Hyperbolic Geometry, Möbius Transformations, and Geometric Optimization

David Eppstein

School of Information and Computer Science
University of California, Irvine

Outline

1. What is hyperbolic geometry?

2. From models to algorithms

3. Points on two planes

4. Optimal Möbius transformation

Outline

1. What is hyperbolic geometry?
2. From models to algorithms
3. Points on two planes
4. Optimal Möbius transformation

What is Hyperbolic geometry?

Uniform space of constant negative curvature (Lobachevski 1837)

Upper Euclidean halfspace acted on by fractional linear transformations
(Klein's Erlangen program 1872)

Satisfies first four Euclidean axioms with different fifth axiom:

1. There exists exactly one straight line through any two points

2. For any line segments AB and CD, there exists a unique E such that ABE occur in that order on a common line and BE is congruent to CD

3. There exists exactly one circle with given center and radius

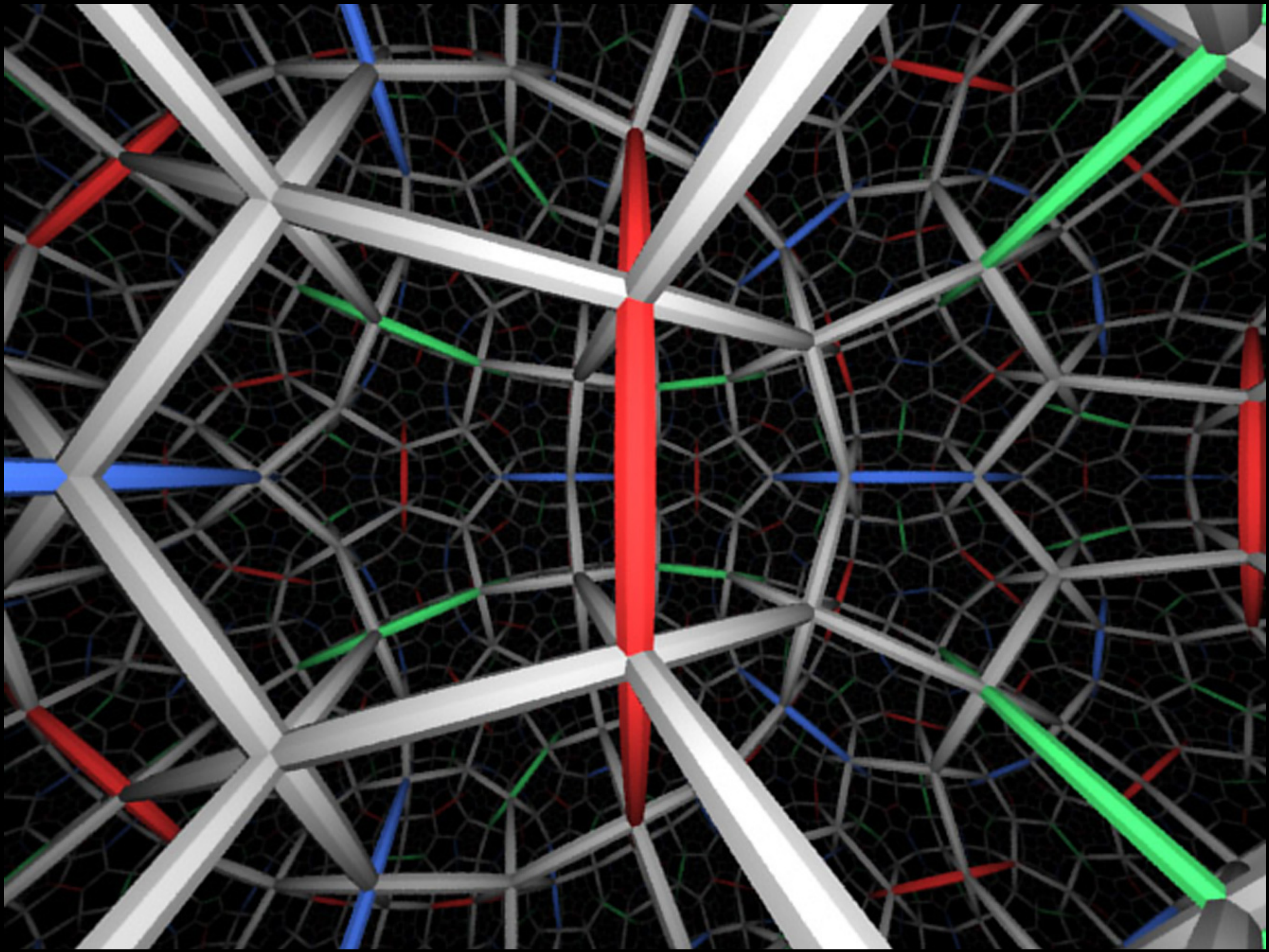
4. Every two right angles are congruent

5. For any point P and line L, P not on L, there exist infinitely many lines through P that do not meet L

Two-dimensional hyperbolic plane viewed from the outside

M. C. Escher, *Circle Limit II*, woodcut, 1959

Three-dimensional hyperbolic space viewed from the inside



From *Not Knot*, Charlie Gunn, The Geometry Center, 1990

Triangles have small area...

From modified Euclidean axiom, sum of interior angles $< \pi$

Deficit = triangle area = π − sum of angles

(Gauss-Bonet theorem)

Ideal triangles (all vertices at infinity)

have all angles zero

and area = π, maximum possible for any triangle

...but circles have large area

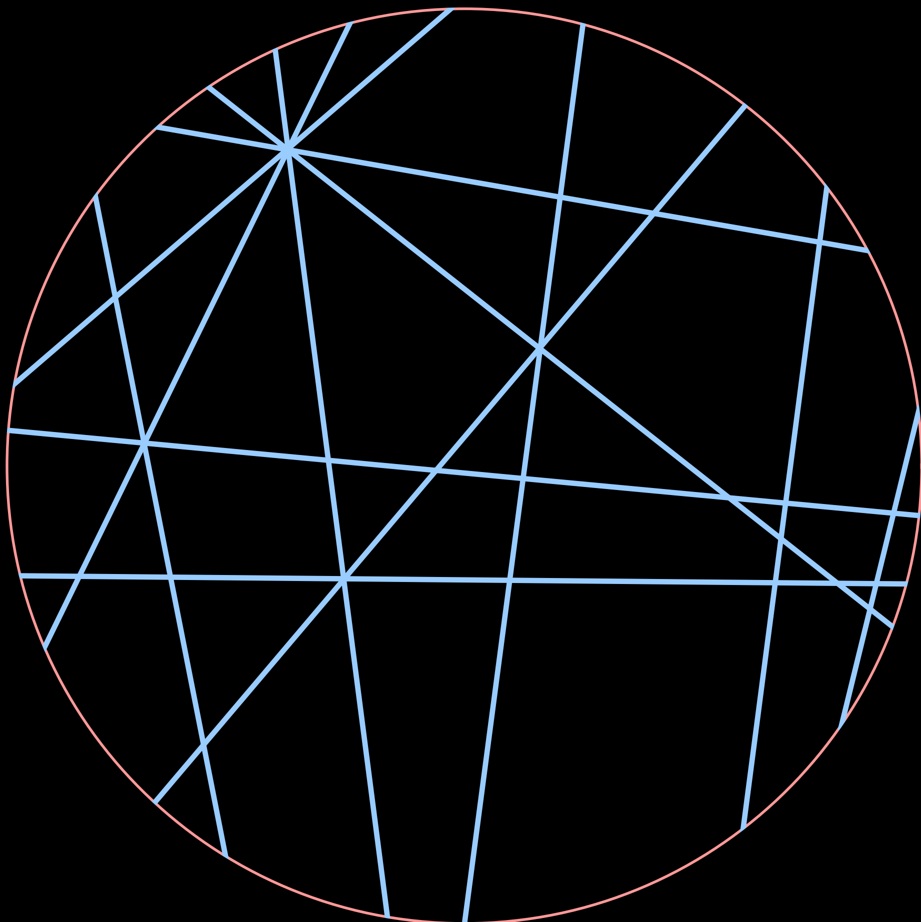
Exponential in circle radius

area/circumference < 1, limit = 1 for large radius circles

Outline

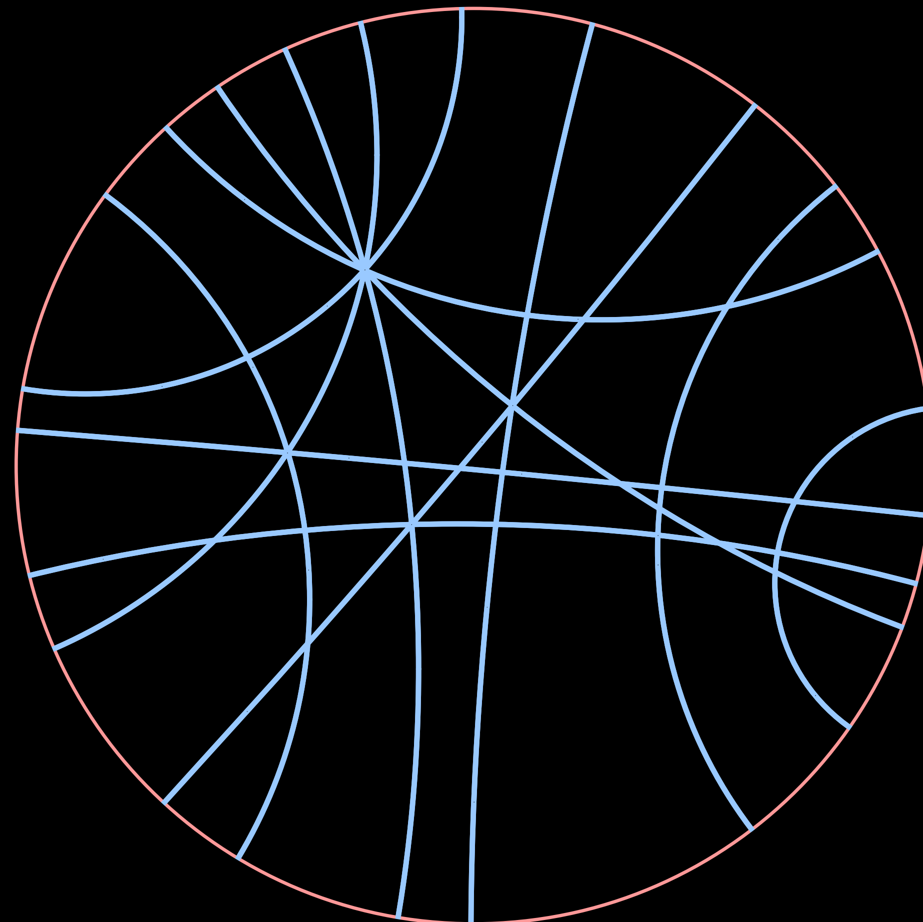
1. What is hyperbolic geometry?
2. From models to algorithms
3. Points on two planes
4. Optimal Möbius transformation

Two Models of Hyperbolic Space



Klein Model

Hyperbolic point = point in unit disk
Hyperbolic line = chord of unit disk



Poincaré Model

Hyperbolic point = point in unit disk
Hyperbolic line = circular arc
meeting unit disk at right angle

Can map each model to/from hyperbolic space (and each other)
preserving point-line incidences (but not all other metric properties)

Klein model

Straight objects in hyperbolic space (lines, etc) correspond to straight objects in the model

Convex objects in hyperbolic space correspond to convex objects in the model

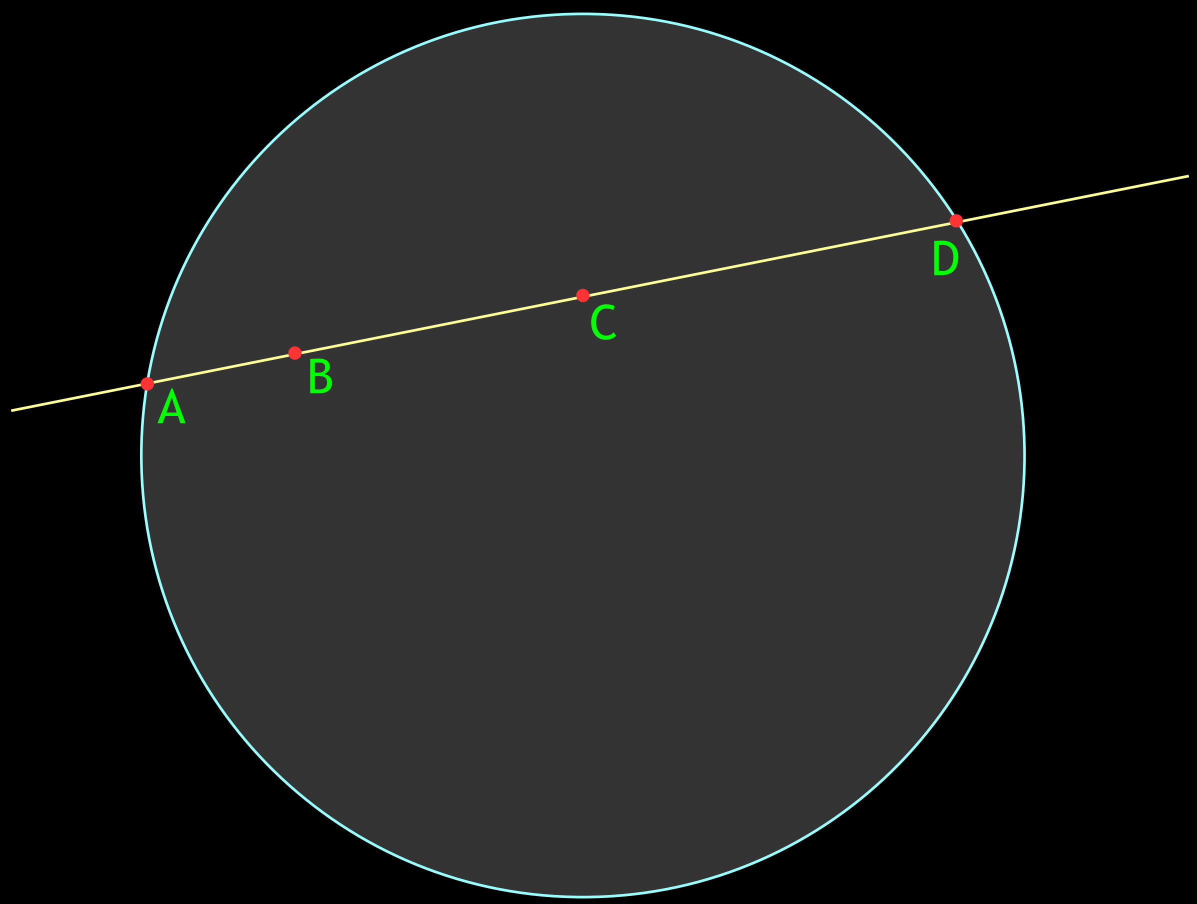
Distance, angles, etc., do not correspond

So...

Any computational geometry algorithm using only straightness and convexity can be immediately adapted to work in hyperbolic space

Convex hulls
Hyperplane arrangements
Polygon triangulation
Simplex range searching
Ray shooting with polygonal obstacles...

Distance formula from Klein model



Distance from B to C = | log((AB • CD) / (AC • BD)) |
where line segment lengths are measured with Euclidean distance

Poincaré model

Circles and spheres in hyperbolic space correspond to circles and spheres in the model

Angles between curves or surfaces correspond to same angles in the model

Distance, straightness, convexity, circle centers, etc., do not correspond

So...

Any computational geometry algorithm using only circles and angles
can be immediately adapted to work in hyperbolic space

Delaunay triangulation
Delaunay-based planar minimum spanning tree construction
Point-sphere incidences
Alpha-shapes...

Outline

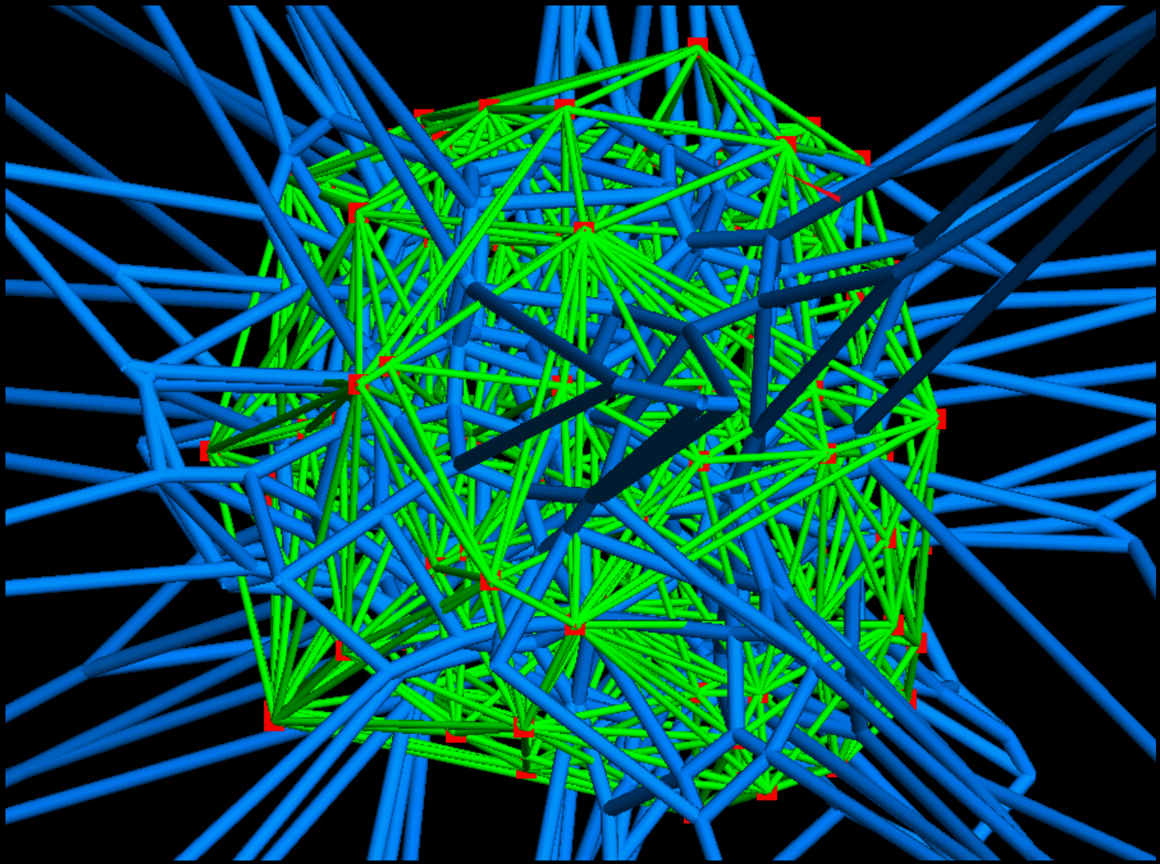
1. What is hyperbolic geometry?
2. From models to algorithms
3. Points on two planes
4. Optimal Möbius transformation

Problem: output-sensitive (Euclidean) 3d Delaunay triangulation

(dually, Voronoi diagram construction)

Delaunay triangulation:
Set of simplices defined by (d+1)-tuples of points
such that circumsphere of the simplex is empty of other points

Voronoi diagram:
Partition of space into cells nearest each of the points



Bruno Levy, CGAL project

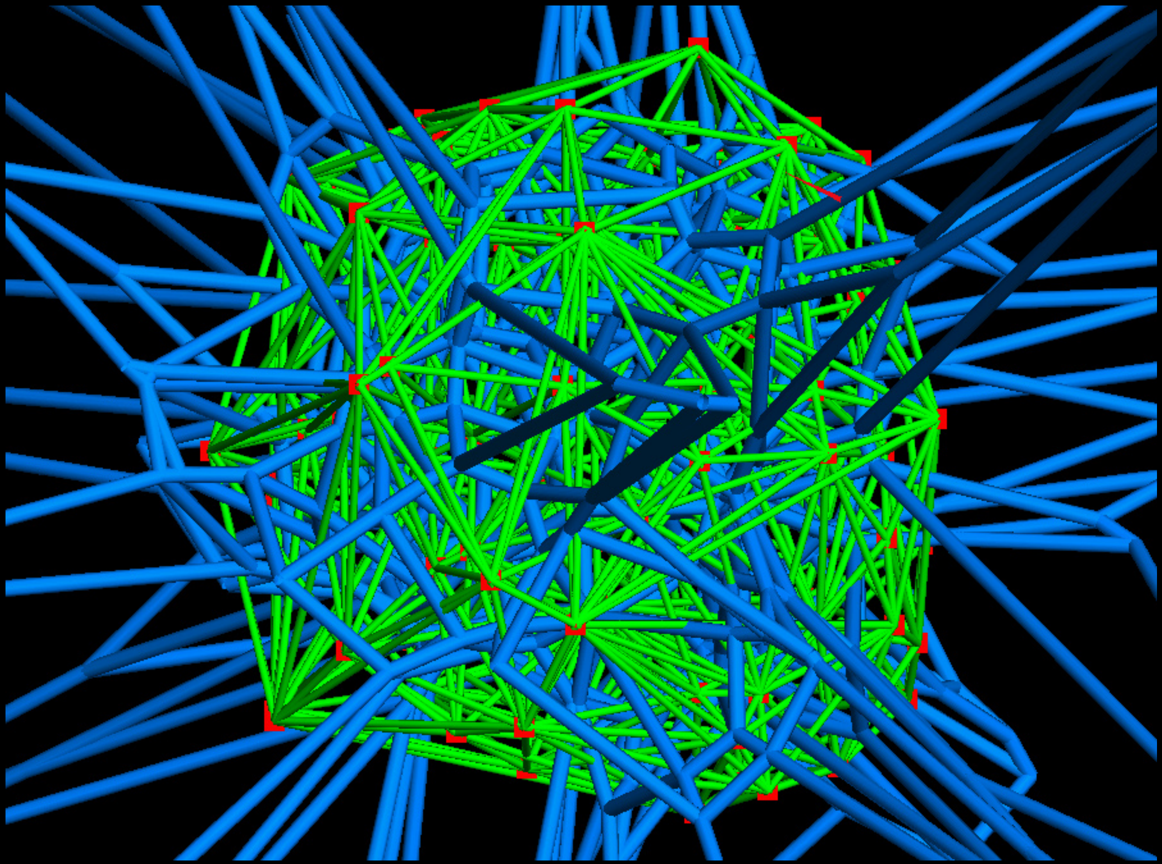
Problem: output-sensitive (Euclidean) 3d Delaunay triangulation

(dually, Voronoi diagram construction)

Output may have quadratic size, but often much closer to linear

Can be solved in time $O((n + t) \log^2 t)$ where t is output size
[Chan, Snoeyink, and Yap, 1997]

Further improvement or simplification for special cases?



Bruno Levy, CGAL project

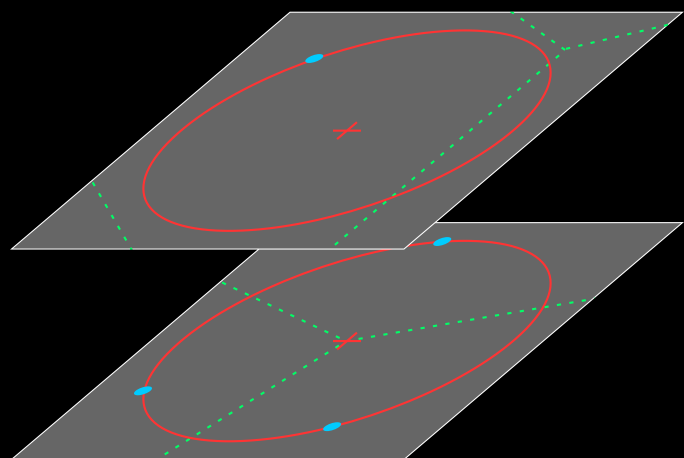
Boissonat, Cerezo, Devillers, and Teillaud, 1996:

Output-sensitive Delaunay for points in two planes

Simpler case: two parallel planes

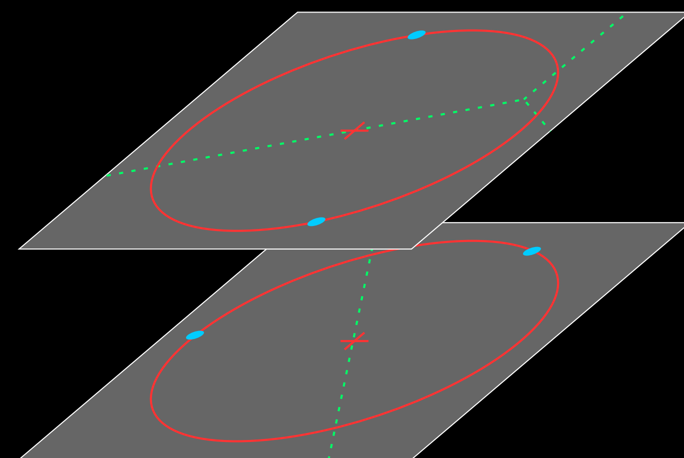
Empty Delaunay sphere intersects each plane in an empty circle
Centers of circles belong to features of Voronoi diagram
Numbers of sites on each circle determine which kind of Voronoi feature

One site on one circle, three on other



Voronoi cell of one diagram
Overlaying Voronoi vertex of the other

Two sites on each circle



Voronoi edge of one diagram
Overlaying Voronoi edge of the other

Algorithm:

1. Compute Euclidean Voronoi diagram in each plane
2. Overlay the two diagrams
3. Construct a tetrahedron for each Voronoi vertex of one diagram in a cell of the other
4. Construct a tetrahedron for each crossing pair of Voronoi edges

What if the points are on two planes but the two planes are not parallel?

Obvious idea:
Instead of overlaying two diagrams, rotate around intersection line

But, this doesn’t work for overlaying Euclidean Voronoi diagrams:
3d Delaunay sphere intersects planes in circles as before
but corresponding Euclidean Voronoi features may not cross

Solution:
Form halfspace Poincaré model with intersection line on boundary
Overlay hyperbolic Voronoi diagrams
Hyperbolic 3d Delaunay sphere center is collinear with 2d centers
so Voronoi feature crossings correspond to Delaunay tetrahedra

Result: Delaunay triangulation of points on two planes in $O(n \log n + t)$

Outline

1. What is hyperbolic geometry?
2. From models to algorithms
3. Points on two planes
4. Optimal Möbius transformation

What are Möbius transformations?

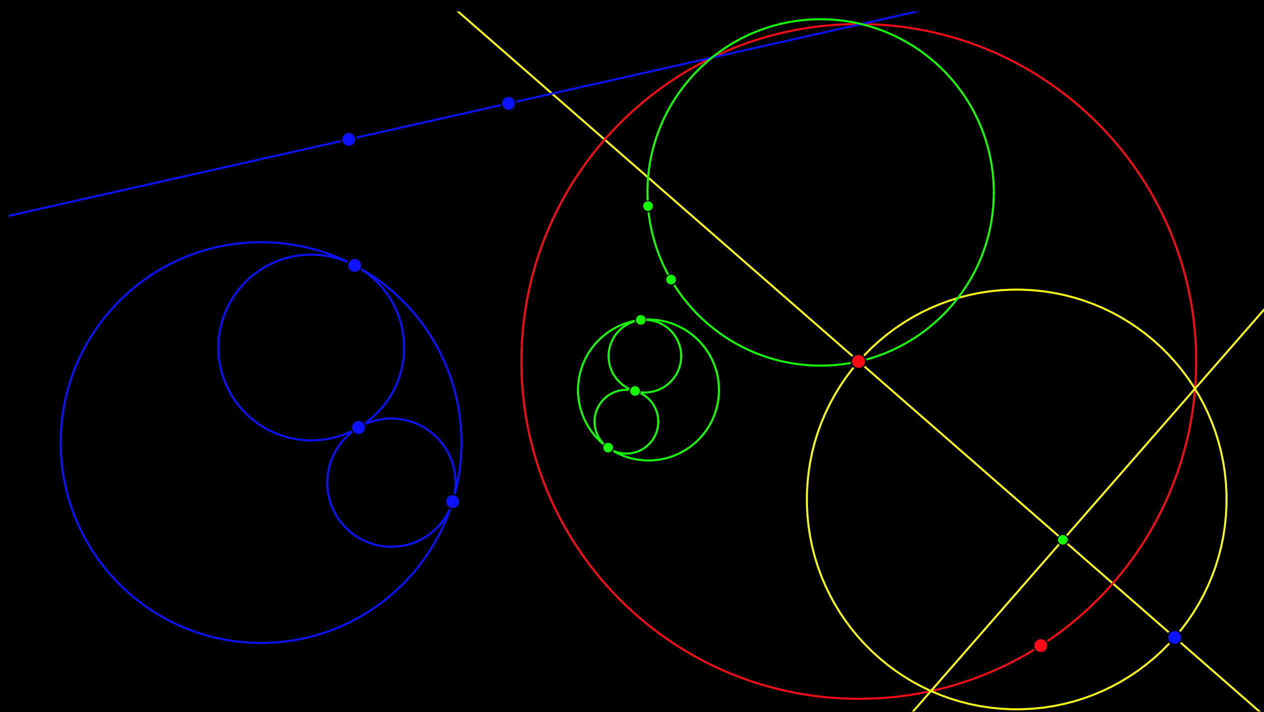
Fractional linear transformations of complex numbers:

$$z \rightarrow (a\,z + b) \,/\, (c\,z + d)$$

But what does it mean geometrically?
How to generalize to higher dimensions?
What is it good for?

Inversion

Given any circle (red below)
map any point to another point on same ray from center
product of two distances from center = radius^2



Circles map to circles
(lines = circles through point at infinity)

Conformal (preserves angles between curves)

Möbius transformations = products of inversions

(or sometimes orientation-preserving products)

Forms group of geometric transformations

Contains all circle-preserving transformations

In higher dimensions (but not 2d) contains all conformal transformations

Optimal Möbius transformation:

Given a planar (or higher dimensional) input configuration

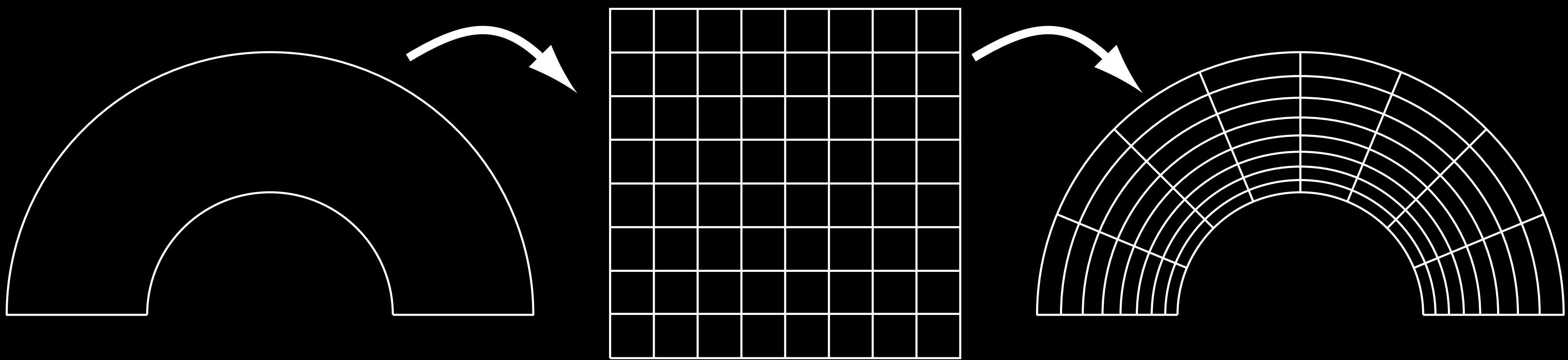
Select a Möbius transformation
from the (six-dimensional or higher) space of all Möbius transformations

That optimizes the shape of the transformed input

Typically min-max or max-min problems:
maximize min(set of functions describing transformed shape quality)

Application: conformal mesh generation

Given simply-connected planar domain to be meshed
Map to square, use regular mesh, invert map to give mesh in original domain



Different points of domain may have different requirements for element size
Want to map regions requiring small size to large areas of square

Conformal map is unique up to Möbius transformation

Optimization Problem:

Find conformal map maximizing min(size requirement * local expansion factor)
to minimize overall number of elements produced

Application: brain flat mapping [Hurdal et al. 1999]

Problem: visualize the human brain
Complicated folded 2d surface

Approach: find quasi-conformal mapping brain → plane
Avoids distorting angles but areas can be greatly distorted

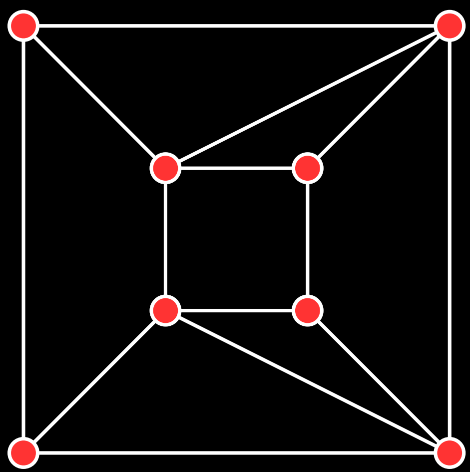
As in mesh gen. problem, mapping unique up to Möbius transformation

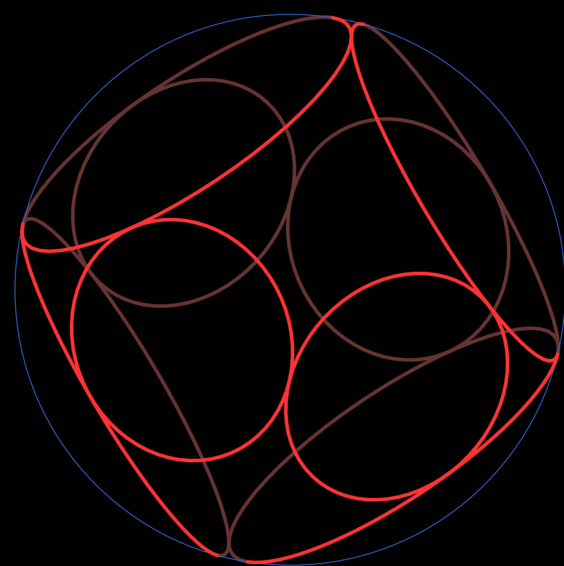
Optimization problem:

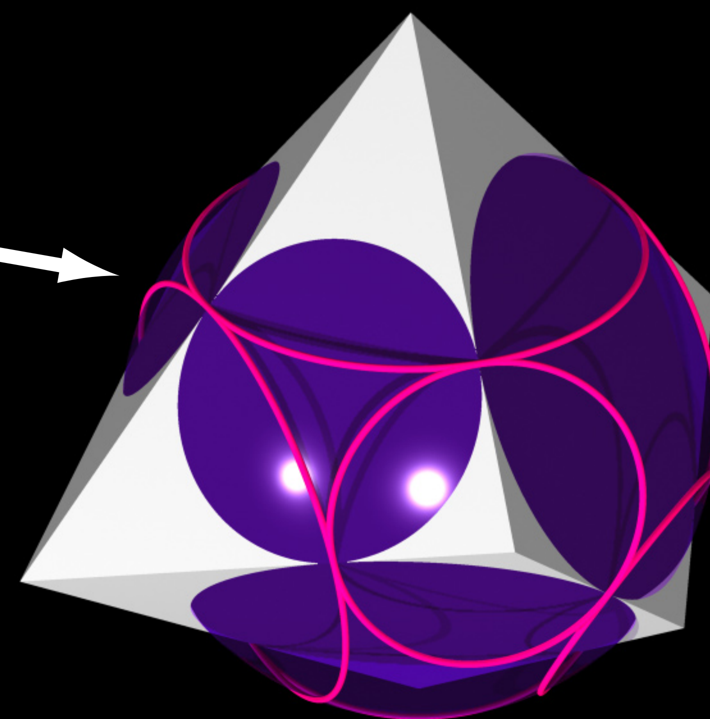
Given map 3d triangulated surface → plane,
find Möbius transformation minimizing max(area distortion of triangle)

Application: coin graph representation

Koebe-Andreev-Thurston Theorem:
vertices and edges of any planar graph can be represented
by disjoint disks and their tangencies on a sphere







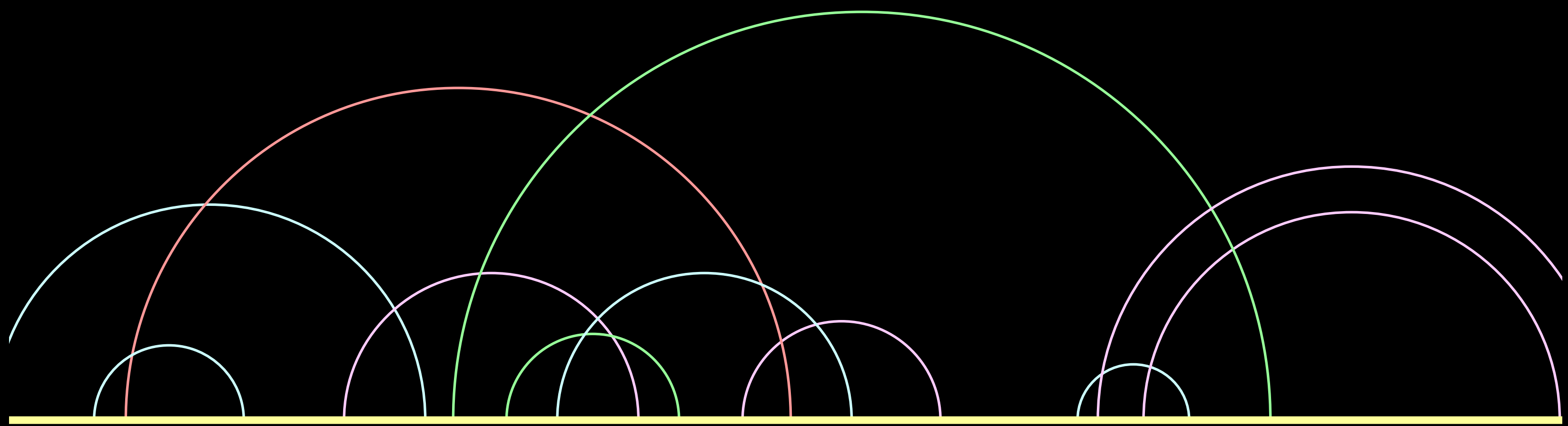
For maximal planar graphs, representation unique up to Möbius transformation

Problem: transform disks to maximize size of smallest disk
Uniqueness of optimal solution leads to display of graph symmetries

Hyperbolic interpretation of Möbius transformations

View *d*-dimensional Euclidean space as boundary of halfspace Poincaré model of hyperbolic $(d + 1)$-dimensional space

View *d*-sphere in $(d + 1)$-dimensional Euclidean space as boundary of unit-disk Poincaré model



Möbius transformations of *d*-space ↔ $(d+1)$-dimensional hyperbolic isometries

Simplify: optimal transformation to optimal location

Möbius transformation preserves spheres and angles
so takes Poincaré model of hyperbolic space
to a different Poincaré model of the same (isometric) space

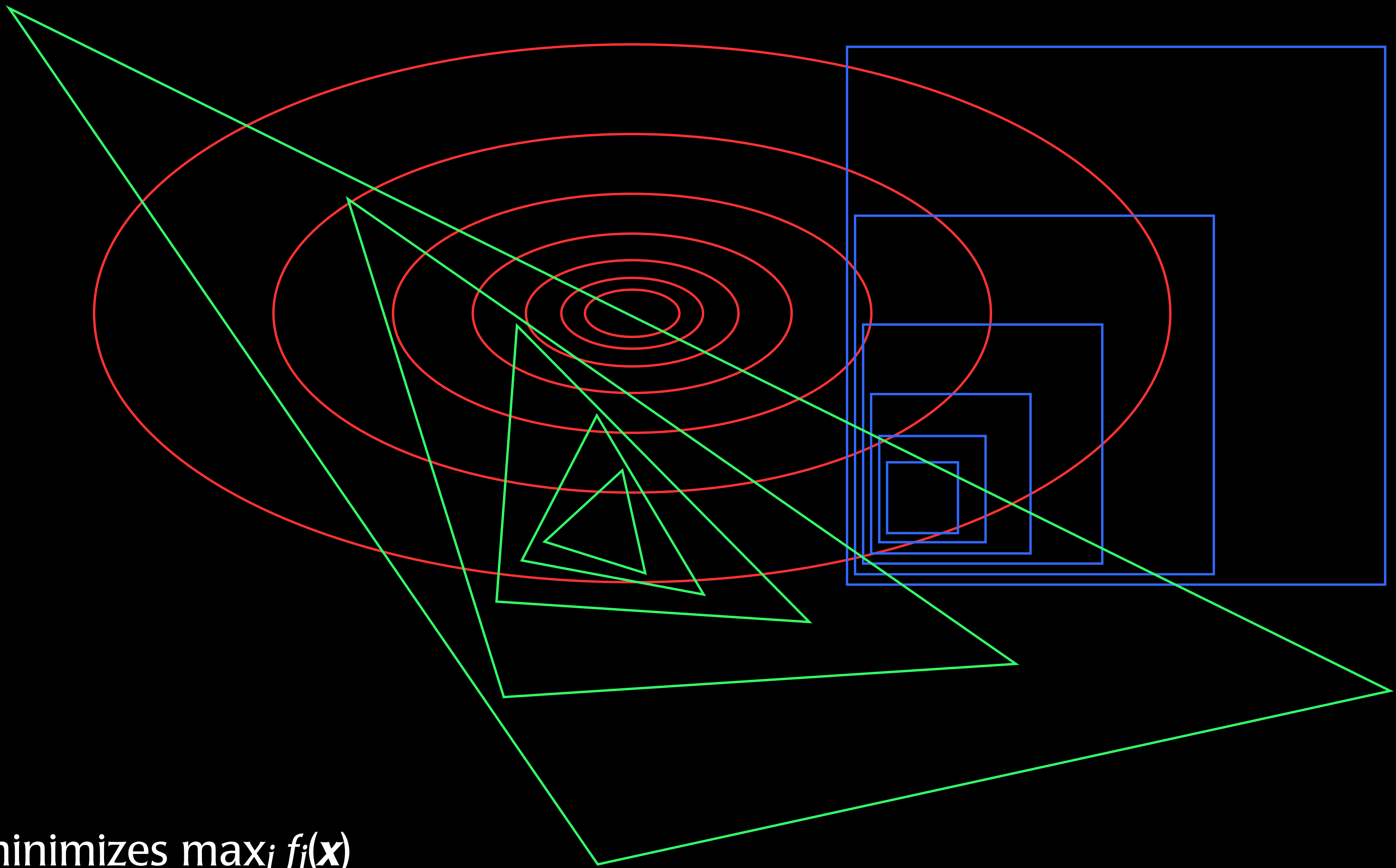
Conversely, given some initial Poincaré model,
choice of any other Poincaré model determines a Möbius transformation

Factor transformations into
choice of center point in hyperbolic model (affects shape)
Euclidean rotation around center point (doesn't affect shape)

Find optimal center point by *quasiconvex programming*

Quasiconvex program:

Input: family of quasiconvex functions $f_i(\boldsymbol{x})$, $\boldsymbol{x}$ in $\mathbf{R}^d$
quasiconvex: level sets $\{\boldsymbol{x}: f_i(\boldsymbol{x}) \leq r\}$ are convex



Output: $\boldsymbol{x}$ that minimizes $\max_i f_i(\boldsymbol{x})$

Some non-hyperbolic applications of quasiconvex programming

Minimum radius containing circle

Finite element mesh smoothing

Color gamut optimization

Asymptotic solution of multivariate recurrences

Quasiconvex programs are generalized linear programs

[Megiddo, Dyer, Clarkson, Seidel, ...]:

Combinatorial algorithms for linear programs
Based on dual simplex (start with infeasible solution, reduce violation amount)

Unlike e.g. Karmarkar or ellipsoid, don't want runtime
to depend on numerical precision of inputs and outputs

Typical results: if dimension is O(1), can solve n-variable LP in O(n) steps
Most steps involve only testing whether current solution violates some constraint
A smaller number of steps involve solving constant-sized LP subproblems

[Amenta, Gärtner, Matoušek, Sharir, Welzl, ...]

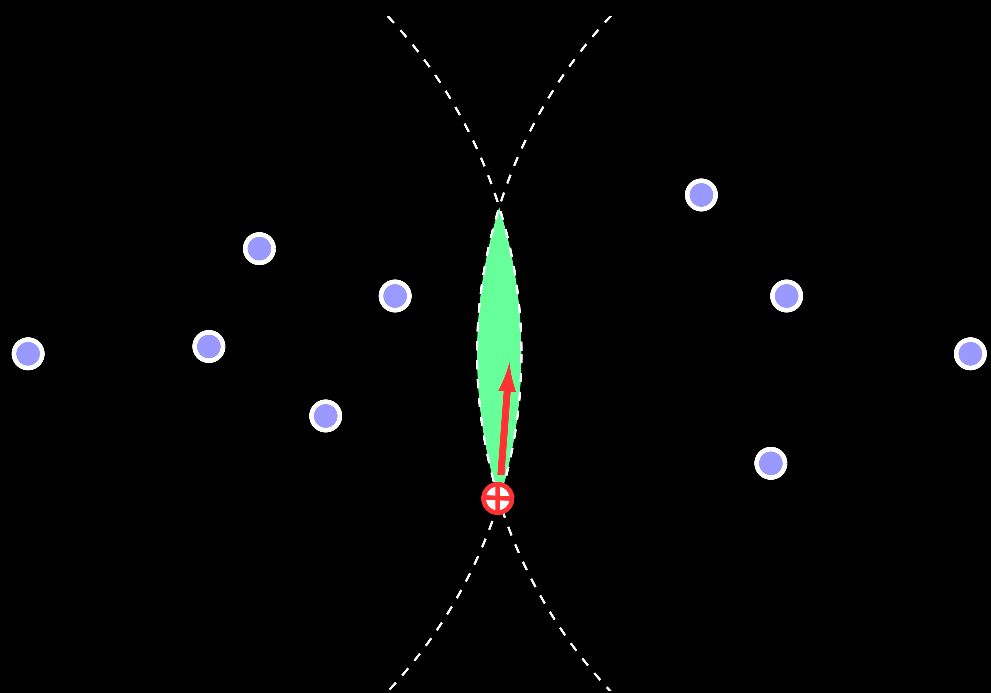
Generalization to nonlinear problems (e.g. circumradius)
Works for any function from subsets to subproblem values
Satisfying certain simple axioms

Quasiconvex programming satisfies GLP axioms
Therefore can be solved with O(n) violation tests (easy...)
and fewer basis-change operations (not so easy...)

Numerical search for quasiconvex program value

Objective function $\max_i f_i(\boldsymbol{x})$ is itself quasiconvex

No local optima to get stuck in, so
local improvement techniques will reach global optimum



How to find improvement direction?

May get trapped in sharp corner

e.g. minimum enclosing disk ($f_i(\boldsymbol{x})$ is distance from ith input point), equidistant from diameter points

Smooth quasiconvex programming (multi-gradient descent)

Suppose level sets have unique tangents at all boundary points
e.g. differentiable functions, step functions of smooth convex sets
(in 2d, use left & right tangents without smoothness assumption)

Then can find gradient $\boldsymbol{v}$ s.t. $\boldsymbol{w}$ is improvement direction iff $\boldsymbol{v} \cdot \boldsymbol{w} < 0$

Repeat:

Find gradients of functions within numerical tolerance of current max

Find simultaneous improvement direction $\boldsymbol{w}$ for all gradients
(lower dimensional minimum enclosing disk of few points)
If not found, algorithm has converged to solution

Replace $\boldsymbol{x}$ by $\boldsymbol{x} + \Delta\, \boldsymbol{w}$ for sufficiently small Δ

Select optimal center point by quasiconvex programming

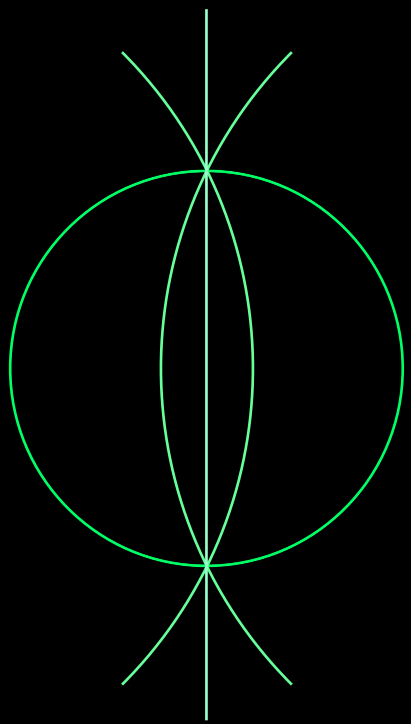
Klein model of hyperbolic geometry preserves convexity
so quasiconvex programming works equally well in hyperbolic space

Quality of Möbius transformation = max of quasiconvex functions

function argument is location of Poincaré model center point

optimal center point determines an optimal transformation

Hard part: proving that our objective functions are quasiconvex



Example: Optimal coin graph representation

Each coin (circle on unit sphere)
is the set of infinite points of a hyperbolic plane

Coin radius is monotonic in hyperbolic distance
from model center to plane

Quasiconvex function to optimize:
minimize max distance from model center to one of these planes

Level set: convex lens shape between two hyperspheres

Result: can use QCP to find optimal Möbius transformations

Unclear: how to represent basis change operations for GLP algebraically?

M. C. Escher, *Dragon*, wood-engraving, 1952

Conclusions

Introduction to computation in hyperbolic spaces

Many Euclidean algorithms can be translated to hyperbolic via Klein and Poincaré models

Hyperbolic viewpoint may help even for Euclidean problems (3d Delaunay, optimal Möbius)

Not much literature yet, plenty more to do