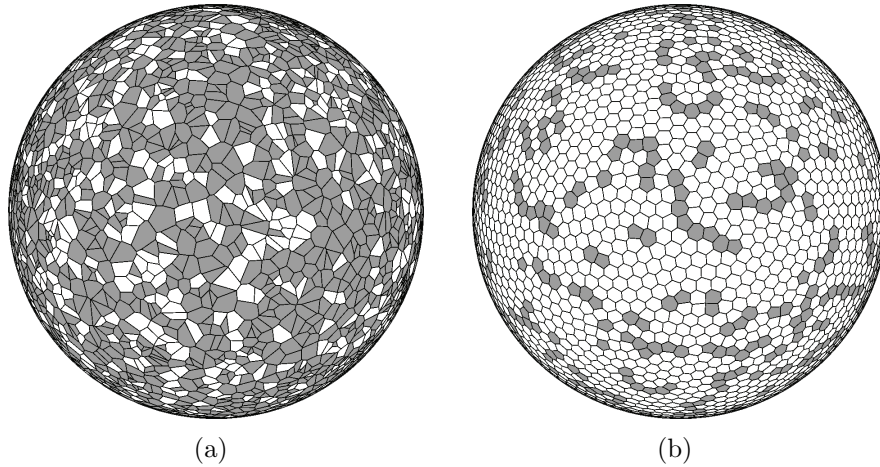


Figure 10: Hexagonal (unshaded) Voronoi cells in (a) the Voronoi diagram of 5,000 random initial sites on a sphere, and (b) the spherical CVT generated from (a). $\rho(\mathbf{p}) \equiv 1$ in this example.

confirmed this conclusion. So more hexagon cells means a better uniformity of the sites. Two examples are shown in Figure 10 and Figure 11 where we show hexagonal Voronoi cells as unshaded. It is clear that most of Voronoi cells in the CVT results are hexagons, and the CVT results have much more hexagon cells than the initial Voronoi diagrams.

Although the above CVT results are all computed with a constant density value $\rho(\mathbf{p}) \equiv 1$, our definition of the CVT is general for any density values. Figure 12 shows two examples with non-constant density values. As we can see, similar to the behavior in Euclidean space, in both spherical and hyperbolic CVTs, sites tend to cluster near the regions with higher density values. This property is critical for many applications in geometric modeling (see Section 6 for details).

6. Applications for Surface Modeling

We have given detailed definitions and analysis of the CVT in spherical, Euclidean, and hyperbolic spaces respectively. In this section, we put these three types of CVT into a unified framework – the *universal covering space*. We also show how to apply the CVT in universal covering space on applications of surface modeling.

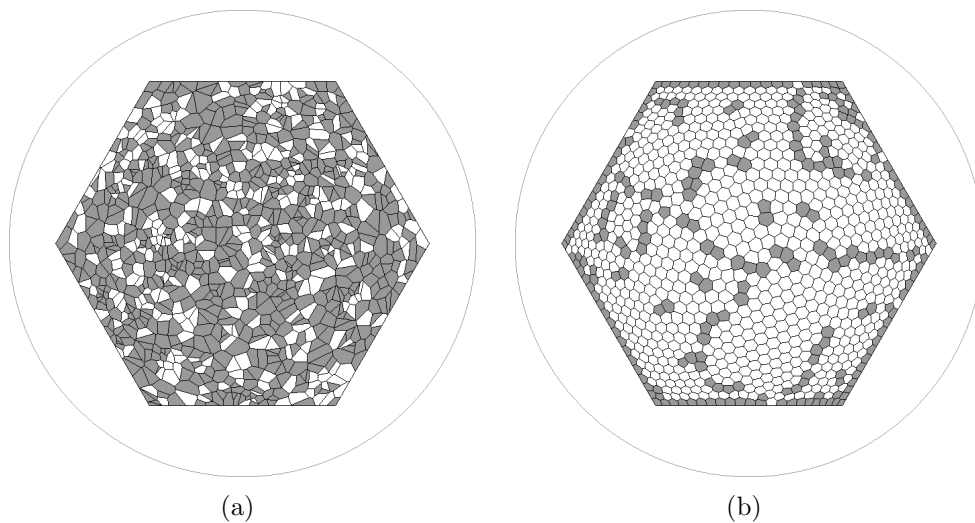


Figure 11: Hexagonal (unshaded) Voronoi cells in (a) the Voronoi diagram of 1,000 random initial sites in a hexagon in Klein disk, and (b) the hyperbolic CVT generated from (a). $\rho(\mathbf{p}) \equiv 1$ in this example.

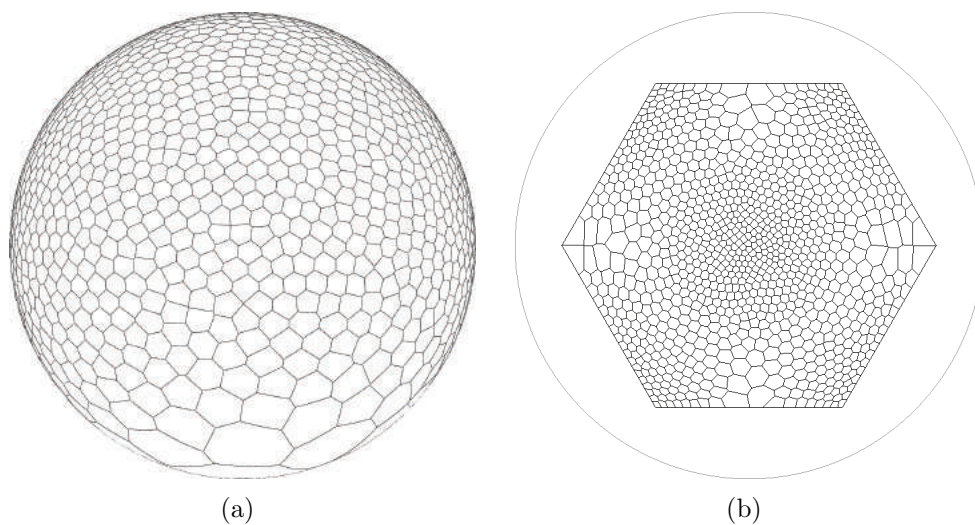


Figure 12: (a) Spherical CVT of 2,562 sites with $\rho(\mathbf{p}) = e^{-6(1-z_{\mathbf{p}})^2}$. (b) Hyperbolic CVT generated from same initial sites as in Figure 11 with $\rho(\mathbf{p}) = e^{-20x_{\mathbf{p}}^2 - 20y_{\mathbf{p}}^2} + 0.05 \sin^2(\pi x_{\mathbf{p}}) \sin^2(\pi y_{\mathbf{p}})$.

6.1. Homotopy Group and Universal Covering Space

This section briefly introduces the necessary background of algebraic topology for the current work. We refer readers to [Munkres, 1996] for details.

Two curves are *homotopic* to each other, if they can deform to each other on the surface. Let \mathcal{S} be a topological surface, and \mathbf{p} be a point of \mathcal{S} . All loops with base point \mathbf{p} are classified by homotopy relations. All homotopy equivalence classes form the homotopy group or fundamental group $\pi(\mathcal{S}, \mathbf{p})$, where the product is defined as the concatenation of two loops through their common base point.

A *covering space* of \mathcal{S} is a space $\bar{\mathcal{S}}$ together with a continuous surjective map $h : \bar{\mathcal{S}} \rightarrow \mathcal{S}$, such that for every $\mathbf{p} \in \mathcal{S}$ there exists an open neighborhood \mathcal{U} of \mathbf{p} such that $h^{-1}(\mathcal{U})$ (the inverse image of \mathcal{U} under h) is a disjoint union of open sets in $\bar{\mathcal{S}}$, each of which is mapped homeomorphically onto \mathcal{U} by h . The map h is called the *covering map*. A simply connected covering space is a *universal covering space*.

A deck transformation of a cover $h : \bar{\mathcal{S}} \rightarrow \mathcal{S}$ is a homeomorphism $f : \bar{\mathcal{S}} \rightarrow \bar{\mathcal{S}}$ such that $h \circ f = h$. All deck transformations form a group, the so-called deck transformation group, which is isomorphic to the fundamental group.

A *fundamental domain* \mathcal{F} is a subset of $\bar{\mathcal{S}}$, such that the universal covering space is the union of conjugates of \mathcal{F} , and any two conjugates have no interior point in common. For a genus g closed surface \mathcal{S} , one can find a set of canonical homotopy group generators $a_1, b_1, a_2, b_2, \dots, a_g, b_g$, such that 1) all the curves meet at a single base point v ; 2) each pair of curves a_i, b_i algebraically intersect each other exactly once; and 3) no curve in another pair a_j, b_j algebraically intersects either of a_i, b_i . We can slice \mathcal{S} along the set of curves and get a fundamental domain with boundary $a_1 b_1 a_1^{-1} b_1^{-1} a_2 b_2 a_2^{-1} b_2^{-1} \dots a_g b_g a_g^{-1} b_g^{-1}$.

For any surface \mathcal{S} , its uniformization metric is also a metric for its universal covering space $\bar{\mathcal{S}}$ which can be isometrically embedded into one of the three canonical spaces: sphere, plane and hyperbolic space, given by the following Uniformization Theorem which was first proved by Koebe and Poincaré independently in 1907. More details can be found in [Chan, 2004].

Theorem 1 (Uniformization Theorem). *Let $(\mathcal{S}, \mathbf{g})$ be a compact two-dimensional surface with a Riemannian metric \mathbf{g} , then there is a metric $\bar{\mathbf{g}}$ conformal to \mathbf{g} with constant Gaussian curvature everywhere; the constant can be made to be one of $\{-1, 0, 1\}$.*

We call such a metric the *uniformization metric* of \mathcal{S} . According to the Gauss-Bonnet theorem, the sign of the constant Gaussian curvature must match the sign of the Euler number χ of the surface: -1 for $\chi(\mathcal{S}) < 0$, 0 for $\chi(\mathcal{S}) = 0$, and $+1$ for $\chi(\mathcal{S}) > 0$.

Since conformal metric only introduces area distortion, which can be easily compensated by modifying the density function expressed in the embedding domain [Alliez et al., 2005], we use uniformization metric to parameterize surfaces of different genera to universal covering space in different spaces. The usage of the universal covering space allows sites to move freely everywhere on the surface: a site can cross the boundary of the fundamental domain, and come into the fundamental domain from the “opposite” side of the boundary. So there is no artifacts along the cutting edges any more. This is also the major advantage of our method over Alliez et al.’s algorithm [Alliez et al., 2005].

For a given closed genus-0 surface, its universal covering space is itself. We apply the spherical harmonic method [Gu et al., 2004] to compute its conformal embedding in unit sphere. A finite portion of the universal covering space of a given closed genus-1 surface is computed and conformally embedded in Euclidean plane based on discrete Euclidean Ricci flow [Jin et al., 2008]. For a given closed high-genus surface, we use discrete hyperbolic Ricci flow [Jin et al., 2006] to compute its hyperbolic uniformization metric and construct the embedding of a finite portion of its universal covering space in Klein disk. We show an example of a genus-2 surface and the corresponding universal covering space in hyperbolic space to better illustrate the concept of universal covering space.

A double torus surface (genus-2) is given in Figure 13(a), with a set of canonical fundamental group generators (black loops a_1, b_1, a_2, b_2) which cut the surface into a topological disk, the fundamental domain, with 8 sides: $a_1, b_1, a_1^{-1}, b_1^{-1}, a_2, b_2, a_2^{-1}, b_2^{-1}$. The fundamental domain is isometrically embedded in Klein disk with marked sides as shown in the unshaded region in Figure 13(b). A finite copies of the fundamental domain (shaded regions in Figure 13(b)) are glued coherently by applying corresponding Möbius transformations. Note that any two domains in universal covering space only differ by a rigid motion of that embedding space. In hyperbolic space, the rigid motion is a Möbius transformation, which can be computed quickly as in [Jin et al., 2006]. In Euclidean space, the rigid motion is simply a translation, which can be easily computed from any corresponding pair of points from the two domains. In spherical space, one fundamental domain, also the surface

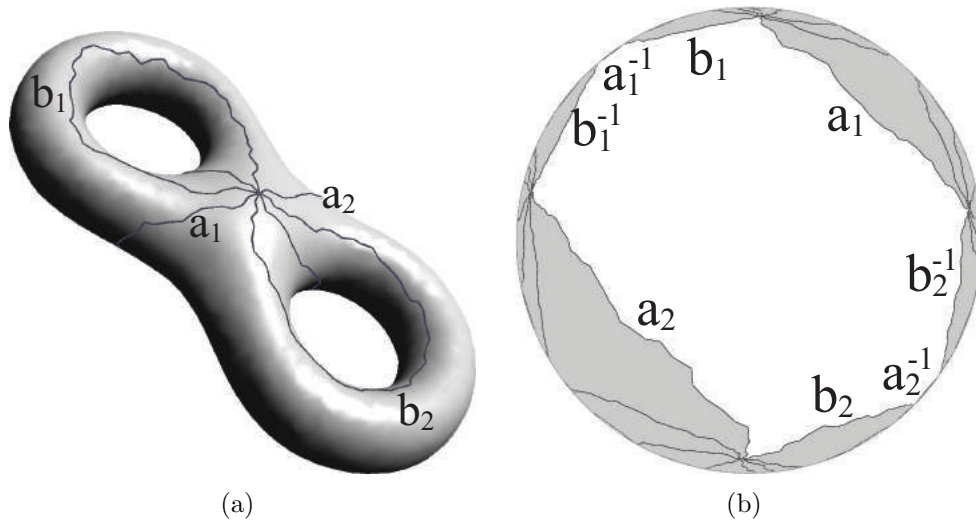


Figure 13: Canonical fundamental group generators on (a) the surface and (b) Klein disk. Fundamental domain is shown unshaded, and neighbor domains shaded.

itself, covers the whole space, we do not need to compute the rigid motion.

6.2. Centroidal Voronoi Tessellation in Universal Covering Space

We adapt Lloyd’s algorithm to compute the CVT of a set of initial sites \mathbf{S} in the embedded fundamental domain of the universal covering space. For Euclidean space and hyperbolic space, by applying the rigid motion on sites in \mathbf{S} , we compute the corresponding points in neighbor domains (those sharing an edge or a vertex with the fundamental domain) and denote the union of them by \mathbf{S}' ($\mathbf{S}' = \emptyset$ for spherical space). The Voronoi diagram of $\mathbf{S} \cup \mathbf{S}'$ is computed, and the centroids of Voronoi cells of sites in \mathbf{S} are located. If a centroid is outside of the fundamental domain by one side, say a_1 (see Figure 13(b)), by performing a corresponding rigid motion we move it to the “opposite” side a_1^{-1} of the fundamental domain. The adjusted centroids are all inside the fundamental domain and are used as the new sites in the next iteration.

Back to the example of the double torus surface and the conformal embedding of the universal covering space in Klein disk, Figure 14(a) shows the Voronoi diagram of 100 random initial sites (solid black dots) inside the fundamental domain and their corresponding points (solid white dots) in neighbor domains in universal covering space. The adjusted centroids (hol-

low dots) for Voronoi cells of sites in \mathbf{S} are all inside the fundamental domain.

As discussed before, conformal metric introduces area distortion only, which can be compensated by assigning an appropriate density value at each point expressed in the embedding domain.

The *conformal factor* cf is defined to measure the local scaling of conformal mapping. For each vertex \mathbf{v} on the surface, cf can be computed as the ratio of the area sum of its incident triangles in 3D space and in 2D parametrization domain, i.e. $cf(\mathbf{v}) = \frac{A_{3D}(\mathbf{v})}{A_{2D}(\mathbf{v})}$. For a non-vertex point on the surface, cf can be computed by linearly interpolating the computed conformal factors of the three vertices of the triangle containing that point.

We then define a *sizing field* where every point \mathbf{p} on the surface has a desired size $\mu(\mathbf{p})$. For every triangle t , its sizing ratio is computed as $sr(t) = \frac{\text{longest_edge}(t)}{\mu(\text{centroid}(t))}$. For a given sizing field, we say a triangle mesh satisfies it if the sizing ratio of every triangle is less than or equal to 1. For the ideal case, the sizing ratios of all triangles should be 1 to minimize the number of triangles used. So the length of the longest edge of every triangle is approximately equal to the sizing at its centroid. The area of the triangle is thus proportional to the square of the sizing at its centroid, i.e. $A(t) \sim \mu(\text{centroid}(t))^2$. Since we want the sites uniformly distributed on the surface, the sizing field on the surface should be a constant value everywhere. After a proper normalization, we have $A_{3D}(\mathbf{p}) \sim 1$ and $A_{2D}(\mathbf{p}) \sim \mu(\mathbf{p})^2$, thus the conformal factor $cf(\mathbf{p}) = \frac{1}{\mu(\mathbf{p})^2}$. It is pointed out by Du and Wang [2006] that the dual mesh of a CVT with density values $\rho(\mathbf{p}) = \frac{1}{\mu(\mathbf{p})^{d+2}}$ satisfies the given sizing field, where d is the dimension of the problem. In 2D, we assign the density value at point \mathbf{p} as:

$$\rho(\mathbf{p}) = \frac{1}{\mu(\mathbf{p})^4} = cf(\mathbf{p})^2. \quad (12)$$

So the uniformity of the sites in the CVT result is compensated by the distortion, and thus uniformly distributed on the surface.

With the modulated density values (instead of a constant), the CVT result generated by 100 Lloyd's iterations from the initial sites shown in Figure 14(a) is given in Figure 14(b). The Voronoi diagram of the initial sites and the CVT results on the surface are shown in Figure 14(c) and 14(d) respectively. As we can see the final sites are uniformly distributed on the surface. Figure 15 shows the initial Voronoi diagram and the corresponding

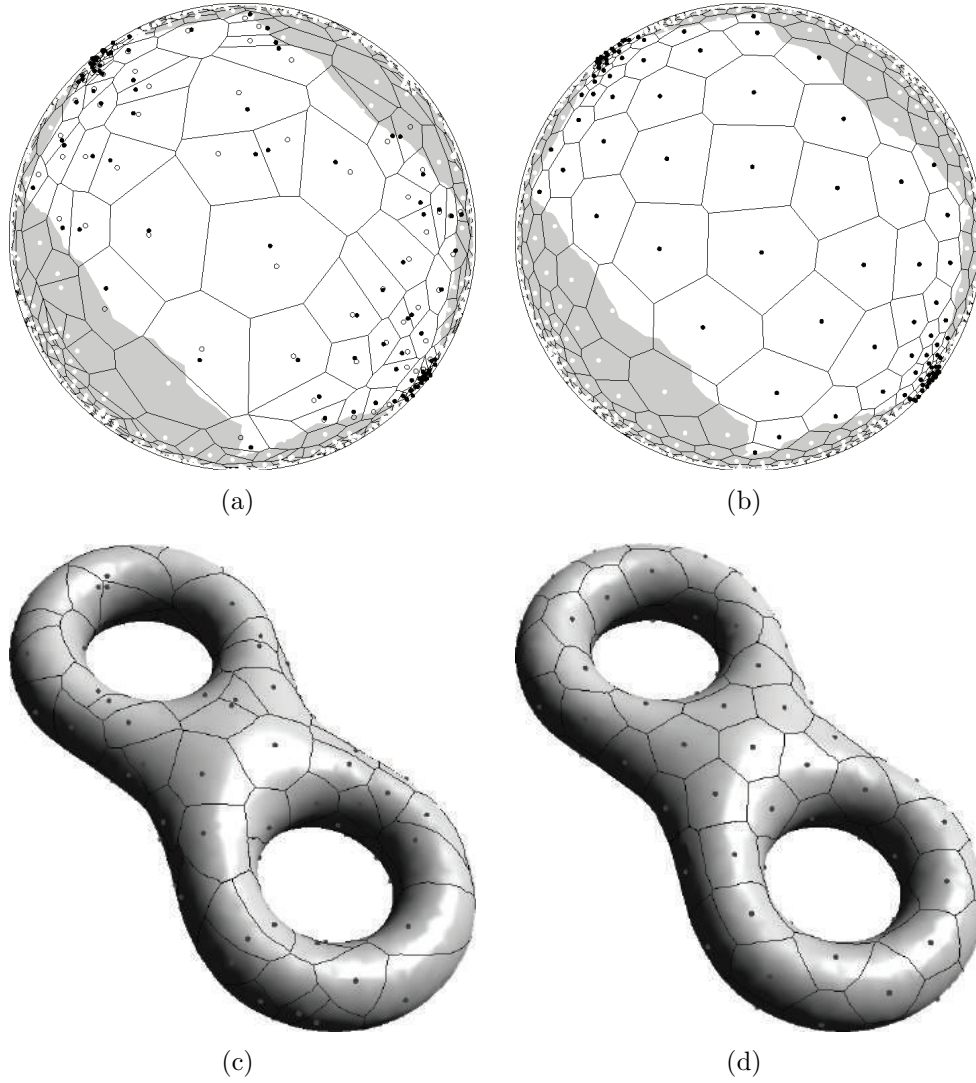


Figure 14: (a) Voronoi diagram of 100 random initial sites in the universal covering space, and the centroids for Voronoi cells of sites in \mathbf{S} . Solid black dots are sites in \mathbf{S} , solid white dots are sites in \mathbf{S}' , and hollow dots are centroids. (b) CVT result generated from (a). (c) Initial Voronoi diagram on surface. (d) CVT result on surface.

Figure	Surface	Genus	#Sites	Time Per Iter. (sec.)	Total Time (sec.)
Fig. 14	Double Torus	2	100	0.38	76.48
Fig. 15	Kitten	1	1000	0.26	52.48
Fig. 16	Igea	0	2000	13.57	2714.01
Fig. 17	Knot	2	2000	25.35	5070.22
Fig. 18(a)	David	3	1000	36.79	7358.53
Fig. 18(b)	Sculpture	3	1000	55.24	11048.36

Table 1: Running time of computation of CVT for different surfaces.

CVT results both in Euclidean parametrization domain and on the genus-1 surface Kitten. Figure 16 shows the results both in spherical parametrization domain and on the genus-0 surface Igea.

Figure 17 shows results for a genus-2 surface Knot with much more complicated shape. It is clear that our algorithm using the hyperbolic CVT can generate high quality results for relatively few sites on complicated surfaces. Two more results for genus-3 surfaces are shown in Figure 18.

We list in Table 1 the running time of our programs on a platform with Intel Core 2 Duo 2.93GHz CPU and 2GB DDR2 RAM. Note that our implementations are not optimized, and there is much space to further improve the speed, which is beyond the scope of this paper. For simplicity, in all the experiments, we use a fix number (200) of iteration as the terminating condition.

Figure 19 compares our algorithm with Yan et al.’s fast restricted Voronoi diagram algorithm [Yan et al., 2009], which uses Euclidean distance to compute a 3D Voronoi diagram, and then finds the intersection between the 3D Voronoi diagram and the surface. For the surface Knot, part of the inner knot is very close to the outer surface, but they are far away along the surface. From our experiments, even with 4,000 sites, some Voronoi cells in the result of the fast restricted Voronoi diagram algorithm still span over the inner knot and the outer surface. As a result, in the dual triangle mesh, the inner knot is glued with the outer surface by some non-manifold vertices and edges (marked out in Figure 19(c)). As a comparison, our method using the hyperbolic CVT has no problem to separate these two parts with much fewer

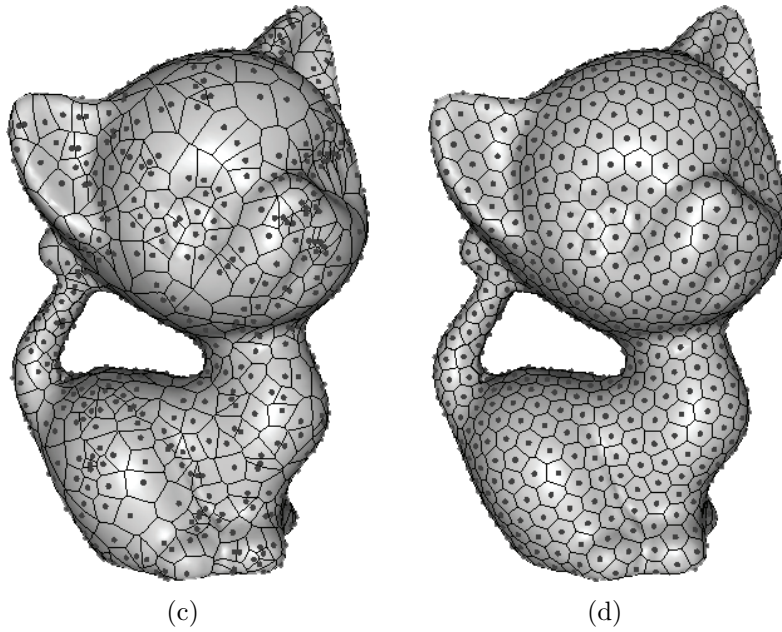
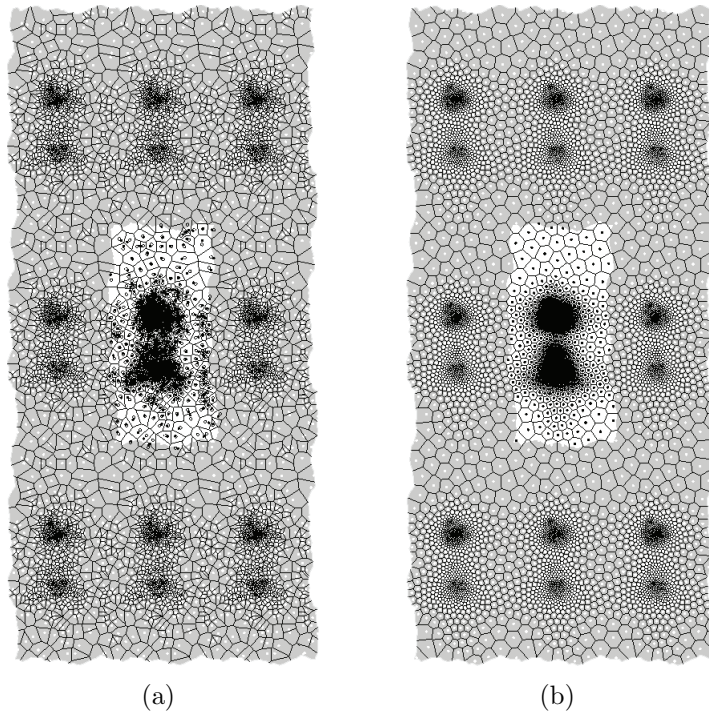


Figure 15: (a) Voronoi diagram of 1,000 random initial sites in the universal covering space, and the centroids for Voronoi cells of sites in \mathbf{S} . Black solid dots are sites in \mathbf{S} , white solid dots are sites in \mathbf{S}' , and hollow dots are centroids. (b) CVT result generated from (a). (c) Initial Voronoi diagram on surface. (d) CVT result on surface.

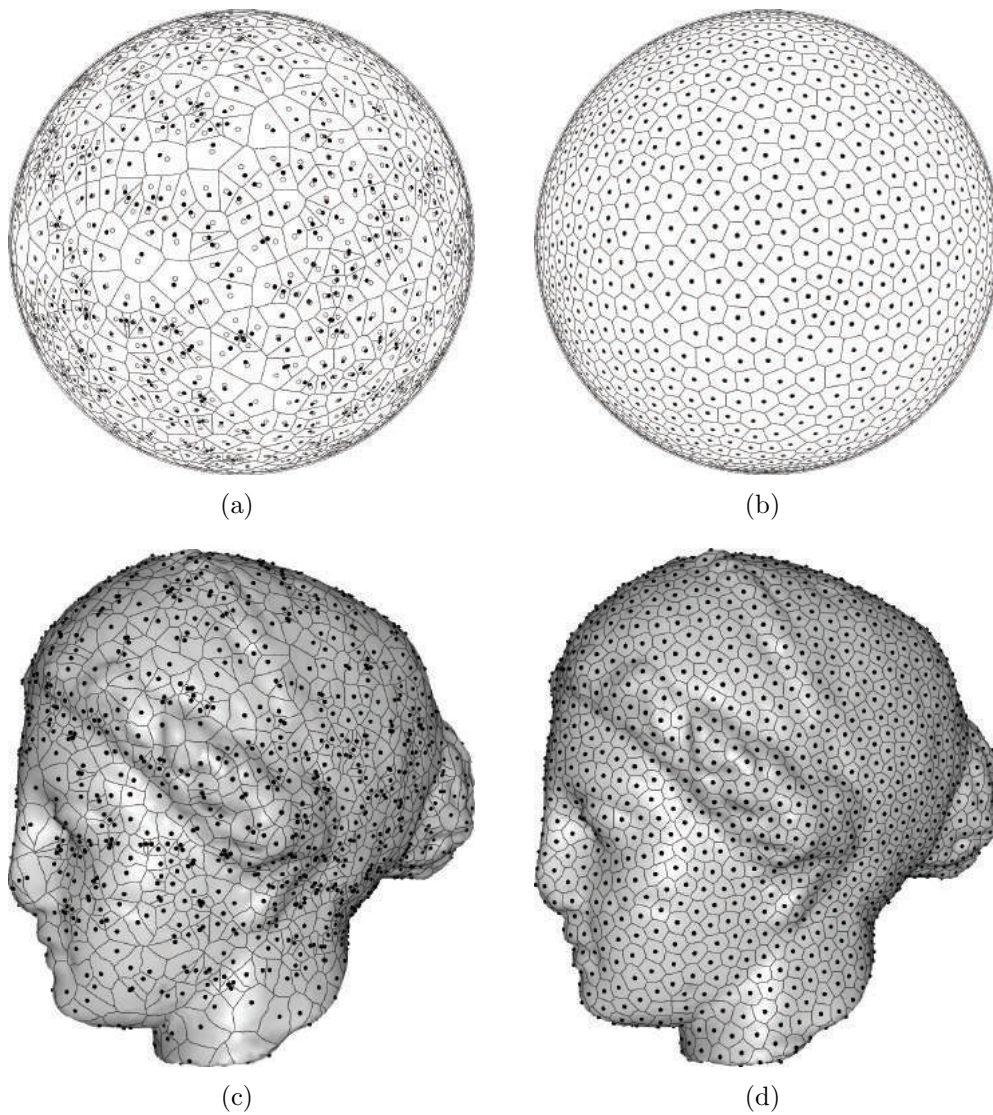


Figure 16: (a) Voronoi diagram of 2,000 random initial sites in the universal covering space, and the centroids for Voronoi cells of sites in \mathbf{S} . Solid dots are sites in \mathbf{S} and hollow dots are centroids. (b) CVT result generated from (a). (c) Initial Voronoi diagram on surface. (d) CVT result on surface.

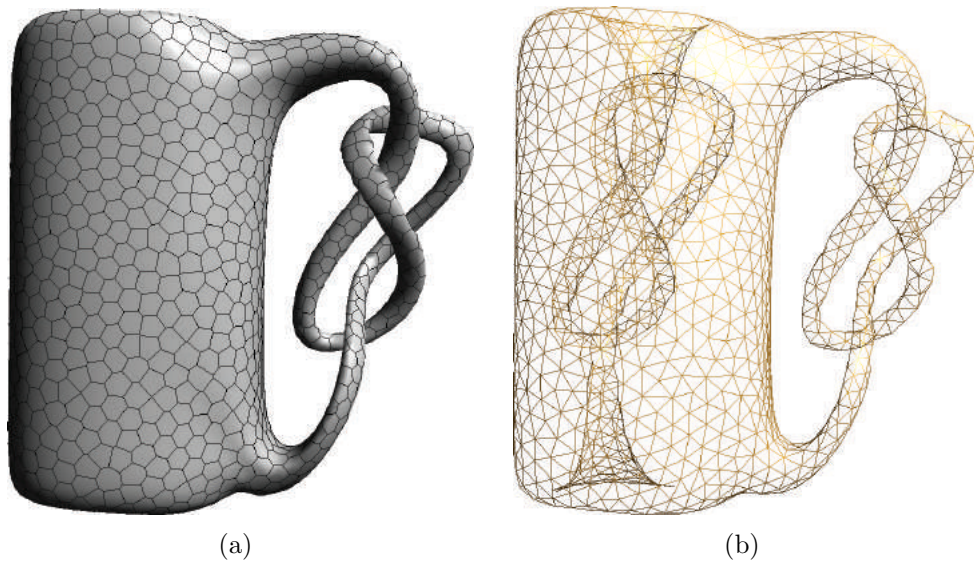


Figure 17: (a) The CVT of 2,000 sites on the surface Knot (genus-2). (b) Dual triangle mesh for Knot. (The wire frame mode is used to show the complex inner structure.)

sites.

Although non-manifold vertices and edges are still possible in the results of our method, we did not have any problems in our experiments with a reasonable number of sites. In our method, the CVT in universal covering space is computed in the running-time, but all the other steps, including computing the uniformization metric, embedding, and computing the distortion can all be pre-computed. The time complexity of computing the spherical Voronoi diagram and the hyperbolic Voronoi diagram is same as that in Euclidean space, i.e. $O(n \log n)$ where n is the number of sites. As a comparison, the algorithms using the exact geodesic distance need to compute the geodesic Voronoi diagram in every iteration, which is much more expensive than the computation of the Voronoi diagram in universal covering space.

7. Conclusion and Future Work

In this paper, the concept of centroidal Voronoi tessellation is extended from Euclidean space to spherical and hyperbolic spaces. New CVT energy functions are defined in these spaces, and the relationship between minimizing these functions and the CVTs in these spaces is proved, which leads to the proof of the convergence of Lloyd's algorithm in these spaces. It is

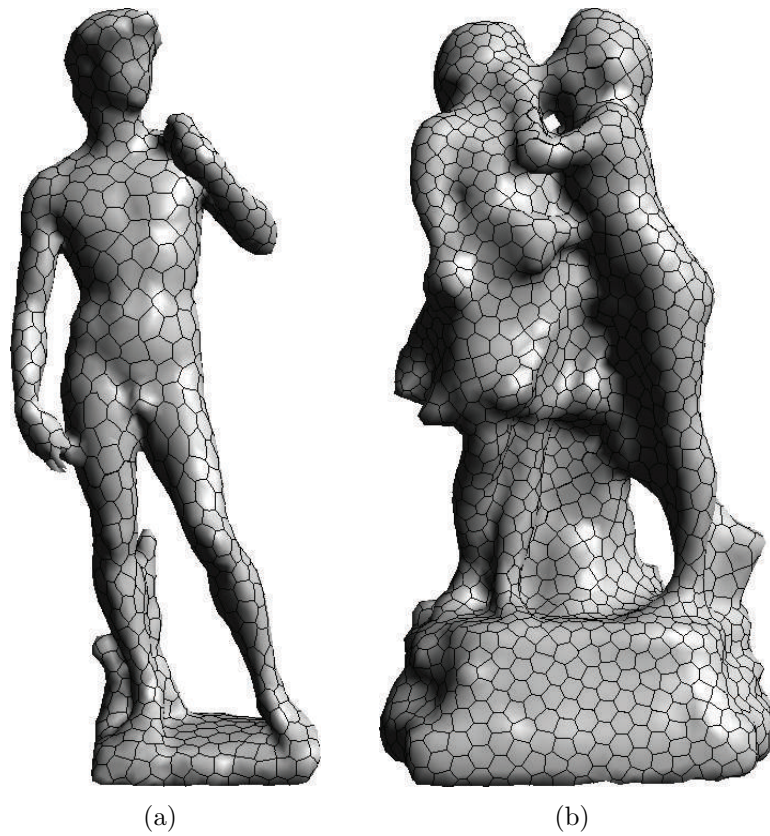


Figure 18: (a) The CVT of 1,000 sites on the surface David (genus-3) and (b)The CVT of 2,000 sites on the surface Sculpture (genus-3).

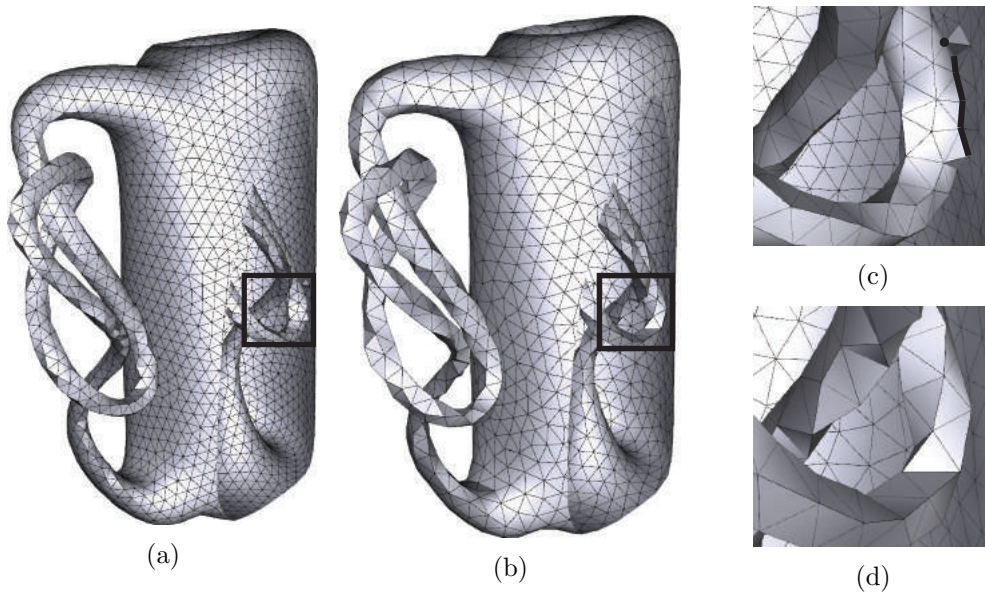


Figure 19: Comparing the dual triangle meshes generated by (a) the fast restricted Voronoi diagram algorithm [Yan et al., 2009] with 4,000 sites and (b) our algorithm with 2,000 sites. The regions where the inner tube is very close to the outer surface (marked by black boxes) are enlarged in (c) and (d). Non-manifold vertex and edges in (c) are marked out.

shown that the spherical CVT and hyperbolic CVT have the same property with its Euclidean counterpart, i.e. the sites are uniformly distributed. Finally, spherical, Euclidean, and hyperbolic CVTs are combined into a unified framework – CVT in universal covering space, and applied to get uniform partitions and high quality remeshing results for genus-0, genus-1, and high-genus (genus>1) surfaces. This is the first work systematically studying the spherical CVT, the hyperbolic CVT, and the CVT in universal covering space. We believe our results would have many more applications in different areas such as surface processing, scientific visualization, pattern recognition, etc.

Due to the lack of a close-form solution for the centroid of a triangle with non-uniform density in spherical or hyperbolic space, we can only use summation to approximate the integral in the computation of the spherical or hyperbolic CVT. This not only leads to degraded quality, but also slows down the performance of our programs. How to directly compute the integral as in Euclidean space is one of our major future work.

In our current implementation, the Voronoi diagram on the universal covering space is computed by using all points including initial sites and their corresponding points in neighbor domains. For a genus- g surface, there are $16g^2 - 8g$ neighbor domains. So the number of sites is quite large. This makes the computation of the Voronoi diagram very slow, which becomes the bottleneck of our current program. For a genus-3 surface with 2,000 sites, our current program needs more than one minute to compute the Voronoi diagram for a single iteration. Optimizing this procedure is critical to increase the program speed. We may borrow the idea of the periodic triangulation package in CGAL [Caroli and Teillaud, 2009, 2011] to accelerate this procedure.

Another possible acceleration approach is to utilize the parallel computability of the programmable graphics processing unit (GPU). GPU has already been used to accelerate the computation of the CVT in Euclidean space [Vasconcelos et al., 2008; Rong et al., 2011]. But the metric of Klein disk is highly non-uniform in the unit Euclidean disk, and is not easy (if not impossible) to be represented by a 2D texture. So none of the existing GPU algorithms can be straightforwardly extended to compute the hyperbolic CVT. We would like to investigate new data structures on the GPU for hyperbolic geometry.

8. Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. Guodong Rong, Liang Shuai, and Xiaohu Guo are partially supported by the National Science Foundation under Grant No. CCF-0727098. Miao Jin is partially supported by Louisiana Board of Regents under Grant No. RD-A-21.

References

- Alliez, P., Colin de Verdière, É., Devillers, O., Isenburg, M., 2005. Centroidal Voronoi diagrams for isotropic surface remeshing. *Graphical Models* 67, 204–231.
- Aurenhammer, F., 1987. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing* 16, 78–96.
- Caroli, M., Teillaud, M., 2009. Computing 3D Periodic Triangulations. Research Report RR-6823. INRIA.
- Caroli, M., Teillaud, M., 2011. Delaunay triangulations of point sets in closed Euclidean d -manifolds, in: *Symposium on Computational Geometry*. To appear.
- Chan, K.T., 2004. Uniformization of Riemann Surfaces. Ph.D. thesis. Mathematics Department, Harvard University.
- Cohen-Steiner, D., Alliez, P., Desbrun, M., 2004. Variational shape approximation. *ACM Transactions on Graphics* 23, 905–914.
- Du, Q., Emelianenko, M., 2006. Acceleration schemes for computing centroidal Voronoi tessellations. *Numerical Linear Algebra with Applications* 13, 173–192.
- Du, Q., Emelianenko, M., Ju, L., 2006. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM Journal on Numerical Analysis* 44, 102–119.
- Du, Q., Faber, V., Gunzburger, M., 1999. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review* 41, 637–676.

- Du, Q., Gunzburger, M., Ju, L., 2010. Advances in studies and applications of centroidal Voronoi tessellations. *Numerical Mathematics: Theory, Methods and Applications* 3, 119–142.
- Du, Q., Gunzburger, M.D., Ju, L., 2003. Constrained centroidal Voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing* 24, 1488–1506.
- Du, Q., Ju, L., 2005a. Approximations of a Ginzburg-Landau model for superconducting hollow spheres based on spherical centroidal Voronoi tessellations. *Mathematics of Computation* 74, 1257–1280.
- Du, Q., Ju, L., 2005b. Finite volume methods on spheres and spherical centroidal Voronoi meshes. *SIAM Journal on Numerical Analysis* 43, 1673–1692.
- Du, Q., Wang, D., 2006. Recent progress in robust and quality Delaunay mesh generation. *Journal of Computational and Applied Mathematics* 195, 8–23.
- Fabri, A., 2001. CGAL - the computational geometry algorithm library, in: *Proceedings of the 10th International Meshing Roundtable*, pp. 137–142.
- Fejes Tóth, G., 2001. A stability criterion to the moment theorem. *Studia Scientiarum Mathematicarum Hungarica* 38, 209–224.
- Galperin, G.A., 1993. A concept of the mass center of a system of material points in the constant curvature spaces. *Communications in Mathematical Physics* 154, 63–84.
- Gersho, A., 1979. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory* 25, 373–380.
- Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.T., 2004. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging* 23, 949–958.
- Jin, M., Kim, J., Luo, F., Gu, X., 2008. Discrete surface Ricci flow. *IEEE Transactions on Visualization and Computer Graphics* 14, 1030–1043.

- Jin, M., Luo, F., Gu, X., 2006. Computing surface hyperbolic structure and real projective structure, in: *ACM Symposium on Solid and Physical Modeling*, pp. 105–116.
- Ju, L., Du, Q., Gunzburger, M., 2002. Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations. *Parallel Computing* 28, 1477–1500.
- Karcher, H., 1977. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics* 30, 509–541.
- Klingenberg, W.P.A., 1982. *Riemannian Geometry*. Walter de Gruyter, New York.
- Kobayashi, S., Nomizu, K., 1996. *Foundations of Differential Geometry*. volume 2. Wiley-Interscience.
- Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.M., Lu, L., Yang, C., 2009. On centroidal Voronoi tessellation — energy smoothness and fast computation. *ACM Transactions on Graphics* 28, 101:1–101:17.
- Lloyd, S.P., 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 129–137.
- Lu, L., Levy, B., Wang, W., 2009. Centroidal Voronoi Tessellations for Line Segments and Graphs. technical report. INRIA-ALICE.
- MacQueen, J.B., 1967. Some methods for classification and analysis of multivariate observations, in: *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press. pp. 281–297.
- Munkres, J.R., 1996. *Elements of Algebraic Topology*. Westview Press.
- Newman, D.J., 1982. The hexagon theorem. *IEEE Transactions on Information Theory* 28, 137–139.
- Nielsen, F., Nock, R., 2010. Hyperbolic Voronoi diagrams made easy, in: *Proceedings of the 2010 International Conference on Computational Science and Its Applications*, Washington, DC, USA. pp. 74–80.

- Nilforoushan, Z., Mohades, A., 2006. Hyperbolic Voronoi diagram, in: Computational Science and Its Applications - ICCSA 2006. Springer-Verlag. volume 3984 of *Lecture Notes in Computer Science*, pp. 735–742.
- Okabe, A., Boots, B., Sugihara, K., Chiu, S.N., 1999. Spatial tessellations: concepts and applications of Voronoi diagrams. John Wiley & Sons. 2nd edition.
- Onishi, K., Takayama, N., 1996. Construction of Voronoi diagram on the upper half-plane. IEICE transactions on fundamentals of electronics, communications and computer sciences E79-A, 533–539.
- Peyré, G., Cohen, L., 2004. Surface segmentation using geodesic centroidal tessellation, in: Proceedings of 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, IEEE Computer Society, Washington, DC, USA. pp. 995–1002.
- Renka, R.J., 1997. Algorithm 772: STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere. ACM Transactions on Mathematical Software 23, 416–434.
- Rong, G., Jin, M., Guo, X., 2010. Hyperbolic centroidal Voronoi tessellation, in: Symposium of Solid & Physical Modeling, pp. 117–126.
- Rong, G., Liu, Y., Wang, W., Yin, X., Gu, X., Guo, X., 2011. GPU-assisted computation of centroidal Voronoi tessellation. IEEE Transactions on Visualization and Computer Graphics 17, 345–356.
- Stahl, S., 2007. Mass in the hyperbolic plane. ACM Computing Research Repository abs/0705.3448.
- Surazhsky, V., Alliez, P., Gotsman, C., 2003. Isotropic remeshing of surfaces: A local parameterization approach, in: Proceedings of the 12th International Meshing Roundtable, pp. 215–224.
- Valette, S., Chassery, J.M., Prost, R., 2008. Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams. IEEE Transactions on Visualization and Computer Graphics 14, 369–381.
- Vasconcelos, C.N., Sá, A., Carvalho, P.C., Gattass, M., 2008. Lloyd’s algorithm on GPU, in: Proceedings of the 4th International Symposium on Visual Computing, pp. 953–964.

Yan, D.M., Lévy, B., Liu, Y., Sun, F., Wang, W., 2009. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. *Computer Graphics Forum* 28, 1445–1454. (Proceedings of Symposium on Geometry Processing 2009).