Voronoi Diagram in Hyperbolic Plane
Algorithms & Application

MATH430 Final Project

Cheng-Yuan Lee

Contents

Chapter 1

Introduction

Contents

1.1 Purpose of the Project

As a competitive programmer, I have always been fascinated by computational geometry algorithms and their applications in solving a wide range of problems. Among these algorithms, the Voronoi diagram has particularly captured my interest due to its effectiveness in addressing complex challenges encountered in programming competitions. Additionally, the class have introduced me to various types of geometry, with hyperbolic geometry standing out as especially intriguing.

This curiosity has led me to explore Voronoi diagrams within the hyperbolic plane. I am interested in understanding whether these diagrams can be computed as efficiently as their Euclidean counterparts and what unique problems they might help solve in hyperbolic spaces. This project aims to investigate these questions, delving into the properties and applications of Voronoi diagrams in hyperbolic geometry.

1.2 Overview of Voronoi Diagrams

Given a set S of n points in the two-dimensional plane, a **Voronoi diagram** partitions the plane into distinct regions based on proximity to the points in S. Specifically, each region associated with a point $p \in S$ consists of all points q in the plane that are closer to p than to any other point in S. In other words, a point q belongs to the region of its nearest neighbor in S.

The Voronoi diagram can be visualized as a planar graph formed by the boundaries between these regions. In this graph:

- **Vertices** occur where three or more boundary segments intersect or extend to infinity. These vertices represent points that are equidistant to three or more points in S and correspond to the circumcenters of the triangles formed by these points.
- **Edges** are the boundary segments that connect two vertices. Each edge consists of points that are equidistant to exactly two points in S and lies on the perpendicular bisector of the line segment joining these two points.

This structure not only provides a clear geometric representation of proximity relationships within the set S but also serves as a foundational tool in various applications, ranging from computer graphics and spatial analysis to problems in competitive programming.

1.3 Motivation for Hyperbolic Geometry

In recent years, there has been a significant shift in machine learning towards leveraging non-Euclidean geometries. Traditional Euclidean geometry often falls short when representing complex datasets that exhibit hierarchical or tree-like structures, which are inherently non-Euclidean. Hyperbolic geometry, with its unique properties, provides a natural framework for modeling such data more effectively.

A notable advancement in this area is the development of Hyperbolic Neural Networks by Ganea et al. in 2018. These networks demonstrated improved performance on tasks involving hierarchical data, highlighting the potential of hyperbolic spaces in machine learning applications. This trend underscores the growing importance of understanding non-Euclidean geometries and their algorithms.

Given this momentum, it is crucial to explore how fundamental algorithms, such as Voronoi diagrams, operate within hyperbolic spaces. Voronoi diagrams play a key role in various applications, including nearest neighbor search, clustering, and spatial partitioning. Investigating their behavior in the hyperbolic plane not only enhances our theoretical understanding of hyperbolic geometry but also contributes to the development of more efficient and effective algorithms tailored for complex, real-world datasets.

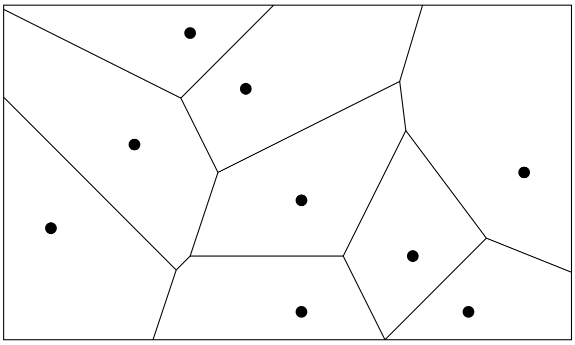

Figure 1.1: Example of a Voronoi Diagram in the Euclidean Plane

Chapter 2

Preliminaries

Contents

2.1 Basics of Hyperbolic Geometry

This study explores the construction of the Voronoi diagram within the Poincaré upper half-plane model. By utilizing Möbius transformations, we can transition between various disk models of hyperbolic geometry, simplifying discussions. In the Poincaré upper half-plane model, each point (a, b) in 2D is viewed as a complex number $z = a + bi$, with the hyperbolic plane defined as $\mathbb{H}^2 = \{a + bi \mid b \geq 0\}$. Within this framework, geodesic lines take the form of either half circles or straight segments. These non-Euclidean lines conform to all Euclidean geometry axioms, excluding the parallel postulate.

We can express each of these two geodesic lines as follows:

1. Half-circle with center at $z = p$ with radius r: $(x - p)^2 + y^2 = r^2$.
2. Straight line: $x = c$.

With these, we can define the distance between two points $p_1 = x_1 + y_1 i, p_2 = x_2 + y_2 i$ by

$$d(p_1, p_2) = \left| \ln \frac{A + \sqrt{A^2 - 4y_1^2 y_2^2}}{A - \sqrt{A^2 - 4y_1^2 y_2^2}} \right|$$

where $A = (x_1 - x_2)^2 + (y_1^2 + y_2^2)$.

We may verify that this definition coincides with the definition of using cross ratios of the points as defined in class.

2.2 Voronoi Diagram

Given a set of points $\{p_1, p_2, \ldots, p_n\}$ in the 2D Euclidean plane, the Voronoi cell $V(p_i)$ for a point p_i is defined as

$$V(p_i) = \left\{ p \in \mathbb{R}^2 \;\middle|\; d(p, p_i) \leq d(p, p_j) \quad \forall j \neq i \right\},$$

where $d(a, b)$ denotes the Euclidean distance between points a and b. Each Voronoi cell is a polygon in the Euclidean plane, and adjacent cells share common edges. Collectively, these polygons partition $\mathbb{R}^2$, forming what is known as the Voronoi diagram. The vertices of these polygons are referred to as Voronoi vertices, and the shared boundaries are called Voronoi edges.

2.3 Delaunay Triangulation

The Delaunay triangulation is the dual graph of the Voronoi diagram. Specifically, for each pair of points p_i and p_j whose Voronoi cells $V(p_i)$ and $V(p_j)$ are adjacent (i.e., they share a Voronoi edge), an edge is drawn between p_i and p_j in the Delaunay triangulation. This process results in a graph that triangulates the convex hull of the original set of points.
To give a formal definition. Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of distinct points in the Euclidean plane $\mathbf{R}^2$. The **Delaunay graph** $\mathcal{D}_{\mathbf{R}^2}(P)$ is an undirected graph where each pair of points (p_i, p_j) is connected by an edge if and only if there exists a **Euclidean circle** passing through both p_i and p_j that contains no other points from P in its interior.

2.4 Algorithms to Compute Voronoi Diagrams in the Euclidean Plane

Several algorithms are commonly employed to compute Voronoi diagrams in the Euclidean plane. Below, we outline some of the most widely used methods:

- **Construct Delaunay Triangulation and Derive Voronoi Diagram**:

  This approach leverages the duality between Voronoi diagrams and Delaunay triangulations. First, the Delaunay triangulation of the given set of points is constructed. Then, the Voronoi diagram is derived by connecting the circumcenters of the Delaunay triangles. This method is efficient and takes advantage of existing algorithms for Delaunay triangulation.

- **Incremental Algorithm**:

  The incremental algorithm builds the Voronoi diagram by adding one point at a time and updating the diagram accordingly. For each new point, the algorithm locates the region where the new Voronoi cell will appear, adjusts the existing cells to accommodate the new cell, and ensures that the diagram remains a valid partition of the plane. This method is relatively simple to implement but can be less efficient for large datasets.

- **Divide and Conquer**:

The divide and conquer algorithm recursively splits the set of points into smaller subsets, computes the Voronoi diagrams for each subset, and then merges them to form the final Voronoi diagram. This approach is highly efficient, especially for large numbers of points, as it reduces the problem size at each recursive step and efficiently combines the partial solutions.

- **Fortune's Algorithm**:

  Fortune's algorithm is a sweep line algorithm that constructs the Voronoi diagram in a single pass from left to right across the plane. It uses a priority queue to manage events and a beach line data structure to keep track of the evolving boundary between processed and unprocessed regions. Fortune's algorithm is optimal with a time complexity of $\mathcal{O}(n \log n)$, making it one of the most efficient algorithms for computing Voronoi diagrams.

In this project, we will be focusing on extending the first algorithm, which we will show that we can construct the Delaunay triangulation in $\mathbb{H}^2$ with existing algorithms to compute the Delaunay triangulation in $\mathbb{R}^2$, then use the constructed Delaunay triangulation to construct the Voronoi diagram. The algorithm is described in the paper ”Construction of Voronoi Diagram on the Upper Half-Plane” by Onishi and Takayama.

Chapter 3

Voronoi Diagrams in the Hyperbolic Plane

Contents

3.1 Compute Geometric Primitives in Hyperbolic Plane

Before we delve into computing the Delaunay triangulation and Voronoi diagram in the hyperbolic plane, we first discuss some fundamental geometric primitives within this context. We will focus on two key concepts: the perpendicular bisector of two points and circles in the hyperbolic plane. The following two theorems formalize these ideas:

Theorem 1. *Let $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ be two distinct points in the hyperbolic plane $\mathbb{H}^2$. The perpendicular bisector of p_1 and p_2 is a geodesic line in the hyperbolic plane.*

Proof. Consider two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ in the hyperbolic plane, represented here using the upper half-plane model. We aim to find the locus of points $p = (x, y)$ that are equidistant from p_1 and p_2, i.e.,

$$d(p, p_1) = d(p, p_2).$$

The hyperbolic distance between two points (x, y) and (x', y') in the upper half-plane model is given by:

$$d\left((x,y),(x',y')\right) = \left|\ln \frac{A + \sqrt{A^2 - 4y^2y'^2}}{A - \sqrt{A^2 - 4y^2y'^2}}\right| = \frac{1}{2}\cosh^{-1}\left(1 + \frac{(x-x')^2 + (y-y')^2}{2yy'}\right).$$

Setting $d(p, p_1) = d(p, p_2)$ and simplifying, we consider two cases:

1. **Horizontal Alignment ($y_1 = y_2$):** If the two points lie on the same horizontal line, the perpendicular bisector simplifies to a vertical line. Specifically,

$$x = \frac{x_1 + x_2}{2}.$$

   This line is a vertical geodesic in the hyperbolic plane.

2. **General Position ($y_1 \neq y_2$):** For points not aligned horizontally, the set of equidistant points forms a semicircle orthogonal to the boundary of the upper half-plane. To derive its equation, we start with:

$$\frac{(x - x_1)^2 + (y - y_1)^2}{2yy_1} = \frac{(x - x_2)^2 + (y - y_2)^2}{2yy_2}.$$

   After simplifying, we obtain:

$$\left(x - \frac{x_1y_2 - x_2y_1}{y_2 - y_1}\right)^2 + y^2 = \frac{y_1y_2\left(\left(\frac{x_1 - x_2}{y_1 - y_2}\right)^2 + 1\right)}{1}.$$

   This represents a semicircle with center at $\left(\frac{x_1y_2 - x_2y_1}{y_2 - y_1}, 0\right)$ and radius squared given by:

$$r^2 = y_1y_2\left(\left(\frac{x_1 - x_2}{y_1 - y_2}\right)^2 + 1\right).$$

   In the upper half-plane model, such a semicircle is a geodesic in the hyperbolic plane.

Therefore, in both cases, the perpendicular bisector is a geodesic line in the hyperbolic plane.

□

Lemma 1. *A hyperbolic circle, defined as the set of points at a fixed hyperbolic distance r from a center point $p_1 = (x_1, y_1)$ in the hyperbolic plane, corresponds to a Euclidean circle in the upper half-plane model. The equation of this circle is:*

$$(x - x_1)^2 + (y - y_1 \cosh r)^2 = (y_1 \sinh r)^2.$$

Additionally, the center point p_1 lies inside this Euclidean circle.

Proof. Let $p = (x, y)$ be any point on the hyperbolic circle centered at $p_1 = (x_1, y_1)$ with hyperbolic radius r. By definition, the hyperbolic distance between p and p_1 satisfies:

$$d(p, p_1) = r.$$

Using the hyperbolic distance formula in the upper half-plane model:

$$\cosh r = 1 + \frac{(x - x_1)^2 + (y - y_1)^2}{2yy_1}.$$

Rearranging terms, we get:

$$(x - x_1)^2 + (y - y_1)^2 = 2yy_1(\cosh r - 1).$$

Using the hyperbolic identity $\cosh^2 r - \sinh^2 r = 1$, we can express $2(\cosh r - 1)$ as $2\sinh^2\left(\frac{r}{2}\right)$ or manipulate it to obtain the standard form of a Euclidean circle. Completing the square for the y-terms leads to:

$$(x - x_1)^2 + (y - y_1\cosh r)^2 = (y_1 \sinh r)^2 .$$

This is the equation of a Euclidean circle with center at $(x_1, y_1 \cosh r)$ and radius $y_1 \sinh r$. Since $\cosh r > 1$ and $\sinh r > 0$ for $r > 0$, the original point $p_1 = (x_1, y_1)$ lies inside this Euclidean circle. □

Before providing the algorithm for constructing the Voronoi diagram in H^2, we will require another geometric primitive. We want to check whether there exists a equidistant point to a given set of 3 points p_1, p_2, p_3 in the hyperbolic plane, which basically is the center of the hyperbolic circle passing through all three points.

Theorem 2. *There exists a equidistant point to a given set of 3 points p_1, p_2, p_3 in the hyperbolic plane if and only if the hyperbolic circle passing through the three points with the form*

$$\alpha(x^2 + y^2) - \beta y + \gamma x - \delta = 0 \ \ (\alpha, \beta, \gamma, \delta \in \mathbf{R})$$

satisfies $\alpha \neq 0,\ \frac{\beta}{2\alpha} > 0$ and $4\alpha\delta + \gamma^2 < 0$

We know that we can compute the Euclidean circumcircle of three points with the following determinant:

$$\begin{vmatrix} 1 & x_1 & y_1 & {x_1}^2 + {y_1}^2 \\ 1 & x_2 & y_2 & {x_2}^2 + {y_2}^2 \\ 1 & x_3 & y_3 & {x_3}^2 + {y_3}^2 \\ 1 & x & y & x^2 + y^2 \end{vmatrix} = 0$$

Hence, by expanding the expression, we can get the center of the Euclidean circle. Let the center be $c = (x_0, y_0)$ and radius be r. Then we can compute the hyperbolic center $c_h = (x_1, y_1)$ and radius r_h by solving

$$\begin{cases} x_0 = x_1 \\ y_0 = y_1 \cosh r_h \\ r = y_1 \sinh r_h \end{cases}$$

This provided us a method to compute the hyperbolic center in $O(1)$.

3.2 Voronoi Diagram & Delaunay Triangulation in Hyperbolic Plane

Similar to the definitions in the Euclidean plane, we will define Voronoi polygons in the hyperbolic plane as

$$V(p_i) = \left\{ p \in \mathbb{H}^2 \;\middle|\; d(p, p_i) \leq d(p, p_j) \quad \forall j \neq i \right\},$$

These polygons form the Voronoi diagram in the hyperbolic plane.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of distinct points in the hyperbolic upper half-plane $\mathbb{H} = \{(x, y) \in \mathbb{R}^2 \mid y > 0\}$ equipped with the hyperbolic metric. The **Delaunay graph** $\mathcal{D}_{\mathbb{H}}(P)$ is an undirected graph where each pair of points (p_i, p_j) is connected by an edge if and only if there exists a **hyperbolic circle** passing through both p_i and p_j that contains no other points from P in its interior with respect to the hyperbolic metric.

Theorem 3. *The Delaunay graph in the hyperbolic upper half-plane $\mathbb{H}$ is a subgraph of the Delaunay graph in the Euclidean plane $\mathbb{R}^2$. Formally, $\mathcal{D}_{\mathbb{H}}(P) \subseteq \mathcal{D}_{\mathbb{R}^2}(P)$.*

Proof. Consider a set of points $P \subset \mathbb{H}$, where $\mathbb{H}$ denotes the hyperbolic upper half-plane model $\{(x, y) \in \mathbb{R}^2 \mid y > 0\}$. Suppose that an edge (p_i, p_j) exists in the hyperbolic Delaunay graph $\mathcal{D}_{\mathbb{H}}(P)$. By the definition of the hyperbolic Delaunay graph, there exists a hyperbolic circle C_H passing through p_i and p_j that contains no other points from P in its interior with respect to the hyperbolic metric.

In the upper half-plane model, hyperbolic circles correspond to either Euclidean circles entirely contained within $\mathbb{H}$ or Euclidean vertical lines (which can be regarded as circles with infinite radius). Since C_H contains no other points from P in its hyperbolic interior, its corresponding Euclidean representation C_E must also be devoid of other points from P in its Euclidean interior. This is because the hyperbolic metric scales distances in a way that if a hyperbolic circle contains no points from P, the corresponding Euclidean circle or line cannot contain any additional points from P within its Euclidean interior.

Therefore, the existence of the Euclidean empty circle C_E passing through p_i and p_j with no other points from P inside implies, by the definition of the Euclidean Delaunay graph, that (p_i, p_j) must also be an edge in $\mathcal{D}_{\mathbb{R}^2}(P)$.

Since this reasoning holds for any arbitrary edge in $\mathcal{D}_{\mathbb{H}}(P)$, it follows that every edge in the hyperbolic Delaunay graph is also present in the Euclidean Delaunay graph. Consequently, we conclude that $\mathcal{D}_{\mathbb{H}}(P) \subseteq \mathcal{D}_{\mathbb{R}^2}(P)$. □

3.3 Construction

Based on the proof provided above, we can develop the following algorithm:

1. Treat the set of points S in $\mathbb{H}^2$ as points in $\mathbb{R}^2$, and compute the Delaunay triangulation in $\mathbb{R}^2$ using the known $O(n \log n)$ algorithm.

2. For each edge in the Delaunay triangulation of $\mathbb{R}^2$, remove those edges that are not present in the Delaunay triangulation of $\mathbb{H}^2$.

3. Finally, compute the Voronoi diagram from the refined Delaunay triangulation obtained in the previous step in $O(n)$ time.

We will show that the second step can be done in $O(n)$.

To do the second step, for all triangles formed by p_1, p_2, p_3 in the Delaunay triangulation in $\mathbb{R}^2$, we can compute the center of the hyperbolic circle that passed through all three points. If there does not exist such a point, we will can simply remove the longest edge of this triangle. We have described the method to compute the center of the hyperbolic circle in the previous section. Since we are computing the determinant of a matrix of size 4×4, we can do this in $O(1)$. Therefore, this algorithm runs in $O(n \log n)$.

References

1. F. Nielsen and R. Nock, “Hyperbolic Voronoi diagrams made easy,” arxiv.org, Mar. 2009, doi: https://doi.org/10.1109/ICCSA.2010.37.

2. F. Nielsen and R. Nock, “Visualizing hyperbolic Voronoi diagrams,” CiteSeer X (The Pennsylvania State University), pp. 90–91, May 2014, doi: https://doi.org/10.1145/2582112.2595647.

3. Onishi, K., Takayama, N.: Construction of Voronoi diagram on the upper halfplane. IEICE Transactions 79-A(4) (1996) 533–539

4. O.-E. Ganea, G. Bécigneul, and T. Hofmann, “Hyperbolic Neural Networks,” arXiv.org, 2018. https://arxiv.org/abs/1805.09112 (accessed Dec. 10, 2024).