Discrete Conformal Mappings via Circle Patterns

LILIYA KHAREVYCH
California Institute of Technology
BORIS SPRINGBORN
Technische Universität Berlin
PETER SCHRÖDER
California Institute of Technology

We introduce a novel method for the construction of discrete conformal mappings from surface meshes of arbitrary topology to the plane. Our approach is based on *circle patterns*, i.e., arrangements of circles—one for each face—with prescribed intersection angles. Given these angles the circle radii follow as the *unique* minimizer of a convex energy. The method supports very flexible boundary conditions ranging from free boundaries to control of the boundary shape via prescribed curvatures. Closed meshes of genus zero can be parameterized over the sphere. To parameterize higher genus meshes we introduce cone singularities at designated vertices. The parameter domain is then a piecewise Euclidean surface. Cone singularities can also help to reduce the often very large area distortion of global conformal maps to moderate levels. Our method involves two optimization problems: a quadratic program and the unconstrained minimization of the circle pattern energy. The latter is a convex function of *logarithmic* radius variables with simple explicit expressions for gradient and Hessian. We demonstrate the versatility and performance of our algorithm with a variety of examples.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling

General Terms: Algorithms; Theory;

Additional Key Words and Phrases: Conformal parameterizations; discrete analytic functions; discrete differential geometry; circle patterns; meshing; texture mapping

1. INTRODUCTION

Surfaces are often represented as collections of samples with connectivity, typically in the form of a simplicial mesh. It is natural and convenient to use the implied piecewise linear mesh as the basis for formulating a variety of computational algorithms, such as parameterization problems or the solution of partial differential equations for purposes of simulation. In this paper we argue that when it comes to computing conformal structures, e.g., conformal parameterizations of surfaces, circles can be a far better basis upon which to formulate the underlying relationships and consequent algorithms (see Figure 1). In particular we advocate the formulation of the *discrete conformal mapping*[1] problem in terms of circles and the angles with which they intersect, so called *circle patterns*.

The idea of using circles to capture a discrete notion of conformality goes back to a conjecture of Thurston's [1985] who posited that one may approximate the Riemann mapping[2] from a given region in the plane to the unit disk through a sequence of increasingly fine, regular (hexagonal) *circle packings*. This conjecture was later proven correct by Rodin

[1]In this article we use the term "discrete conformal map" for piecewise linear maps between meshes that are close to angle preserving.
[2]The Riemann Mapping Theorem asserts that there exists a unique (up to Möbius transformations of the unit disk to itself) conformal map from any region in the plane (open, connected and simply connected, not the whole plane) onto the unit disk.

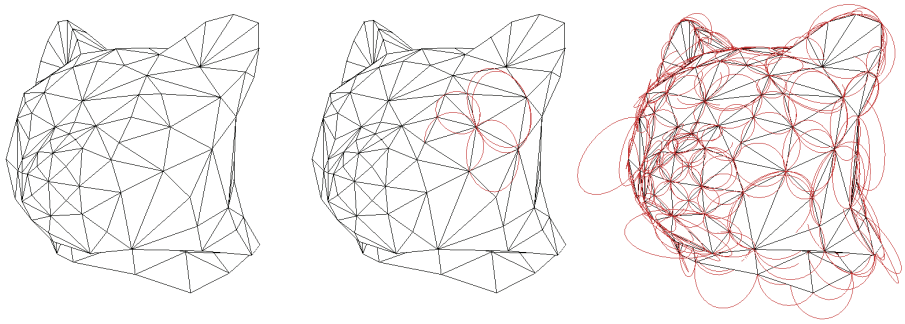

Fig. 1: *Typically a triangle mesh is understood as the piecewise linear interpolation of given vertex coordinates induced by the connectivity of the mesh (left). Alternatively we may also think of the vertices as the unique loci where incident triangle circumcircles intersect (middle;right). The latter point of view is more appropriate for formulating relationships of conformal geometry.*

and Sullivan [1987]. Circle packings assign a circle to each vertex, with pairwise tangency for each edge in the mesh. The task then is to find radii for these circles such that the combinatorially prescribed tangencies are maintained and the resulting arrangement of circles fills the unit disk. A numerical algorithm for the construction of such mappings, based on iterative adjustment of circle radii, was proposed by Thurston [1980] and improved and realized by Collins and Stephenson [2003]. Unfortunately, circle *packings* yield mappings which depend only on the *combinatorics* of the original mesh, while we are seeking methods which depend on the *geometry* of the mesh. One possible avenue to remedy this shortcoming is to use patterns of non-intersecting circles [Bowers and Hurdal 2003].

In contrast, circle *patterns*, which associate a circle with each *face* in the original mesh provide an opportunity to incorporate the intrinsic geometry of the original mesh: each edge is assigned an angle $\theta \in (0, \pi)$ which corresponds to the intersection angle of the two incident face circles. A recent theorem of Bobenko and Springborn [2004] characterizes such circle patterns as the *unique* minimizer of a convex energy expressed in terms of *logarithmic* radius variables (and the given edge angles). Simple, explicit expressions for the energy, its gradient, and Hessian are available and greatly facilitate an efficient implementation.

1.1 Contributions

In this paper we use the theory of Bobenko and Springborn to *realize effective and robust algorithms* for the construction of discrete conformal mappings from triangle meshes of arbitrary topology to the plane. The basic algorithm consists of three stages (Section 3). In a first step each edge of the input mesh is assigned an angle $0 < \theta_e < \pi$. These angles serve to incorporate the original geometry into the circle pattern algorithm. Choosing "good" angles is achieved by solving a quadratic program (Section 3.1). Once the angles have been assigned the circle radii are found as the unique minimum of a convex energy (Section 3.3). Finally the edge angles together with the found radii are used to lay out the mesh in the parameter domain. The mappings are always locally injective. They may fail to be globally injective due to self-overlap of the boundary of the parameter domain. However, this can be avoided since we can prescribe the boundary curvature κ: if for any sequence of consecutive boundary vertices the sum of κs is larger or equal $-\pi$, then there can be no overlap and the method is guaranteed to produce a global embedding.

We develop our method in stages. First the theory is introduced (Section 2) followed by its application to meshes with disk topology (see Figure 11). Such meshes may be given in that form or result after cutting a mesh with non-trivial topology. In the latter case there is a priori no continuity of the mapping across the cut borders. For the topological disk case we introduce boundary conditions which allow explicit control over the boundary curvature (see Figure 2). This can be used to achieve polygonal outlines all the while maintaining discrete conformality. This is of interest, e.g.,

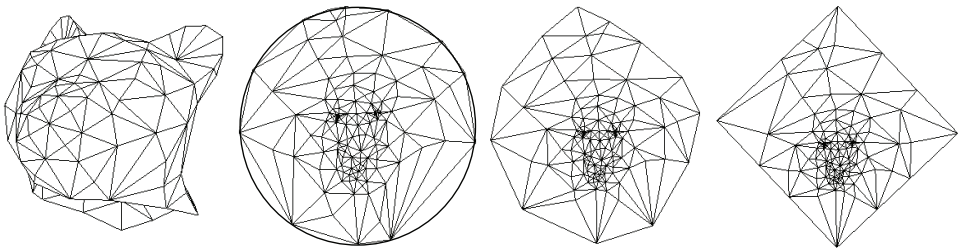

Fig. 2: *A simple mesh (left) mapped to the plane using circle patterns. Boundary shape is controlled through appropriate curvature conditions. Examples: disk boundary; free boundary; and rectangular boundary (left to right).*

in the context of more efficient texture packing (see Figure 11; Max Planck example).

The basic method is then used to compute discrete conformal parameterizations of closed surfaces of genus zero over the sphere (see Figure 7) in Section 4.1 and mappings to the disk in Section 4.2.

Since the circle pattern algorithm always produces planar meshes which are Delaunay, the basic method produces larger than necessary error for input meshes that are far from Delaunay. We adress this issue in Section 5. In a preprocessing step *which does not change the intrinsic geometry of the mesh*, we construct an *intrinsic* Delaunay triangulation to be fed into the basic algorithm (see Figure 9 for a comparison of distortion measures).

For surfaces of higher genus ($g > 1$) we introduce *cone singularities* (Section 6): One may designate a number of vertices as cone vertices with prescribed cone angles. (A cone vertex is characterized as a vertex where the tip angles of incident triangles do not sum to 2π, i.e., vertices with non-vanishing discrete Gauss curvature.) The resulting parameterizations are Euclidean everywhere except at the cone vertices. Parameterizations constructed in this manner are globally continuous. To pack them into a texture plane, they must be cut. However, the placement of the cuts has no impact on either continuity or distortion (this is in contrast to algorithms which cut before parameterizing). Furthermore, diligently placed cone singularities can reduce the area distortion of discrete conformal maps greatly (see Figures 14–18).

Finally we evaluate our approach in terms of performance and distortion measures and compare it with the most closely related alternative method [Sheffer and de Sturler 2000] (which is also based on angle optimization albeit with different mathematical underpinnings).

1.2 Related Work

Most approaches to the construction of conformal mappings for meshes have relied on *discretizations of continuous formulations*. First order finite element approximations of the Cauchy-Riemann equations were used by Levy et al. [2002]. The same equations, the so-called "cotan formula" [Pinkall and Polthier 1993], also result when considering a discrete variational Ansatz based on mesh invariants [Desbrun et al. 2002] or when deriving discrete holomorphy [Mercat 2001] from first principles using Discrete Exterior Calculus (DEC) methods [Hirani 2003]. Such approaches, for example, have been used to compute discrete approximations of Riemann structures for general meshes by Gu and co-workers (see [Gu and Yau 2003] and the references therein).

A great advantage of these methods is that they require only the solution of simple linear systems. However, due to negative cotan weights solutions may lack local injectivity. More troublesome is the lack of flexible boundary conditions: Dirichlet conditions introduce non-conformal distortion (see Figure 3), while "natural" boundaries provide essentially no control over the boundary shape.

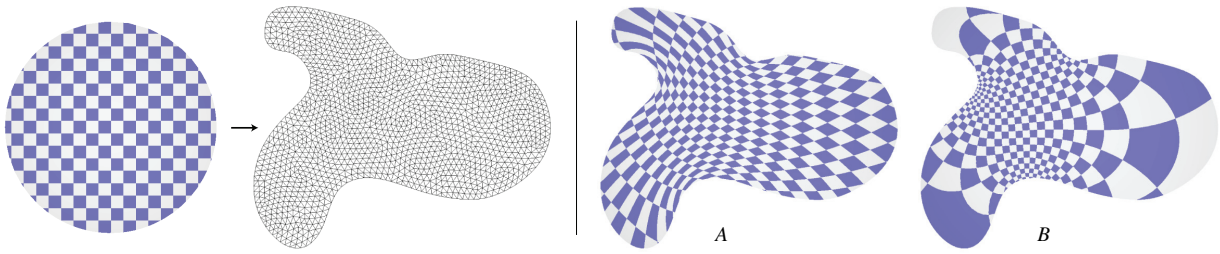

Fig. 3: *Example of a map from the disk (shown with superimposed checker board pattern) to a region shown as a very nicely sampled mesh. Using the cotan formula one must specify Dirichlet boundary conditions. For example, mapping vertices on the boundary of the region to points on the boundary of the disk with matching (relative) secant lengths. The resulting (inverse) mapping is visualized in (A). Note how squares are sheared, indicating severe angle distortion. (B) shows the resulting mapping using our approach which sets appropriate* angle *conditions at the boundary.*

A completely different Ansatz to the construction of conformal maps is based on circle *packing*. Continuous conformal mappings can be characterized as mapping infinitesimal circles to infinitesimal circles. Circle packings replace infinitesimal circles with finite circles. In the limit of refinement the continuous conformal maps are recovered [Rodin and Sullivan 1987]. Collins and Stephenson [2003] have implemented these ideas in their software `CirclePack`. The disadvantage of using circle packings (with tangent circles) is that they depend only on the combinatorics of the original mesh. In particular, if one starts with a planar mesh and parameterizes it, the result is not the original mesh. This is in contrast to our approach. Given a triangulation of a region in the plane satisfying the empty circumcircle property, one simply assigns the observed intersection angles of circumcircles (at the boundary one adds infinite circles, i.e., straight lines) to each edge and our variational approach will return the identity map as a conformal map of the region to itself. This is because the Delaunay triangulation is uniquely determined by the abstract triangulation and the intersection angles, hence the original mesh is the only solution.

An extension of Stephenson's original circle packing scheme that takes the geometry of the original mesh into account is based on patterns of non-intersecting circles, so called *inversive distance circle packings* [Bowers and Hurdal 2003]. (Non-intersecting circles have imaginary "intersection angles.") The idea is to construct patterns of non-intersecting circles, each circle corresponding to a vertex of a triangulation, with prescribed inversive distances between circles corresponding to neighboring vertices. But whereas practical conditions for the existence and uniqueness of circle patterns with intersecting circles are known (see, e.g., Section 2), virtually nothing is known regarding the existence and uniqueness of inversive distance packings.

The first variational principle for circle packings was presented in a seminal paper by Colin de Verdière [1991]. The variables are the circle radii. However, a closed formula is presented only for the derivative of the energy, not for the energy itself. Since then, different variational principles for circle *packings* [Brägger 1992] as well as circle *patterns* [Rivin 1994; Leibon 2002] were discovered. In these, the variables are the angles of a triangulation, subject to numerous linear constraints (one per edge and one per face for Rivin's energy). Leibon's energy deals with patterns in the hyperbolic plane. In this paper, we use the most general functional which was given by Bobenko and Springborn [2004]. In their setup, the variables are logarithmic circle radii. Most importantly, they are not subject to any constraints.

Most closely related to our approach is the angle based flattening (ABF) algorithm of Sheffer and de Sturler [2000]. They flatten a given mesh by formulating a constrained quadratic minimization problem which seeks to find angles at the corners of triangles which are close to desired angles in a weighted l_2 norm. The constraints capture the angle sum conditions at all faces (sum of angles $= \pi$) and vertices (sum of angles $= 2\pi$) as well as a non-linear condition on the product of sines of angles. The resulting minimization problem has local minima and does not have a unique solution.

The resulting mappings are of excellent quality but only free boundary conditions are provided (for a formulation of ABF using additional boundary curvature constraints see [Zayer et al. 2003]). It may also be possible to incorporate cone singularities into the ABF method, but this has not yet been demonstrated. No less, ABF is similar in spirit to our algorithm as we also optimize angles. We discuss the similarities and differences of both schemes in Section 3.2.

We defer the discussion of related work as regards the use of cone singularities to Section 6.

2. CIRCLE PATTERNS

For computational problems in Euclidean geometry the use of triangles as a basic primitive is convenient and natural, since triangles are the basic invariant "building blocks" of Euclidean geometry. When one is interested in *conformal* geometry the picture changes. The basic invariants of conformal geometry are circles and the angles they make with one another. In the case of triangle meshes these differing points of view are naturally compatible. For example, the vertices in a triangle mesh are the unique loci where the circumcircles of the incident triangles intersect (see Figure 1). Similarly the empty circumcircle property, which corresponds to non-negative intersection angles between circumcircles incident on an edge, is a defining feature of Delaunay triangulations. While we may use triangles for tasks such as interpolation and rendering, conformal relationships between vertices are better captured through expressions involving the circumcircles they define and the angles these circles make with one another. A benefit of this different point of view is that much mathematical machinery from conformal geometry carries over to the discrete computational setting. For example, existence and uniqueness properties of conformal maps are reflected in the existence and uniqueness properties of circle patterns.

We begin this section by defining circle patterns in the plane and describing their characterization as minimizers of a variational energy. The latter forms the basis for our approach. While we only deal with triangle meshes here, the theory extends to polyhedral meshes [Bobenko and Springborn 2004].

Consider a Delaunay triangulation $\mathscr{T} = (V,E,T)$ of finitely many points $P = \{p_i\}$ in the plane. Here $V = \{v_i\}$, $E = \{e_{ij}\}$, and $T = \{t_{ijk}\}$ denote the sets of vertices, edges, and triangles respectively with p_i the point position of vertex v_i. From this Delaunay triangulation, we may read off the following edge weights

$$\forall e_{ij} \in E : \theta_e = \begin{cases} \pi - \alpha_{ij}^k - \alpha_{ij}^l & \text{for interior edges} \\ \pi - \alpha_{ij}^k & \text{for boundary edges} \end{cases}, \tag{1}$$

where α_{ij}^k (and α_{ij}^l) are the angle(s) opposite e_{ij} in the adjacent triangle(s) t_{ijk} (and t_{jil}). The θ-weight of an interior edge is the (exterior) intersection angle of the circumcircles of the incident triangles (see Figure 4). The θ-weight of

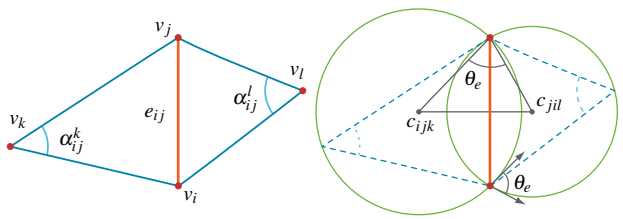

Fig. 4. Notation: The angles α_{ij}^k and α_{ij}^l opposite a given edge e_{ij} and incident on a particular vertex v_k respectively v_l. The edge angle θ_e denotes the exterior intersection angle of the incident circumcircles or equivalently the angle between the two radii at v_i (or v_j).

a boundary edge is the intersection angle of the circumcircle of the incident triangle with the straight line containing the boundary edge. (View this line as a circle with infinite radius.)

Assume that the Delaunay triangulation is unique (points are in general position). Then

$$\forall e_{ij} \in E : 0 < \theta_e < \pi. \tag{2}$$

(Otherwise, some θ_e may be 0.) For an interior vertex v_i, the sum of edge weights on the incident edges is 2π:

$$\forall v_i \in V_{\text{int}} : \sum_{e \ni v_i} \theta_e = 2\pi, \tag{3}$$

while for a boundary vertex v_i, the defect

$$\forall v_i \in V_{\text{bdy}} : \kappa_i = 2\pi - \sum_{e \ni v_i} \theta_e \tag{4}$$

is the curvature angle of the polygonal boundary at that vertex (see Figure 5). Since the Delaunay triangulation

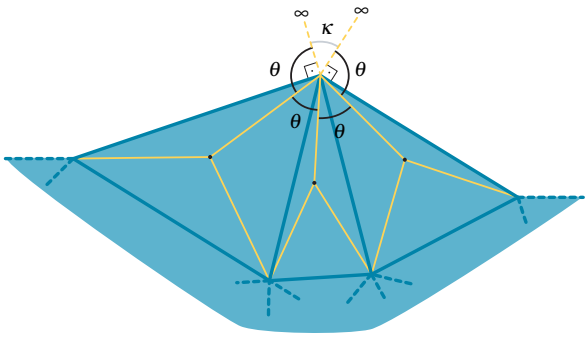

Fig. 5: *The θ angle sum at the boundary contains the discrete curvature term κ.*

triangulates the convex hull of the sites, $0 \leq \kappa_i < \pi$. However, we want to consider a slightly more general setup: *Instead of a Delaunay triangulation, we may start with a flat PL-surface that is topologically a disk and that is triangulated in such a way that the edge weights θ satisfy the local Delaunay condition (Equation 2).* That is, we allow "Delaunay triangulations" of non-convex regions with polygonal boundary. Hence, κ_i may be negative, and we speak of *circle patterns* instead of Delaunay triangulations.

Now the idea is to reconstruct a circle pattern from its abstract triangulation and the intersection angles.

Circle Pattern Problem. Given an abstract triangulation $\mathscr{T}$ of a topological disk and a function $\theta \in \mathbb{R}^E$ on the edge set E that satisfies Equation 2 and the angle sum condition for interior vertices (Equation 3), *find* a circle pattern that is combinatorially equivalent to $\mathscr{T}$ and has the given edge weights θ.

The circle pattern problem has a solution (unique up to scale) *if and only if* a coherent angle system exists [Rivin 1994; Bobenko and Springborn 2004]. A *coherent angle system* is an assignment of angles $\hat{\alpha}_{ij}^k$ for all triangles such that

(i) they are all positive

$$\hat{\alpha}_{ij}^k > 0,$$

(ii) they sum to π in each triangle

$$\forall t_{ijk} \in T : \hat{\alpha}_{ij}^k + \hat{\alpha}_{jk}^i + \hat{\alpha}_{ki}^j = \pi,$$

(iii) they satisfy Equations 1 (with $\hat{\alpha}$ instead of α) for the *given* θ-weights.

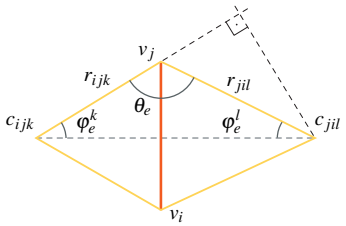

Fig. 6. Geometry around an edge (left). The kite formed by the edge endpoints and the incident face circumcircle centers (c_{ijk}, c_{jil}) allows us to determine φ_e^k as a function of the given θ_e and unknown radii r_{ijk} and r_{jil} (see Equation 6).

The solvability question for a circle pattern problem is therefore reduced to a linear feasibility problem with $3|T|$ variables, $3|T|$ inequality constraints and $|T|+|E|$ equality constraints.

Conditions (i) and (ii) imply that all $\hat{\alpha}_{ij}^k < \pi$. Condition (iii) implies that the $\hat{\alpha}_{ij}^k$ sum to 2π around interior vertices and to $\pi - \kappa_i$ for boundary vertices (with κ_i defined by Equation 4). This may give the false impression that finding a coherent angle system is equivalent to solving the circle pattern problem—in the sense that one could construct triangles with angles $\hat{\alpha}_{ij}^k$ and lay them out. This is not so. The angles determine the triangles only up to scale. In general it is not possible to determine the size of each triangle in such a way that they all fit together. This observation is also what lead Sheffer and de Sturler [2000] to add their non-linear constraints on the product of sines of triangle angles.

Conditions (i), (ii) and (iii) combined imply that a necessary condition for the solvability of a circle pattern problem is that

$$\sum_{v_i \in V_{\text{bdy}}} \kappa_i = 2\pi, \tag{5}$$

with κ_i defined by Equation 4.

Local Geometry of an Edge. To elucidate the role of terms which make up the variational energy characterizing the solution to the circle pattern problem we consider the local geometry around a given edge (see Figure 6). The basic building blocks of the parameterization are the *kites* formed by the endpoints of an edge (v_i, v_j) and the incident face circumcircle centers c_{ijk} and c_{jil} for triangles t_{ijk} and t_{jil} respectively (at the boundary t_{jil} is missing). Since all relations are scale invariant it is convenient to introduce the *logarithmic* radius variables $\rho_t = \log r_t$ for $t \in T$. With these definitions the angle φ_e^k induced at c_{ijk} by e_{ij} follows as

$$\varphi_e^k = \begin{cases} f_e(x) = \operatorname{atan2}(\sin\theta_e, e^x - \cos\theta_e) & e \in E_{\text{int}} \\ \pi - \theta_e & e \in E_{\text{bdy}} \end{cases} \tag{6}$$

where $x = \rho_{ijk} - \rho_{jil}$ (see Figure 6). The condition that every face in the parameterization should be flat constrains the ρ_t as functions of the θ_e through the system of non-linear equations

$$\forall t \in T : 0 = 2\pi - \sum_{e \in t} 2\varphi_e^t, \tag{7}$$

i.e., summing around all edges incident on a face the resulting total angle must be 2π.

The basic idea of the variational energy formulation of Bobenko and Springborn now is to give an energy $S(\rho)$ for which the face flatness Equations 7 are equivalent to the vanishing of the energy gradient, $\nabla_\rho S = 0$.

Variational Characterization of Circle Patterns. To arrive at the desired energy Bobenko and Springborn used the fact that

$$F_e(x) = \int_{-\infty}^{x} f_e(\xi)\, d\xi = \operatorname{Im}\operatorname{Li}_2(e^{x+i\theta_e}),$$

where

$$\mathrm{Li}_2(x) = -\int_0^x \frac{\log(1-\xi)}{\xi}\, d\xi \overset{|x|\le 1}{=} \sum_{k=1}^{\infty} \frac{x^k}{k^2}$$

denotes the dilogarithm function. The imaginary part of the dilogarithm function of a complex argument can be expressed in terms of a 2π-periodic real function (Clausen's integral) that can be computed efficiently with high accuracy. Using ρ_k and ρ_l as shorthand for ρ_{ijk} (respectively ρ_{jil}) the energy is

$$S(\rho) = \sum_{e\in E_{\mathrm{int}}} \left(\mathrm{Im}\,\mathrm{Li}_2(e^{\rho_k-\rho_l+i\theta_e}) + \mathrm{Im}\,\mathrm{Li}_2(e^{\rho_l-\rho_k+i\theta_e}) - (\pi-\theta_e)(\rho_k+\rho_l)\right) - \sum_{e\in E_{\mathrm{bdy}}} 2(\pi-\theta_e)\rho_k + 2\pi\sum_{t\in T}\rho_t. \tag{8}$$

The partial of this energy with respect to ρ_k is

$$\frac{\partial S}{\partial \rho_k} = 2\pi - \sum_{\{e\in t_{ijk}\}\cap E_{\mathrm{int}}} 2f_e(\rho_k-\rho_l) - \sum_{\{e\in t_{ijk}\}\cap E_{\mathrm{bdy}}} 2(\pi-\theta_e), \tag{9}$$

giving us the desired equivalence of $\nabla_\rho S = 0$ and Equation 7. The Hessian of the energy is

$$d\rho^T(\mathrm{Hess}\,S)d\rho = \sum_{e\in E_{\mathrm{int}}} \frac{\sin\theta_e}{\cosh(d\rho_k - d\rho_l) - \cos\theta_e}(d\rho_k - d\rho_l)^2. \tag{10}$$

In particular from this expression we can see that the energy is convex except along the scaling "direction," i.e., the Hessian has a null space spanned by the constant vector $d\rho = (1,1,1,\ldots)$ and is otherwise positive. (This immediately implies the uniqueness of solutions of the circle pattern problem up to scale.)

3. BASIC ALGORITHM

To exploit the theory laid out in the previous section for a practical algorithm we need three basic stages: (1) setting the θ angles; (2) minimizing the energy; and (3) generating the layout. We discuss these in turn.

3.1 Edge Angles

As a first step of the algorithm, θ_e angles need to be assigned to all edges of the mesh. These must satisfy the bounds constraints (Equation 2), sum conditions (Equation 3) at interior vertices, and, in the case of prescribed boundary curvatures, the boundary curvature conditions (Equations 4 and 5). Last but not least, a coherent angle system must exist for them. Of course, the θ angles should also reflect the conformal structure of the original mesh as well as possible. Let α_{ij}^k denote the angles in the original mesh. Ideally, one would like to assign $\theta_e = \pi - \alpha_{ij}^k - \alpha_{ij}^l$, where α_{ij}^k and α_{ij}^l are the angles opposite an interior edge e (and $\theta_e = \pi - \alpha_{ij}^k$ for a boundary edge.) Then the intersection angles in the circle pattern would be the same as the intersection angles of the circumcircles in the mesh. This is too much to ask, though, because the conditions on θ_e will be violated (after all the original mesh is not flat).

Our aim is to find angles $\hat{\alpha}_{ij}^k$ close to α_{ij}^k that can serve as a coherent angle system for the θ-angles that we read off in the usual way. To this end, we minimize the objective function

$$Q(\hat{\alpha}) = \sum |\hat{\alpha}_{ij}^k - \alpha_{ij}^k|^2$$

subject to the following constraints:

—positivity: $\forall \hat{\alpha}_{ij}^k : \hat{\alpha}_{ij}^k > 0$,

—local Delaunay condition: $\forall e_{ij} \in E_{\mathrm{int}} : \hat{\alpha}_{ij}^k + \hat{\alpha}_{ij}^l < \pi$,

—triangle sum condition: $\forall t_{ijk} \in T : \hat{\alpha}_{ij}^k + \hat{\alpha}_{jk}^i + \hat{\alpha}_{ki}^j = \pi$,

—vertex sum condition: $\forall v_k \in V_{\text{int}} : \sum_{t_{ijk} \ni v_k} \hat{\alpha}_{ij}^k = 2\pi$.

If we want free boundary conditions (see for example the face in Figure 11 and the lion and Max Planck examples in Figures 12, 13), we add the constraint

—free boundary condition: $\forall v_k \in V_{\text{bdy}} : \sum_{t_{ijk} \ni v_k} \hat{\alpha}_{ij}^k < 2\pi$.

If we want to prescribe the boundary curvature κ at boundary vertices (see the straightline layout for Max Planck in Figure 11), we add the constraint

—prescribed boundary curvature: $\forall v_k \in V_{\text{bdy}} : \sum_{t_{ijk} \ni v_k} \hat{\alpha}_{ij}^k = \pi - \kappa_k$.

We solve this quadratic minimization problem with linear inequality constraints (on the $\hat{\alpha}_{ij}^k$) with the software [Mosek 2005]. Since both the bounds constraints and the objective are convex and the additional constraints linear we have experienced no difficulty finding solutions efficiently (see Section 7).

Then we set $\theta_e = \pi - \hat{\alpha}_{ij}^k - \hat{\alpha}_{ij}^l$ on interior edges and $\theta_e = \pi - \hat{\alpha}_{ij}^k$ on boundary edges and proceed to the next stage of energy minimization.

3.2 Discussion

It is theoretically possible that the constraints in the above quadratic program are not feasible. In the case of free boundary conditions (as well as mapping to the disk; see Section 4.1) this is due to the (counterintuitive) fact that there exist triangulations of the topological sphere that cannot be realized as convex polyhedra with vertices on the unit sphere (see, e.g., Grünbaum [2003]). These triangulations are rather special; fairly weak sufficient conditions for Delaunay-realizability are given by Dillencourt and Smith [1996]. We do not expect to encounter non-Delaunay-realizable triangulations in practice. In any case, one can show that a 4 to 1 refinement applied once to all triangles *always* results in a legal triangulation. (Indeed, it is known that any triangulation of genus 0 can be realized as a convex polyhedron with edges tangent to the sphere; see [Ziegler 2004] and the references therein. From this realization, one can easily construct a Delaunay realization of the 4 to 1 refinement.) In the case of prescribed boundary curvatures, it may happen that the constraints become infeasible. We have not encountered any problems in all the examples we have computed.

Angle optimization is also at the heart of the work of Sheffer and de Sturler [2000], who formulate their flattening problem as one which minimizes the (weighted) least squares deviation of the measured α_{ij}^k (scaled to satisfy the angle sum condition around a vertex) from the realizable $\hat{\alpha}_{ij}^k$ with flatness sum conditions for each triangle and each vertex. Unfortunately, when using the $\hat{\alpha}_{ij}^k$ to determine the final shape of the triangles in the flat mesh, one must additionally include non-linear conditions on the quotients of sines of $\hat{\alpha}_{ij}^k$ around each vertex (due to the law of sines). This additional non-linear condition makes the minimization problem numerically much harder since it becomes non-convex (for recent significant progress in the numerical treatment see [Sheffer et al. 2004]). Due to the lack of convexity, it cannot be expected that there is only one local minimum.

In contrast to the approach of Sheffer and de Sturler, our method splits the optimization into two stages. First we change the angles of the triangles, but only so much that we can read off valid θ-angles. The deviation from the ideal angles is measured with a quadratic objective just as in the work of Sheffer and de Sturler. The critical observation is that the sum of two α-angles across an edge is a Möbius invariant (because it measures the angle of intersection between the circumcircles), while the α-angles themselves are not. In other words, the best one can hope for in a discrete conformal mapping is conservation of θ-angles, not α-angles. In the second stage (see below) the given

θ-angles are then used to find appropriate radii. This stage is an *unconstrained* convex minimization problem. Due to the convexity of the energy the solution is unique: the final circle pattern in the plane is completely determined by the θ-angles.

In summary, Sheffer and de Sturler optimize the α-angle deviation subject to non-linear constraints (the quotients of sines), which encode the compatibility of edge lengths. We first optimize θ-angles with only linear constraints (a quadratic program) followed by solving for compatible edge lengths through finding appropriate radii (a simple convex minimization without any constraints).

3.3 Energy Minimization

With a given valid assignment of θ_e for all edges the unconstrained energy minimization, with the unknown (logarithmic) circle radii as variables, is straight forward (using Equations 8, 9, and 10). Since gradient and Hessian of the energy are available, and furthermore the Hessian is non-negative, standard Newton methods will easily find the minimum. Instead of writing our own solver we have relied on a black box energy minimizer [Mosek 2005] with excellent results (see Section 7).

3.4 Layout Generation

Once all radii have been determined through the energy minimization the length of each edge follows easily from a *local* computation

$$|e_{ij}| = 2r_k \sin \varphi_e^k = 2r_l \sin \varphi_e^l$$

(see Figure 6). We begin the layout with some interior edge, starting it at the origin and orienting it along the positive x-axis and push it onto an empty stack. When popping an edge off the stack we lay out any vertices not yet laid out in the incident triangles (at most two). This results in at most four edges being finished (second end point fixed); such edges are pushed onto the stack. The process repeats until the stack is empty. There may be concern that such a procedure might lead to cumulating error as one proceeds. We have found consistently that we achieve accuracies on the order of 10^{-8} with double precision data (for example for the Feline, Hand, Letter "a", Max Planck, and Lion datasets). Should accuracy of the layout become an issue we recommend the procedure given in [Sheffer et al. 2004], which solves for all vertex positions simultaneously with a simple (least squares induced) linear system.

4. MAPPING TO THE SPHERE AND TO THE DISK

In this Section we describe how we map a closed genus zero mesh to the sphere and a mesh with disk topology to a circular disk. By means of the stereographic projection, these problems are both reduced to mapping the original mesh (sans a single vertex and its incident triangles) to a region in the plane with a boundary as desribed in Section 3.

4.1 Mapping to the Sphere

Consider a Delaunay triangulation of the whole sphere and the circle pattern formed by the circumcircles of all triangles. Choose one vertex as designated north pole and project the sphere stereographically to the equatorial plane. Since stereographic projection is conformal and maps circles to circles (and lines), we get a circle pattern in the plane with the same circle intersection angles. The north pole is mapped to infinity. All the circles through the north pole are mapped to straight lines. The result is thus a planar Delaunay triangulation, the faces of which correspond to the faces of the spherical triangulation so long as they are not incident to the north pole. Note that even though all triangles and

edges incident to the designated north pole vertex have been removed this data can be recovered from the fact that the north pole vertex gets mapped to infinity.

To construct a spherical circle pattern, we go the other way: Construct a planar pattern and project to the sphere.

Möbius Ambiguity and Normalization. Mesh combinatorics and intersection angles determine a planar circle pattern up to similarity. But they determine a spherical circle pattern only up to Möbius transformations (which preserve circles and angles). (Recall that the only conformal mappings of the plane $\mathbb{R}^2$ to itself are the fractional linear transformations. These are generated by the inversions in circles.) Suppose we applied a Möbius transformation to the spherical circle pattern before we project to the plane (choosing the same vertex as north pole). The result would be a planar pattern that differs only by a similarity transformation, i.e., a pattern that is translated, rotated, and uniformly scaled. Conversely, if we applied a similarity transformation to the planar pattern before projecting to the sphere, we would obtain a pattern on the sphere that differs by a Möbius transformation.

This Möbius ambiguity can (and should) be used to normalize the spherical parameterizations. For example, by applying a (unique) suitable Möbius transformation one can ensure that the barycenter of all vertices of the parameter mesh is at the center of the sphere [Springborn 2005]. This in effect achieves a notion of "uniform" distribution of vertices and it is the normalization used for the examples in this paper. An alternative normalization was proposed by Bern and Eppstein [2001]. They determined the unique Möbius transformation which maximizes the smallest circle radius. This may be useful for downstream numerical computing tasks.

Algorithm. The input is a triangle mesh of genus 0.

(1) Delete one (arbitrary) vertex from the mesh together with all incident faces. The remaining mesh is a topological disk.

(2) Generate a free boundary parameterization as described in Section 3

(3) Project stereographically to the sphere.

(4) Fill the hole in the mesh by adding a vertex at the north pole and completing the mesh.

(5) Normalize by applying a suitable Möbius transformation.

Example results of this method can be seen in Figure 7.

4.2 Mapping to the Disk

Our method to map a mesh to the disk should be seen as an extension of mapping to the sphere. The basic idea is to add an ideal (non-triangular) face to the mesh to turn it into a topological sphere. Then map to the sphere. The whole mesh is contained in the circle corresponding to the virtual face. Project stereographically to the plane to get a mesh with circular boundary. In practice, we can short cut this process and avoid the projection from the plane to the sphere and then back again.

Algorithm. The input is a triangle mesh that is topological disk.

(1) Pick a vertex at the boundary of the mesh and remove it together with all incident faces (see Figure 8, left).

(2) Map the resulting mesh to a convex polygon in the plane with prescribed boundary curvature, as described in Section 3. At boundary vertices that were not incident to any of the removed faces, fix the curvature to be 0. At

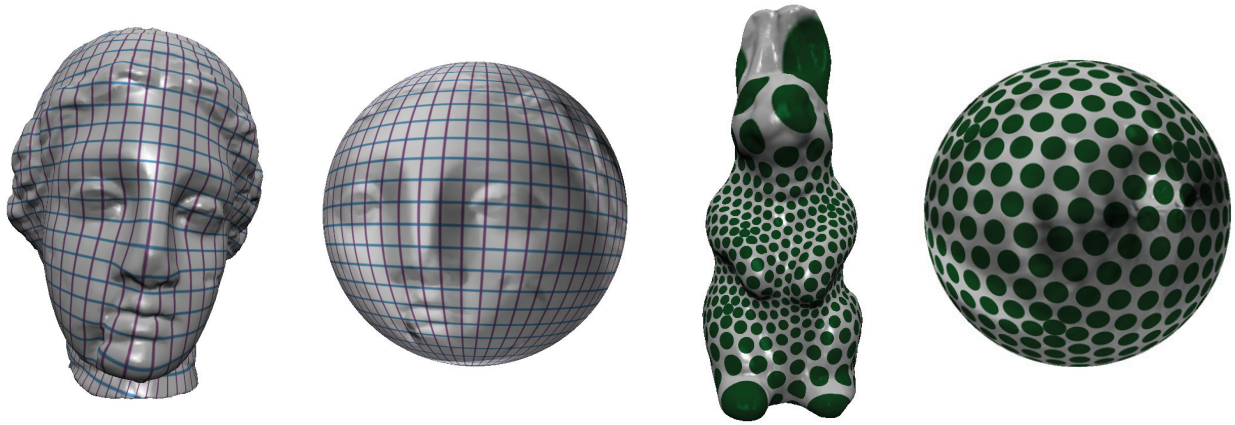

Fig. 7: *The Hygeia (50K triangles) and rabbit (26K triangles) models parameterized over the sphere with the parameterization visualized through textures. In the case of Hygeia a grid of lattitude/longitude lines. The rabbit is textured with points in an icosahedral pattern. Note the typical area distortion when mapping the head and ear regions to the sphere. No less, the roundness of the texture dots is well preserved through the mapping.*

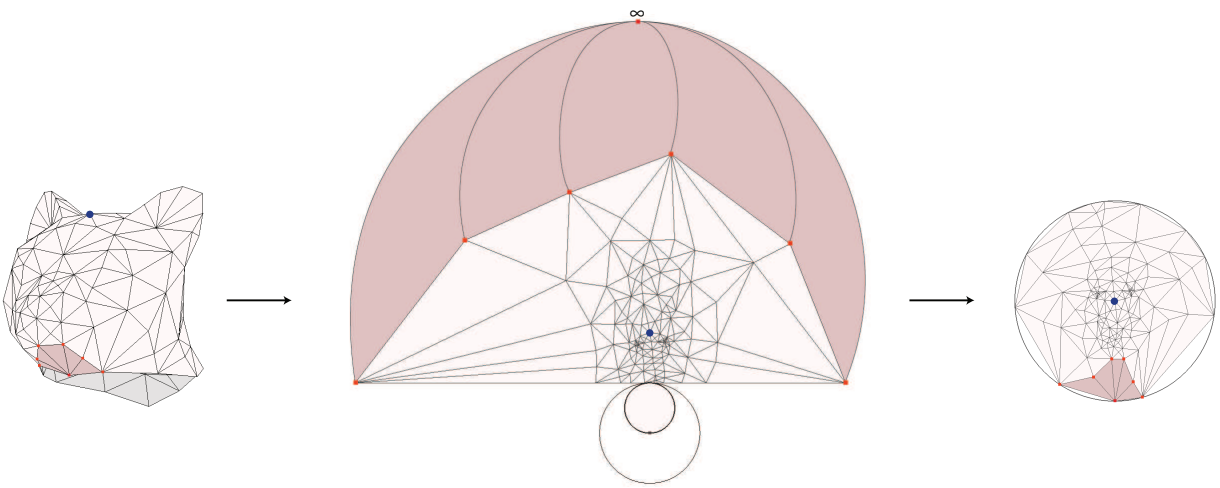

Fig. 8: *Mapping a mesh to the disk. (1) Remove a boundary vertex together with all adjacent faces, shown in dark pink (left). (2) Map the mesh to a convex polygon where all the original boundary edges lie on a straight line (middle). The removed vertex (which should be imagined at infinity) and its incident triangles are shown schematically (dark pink and ∞ symbol). (3) Invert in a circle and reinsert the missing vertex to complete the mesh (right). In this inversion we get to pick a vertex which will map to the center of the disk. In our example this vertex is the marked vertex between the ears of the cat.*

the others, bound the curvature to be positive. These conditions ensure that the original mesh boundary will be mapped to a straight line (Figure 8, middle).

(3) Reflect in a circle whose center lies on the other side of the straight line. This maps the boundary line to a boundary circle.

(4) Reinsert the removed vertex at the center of the reflection circle (it lies on the boundary circle) and complete the mesh (Figure 8, right).

Note that the circle reflection maps the other edges of the polygon to the circumcircles of the reinserted triangles. If we want to have a particular interior vertex at the center of the circular map, this can be achieved by an appropriate choice of reflection circle (or equivalently by a Möbius transformation of the disk to itself).

An alternative approach for the construction of mappings to the disk in the context of discrete harmonic mappings (cotan formula) was proposed by Jin and co-workers [2004] (other approaches, not necessarily focusing on discrete conformality, are possible as well; see [Friedel et al. 2005] and the references therein). They "welded" a mesh with its double at the boundary to create a sphere topology and then proceeded to map to the sphere. Enforcing symmetry in this mapping turns the original boundary into an equator. They had to effectively double the degrees of freedom while in our approach the number of degrees of freedom is (essentially) constant. As Jin observed, being able to map to a canonical region such as a disk one can effectively map from any region to any other through composition of one map to the disk with the inverse of another map to the disk (though the meshes in general do not match up and some interpolation in the disk is required).

5. INTRINSIC DELAUNAY TRIANGULATIONS

The parameterizations produced by our method always result in triangulations in the parameter plane which satisfy the local Delaunay criterion. This follows from the fact that $0 < \theta_e = \pi - \hat{\alpha}^l_{ij} - \hat{\alpha}^k_{ij} < \pi$, which is equivalent to saying that the edge e_{ij} satisfies the empty circumcircle property. If the input mesh is such that $\alpha^k_{ij} + \alpha^l_{ij} > \pi$ for some edge, the angle optimization forces $\hat{\alpha}^k_{ij}$ and $\hat{\alpha}^l_{ij}$ to be sufficiently different from α^k_{ij} (resp. α^l_{ij}) to get θ_e into the range $(0, \pi)$. As a consequence one can observe increased angle distortion in the vicinity of such edges (see the comparison in Figure 9).

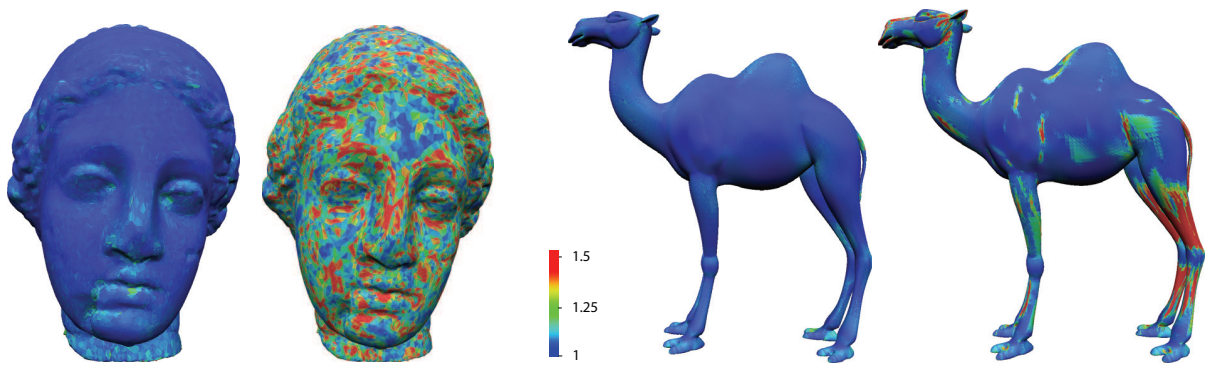

Fig. 9: *Two examples (Hygeia, genus zero; cut camel with free boundaries) comparing quasi-conformal distortion (see Section 7) for intrinsically Delaunay (left) and original triangulation (right). In both cases the quasi-conformal distortion is markedly improved. Note that the original input surface is* not *changed in this process.*

To address this issue one may use *intrinsic Delaunay* triangulations [Rivin 1994; Indermitte et al. 2001; Bobenko and Springborn 2005] of the input surface as the data given to the circle pattern method. To see this distinction we must first rigorously separate the surface with its intrinsic properties from the triangles we usually visualize when we think of an embedded triangle mesh. The embedded triangle mesh is defined by assigning point coordinates to each *abstract* vertex and then using linear interpolation of these coordinates over each *abstract* triangle. The basic idea now is that our problem—mapping a 3D mesh to the plane—depends *only* on the *intrinsic* geometry of the surface, not on the way it is embedded in space. In other words, what would a flatlander, living "inside" the surface, notice? Apparently the

surface is locally flat everywhere except at the vertices. In particular when the flatlander crosses a "crease" (as seen by us) in the surface—originally induced by an edge of the triangulation together with the coordinates associated to the vertices—the flatlander would not notice anything. There is one thing though they would notice: upon walking in a closed loop (of index 1) around a vertex they would notice that angles do not add up to 2π. In summary: the intrinsic geometry, i.e., what the flatlander notices, is completely summarized by giving *some* triangulation of the abstract vertices together with the lengths of straight paths between the vertices. One such example is the original abstract triangulation together with the original edge lengths.

The mathematical abstraction describing the intrinsic geometry of the mesh is a surface equipped with a flat metric with cone singularities at the vertices, a *piecewise flat (PF) surface*. An abstract triangulation with edge lengths (satisfying all triangle inequalities) describes a particular triangulation of a PF surface. It may coincide with the original triangulation which defined the embedding *or* it may be another triangulation so long as the corresponding flatlander distances between vertices are recorded correctly. In this sense, *intrinsic edge flips* change the *intrinsic triangulation* but not the underlying PF surface.

To intrinsically flip an (interior) edge, read off its length and the lengths of the four remaining edges of the two adjacent triangles. These lenghts determine the shape of a Euclidean quadrilateral with diagonal. Now perform a combinatorial edge flip in the abstract triangulation and label the flipped edge with the length of the other diagonal in this quadrilateral. A flatlander sees this new triangulation as simply another triangulation of the same surface since they cannot detect any change in their world. For us as outside observers we see different graphs drawn on the same fixed surface in space. Only the original graph followed apparent creases on the surface. After a flip the graph generally does not follow the creases anymore, but the surface itself is completely unchanged.

An ordinary planar triangulation is Delaunay if and only if each interior edge satisfies the local Delaunay criterion, and this can be checked using only the length of the given edge and the lengths of the remaining edges in the incident triangles. This leads to a natural definition of Delaunay triangulations of PF surfaces. It turns out that just as in the planar case, the Delaunay triangulation of a PF surface is generically unique [Bobenko and Springborn 2005], and can be constructed using the intrinsic edge flipping algorithm: search for an edge that violates the local Delaunay criterion, flip it, and continue until all edges are Delaunay. (It is actually not hard to show that this algorithm terminates not just in the standard but also in the PF setting and that local satisfaction of the Delaunay criterion is equivalent to global satisfaction [Rivin 1994; Indermitte et al. 2001; Bobenko and Springborn 2005].)

We emphasize that throughout this procedure the intrinsic geometry of the original embedded surface remains unchanged: a flatlander sees no change to what they perceive as the surface they live in; they merely see different triangulations and, upon termination of the flipping procedure, one that they will agree is Delaunay.

In this new triangulation all edges satisfy $0 < \pi - \alpha_{ij}^k - \alpha_{ij}^l < \pi$. This triangulation is now used as input to the circle pattern algorithm, resulting (after layout) in an assignment of parametric coordinates to each original vertex. The original triangulation of the input mesh can now be recovered by running all edge flips in reverse order (without modifying any of the parametric assignments to vertices). In theory this process can get stuck if one attempts to un-flip an edge which, taken together with its two incident triangles, is interior to a non-convex quad in the parameter plane. In all our examples this has occured for only a single edge (in the camel model). In that case the edge must be split and a new vertex is introduced whose position (in 3D) and parametric assignment are barycentrically interpolated from the surrounding quadrilateral.

It is conceivable (certainly with an adversary) that this case occurs many times over in a given mesh. The intuition behind the observation that this event occurs exceedingly rarely in practice is as follows. Consider an edge in the original 3D mesh and its two incident triangles. This edge will only be a candidate for intrinsic flipping if it is interior to an intrinsically convex quadrilateral formed by the two incident triangles (edges interior to a concave quadrilateral

always satisfy the Delaunay criterion). Given that the circle pattern method (nearly) preserves angles the mapped quadrilateral is also (with high probability) convex, allowing the later un-flip to proceed as desired.

The impact of using intrinsic Delaunay triangulations for the parameterization of meshes with many initial edges which are not intrinsically Delaunay is demonstrated in Figure 9. As the following table indicates both quasi-conformal distortion (see Section 7) and stretch distortion are significantly reduced when using the intrinsic Delaunay triangulation to compute the circle pattern.

Model	quasi-conformal no flipping	quasi-conformal with flipping	stretch no flipping	stretch with flipping	number of flips
Camel	1.1762	1.0266	1.8584	1.7597	17074
Hygeia	1.2157	1.0209	1.1333	1.0962	10246

6. CONE SINGULARITIES

Global parameterizations of triangle meshes often suffer from severe area distortion (see Figure 14). If one insists on flattening the entire model (in this case a topological disk) into the plane, triangles are severely compressed. Similar observations apply to examples such as mapping the hand (Figure 15) and the letter "a" (a topological torus, Figure 17). In particular long "appendages" are often troublesome. In case of higher genus surfaces this problem is further compounded as there is no obvious Euclidean domain over which to parameterize the surface. One could map to the hyperbolic plane in this case, however this would not address the area distortion issue.

To address both of these issues (area distortion and arbitrary topology) we advocate the use of cone singularities. Consider the simple example shown in Figure 10. If one parameterizes the cathead over a Euclidean domain with two cone singularities (the "coffee filter" domain) the distortion is greatly reduced over the case of mapping to a flat disk. (Figure 14 shows a similar example involving a more complicated mesh.) As the example of a coffee filter demonstrates it can be laid entirely flat if one introduces cuts (and one has great freedom to so). A benefit of the circle pattern method is that one can introduce cone singularities *without* the need to introduce cuts a priori. While cuts may be needed after the fact to lay out the parameterization in a single texture domain, the parameterizations themselves are independent of the later placement of cuts. In particular they are globally continuous. All of this can be achieved with only a small change to the algorithm described in Section 3.

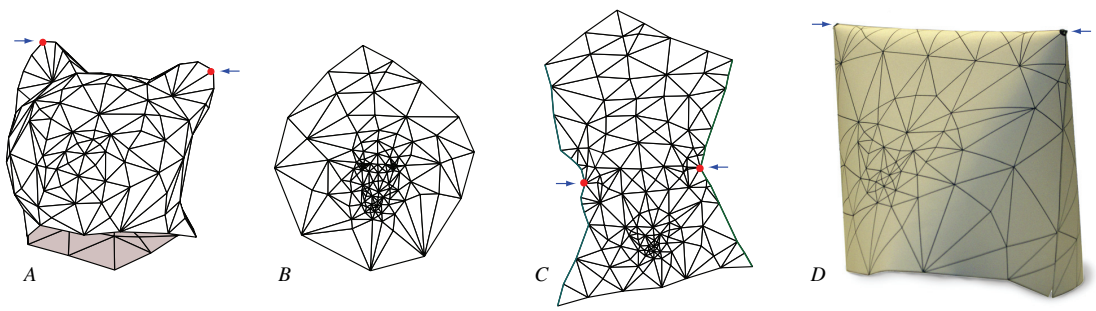

Fig. 10: *The simple cat head mesh (A) is parameterized without (B) and with two cone singularities (C). In the latter case, the parameter domain has zero Gaussian curvature everywhere except at the two marked cone points(blue arrows). The photo (D) shows it as an embedded surface in space. To flatten the parameter domain, it has to be cut. Notice however how the parts of the boundary marked with same colors (in C), corresponding to different sides of the same cut, fit together as expected.*

To summarize, the advantages of introducing cone singularities are

—area distortion can be greatly reduced (see Figure 15) while simultaneously ensuring that the parameterization is discrete conformal, except at the cone points themselves (see Figure 19);

—surfaces of arbitrary genus can be parameterized over *Euclidean domains* with cone singularities (see Figure 17);

—even though the parameterizations have to be cut to be flattened in the plane *a posteriori*, they are still *global* parameterizations in the sense that they are continuous everywhere (see Figures 16, 14, and 17).

6.1 Related Work

Design of parameterizations for complex shapes while controlling distortion through the judicious introduction of cuts *prior* or *during* parameterization was addressed by Gu and co-workers [Gu et al. 2002]. The geometry images produced in this way have complex identifications along cuts and require constrainted parameterizations to ensure continuity across cuts (and the boundary of the geometry image). In applications these identifications can be cumbersome and are, in any case, a source of additional distortion. A less severe setting—not attempting to find a single global parameterization—leads to chart based approaches (for examples see [Sander et al. 2003] and [Zhang et al. 2005] and the references therein). Since the general cutting problem is quite difficult [Erickson and Har-Peled 2004], good heuristics are crucial (for examples see [Katz and Tal 2003] and [Garland et al. 2001] and references therein). Alternatively one may focus on ways to "hide" the seams [Sheffer and Hart 2002] or pursue joint discovery of a parameterization and suitable patches [Sorkine et al. 2002]. Approaches based on identifying a topologically equivalent (to the original surface) base domain with good properties leads to methods such as in [Khodakovsky et al. 2003] (and references therein). Fundamental to all of these methods are the issues associated with ensuring continuity across patch boundaries while minimizing the number of patches, finding effective layouts for them in the texture domain, and keeping distortion to a tolerable level.

Focusing on discrete conformal mappings in particular, Gu and Yau [2003] proposed the use of discrete holomorphic forms for the construction of global parameterizations of surfaces with genus $g \geq 1$. This leads to discrete conformal parameterizations with cone singularities as well. However, as a consequence of the underlying mathematical theory, the number of cone points is determined by the genus of the surface: there are $2g-2$ of them. The cone angles are always 4π. Control over the placement of these singularities is only provided indirectly through a weighted optimization of the local conformal factor [Jin et al. 2004]. In their approach it is not possible to assign arbitrary points on the surface as cone points: neither their number nor their individual cone angles are free parameters. Gu and co-workers have to artificially increase the genus of the mesh if they want to exploit cone singularities to bring the area factor down (see their examples that involve placing topological punctures at the tip of long appendages).

A different approach to globally smooth parameterizations without requiring cuts was recently suggested by Ray and co-workers [Ray et al. 2005]. They use singularities of a vector field as, effectively, cone points and employ vector field optimization techniques to achieve favorable cone point placement.

Finally we mention Poly-Cube maps [Tarini et al. 2004]. They generalize the notion of a cube map to allow multiple cubes which more closely follow the original geometry and thus, when used as a texture domain, admit parameterizations with lower distortion. In fact, the parameterization produced by Poly-Cube maps is also Euclidean with cone singularities, it is however not discrete conformal and the cone angles are restricted to certain multiples of $\pi/2$.

On the theoretical side, Troyanov [1991] answered the following question. Given a compact Riemann surface, i.e., a compact two-dimensional manifold with a conformal structure, under what conditions on their placement and cone angles does there exist, in the conformal class of the Riemann surface, a flat metric with prescribed cone singularities. It turns out that only the Gauss-Bonnet formula has to be satisfied; then a cone metric exists and is unique up to scale.

(More generally, Troyanov investigates the conditions under which one may prescribe isolated cone singularities and Riemannian curvature, as a smooth function, on a Riemann surface. The proofs are not constructive.) His work can be seen as a mathematical justification for considering conformal maps with arbitrarily placed cone singularities.

6.2 Changes to the Algorithm

The algorithm takes as additional *input*

—a cone angle $\Theta_i > 0$ for each vertex v_i ($\Theta_i \neq 2\pi$ for a few designated cone vertices, otherwise $\Theta_i = 2\pi$).

The vertex sum condition of Section 3 is changed to

$$\forall v_k \in V_{\text{int}} : \sum_{t_{ijk} \ni v_k} \hat{\alpha}_{ij}^k = \Theta_i.$$

Introducing cone singularities does not compromise the underlying theory: If there is a coherent angle system, then a solution to the circle pattern problem exists (see Section 2). One easily sees the Gauss-Bonnet Equation

$$\sum_{\text{cone vertices } v_i} K_i + \sum_{\text{boundary vertices } v_i} \kappa_i = 2\pi\chi, \tag{11}$$

where $K_i = 2\pi - \Theta_i$ is the Gaussian curvature, is a *necessary* condition for the existence of a solution. An obvious further necessary condition is

$$\Theta_i < \pi \,\text{degree}(v_i),$$

since the angles in a triangle are less than π.

If these conditions are satisfied, it could nevertheless happen that a particular choice of cone singularities renders the quadratic programming problem of Section 3.1 infeasible. (See also Section 3.2.) However, we have not encountered this in practice.

6.3 Choosing the Cone Vertices and their Cone Angles

To choose the location of the cone vertices and their cone angles, it is sometimes helpful to draw a rough polygonal outline of a reasonable parameter domain. For example, to set the cone points for the hand model (Figures 15 and 16) we started with a polygonal sketch that looked qualitatively much like the outline of the final parameter domain (Figure 15; right). Now place cone singularities at vertices of the mesh which roughly correspond to the corners of the sketched polygonal outline. To determine approximate values for the cone angles, note that a cone vertex in the mesh corresponds to one or more corners of the polygonal outline. The sum of the interior angles at these corners is the cone angle. For example, on the hand model we placed two cone singularities with cone angle π at each finger tip. Each of these cone vertices corresponds to two corners with angle $\frac{\pi}{2}$ in the idealized polygonal outline of the parameter domain.

After cone singularities and angles have been set and a parameterization is generated, one can adjust the cone angles, for example, to lower the area distortion further: *De*creasing the cone angle will *enlarge* a neighborhood of the cone vertex in the parameter domain, *in*creasing the cone angle will *reduce* it. (This effect is analogous to the behavior of the complex map $z \mapsto z^\alpha$. For $0 < \alpha < 1$, it decreases radial angles at the origin, but increases length there. For $\alpha > 1$ the opposite is the case.) This is how we arrived at the particular values for the cone angles in the hand model example (see Section 7).

If one chooses to change the location of the cuts, for example to avoid self overlap, one need only redo the layout, (Stage 3 of the algorithm).

The Gauss-Bonnet formula (Equation 11) is a relation between the cone angles, the curvature of the boundary, and the Euler characteristic of the mesh. If the mesh has a boundary, free boundary angles will ensure that it is satisfied. However, when setting the cone singularities for a closed surface of genus g, the sum of the Gaussian curvatures has to be $2\pi(2-2g)$. For example, if the mesh is topologically a torus like the letter "a" in Figure 17, the total Gaussian curvature has to be zero.

7. RESULTS

We have applied our method to a variety of example meshes ranging from simple disk topologies, with a variety of boundary conditions, to higher genus surfaces using cone singularities.

Figure 11 shows meshes of disk topology with varying boundary treatments: disk (lion), polygonal (Max Planck), free (face). The mapping of the Max Planck head to a simple polygonal domain was achieved by first cutting the mesh and

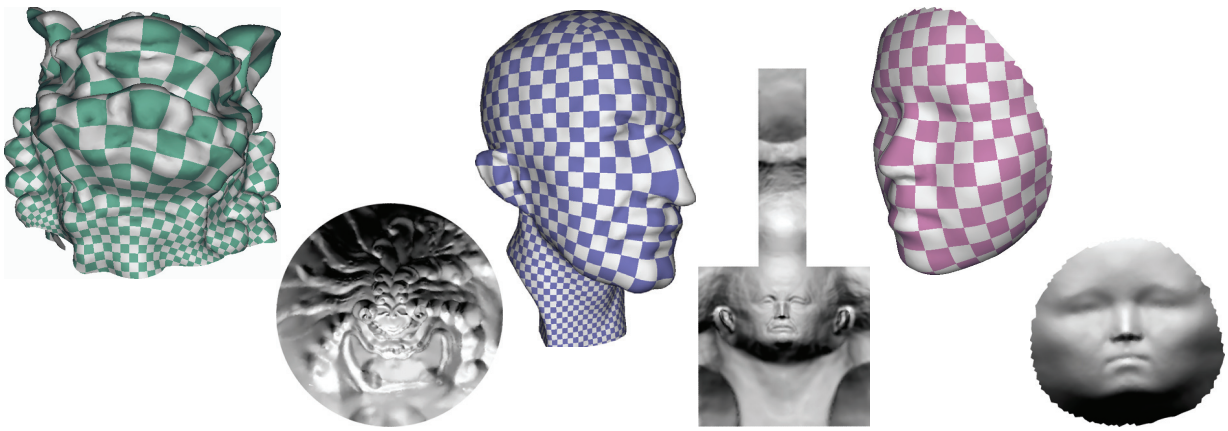

Fig. 11: *Examples of discrete conformal maps for surfaces with disk topology and varying boundary conditions. Lion: disk boundary; Max Planck: polygonal boundary; and Face mask: free boundaries. Next to each textured model is a visualization of the planar region over which the surface is parameterized (using shading from the 3D mesh).*

then prescribing the boundary curvature of the parameter domain: it was set to zero at all boundary vertices, except for eight designated corner vertices, where it was set to $\pm\pi/2$. Because the mesh was cut before parameterization there is no continuity across the cut boundaries. This is in contrast to global parameterizations with cone singularities which do not require a priori cuts (see Figure 14).

Figures 12 and 13 visualize the *quasi-conformal* distortion induced by our mappings, and in particular compare free and constrained boundaries. The quasi-conformal distortion is computed and visualized per triangle as $\frac{\Gamma}{\gamma}$, where Γ and γ are larger and smaller eigen values of the Jacobian as in Sander et al. [2001]. This quotient is at least 1 and 1 only if the mapping is conformal. In other words, the *quasi-conformal* distortion measures the deviation from the ideal of perfect conformality when this quotient would be 1 everywhere in the domain. The average quasi-conformal distortion of the mesh is computed as

$$E_{\text{qc}} = \sum_T \frac{\Gamma}{\gamma} A_{3D}^T / \sum_T A_{3D}^T.$$

Where A_{3D}^T is the area of a triangle in the original mesh. The maximum quasi-conformal distortion is the largest $\frac{\Gamma}{\gamma}$ in the mesh.

For the Max Planck head with free boundaries we get an average of 1.0157 (maximum quotient: 1.4). With polygonal boundary conditions the maximum distortion goes up to 4.05 while the average grows only slightly (1.0265). In particular we note that the high distortion is concentrated at the boundary and spreads very little inwards. The far more convoluted lion head shows low quasi-conformal distortion in most places with a few hot spots of high distortion. The free boundary gives an average of 1.045 (maximum 3.46), while the constrained (disk) boundary increases this to 1.050 (average) and 11.16 (maximum) respectively. In the following table we give runtimes for angle optimization and

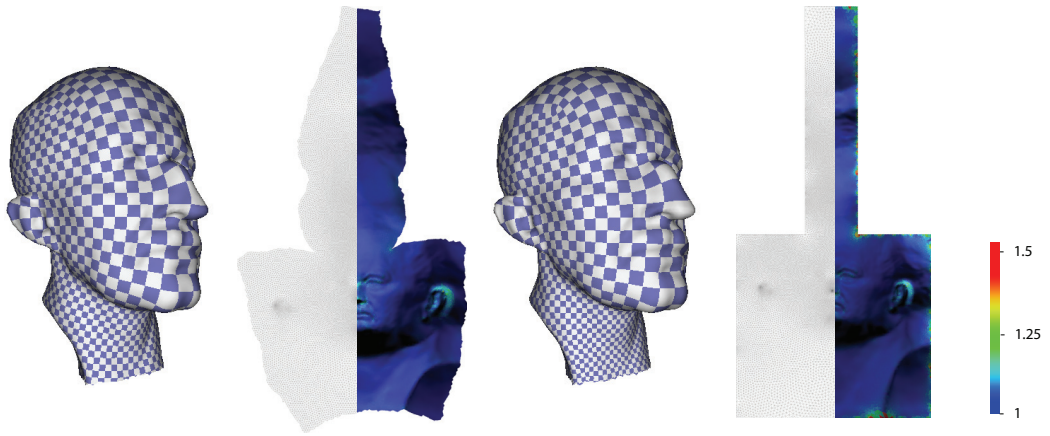

Fig. 12: *Comparison of two different mappings of the cut Max Planck model: free and polygonal boundary. Error plots indicate the quasiconformal distortion.*

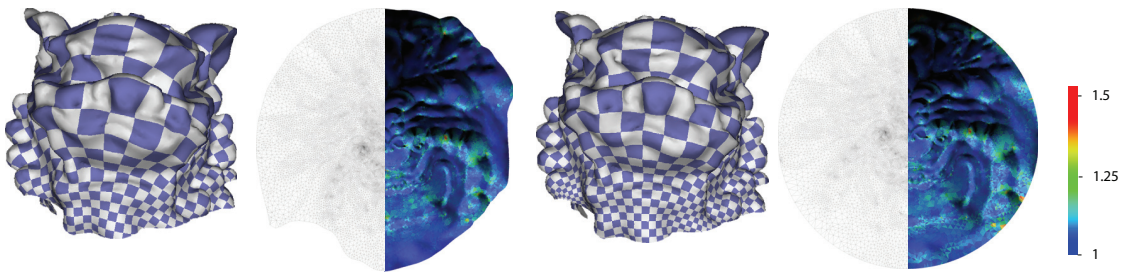

Fig. 13: *Comparison of two different mappings of the lion model: free and disk boundary. Error plots indicate the quasiconformal distortion.*

energy minimization. Layout and intrinsic edge-flipping (where applicable) timings are negligible at approximately 0.5s each. These timings were measured on a 3GHz Pentium IV running Windows XP. Aside from the average quasi-conformal distortion measure we also include the popular stretch distortion measure of Sander et al. [2001].

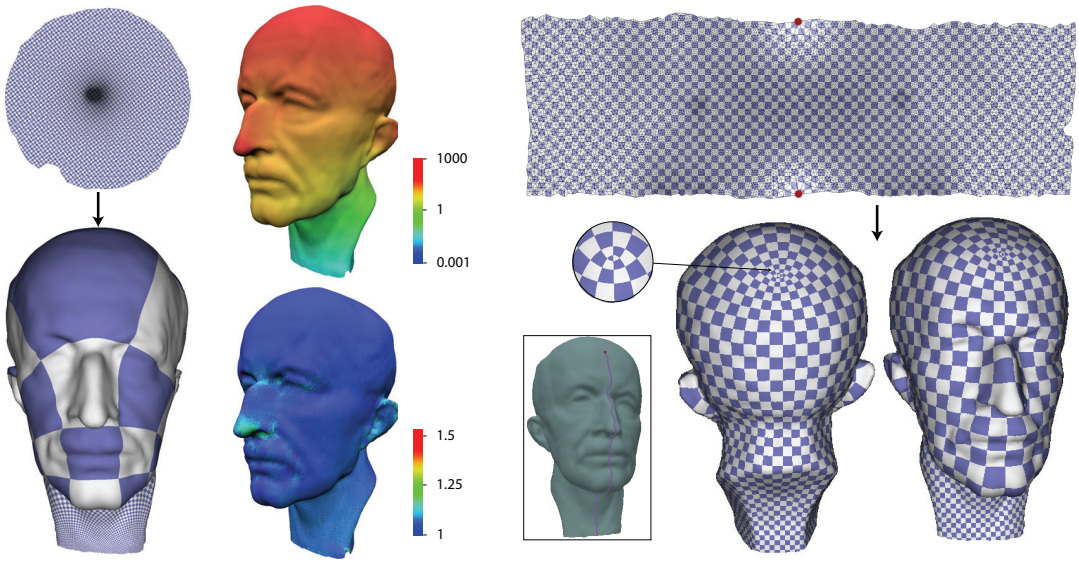

Fig. 14: *Left: The Max Planck model parameterized without cone singularities. The error plots show area distortion (top) and quasiconformal distortion (bottom). Right: The Max Planck model parameterized with two cone singularities: one on the forehead the other at the back of the scalp, both having cone angle π. See Figure 19 for error plots.*

Model	Triangles	Angle opt.	Energy min.	Total time	quasi-conf.	stretch
Face	12.6K	7s	2s	9s	1.013	1.063
Max Planck (free)	37.9K	35s	9s	45s	1.016	1.197
Max Planck (poly)	37.9K	35s	9s	45s	1.027	1.367
Lion Head (free)	39.6K	32s	6s	39s	1.045	2.043
Lion Head (disk)	39.6K	35s	6s	42s	1.050	2.018
Hygeia	50K	56s	10s	67s	1.021	1.096
Rabbit	26K	27s	6s	34s	1.043	5.123

Next we consider parameterizations with cone singularities. Figure 14 compares a parameterization of the Max Planck head (this time without any cuts) without and with cone singularities (left/right side). While the quasi-conformal distortion is low even when no cone singularities are used, the area factor, i.e., the ratio of corresponding triangle areas in the original mesh and the parametric domain, is quite large (2900). The area factor drops to 47 with two cone singularities. These are placed on the front and back of the scalp with a cone angle of π each. Two cuts, introduced *after* parameterization, make it possible to flatten the parameter domain: one runs from the front singularity down the middle of the face to the boundary. A similar cut is placed on the backside. To visualize how the parameter domain is mapped to the head, imagine drawing a line between the cone points marked in red. Fold along this line and glue the two parts of the top boundary and the two parts of the bottom boundary together to form a bag. Then put this bag over the head. Note that the texture is continuous across the cuts as expected.

A more complex example using cone singularities is shown in Figures 15 (right) and 16. In this case there are two cone singularities with Gaussian curvature π on the tip of each digit. Additionally two cone singularities are placed between digits. Their curvatures are, from little finger to thumb: $-\pi$, $-\pi$; -0.9π, -0.9π; $-\pi$, -0.9π; -0.3π, -1.15π. (The initial value of $-\pi$ for all of them was adjusted twice in the manner discussed in Section 6.3.) To flatten the parameter mesh it was cut along a sequence of edges traversing cone points and running from the tip of the little finger over

all digits to the boundary below the thumb (see Figure 16, left). In Figure 16 the parameterization is used to texture the hand to help visualize both conformal quality (circles remain circles; angles are preserved) and area distortion (equisized disks in the parameter domain map to disks of small size variation on the mesh). To produce such a texture image in the parametric layout, one has to ensure that the texture elements fit together across the cuts, which is easily enforced through appropriate identifications. The result is a seamless texture (see the inset view of the outer side of the ring finger).

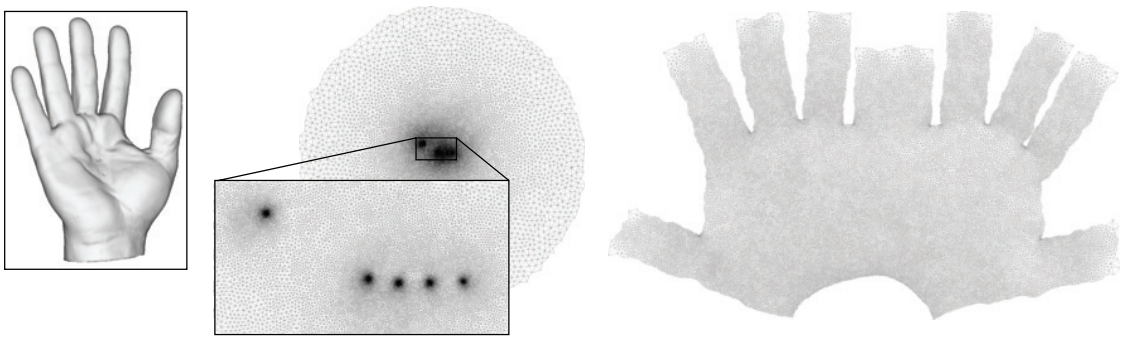

Fig. 15: *Model of a hand (left). A global discrete conformal parameterization leads to a parameter mesh with ridiculously large area distortion (middle). Placing cone singularities at the tip of each digit as well as between the digits reduces area distortion to a moderate level (right). (See also Figures 16 and 19.)*

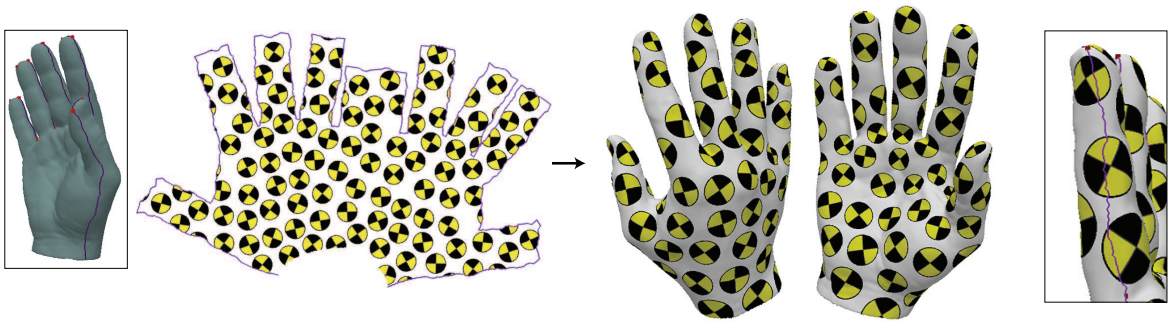

Fig. 16: *The same parameterization as shown in Figure 15 is used to put a texture on the hand model. The circles in the texture image are all the same size. They remain circular on the 3D mesh and do not greatly differ in size. This demonstrates the near conformality of the parameterization and the moderate area distortion. The cuts that were used to flatten the texture image (left) are not visible in the texturing (outer side of ring finger; right).*

Figure 17 shows a parameterization of a mesh that is topologically a torus. There are four cone singularities with cone angle π; two are at the end of the long appendage and two on the bottom of the "foot." Four cone points with angle 3π are located at the two places where the belly of the letter "a" is connected to the stem.

Our most complex example with cone singularities is the feline model with genus 2 (Figure 18). To choose the location of cone singularities we proceeded by first drawing a rough outline of the desired parameterization (see Figure 18, bottom left). This is best visualized by imagining suitable cuts, even though we do not perform cuts before parameterization. We assume that the cut starts on the horns, goes down the belly and legs and finishes at the tail.

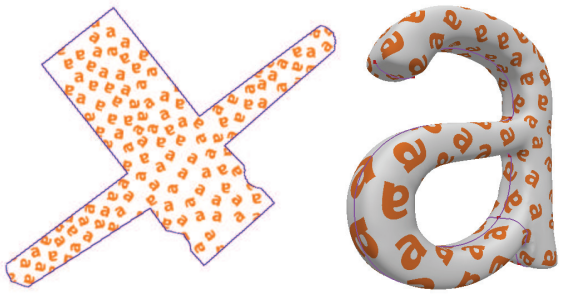

Fig. 17. Topologically, this letter "a" is a torus. Note in particular the appendage which would normally suffer from great area distortion if the parameter domain was a flat torus without cone singularities.

There are a few extra cuts on the tail to open up the topological handles. These are shown in blue (Figure 18, bottom left). From the outline we can compute initial values for the cone angles and approximate locations for the cone singularities. This outline suggests one cone singularity between the horns (A), two singularities on the horns (B_1, B_2), two on the belly (C, D), two on each leg $(K_i, L_i)_{i=1\ldots4}$ and four on the tail (E, F, G, H) are suitable. We increased the number of singularities by considering the geometry of the model, e.g., we added four extra cone vertices on each wing, added cone vertices to the legs and belly to distribute curvature among them, and added three extra cone singularities on the face to decrease the area distortion there (red arrows on Figure 18, bottom right). One can choose to put more cone singularities (for example on the ears), or less depending on how much area preservation is important. After setting initial cone singularities and their angles, we performed several iterations of parameterizing the surface, plotting area distortion, and adjusting the values of cone angles as described in Section 6.3. In regions with a high concentration of area distortion but no cone singularities, one can add a new singularity and adjust the values to satisfy the sum condition for the overall Gaussian curvatures. This way we added two new cone singularities at the base of the tail (red arrows on Figure 18, bottom right).

Runtimes and distortion for these examples are shown in the following table.

Model	Triangles	Angle opt.	Energy min.	Total time	quasi-conf.	stretch
Letter "a"	35.3K	40s	10s	51s	1.0201	1.1558
Max Planck	38.3K	50s	11s	62s	1.0180	1.2967
Hand	65.7K	76s	16s	93s	1.0159	1.0664
Feline	77.5K	104s	35s	140s	1.0345	1.1973

Figure 19 shows plots of the area factor and quasi-conformal distortion for the four examples with cone singularities (compare also with Figure 14 for Max Planck). We can see that these are fairly low and uniform except for hotspots in the immediate vicinity of the cone singularities (and some increased area distortion around the nose, chin, and ears of the Max Planck model as would be expected).

Finally, Figure 20 shows the results of an experiment to see how sensitive our method is to abruptly differing sampling rates. The geometry of the head is symmetric while the sampling rate doubles at the right/left symmetry line. Examining the flattened mesh (using free boundary conditions) we observe that the left/right symmetry is very well preserved (see also the resulting texture mapping on the original surface).

7.1 Comparison with ABF

The most closely related method to our approach is the work of Sheffer and de Sturler (ABF). While the underlying mathematics are entirely different their method also uses $\hat{\alpha}$-angle optimization. We compare here the quasi-conformal distortion and Sander et al. stretch metric for the cut camel, rocker, and horse models (as provided by Sheffer).

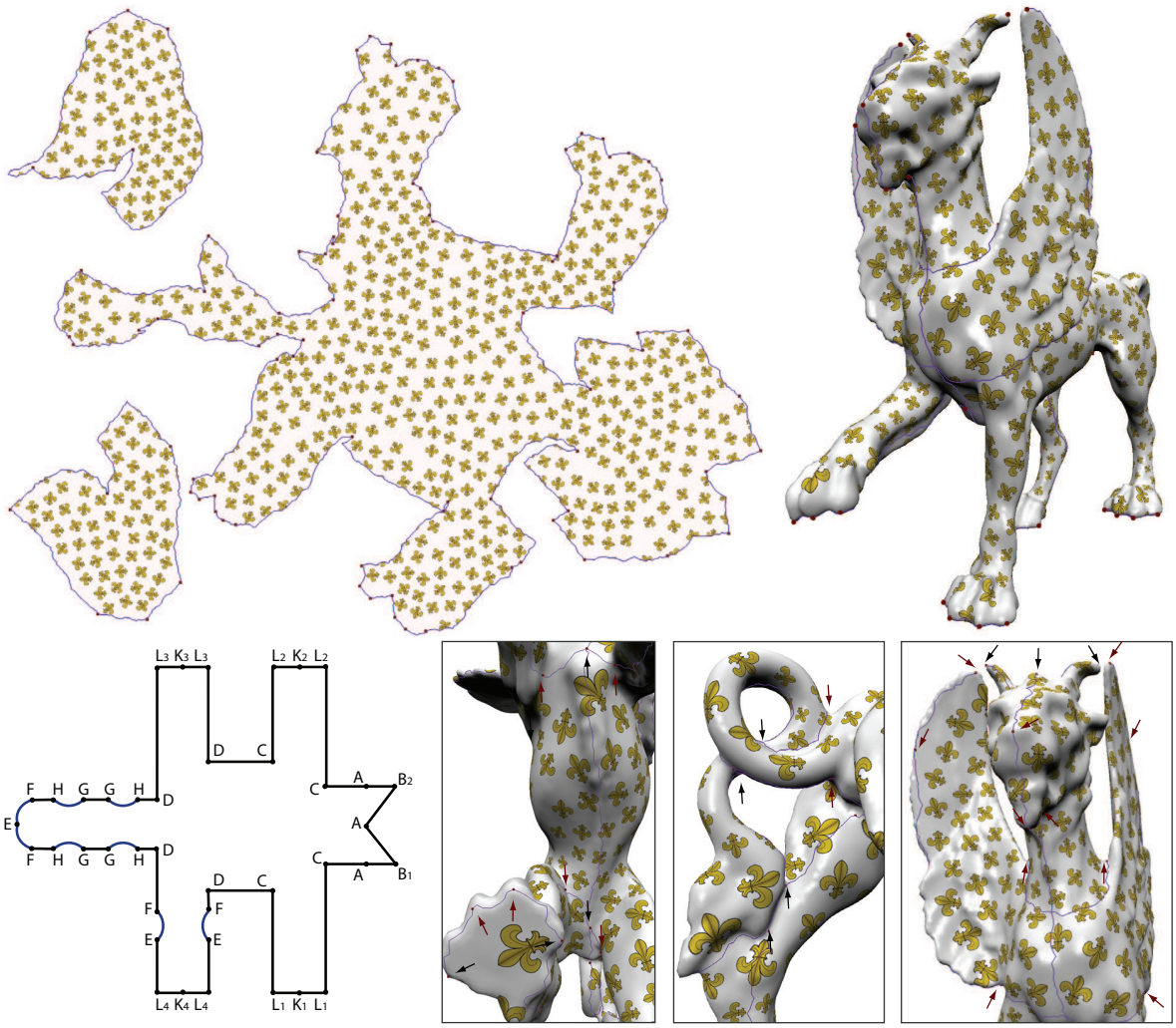

Fig. 18: *Feline model of genus 2 parameterized with cone singularities. Rough outline of the parameter domain (bottom left) helps to compute cone singularities. Fleur-de-Lis texture from the 2D parameter domain (top left) is used to texure map the feline model (right). Purple lines on the 3D model correspond to cuts in the parameter domain.*

Model	quasi-conformal ABF	quasi-conformal circle pattern	stretch ABF	stretch circle pattern
Camel	1.0249	1.0266	1.4864	1.7597
Rocker	1.0129	1.0133	1.0905	1.0777
Horse	1.0235	1.0256	1.3981	1.2713

The quasi-conformal distortion is essentially the same for both methods, while for the stretch metric the differences are more pronounced, though with no clear trend.

Fig. 19: *Error plots of area factor (left) and quasiconformal distortion (right).*

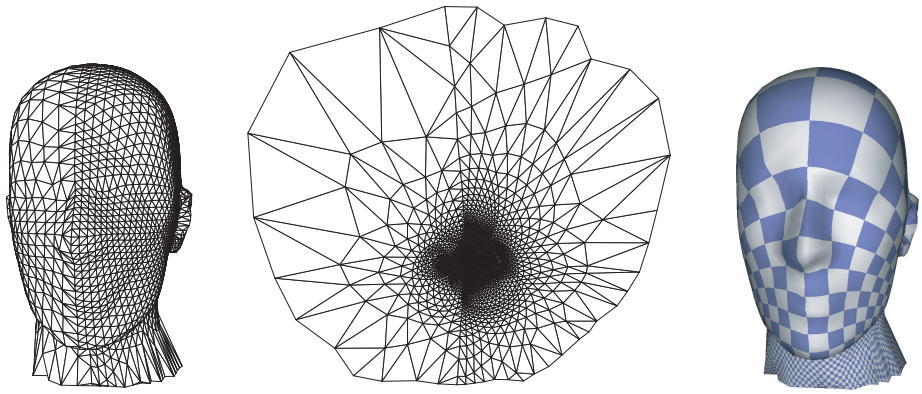

Fig. 20: *Our method is robust to varying sampling rates. Here a symmetric (left/right) geometry sampled at different rates. The parameterization (with free boundary) maintains the symmetry of the geometry.*

8. CONCLUSION

We have presented a new method to parameterize arbitrary topology surface meshes. It is based on the mathematical theory of *circle patterns*. In the case of bounded domains the shape of the boundary may be determined by free boundary conditions or by prescribing the curvature of the boundary. This provides a high degree of flexibility in controlling the boundary shape ranging from disks and simple polygonal outlines to more complex boundary arrangements. In-

troducing cone singularities we are able to mitigate the usually high area factor in conformal parameterizations. Cone singularities are also the key in our approach to dealing with globally continuous parameterizations of arbitrary topology meshes over Euclidean domains with cone singularities. The examples show that our method is efficient, provides good parameterizations even for large and complex meshes, and that the result is insensitive to the sampling rate of the surface. For input meshes with many edges which do not verify the Delaunay criterion we advocate the use of *intrinsic* Delaunay triangulations to further lower the quasi-conformal distortion in the mappings.

Our basic algorithm works in three stages: first, we solve a quadratic programming problem to obtain circumcircle intersection angles. These are the input for the second stage that consists in the unconstrained minimization of a convex energy. The output of this stage (the radii of the circumcircles) and the intersection angles determine the shape of all triangles which are then laid out. In every stage, the solution is unique and depends continuously on the input.

At the moment, we use general purpose solvers in the first two stages, and a very simple layout algorithm in the third. Efficiency could be improved by customizing and tuning the minimizers, and by switching to hierarchical methods [Sheffer et al. 2004]. Their sophisticated layout scheme—it minimizes the global layout error—can also be used without change as the third stage in our algorithm.

To avoid excess angle distortion, if the input mesh is far from Delaunay, we used a preprocessing step to pass to the intrinsic Delaunay triangulaton as input to the basic algorithm. This does not change the intrinsic mesh geometry but can greatly decrease the quasi-conformal error.

Diligent placement of cone singularities in the parameter domain can be used to greatly reduce the area distortion of our parameterization. The use of cone singularities also allows the parameterization of surface meshes with arbibtrary topology over Euclidean domains with cone singularities. Even though the parameter domain has to be cut to be flattened in the plane, the parameterization is continuous across the cuts. Again, this is achieved with only a minor modification (specification of cone angles) to the basic algorithm.

When we parameterize a mesh with prescribed boundary curvature, we have to set the curvature at each boundary vertex. At present, we do this manually. A user friendlier graphical interface is desirable. Since it is most likely that the performance of our method can be improved further, it is not unreasonable to envision an interactive interface that lets the user manipulate the parameter domain while the parameterization is incrementally recomputed.

The largest outstanding issue is a more complete study of the relationship between the placement of cone singularities and reduction in area distortion. A clear mathematical relationship between these could be the foundation of an automatic method for the placement of cone singularities and choice of cone angles. Especially for higher genus surfaces such an automatic procedure might be preferable over "hand" placement of cone singularities. (One possible approach for automatic placement was recently suggested by Ray et al. [2005].)

ACKNOWLEDGMENTS

This work was supported in part by NSF (DMS-0220905, DMS-0138458, ACI-0219979), DFG Research Center MATHEON "Mathematics for Key Technologies", DOE (W-7405-ENG-48/B341492), Center for the Mathematics of Information, Alias, and Pixar. Special thanks to Alexander Bobenko, Mathieu Desbrun, Ilja Friedel, Nathan Litke, and Cici Koenig.

REFERENCES

BERN, M. AND EPPSTEIN, D. 2001. Optimal Möbius Transformations for Information Visualization and Meshing. In *Algorithms and data structures (Providence, RI, 2001)*. Lecture Notes in Comput. Sci., vol. 2125. Springer-Verlag, Berlin, 14–25.

BOBENKO, A. I. AND SPRINGBORN, B. A. 2004. Variational Principles for Circle Patterns and Koebe's Theorem. *Transactions of the American Mathematical Society 356*, 659–689.

BOBENKO, A. I. AND SPRINGBORN, B. A. 2005. A Discrete Laplace-Beltrami Operator for Simplicial Surfaces. Preprint http://arxiv.org/abs/math.DG/0503219.

BOWERS, P. L. AND HURDAL, M. K. 2003. Planar Conformal Mappings of Piecewise Flat Surfaces. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Mathematics and Visualization. Springer-Verlag, Berlin, 3–34.

BRÄGGER, W. 1992. Kreispackungen und Triangulierungen. *L'Enseignement Mathématique 38*, 201–217.

COLIN DE VERDIÈRE, Y. 1991. Un principe variationnel pour les empilements de cercles. *Inventiones Mathematicae 104*, 655–669.

COLLINS, C. AND STEPHENSON, K. 2003. A Circle Packing Algorithm. *Computational Geometry: Theory and Applications 25*, 233–256.

DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic Parameterizations of Surface Meshes. *Computer Graphics Forum (Proceedings of Eurographics 2002) 21,* 3, 209–218.

DILLENCOURT, M. B. AND SMITH, W. D. 1996. Graph-theoretical conditions for inscribability and Delaunay realizability. *Discrete Mathematics 161*, 63–77.

ERICKSON, J. AND HAR-PELED, S. 2004. Optimally Cutting a Surface into a Disk. *Discrete & Computational Geometry 31,* 1, 37–59.

FRIEDEL, I., SCHRÖDER, P., AND DESBRUN, M. 2005. Unconstrained Spherical Parameterization. Submitted for Publication (http://multires.caltech.edu/pubs/SParam_JGT.pdf).

GARLAND, M., WILLMOTT, A., AND HECKBERT, P. S. 2001. Hierarchical Face Clustering on Polygonal Surfaces. In *Proceedings of the Symposium on Interactive 3D Graphics*. ACM Press, 49–58.

GRÜNBAUM, B. 2003. *Convex Polytopes*, Second ed. Graduate Texts in Mathematics, vol. 221. Springer-Verlag, New York.

GU, X., GORTLER, S. J., AND HOPPE, H. 2002. Geometry Images. *ACM Transactions on Graphics 21,* 3, 355–361.

GU, X. AND YAU, S.-T. 2003. Global Conformal Surface Parameterization. In *Proceedings of the Symposium on Geometry Processing*. Eurographics Association, 127–137.

HIRANI, A. N. 2003. Discrete Exterior Calculus. Ph.D. thesis, California Institute of Technology.

INDERMITTE, C., LIEBLING, T., TROYANOV, M., AND CLÉMENÇON, H. 2001. Voronoi Diagrams on Piecewise Flat Surfaces and an Application to Biological Growth. *Theoretical Computer Science 263,* 1–2, 268–274.

JIN, M., WANG, Y., YAU, S.-T., AND GU, X. 2004. Optimal Global Conformal Surface Parameterizations. In *Proceedings of IEEE Visualization*. IEEE Computer Society, 267–274.

KATZ, S. AND TAL, A. 2003. Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts. *ACM Transactions on Graphics 22,* 3, 954–961.

KHODAKOVSKY, A., LITKE, N., AND SCHRÖDER, P. 2003. Globally Smooth Parameterizations with Low Distortion. *ACM Transactions on Graphics 22,* 3, 350–357.

LEIBON, G. 2002. Characterizing the Delaunay Decompositions of Compact Hyperbolic Surfaces. *Geometry and Topology 6*, 361–391.

LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Transactions on Graphics 21,* 3, 362–371.

MERCAT, C. 2001. Discrete Riemann Surfaces and the Ising Model. *Communications in Mathematical Physics 218,* 1, 177–216.

MOSEK. 2005. Constrained quadratic minimization software. http://www.mosek.com/. Version 3.1r42.

PINKALL, U. AND POLTHIER, K. 1993. Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics 2,* 1, 15–36.

RAY, N., LI, W. C., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2005. Periodic Global Parameterizations. Submitted for publication (http://www.loria.fr/%7elevy/publications/papers/2005/PGP/pgp.pdf).

RIVIN, I. 1994. Euclidean Structures on Simplicial Surfaces and Hyperbolic Volume. *Annals of Mathematics 139,* 3, 553–580.

RODIN, B. AND SULLIVAN, D. 1987. The convergence of circle packings to the Riemann mapping. *Journal of Differential Geometry 26,* 2, 349–360.

SANDER, P. V., SNYDER, J., GORTLER, S. J., AND HOPPE, H. 2001. Texture Mapping Progressive Meshes. In *Proceedings of SIGGRAPH 2001*. 409–416.

SANDER, P. V., WOOD, Z. J., GORTLER, S. J., SNYDER, J., AND HOPPE, H. 2003. Multi-Chart Geometry Images. In *Proceedings of the Symposium on Geometry Processing*. Eurographics Association, 146–155.

SHEFFER, A. AND DE STURLER, E. 2000. Surface Parameterization for Meshing by Triangulation Flattening. In *Proceedings of the 9th International Meshing Roundtable*. Sandia National Labs, 161–172.

SHEFFER, A. AND HART, J. C. 2002. Seamster: Inconspicuous Low-Distortion Texture Seam Layout. In *Proceedings of IEEE Visualization*. IEEE Computer Society, 291–298.

SHEFFER, A., LÉVY, B., MOGILNITSKI, M., AND BOGOMYAKOV, A. 2004. ABF++: Fast and Robust Angle Based Flattening. *ACM Transactions on Graphics 24,* 2, 311–330.

SORKINE, O., COHEN-OR, D., GOLDENTHAL, R., AND LISCHINSKI, D. 2002. Bounded-Distortion Piecewise Mesh Parameterization. In *Proceedings of IEEE Visualization*. IEEE Computer Society, 355–362.

SPRINGBORN, B. A. 2005. A Unique Representation of Polyhedral Types. Centering via Möbius Transformations. *Mathematische Zeitschrift 249,* 3, 513–517.

TARINI, M., HORMANN, K., CIGNONI, P., AND MONTANI, C. 2004. PolyCube-Maps. *ACM Transactions on Graphics 23,* 3, 853–860.

THURSTON, W. P. 1980. The Geometry and Topology of Three-Manifolds. Available at http://www.msri.org/publications/books/gt3m/.

THURSTON, W. P. 1985. The Finite Riemann Mapping Theorem. Invited talk at the symposium on the occasion of the proof of the Bieberbach conjecture held at Purdue University, March 1985.

TROYANOV, M. 1991. Prescribing curvature on compact surfaces with conical singularities. *Transactions of the American Mathematical Society 324*, 793–821.

ZAYER, R., RÖSSL, C., AND SEIDEL, H.-P. 2003. Convex Boundary Angle Based Flattening. In *Proceedings of Vision, Modeling and Visualization*. 281–288.

ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2005. Feature-Based Surface Parameterization and Texture Mapping. *ACM Transactions on Graphics 24,* 1, 1–27.

ZIEGLER, G. M. 2004. Convex polytopes: Extremal constructions and f-vector shapes. To appear in IAS/Park City Mathematics Series, vol. 14. Preprint http://arxiv.org/abs/math/0411400.

Received Month Year; revised Month Year; accepted Month Year