

Optimal Möbius Transformations for Information Visualization and Meshing

Marshall Bern¹ and David Eppstein²

¹ Xerox PARC, 3333 Coyote Hill Rd., Palo Alto, CA 94304, USA
`bern@parc.xerox.com`

² Dept. Inf. & Comp. Sci., UC Irvine, CA 92697-3425, USA
`eppstein@ics.uci.edu`

Abstract. We give linear-time quasiconvex programming algorithms for finding a Möbius transformation of a set of spheres in a unit ball or on the surface of a unit sphere that maximizes the minimum size of a transformed sphere. We can also use similar methods to maximize the minimum distance among a set of pairs of input points. We apply these results to vertex separation and symmetry display in spherical graph drawing, viewpoint selection in hyperbolic browsing, element size control in conformal structured mesh generation, and brain flat mapping.

1 Introduction

Möbius transformations of d -dimensional space form one of the fundamental geometric groups. Generated by inversions of spheres, they preserve spherical shape as well as the angles between pairs of curves or surfaces. We consider here problems of finding an *optimal Möbius transformation*:

- Given a set of $(d - 1)$ -dimensional spheres in a d -dimensional unit ball, find a Möbius transformation that maps the ball to itself and maximizes the minimum radius among the transformed spheres.
- Given a set of $(d - 1)$ -dimensional spheres on a sphere \mathbb{S}^d , find a Möbius transformation of \mathbb{S}^d that maximizes the minimum radius among the transformed spheres.
- Given a graph connecting a set of vertices on \mathbb{S}^d or in the unit ball in \mathbb{E}^d , find a Möbius transformation that maximizes the minimum edge length.

We develop efficient algorithms for these problems, by formulating them as quasiconvex programs in hyperbolic space. The same formulation also shows that simple hill-climbing methods are guaranteed to find the global optimum; this approach is likely to work well in practice as an alternative to our more complicated quasiconvex programs. We apply these results to the following areas:

- Spherical graph drawing [15]. Any embedded planar graph can be represented as a collection of tangent circles on a sphere \mathbb{S}^2 ; this representation

- is unique for maximal planar graphs, up to Möbius transformation. Our algorithms can find a canonical spherical realization of any planar graph that optimizes the minimum circle radius or the minimum separation between two vertices, and that realizes any symmetries implicit in the given embedding.
- Hyperbolic browsing [16]. The Poincaré model of the hyperbolic plane has become popular as a way of displaying web sites and other graph models too complex to view in their entirety. This model permits parts of the site structure to be viewed in detail, while reducing the size of peripheral parts. Our algorithms can find a “central” initial viewpoint for a hyperbolic browser, that allows all parts of the sites to be viewed at an optimal level of detail.
 - Mesh generation [3, 24]. A principled method of structured mesh generation involves conformal mapping of the problem domain to a simple standardized shape such as a disk, construction of a uniform mesh in the disk, and then inverting the conformal mapping to produce mesh elements in the original domain. Möbius transformations can be viewed as a special class of conformal maps that take the disk to itself. Our algorithms can find a conformal mesh that meets given requirements of element size in different portions of the input domain, while using a minimal number of elements in the overall mesh.
 - Brain flat mapping [13]. Hurdal et al. proposed a system for visualizing convoluted brain surfaces, by approximate conformal mapping of those surfaces to a Euclidean disk, sphere, or hyperbolic plane. Our algorithms can find a conformal mapping that minimizes the resulting areal distortion.

We assume throughout that the dimension d of the spaces we deal with is a constant; most commonly in our applications, $d = 2$ or $d = 3$. We omit many details in this extended abstract; see the full paper [2] for details.

2 Preliminaries

2.1 Möbius Transformation and Hyperbolic Geometry

An *inversion* of the set $\mathbb{R}^d \cup \{\infty\}$, generated by a sphere C with radius r , maps to itself every ray that originates at the sphere’s center. Each point is mapped to another point along the same ray, so that the product of the distances from the center to the point and to its image equals r^2 . The center of C is mapped to ∞ and vice versa. An inversion maps each point of C to itself, transforms spheres to other spheres, and preserves angles between pairs of curves or surfaces. Repeating an inversion produces the identity mapping on $\mathbb{R}^d \cup \{\infty\}$. The set of products of inversions forms a group, the group of *Möbius transformations* on the Euclidean space \mathbb{E}^d . If we restrict our attention to the subgroup that maps a given sphere \mathbb{S}^{d-1} to itself, we find the group of Möbius transformations on \mathbb{S}^{d-1} .

Although our problem statements involve Euclidean and spherical geometry, our solutions involve techniques from hyperbolic geometry [14], and in particular the classical methods of embedding hyperbolic space into Euclidean space: the *Poincaré model* and the *Klein model*.

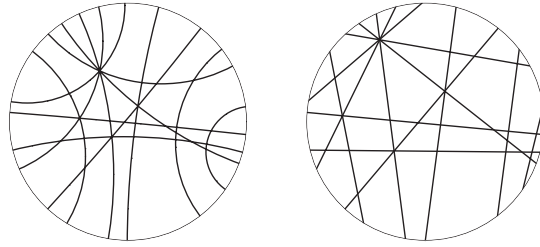


Fig. 1. Poincaré (left) and Klein (right) models of the hyperbolic plane. Analogous models exist for any higher dimensional hyperbolic space.

In both the Poincaré and Klein models, the d -dimensional hyperbolic space \mathbb{H}^d is viewed as an open unit ball in a Euclidean space \mathbb{E}^d , while the unit sphere bounding the ball forms a set of points “at infinity”. In the Poincaré model (Figure 1, left), the lines of the hyperbolic space are modeled by circular arcs of the Euclidean space, perpendicular to the unit sphere. Hyperplanes are modeled by spheres perpendicular to the unit sphere, and hyperbolic spheres are modeled by spheres fully contained within the unit ball. Two more classes of surfaces are also modeled as Euclidean spheres: *hyperspheres* (surfaces at constant distance from a hyperplane) are modeled by spheres crossing the unit sphere non-perpendicularly, and *horospheres* are modeled by spheres tangent to the unit sphere. The Poincaré model thus preserves spherical shape as well as the angles between pairs of curves or surfaces. Any horosphere or hypersphere divides \mathbb{H}^d into a convex and a nonconvex region; we define a *horoball* or *hyperball* to be the convex region bounded by a horosphere or hypersphere respectively.

In the Klein model (Figure 1, right), the lines of the hyperbolic space map to line segments of the Euclidean space, formed by intersecting Euclidean lines with the unit ball. Hyperplanes thus map to hyperplanes, and convex bodies map to convex bodies. Although the Klein model does not preserve spherical shape, it does preserve flatness and convexity. In the Klein model, spheres are modeled as ellipsoids contained in the unit ball, horospheres are modeled as ellipsoids tangent at one point to the unit ball, and hyperspheres are modeled as halves of ellipsoids tangent in a $(d - 1)$ -sphere to the unit ball.

The Poincaré and Klein models are not intrinsic to hyperbolic space, rather there can be many such models for the same space. The choice of model is determined by the hyperbolic point mapped to the model’s center point, and by an orientation of the space around this center. We call this central point the *viewpoint* since it determines the Euclidean view of the hyperbolic space.

The connection between hyperbolic space and Möbius transformations is this: the isometries of the hyperbolic space are in one-to-one correspondence with the subset of Möbius transformations of the unit ball that map the unit ball to itself, where the correspondence is given by the Poincaré model [14, Theorem 6.3]. Any hyperbolic isometry (or unit-ball preserving Möbius transformation) can be factored into a hyperbolic translation mapping some point of the hyperbolic space

to the viewpoint, followed by a rotation around the viewpoint [14, Lemma 6.4]. Since rotation does not change the Euclidean shape of the objects on the model, our problems of selecting an optimal Möbius transformation of a Euclidean or spherical space can be rephrased as finding an optimal hyperbolic viewpoint.

2.2 Quasiconvex Programming

The viewpoint we seek in our optimal Möbius transformation problems will be expressed as the pointwise maximum of a finite set of *quasiconvex functions*; that is, functions for which the level sets are all convex. To find this point, we use a generalized linear programming framework of Amenta et al. [1] called *quasiconvex programming*.

Define a *nested convex family* to be a map $\kappa(t)$ from nonnegative real numbers to compact convex sets in \mathbb{E}^d such that if $t < t'$ then $\kappa(t) \subset \kappa(t')$, and such that for all t , $\kappa(t) = \bigcap_{t' > t} \kappa(t')$. Any nested convex family κ determines a function $f_\kappa(x) = \inf \{ t \mid x \in \kappa(t) \}$ on \mathbb{R}^d , with level sets that are the boundaries of $\kappa(t)$. If f_κ does not take a constant value on any open set, and if $\kappa(t')$ is contained in the interior of $\kappa(t)$ for any $t' < t$, we say that κ is *continuously shrinking*. Conversely, the level sets of any quasiconvex function form the boundaries of the convex sets in a nested convex family, and if the function is continuous and not constant on any open set then the family will be continuously shrinking.

If $S = \{\kappa_1, \kappa_2, \dots, \kappa_n\}$ is a set of nested convex families, and $A \subset S$, let

$$f(A) = \inf \left\{ (t, x) \mid x \in \bigcap_{\kappa_i \in A} \kappa_i(t) \right\}$$

where the infimum is taken in the lexicographic ordering, first by t and then by the coordinates of x . Amenta et al. [1] defined a *quasiconvex program* to be a finite set S of nested convex families, with the objective function f described above, and showed that generalized linear programming techniques could be used to solve such programs in linear time for any constant dimension. Due to the convexity-preserving properties of the Klein model, we can replace \mathbb{E}^d by \mathbb{H}^d in the definition of a nested convex family without changing the above result.

3 Algorithms

We now describe how to apply the quasiconvex programming framework described above in order to solve our optimal Möbius transformation problems. In each case, we form a set of nested convex families κ_i , where each family corresponds to a function f_i describing the size of one of the objects in the problem (e.g., the transformed radius of a sphere) as a function of the viewpoint location. The solution to the resulting quasiconvex program then gives the viewpoint maximizing the minimum of these function values. The only remaining question in applying this technique is to show that, for each of the problems we study, the functions of interest do indeed have convex level sets.

3.1 Maximizing the Minimum Sphere

We begin with the simplest of the problems described in the introduction: finding a Möbius transformation that takes the unit ball to itself and maximizes the minimum radius among a set of transformed spheres. Equivalently, we are given a set of spheres in a hyperbolic space, and wish to choose a viewpoint for a Poincaré model of the space that maximizes the minimum Euclidean radius of the spheres in the model. By symmetry, the radius of a sphere in the Poincaré model depends only on its hyperbolic radius, and on the hyperbolic distance from its center to the viewpoint. Thus, in this case, the level sets of the transformed radius are just concentric hyperbolic spheres.

Theorem 1. *Suppose we are given as input a set of n spheres, all contained in the unit ball in \mathbb{E}^d . Then we can find the Möbius transformation of the unit ball that maximizes the minimum radius of the transformed spheres, in $\mathcal{O}(n)$ time, by quasiconvex programming.*

If we view the unit sphere itself as being one of the input spheres, then Theorem 1 can be viewed as minimizing the ratio between the radii of the largest and smallest transformed spheres.

Open Problem 1. *Is there an efficient algorithm for finding a Möbius transformation of \mathbb{E}^d minimizing the ratio between the radii of the largest and smallest transformed spheres, when the input does not necessarily include one sphere that contains all the others?*

We can also prove a similar result for radius optimization on the sphere:

Theorem 2. *Suppose we are given as input a set of n spheres in \mathbb{S}^d . Then we can find the Möbius transformation of \mathbb{S}^d that maximizes the minimum radius of the transformed spheres, in $\mathcal{O}(n)$ time, by quasiconvex programming.*

3.2 Vertex Separation

We next consider problems of using Möbius transformations to separate a collection of points.

Theorem 3. *Suppose we are given as input a graph with n vertices and m edges, with each vertex assigned to a point on the sphere \mathbb{S}^d or the unit disk in \mathbb{R}^d , and with edges represented as great circle arcs or straight line segments respectively. Then we can find the Möbius transformation that maximizes the minimum length of the transformed graph edges, in $\mathcal{O}(m)$ time, by quasiconvex programming.*

Similarly, we can find the Möbius transformation maximizing the minimum distance among a set of n transformed points by applying Theorem 3 to the complete graph K_n . However, the input size in this case is n , while the algorithm of Theorem 3 takes time proportional to the number of edges in K_n , $\mathcal{O}(n^2)$. With care we can reduce the time to near-linear:

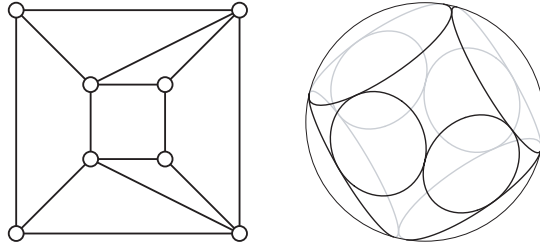


Fig. 2. A planar graph (left) and its coin graph representation (right).

Theorem 4. *Suppose we are given n points in \mathbb{S}^2 or the unit disk in \mathbb{R}^2 . Then we can find the Möbius transformation that maximizes the minimum distance among the transformed points in $\mathcal{O}(n \log n)$ time.*

Proof. The Delaunay triangulation of the points can be computed in $\mathcal{O}(n \log n)$ time, is Möbius-invariant (due to its definition in terms of empty circles), forms a planar graph with $\mathcal{O}(n)$ edges, and is guaranteed to contain the shortest transformed distance among the points. Therefore, applying Theorem 3 to the Delaunay triangulation gives the desired result. \square

In higher dimensions, the Delaunay triangulation may be complete, and so gives us no advantage. However we can again reduce the time from quadratic by using a random sampling scheme similar to one from our work on inverse parametric optimization problems [9].

Theorem 5. *Suppose we are given n points in \mathbb{S}^d . Then we can find the Möbius transformation that maximizes the minimum distance among the transformed points in randomized expected time $\mathcal{O}(n \log n)$.*

Open Problem 2. *Is there an efficient deterministic algorithm for maximizing the minimum distance among n points in \mathbb{S}^d , $d \geq 3$?*

4 Applications

4.1 Spherical Graph Drawing

As is by now well known, any planar graph can be represented by a set of disjoint circles in \mathbb{S}^2 , such that two vertices are adjacent exactly when the corresponding two circles are tangent [6, 15, 21]. We call such a representation a *coin graph*; Figure 2 shows an example. Although it seems difficult to represent the positions of the coins exactly, fast algorithms for computing numerical approximations to their positions are known [7, 18, 22]. By polar projection, we can transform any coin graph representation on the sphere to one in the plane or vice versa. See Ken Stephenson’s web site <http://www.math.utk.edu/~kens/> for more information, including software for constructing coin graph representations and a bibliography of circle packing papers.

It is natural to ask for the planar or spherical coin graph representation in which all circles are most nearly the same size. It is NP-hard to determine whether a planar coin graph representation exists in which all circles are equal, or in which the ratio between the maximum and minimum radius satisfies a given bound [5]. However, if the graph is maximal planar, its coin graph representation is unique up to Möbius transformation, and we can apply Theorem 2 to find the optimal spherical coin graph representation. There is also a natural way of obtaining a canonical coin graph representation from a non-maximal embedded planar graph: add a new vertex in each face, connected to all the vertices of the face. Find the coin graph representation of the augmented graph, and delete the circles representing the added vertices. Again, Theorem 2 can then find the optimal Möbius transformation of the resulting coin graph.

Due to the fact that a quasiconvex program only has a single global optimum, the transformed coin graph will display any symmetries present in the initial graph embedding. That is, any homeomorphism of the sphere that transforms the initial embedded graph into itself becomes simply a rotation or reflection of the sphere in the optimal embedding. If the graph has a unique embedding then any isomorphism of the graph becomes a rotation or reflection. For instance, the coin graph representation in Figure 2 (right) has the full symmetry of the underlying graph, while the planar drawing on the left of the figure does not show the symmetries that switch the vertices in the inner and outer squares.

Alternatively, by representing each vertex by the center of its circle, a coin graph representation can be used to find a straight-line drawing of a planar graph, or a drawing on the sphere in which the edges appear as non-crossing great-circle arcs. The algorithms in Section 3.2 can then be used to find a representation maximizing the minimum vertex separation, among all Möbius transformations of the initial vertex positions.

4.2 Hyperbolic Browser

There has been quite a bit of recent work in the information visualization community on *hyperbolic browsers*, techniques for using hyperbolic space to aid in the visualization of large graphs or graph-like structures such as the World-Wide Web [20]. In these techniques, a graph is arranged within a hyperbolic plane [16] or three-dimensional hyperbolic space [19], and viewed using the Klein or Poincaré models, or rendered as it would be seen by a viewer within the hyperbolic space. The main advantage of hyperbolic browsers is that they provide a “fish-eye” view [10] that allows both the details of the current focus of interest and the overall structure of the graph to be viewed simultaneously. In addition, the homogeneous and isotropic geometry of hyperbolic space allows for natural and smooth navigation from one view to another.

Although there are many interesting problems in graph layout for hyperbolic spaces, we are interested in a simpler question: where should one place one’s initial focus, in order to make the overall graph structure as clear as possible? Previous work has handled this problem by the simplistic approach of laying out the graph using a rooted subtree such as a breadth-first or depth-first tree, and

placing the focus at the root of the tree. This approach will work well if the tree is balanced, but otherwise deeper parts of the tree may be given a much more crowded initial view. Instead, one could use our techniques to find a viewpoint that shows the whole graph as clearly as possible.

We assume we are given a graph, with vertices placed in a hyperbolic plane. As in [16], we further assume that each vertex has a circular *display region* where information related to that vertex is displayed; different nodes may have display regions of different sizes. It is straightforward to apply Theorem 1 to these display regions; the result is a focus placement for the Poincaré model that maximizes the minimum size of any display region. We can similarly find the Klein model maximizing the minimum diameter or width of a transformed circle, since the level sets of these functions are again simply concentric balls. By applying Theorem 3 we can instead choose the focus for a Poincaré model that maximizes the minimum distance between vertices, either among pairs from the given graph or among all possible pairs.

Open Problem 3. *Is there an efficient algorithm for choosing a Klein model of a hyperbolically embedded graph that maximizes the minimum Euclidean distance between adjacent vertices?*

We can apply similar methods to find a focus in 3-dimensional hyperbolic space that maximizes the minimum solid angle subtended by any display region.

Open Problem 4. *Does there exist an efficient algorithm to find a viewpoint in 3-dimensional hyperbolic space maximizing the minimum angle separating any pair among n given points?*

Even the 2-dimensional Euclidean version of this maxmin angle separation problem is interesting [17]: the level sets are nonconvex unions of two disks, so our quasiconvex programming techniques do not seem to apply.

4.3 Conformal Mesh Generation

One of the standard methods of two-dimensional structured mesh generation [3, 24] is based on *conformal mapping* (that is, an angle-preserving homeomorphism). The idea is to find a conformal map from the domain to be meshed into some simpler shape such as a disk, use some predefined template to form a mesh on the disk, and invert the map to lift the mesh back to the original domain. Conformal meshes have significant advantages: the orthogonality of the grid lines means that one can avoid certain additional terms in the definition of the partial differential equation to be solved [24]. Nevertheless, despite much work on algorithms for finding conformal maps [8, 12, 22, 23, 25] conformal methods are often avoided in favor of quasi-conformal mesh generation techniques that allow some distortion of angles, but provide greater control of node placement [3, 24].

Möbius transformations are conformal, and any two conformal maps from a simply connected domain to a disk can be related to each other via a Möbius

transformation. However, different conformal maps will lead to different structured meshes: the points of the domain mapped on or near the center of the disk will generally be included in mesh elements with the finest level of detail, and points near the boundary will be in coarser mesh elements. Therefore, as we now describe, we can use our optimal Möbius transformation algorithms to find the conformal mesh that best fits the desired level of detail at different parts of the domain, reducing the number of mesh elements created and providing some of the node placement control needed to use conformal meshing effectively.

We formalize the problem by assuming an input domain in which certain interior points p_i are marked with a desired element size s_i . If we find a conformal map f from the domain to a disk, the gradient of f maps the marked element sizes to desired sizes s'_i in the transformed disk: $s'_i = \|f'(p_i)\|$. We can then choose a structured mesh with element size $\min_i s'_i$ in the disk, and transform it back to a mesh of the original domain. The goal is to choose our conformal map in a way that maximizes $\min_i s'_i$, so that we can use a structured mesh with as few elements as possible. Another way of interpreting this is that s'_i can be seen as the radius of a small disk at $f(p_i)$. What we want is the viewpoint that maximizes the minimum of these radii.

By applying a single conformal map, found using one of the aforementioned techniques, we can assume without loss of generality that the input domain is itself a disk. Since the conformal maps from disks to disks are just the Möbius transformations, our task is then to find the Möbius transformation maximizing $\min_i s'_i$. Since s'_i depends only on the hyperbolic distance of the viewpoint from p_i , the level sets for this problem are themselves disks, so we can solve this problem by the same quasiconvex programming techniques as before. Indeed, we can view this problem as a limiting case of Theorem 1 for infinitesimally small (but unequal) sphere radii.

Open Problem 5. *Because of our use of a conformal map to a low aspect ratio shape (the unit disk), rotation around the viewpoint does not significantly affect element size. Howell [12] describes methods for computing conformal maps to high-aspect-ratio shapes such as rectangles. Can one efficiently compute the optimal choice of conformal map to a high-aspect-ratio rectangle to maximize the minimum desired element size? What if the rectangle aspect ratio can also be chosen by the optimization algorithm?*

4.4 Brain Flat Mapping

In order to visualize and understand the complicated structure of the brain, neuroanatomists have sought methods for stretching its convoluted surface folds onto a flat plane. Hurdal et al. [13] proposed a principled way of performing this stretching via conformal maps: since the surfaces of major brain components such as the cerebellum are topologically disks, one can conformally map these surfaces onto a Euclidean unit disk, sphere, or hyperbolic plane. Hurdal et al. approximate this conformal map by using a fine triangular mesh to represent the brain surface, and forming a coin graph representation of this mesh.

Necessarily, any flat mapping of a curved surface such as the brain’s involves some distortions of area, but the distortions produced by conformal mapping can be severe; thus, it would be of interest to choose the mapping in such a way that the distortion is minimized. As we already noted in section 4.3, the remaining degrees of freedom in choosing a conformal mapping can be described by a single Möbius transformation. Thus, we need to formulate a measure of distortion, and find the transformation optimizing that measure.

Since we want to measure area, and the mapping constructed by the method of Hurdal et al. is performed on triangles of a mesh, the most natural quality measure for this purpose seems to be in terms of those triangles: we want to minimize the maximum ratio a/a' where a is the area of a triangle in the initial three-dimensional map, and a' is the area of its image in the flat map. Unfortunately, we have not yet been able to extend our techniques to this quality measure. A positive answer to the following question would allow us to apply our quasiconvex programming algorithms:

Open Problem 6. *Let T be a triangle in the unit disk or sphere, and let C be the set of viewpoints for Möbius transformations that transform T into a triangle of area at least A . Is C necessarily convex?*

Instead of attempting to optimize the area of the triangles, it seems simpler (although perhaps less accurate) to optimize the area of the disks in the coin graph. Under the assumption that the initial triangular mesh has elements of roughly uniform size, it would be desirable that the coin graph representation similarly uses disks of as uniform size as possible. This can be achieved by our linear-time algorithms by applying Theorem 1 in the unit disk, or Theorem 2 in the sphere. In case the triangular mesh is nonuniform, it may be appropriate to apply a weighted version of these theorems, where the weight of each disk is computed from the lengths of edges incident to the corresponding mesh vertex.

5 Conclusions

We have identified several applications in information visualization and structured mesh generation for which it is of interest to find a Möbius transformation that optimizes an objective function, typically defined as the minimum size of a collection of geometric objects. Further, we have shown that these problems can be solved either by local optimization techniques, or by linear-time quasiconvex programming algorithms. For the problems where the input to the quasiconvex program is itself superlinear in size (maximizing the minimum distance between transformed points) we have described Delaunay triangulation and random sampling techniques for solving the problems in near-linear time.

We have listed open problems arising in our investigations throughout the paper. There is also an important problem in the practical application of our algorithms: although there should be little difficulty implementing local optimization techniques for our problems, the linear-time quasiconvex programming

algorithms are based on two primitives that (while constant time by general principles) have not been specified in sufficient detail for an implementation, one to test a new constraint against a given basis and the other to find the changed basis of a set formed by adding a new constraint to a basis. If the basis representation includes the value of the objective function, testing a new constraint is simply a matter of evaluating the corresponding object size and comparing it to the previous value. However, the less frequent basis change operations require a more detailed examination of the detailed structure of each problem, which we have not carried out. For an example of the difficulty of this step see [11]. In practice it may be appropriate to combine the two approaches, using local optimization techniques to find a numerical approximation to the basis change operations needed for the quasiconvex programming algorithms. Especially in the coin graph application, the input to the quasiconvex program is already itself a numerical approximation, so this further level of approximation should not cause additional problems, but one would need to verify that a quasiconvex programming algorithm can behave robustly with approximate primitives.

More generally, there are very few computational geometry algorithms involving hyperbolic geometry (a notable exception being [4]) although many Euclidean constructions such as the Delaunay triangulation or hyperplane arrangements can be translated to the hyperbolic case without difficulty using the Poincaré or Klein models. We expect many other interesting problems and algorithms to be discovered in this area.

Acknowledgements

We thank Hervé Brönnimann, Fan Chung, and Ken Stephenson for helpful discussions. Work of Eppstein was done in part while visiting Xerox PARC and supported in part by NSF grant CCR-9912338.

References

1. N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. *J. Algorithms* 30(2):302–322, February 1999, cs.CG/9809081.
2. M. Bern and D. Eppstein. Optimal Möbius transformations for information visualization and meshing. ACM Computing Research Repository, 2001, cs.CG/0101006.
3. M. Bern and P. E. Plassmann. Mesh generation. *Handbook of Computational Geometry*, chapter 6, pp. 291–332. Elsevier, 2000.
4. J.-D. Boissonnat, A. Cérézo, O. Devillers, and M. Teillaud. Output-sensitive construction of the Delaunay triangulation of points lying in two planes. *Int. J. Comp. Geom. & Appl.* 6(1):1–14, 1996.
5. H. Brey and D. G. Kirkpatrick. On the complexity of recognizing intersection and touching graphs of disks. *Proc. 3rd Int. Symp. Graph Drawing*, pp. 88–98. Springer-Verlag, Lecture Notes in Comput. Sci. 1027, 1995.
6. G. R. Brightwell and E. R. Scheinerman. Representations of planar graphs. *SIAM J. Discrete Math.* 6(2):214–229, 1993.

7. C. Collins and K. Stephenson. A circle packing algorithm. Manuscript, September 1997, <http://www.math.utk.edu/~kens/ACPA/ACPA.ps.gz>.
8. T. A. Driscoll and S. A. Vavasis. Numerical conformal mapping using cross-ratios and Delaunay triangulation. *SIAM J. Sci. Comput.* 19(6):1783–1803, 1998, <ftp://ftp.cs.cornell.edu/pub/vavasis/papers/crdt.ps.gz>.
9. D. Eppstein. Setting parameters by example. *Proc. 40th Symp. Foundations of Computer Science*, pp. 309–318. IEEE, October 1999, cs.DS/9907001.
10. A. Formella and J. Keller. Generalized fisheye views of graphs. *Proc. 3rd Int. Symp. Graph Drawing*, pp. 243–253. Springer-Verlag, Lecture Notes in Comput. Sci. 1027, 1995, <http://www-wjp.cs.uni-sb.de/~formella/dist2.ps.gz>.
11. B. Gärtner and S. Schönherr. Exact primitives for smallest enclosing ellipses. *Inf. Proc. Lett.* 68:33–38, 1998.
12. L. H. Howell. *Computation of Conformal Maps by Modified Schwarz-Christoffel Transformations*. Ph.D. thesis, MIT, 1990, <http://www.llnl.gov/CASC/people/howell/lhhphd.ps.gz>.
13. M. K. Hurdal, P. L. Bowers, K. Stephenson, D. W. L. Summers, K. Rehm, K. Shaper, and D. A. Rottenberg. Quasi-conformally flat mapping the human cerebellum. Tech. Rep. FSU-99-05, Florida State Univ., Dept. of Mathematics, 1999, <http://www.math.fsu.edu/~aluffi/archive/paper98.ps.gz>.
14. B. Iversen. *Hyperbolic Geometry*. London Math. Soc. Student Texts 25. Cambridge Univ. Press, 1992.
15. P. Koebe. Kontaktprobleme der konformen Abbildung. *Ber. Verh. Sächs. Akad. Wiss. Leipzig Math.-Phys. Kl.* 88:141–164, 1936.
16. J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for viewing large hierarchies. *Proc. Conf. Human Factors in Computing Systems*, pp. 401–408. ACM, 1995, <http://www.parc.xerox.com/istl/projects/uir/pubs/pdf/UIR-R-1995-04-Lamping-CHI95-FocusContext.pdf>.
17. J. McKay. Sighting point. sci.math, 20 April 1989, <http://www.ics.uci.edu/~eppstein/junkyard/maxmin-angle.html>.
18. B. Mohar. A polynomial time circle packing algorithm. *Discrete Math.* 117(1–3):257–263, 1993.
19. T. Munzner. Exploring large graphs in 3D hyperbolic space. *IEEE Comp. Graphics & Appl.* 18(4):18–23, 1997, <http://graphics.stanford.edu/papers/h3cga/>.
20. T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3D hyperbolic space. *Proc. VRML '95*, pp. 33–38. ACM, 1995, <http://www.geom.umn.edu/docs/research/webviz/webviz/>.
21. H. Sachs. Coin graphs, polyhedra, and conformal mapping. *Discrete Math.* 134(1–3):133–138, 1994.
22. W. D. Smith. Accurate circle configurations and numerical conformal mapping in polynomial time, <http://www.neci.nj.nec.com/homepages/wds/braegger.ps>. Manuscript, December 1991.
23. F. Stenger and R. Schmidlein. Conformal maps via sinc methods. *Proc. Conf. Computational Methods in Function Theory*, pp. 505–549. World Scientific, 1997, <http://www.cs.utah.edu/~stenger/PAPERS/stenger-sinc-comformal-maps.ps>.
24. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. *Numerical Grid Generation: Foundations and Applications*. North-Holland, 1985.
25. L. N. Trefethen. Numerical computation of the Schwarz-Christoffel transformation. *SIAM J. Sci. Stat. Comput.* 1(1):82–102, 1980.