Inter-Surface Maps via Constant-Curvature Metrics

PATRICK SCHMIDT, RWTH Aachen University
MARCEL CAMPEN, Osnabrück University
JANIS BORN, RWTH Aachen University
LEIF KOBBELT, RWTH Aachen University

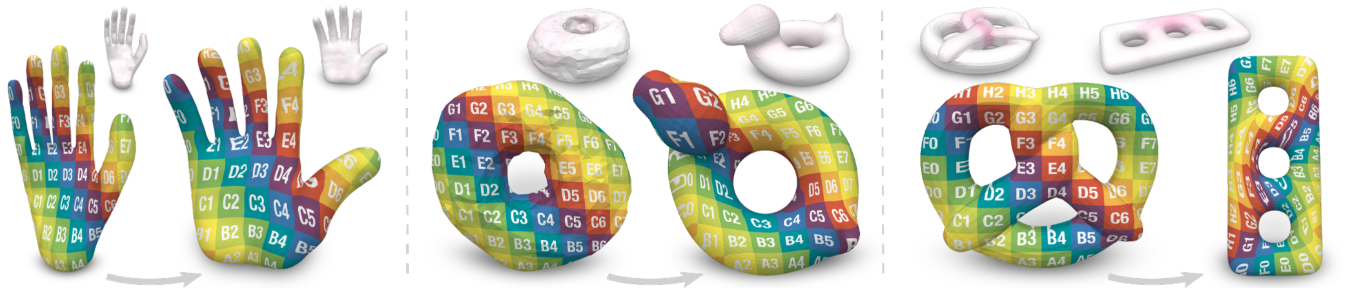


Fig. 1. Visualization of inter-surface maps for pairs of surfaces of varying genus, optimized for low distortion while guaranteeing bijectivity. We represent and optimize such maps flexibly and compactly via discrete constant-curvature metrics of spherical (genus 0), flat (genus 1), or hyperbolic (genus 2+) type.

We propose a novel approach to represent maps between two discrete surfaces of the same genus and to minimize intrinsic mapping distortion. Our maps are well-defined at every surface point and are guaranteed to be continuous bijections (surface homeomorphisms). As a key feature of our approach, only the images of vertices need to be represented explicitly, since the images of all other points (on edges or in faces) are properly defined implicitly. This definition is via unique geodesics in metrics of constant Gaussian curvature. Our method is built upon the fact that such metrics exist on surfaces of arbitrary topology, without the need for any cuts or cones (as asserted by the uniformization theorem). Depending on the surfaces' genus, these metrics exhibit one of the three classical geometries: Euclidean, spherical or hyperbolic. Our formulation handles constructions in all three geometries in a unified way. In addition, by considering not only the vertex images but also the discrete metric as degrees of freedom, our formulation enables us to simultaneously optimize the images of these vertices and images of all other points.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; **Mesh models**; **Mesh geometry models**; *Shape modeling*.

Additional Key Words and Phrases: cross-parametrization, surface parametrization, mesh overlay, bijection, discrete homeomorphism

ACM Reference Format:
Patrick Schmidt, Marcel Campen, Janis Born, and Leif Kobbelt. 2020. Inter-Surface Maps via Constant-Curvature Metrics. *ACM Trans. Graph.* 39, 4, Article 119 (July 2020), 15 pages. https://doi.org/10.1145/3386569.3392399

Author's addresses: Patrick Schmidt, Janis Born, Leif Kobbelt, Visual Computing Institute, RWTH Aachen University, Germany; Marcel Campen, Institute of Computer Science, Osnabrück University, Germany.

1 INTRODUCTION

Maps between surfaces have a variety of uses in Computer Graphics and Geometry Processing. Classical applications include the transfer of various types of information between surfaces, such as textures, geometric detail, deformations, or tessellations. The parametrization or registration of exemplars over a common base model is another application scenario. Such inter-surface maps are furthermore of increasing importance for advanced shape processing tasks, in the context of co-processing of shape collections, or the analysis of frame sequences of time-varying or animated shapes.

In these various fields, inter-surface maps are used as fundamental building blocks of complex methods. Being able to reliably compute, optimize, and provide such maps is therefore of significant practical interest. Properties of maps that commonly are relevant in such applications are bijectivity, continuity, and low distortion.

We present a novel approach to represent inter-surface maps with guaranteed bijectivity and continuity (i.e., surface homeomorphisms) and a method to optimize such maps for low distortion in a direct manner. Our approach is general in that it supports discrete surfaces (triangle meshes) of arbitrary genus.

1.1 Piecewise Maps

Piecewise-linear maps are a common choice for mapping from a discrete surface (triangle mesh) to the plane, e.g. for surface parametrization purposes. These maps are easily represented by the images of the vertices alone, i.e. a finite list of 2D point coordinates. The rest of the map is defined implicitly via barycentric interpolation: the image of any non-vertex surface point $\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + (1 - \alpha - \beta)\mathbf{c}$, lying in a triangle with vertices $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, is defined as

$$f(\mathbf{p}) = \alpha f(\mathbf{a}) + \beta f(\mathbf{b}) + (1 - \alpha - \beta) f(\mathbf{c}).$$

The fact that the map can be represented in this compact manner, and that important quantities like the map's Jacobian are linear in the vertex parameters, is a cornerstone for many map generation and map optimization algorithms.

When considering maps between two discrete surfaces (i.e., inter-surface maps) such a convenient representation is unavailable: the above representation, based on barycentric interpolation, exploits the fact that the Euclidean plane is a linear space. But this is not the case for a surface (a Riemannian manifold). In particular, there is no natural notion of interpolation between two or three points on an arbitrary surface. Methods concerned with formally well-defined inter-surface maps thus often operate with the plane as an intermediate domain—either globally [Aigerman et al. 2014; Litke et al. 2005; Schmidt et al. 2019], or locally [Kraevoy and Sheffer 2004; Schreiner et al. 2004]. This comes with disadvantages though, such as topological restrictions, limited distortion control, or the inability to efficiently optimize on a global scale.

A piecewise-linear map to the plane can equivalently be defined geometrically instead of algebraically (thereby not exploiting the vector space nature of the plane): edge images are geodesics in the plane, and for points in the interior of a triangle, the notion of barycentric coordinates can be viewed as a weighted Fréchet mean [Panozzo et al. 2013]. However, also this construction fails to directly extend to surfaces: neither geodesics nor Fréchet means are unique on a general Riemannian surface [Pennec 2006; Rustamov 2010] and therefore do not globally well-define the map in general.

1.2 Constant-Curvature Metrics

If, however, the Riemannian metric is flat, geodesics and the Fréchet mean *are* unique (and continuosly varying), hence the map is well-defined. Note that intermediate plane methods can be interpreted as equipping the surface with a flat metric.

This intrinsic metric view allows for further generalization: any constant curvature metric, including spherical and hyperbolic metrics [Bacák 2014], has (under mild assumptions) unique geodesics and Fréchet means. The latter is exploited in some mapping methods, such as [Aigerman et al. 2017; Aigerman and Lipman 2016]. The uniformization theorem states that for a Riemannian surface of any genus such a metric of constant Gaussian curvature K exists globally [Farkas and Kra 1992]. Further, the Gauss-Bonnet theorem $\int K \mathrm{d}A = 4\pi(1-g)$ relates the total curvature to the genus g of the closed surface. Thus, depending on its genus, any closed surface admits a

- $g = 0$: spherical metric (constant curvature $K > 0$),
- $g = 1$: flat metric (constant curvature $K = 0$),
- $g \geq 2$: hyperbolic metric (constant curvature $K < 0$).

A map from a triangle mesh to a surface equipped with a constant-curvature metric can again be expressed in a piecewise manner. As in the planar case, we *explicitly* define the images of the **vertices** on the target surface, while the images of **edges** are defined *implicitly* as the (unique) geodesics in the constant curvature metric. Once the images of edges are defined, the map can be extended rather easily to the interior of triangles, as we discuss later.

Note that the paths that edge images take on the target surface are fully determined by its metric. This metric, however, is not unique; there is an infinite space of discrete constant-curvature metrics for a given surface (e.g. every spherical embedding in the genus 0 case). We take both the vertex images *and* the metric of the target surface into account as degrees of freedom in our map representation and optimization.

1.3 Contribution

We provide a novel representation and an optimization algorithm for maps between two discrete closed and orientable surfaces.

- Both the map and its inverse are well-defined at every surface point and guaranteed to be continuous bijections. I.e., we produce a discrete surface homeomorphism (as opposed to a mere vertex map).
- Our representation supports surfaces of arbitrary genus in a unified formulation. No cut graphs, cone singularities, or transition functions are introduced.
- We optimize map quality using global second-order optimization techniques. Common intrinsic distortion measures (such as the symmetric Dirichlet energy) are supported and are evaluated in a direct surface-to-surface manner.
- By adjusting the underlying map-defining metric, the optimization controls not only the embedding of vertices on the target surface, but also of all other points. It is not required (but supported) to pin down any vertices during optimization.

1.4 Method Overview

In our method, a map from a source to a target triangle mesh is represented by a vertex-to-surface map in combination with a constant-curvature metric on the target surface. Given such a representation, our iterative map optimization algorithm repeatedly performs the following steps:

(1) Extract a common tessellation (mesh overlay) of both surfaces by computing geodesics in the constant-curvature metric.
(2) Compute geometric embeddings of the tessellation on both surfaces, defining the map in a piecewise-linear manner.
(3) Evaluate a distortion measure of the map and compute derivatives w.r.t. the vertex images on the target surface.
(4) Take a modified Newton step, i.e. compute per-vertex updates as tangent vectors and trace them along the target surface.
(5) Switch the roles of source and target, i.e., obtain the current vertex-to-surface map in reverse direction as well as the current metric of the source mesh. Both are uniquely defined.

Step (5) contains the key ingredient to optimizing both the vertices' images and the metric: Modifying the vertex-to-surface map in one direction implies a change of metric in the reverse direction (which is also directly driven by the map distortion measure). Alternating between the two representation directions exploits all available degrees of freedom, while allowing for a simple choice of variables in each step. It can be seen as a block descent approach.

After giving a precise map representation in Sec. 3 and providing the background required for the computation of geodesics in the different types of metrics in Sec. 4, we describe the algorithm to formally evaluate a constant-curvature-based map in Sec. 5. This covers steps (1) and (2) listed above. In Sec. 6 the description of the optimization procedure built on top of this framework, steps (3), (4), and (5), is elaborated, supplemented by implementation details.

2 RELATED WORK

Relaxed Surface Maps. The majority of previous approaches to the problem of inter-surface map representation, computation, and optimization build on relaxed variants of surface homeomorphisms. Those can be simpler to represent and easier to optimize, but commonly do not provide properties like continuity, bijectivity, or invertibility.

One class of methods focuses on discrete maps, defining the map for the vertices of a source surface only. Images can be restricted to target surface vertices (vertex-to-vertex maps), e.g. [Rodolà et al. 2015], or to arbitrary surface points (vertex-to-mesh maps), e.g. [Ezuz et al. 2019a,b; Yang et al. 2018]. Images of points on edges or in faces are not inherently defined. Extending the map to those in a consistent and robust manner is a major non-trivial challenge.

Further classes of methods build on maps which are not defined in a sharp manner, i.e., which do not map a point onto a point. Examples are functional maps [Ovsjanikov et al. 2012], soft maps [Solomon et al. 2012], or maps based on optimal transport principles [Mandad et al. 2017]. While there are ways to extract sharp maps out of such distributional maps [Ovsjanikov et al. 2012; Rodolà et al. 2015], the problem of general and robust conversion to an actual surface homeomorphism remains open.

A different perspective is taken by surface registration methods, which commonly approach the problem extrinsically. The idea is to deform one discrete surface onto the other [Bouaziz et al. 2013; Huang et al. 2008; Sharf et al. 2006; Tam et al. 2013; Wu et al. 2007; Yang et al. 2018; Zhang et al. 2006]. Similar as with the above vertex-defined maps, issues come into play when a precise continuous and bijective map is to be deduced from the piecewise linear, thus imprecise, deformation.

Strict Surface Homeomorphisms. More closely related to the problem we address herein are methods that are concerned with continuously defined maps between discrete surfaces. Many of these actually exploit the simplicity of defining homeomorphisms to the plane, cf. Sec. 1.1, and define inter-surface maps indirectly as a composition of two such maps [Aigerman and Lipman 2015; Aigerman et al. 2014, 2015; Kanai et al. 1997; Kim et al. 2011; Lipman and Funkhouser 2009; Litke et al. 2005; Schmidt et al. 2019; Tierny et al. 2011]. Due to this definition via the plane such methods are often restricted to surfaces of disk topology. Alternatively, other intermediate domains, in particular the sphere [Aigerman et al. 2017; Baden et al. 2018], can be made use of to define maps between pairs of surfaces of different, but again specific, genera. The introduction of cuts and transitions is required to enable some form of generic extension to arbitrary topology [Aigerman and Lipman 2016; Aigerman et al. 2014, 2015; Li et al. 2008] in this context.

A major challenge, in particular when employing such indirect definitions, and even more so when cuts are involved, is the optimization of maps between surfaces in an end-to-end manner. The relevance of this aspect has recently been discussed in detail in [Schmidt et al. 2019]. The method described in that paper enables distortion optimization with respect to the actual distortion measured between the surfaces (instead of between surface and intermediate domain), but again is restricted to surfaces of disk topology. A further limitation of several of these methods based on indirect definitions is their reliance on constrained corresponding points: a certain number of corresponding points may have to be known precisely in advance, because the proposed optimization procedures inherently preserve them.

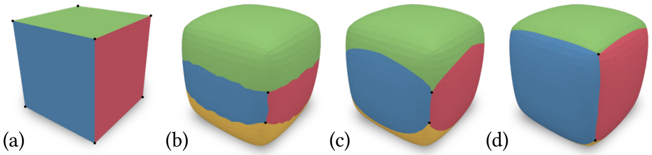


Fig. 2. a) Surface $\mathcal{A}$, a cube with distinctly colored faces. b) Image of the cube on $\mathcal{B}$ (a rounded cube) under a map defined by tuple (ϕ, h, ℓ). c) By changing ℓ (the constant-curvature metric associated with $\mathcal{B}$), the implicitly defined edge images of $\mathcal{A}$ on $\mathcal{B}$ change. d) By additionally changing ϕ, the explicitly represented vertex images are relocated.

Distortion-Minimizing Maps for Arbitrary Genus. When focusing on methods defining continuous inter-surface maps that support arbitrary genus *and* direct (i.e. end-to-end) distortion optimization, it comes down to essentially just one approach: the method described by Schreiner et al. [2004] and to some extent the similar concurrent method by Kraevoy et al. [2004]. While being very generic and flexible, the approach has one central disadvantage: inherently, the map can only be optimized in a local fashion. This is because it is defined in a piecewise manner, essentially per 1-ring, through small local parametrizations via the plane. These are mutually incompatible, hindering the definition of global optimization operators. In our approach, maps are represented in a globally consistent manner. We demonstrate that this enables the application of modern global optimization techniques and believe that it will open up further possibilities in the context of efficient processing and use of inter-surface maps.

Due to its use of constant curvature metrics, our method shares some aspects with certain (quasi-)conformal mapping methods based on uniformization, e.g., [Li et al. 2008]. Quite differently, though, they are tied to conformal distortion consideration. Our goal is to support generic distortion measures of practical relevance, which do not solely focus on angle-preservation but offer a flexible balance between length and angle distortion.

3 MAP REPRESENTATION

Consider two input triangle meshes, defining closed surfaces of the same genus: $\mathcal{A} = (\mathcal{V}_\mathcal{A}, \mathcal{E}_\mathcal{A}, \mathcal{T}_\mathcal{A})$ and $\mathcal{B} = (\mathcal{V}_\mathcal{B}, \mathcal{E}_\mathcal{B}, \mathcal{T}_\mathcal{B})$ with their respective sets of vertices, edges and triangles. In slight abuse of notation, we will also refer via $\mathcal{A}$ and $\mathcal{B}$ to the two surfaces as sets of points in $\mathbb{R}^3$.

Our core map representation is defined in a directional manner, i.e., from $\mathcal{A}$ to $\mathcal{B}$, or from $\mathcal{B}$ to $\mathcal{A}$. The resulting map, obtained from this representation, is symmetric, i.e. it can be evaluated in both directions and neither $\mathcal{A}$ nor $\mathcal{B}$ is labeled as source or target. When the map is bijective (as we always ensure in our method) we can easily switch between representations of the same map in both directions (cf. Sec. 6.3).

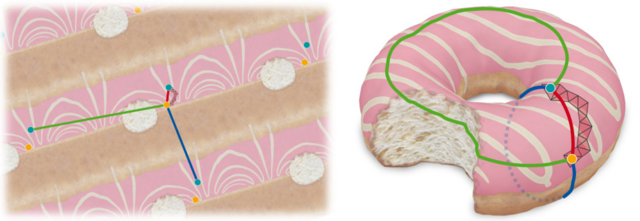


Fig. 3. A genus 1 object and its tiling (universal cover) of the plane as implied by a constant curvature metric ℓ. Two points on the surface are connected via geodesics (w.r.t. to ℓ) in different homotopy classes. Given an instance of one endpoint (orange), paths in different path homotopy classes correspond to straight line segments to different instances of the second point (turquoise) in the plane. The homotopy map h assigns to each edge a triangle strip (red) of the underlying mesh, representing the intended homotopy class.

3.1 Vertex-Based Map Representation

We represent a map from $\mathcal{A}$ to $\mathcal{B}$ as a tuple (ϕ, h, ℓ), with:

$$\phi : \mathcal{V}_\mathcal{A} \to \mathcal{T}_\mathcal{B} \times \mathbb{R}^2 \qquad \text{(vertex-to-surface map)}$$

$$h : \mathcal{E}_\mathcal{A} \to D_\mathcal{B} \qquad \text{(edge homotopy classes)}$$

$$\ell : \mathcal{E}_\mathcal{B} \to \mathbb{R}^{>0} \qquad \text{(discrete metric)}$$

The vertex-to-surface map ϕ assigns to each vertex of $\mathcal{A}$ a point in a triangle of $\mathcal{B}$, expressed in a barycentric coordinate system. The constant curvature metric ℓ on $\mathcal{B}$ can be expressed (due to the discrete setting) as a map assigning to each edge of $\mathcal{B}$ its length under this metric. The image of an edge (v_0, v_1) of $\mathcal{A}$ on $\mathcal{B}$ is defined as the geodesic path under the metric ℓ between the points $\phi(v_0)$ and $\phi(v_1)$. Given the constant curvature property of ℓ, this geodesic is unique—up to its path homotopy class, cf. Fig. 3. For disambiguation, the homotopy map h assigns to each edge of $\mathcal{A}$ the intended path homotopy class in $\mathcal{B}$. A path homotopy class is expressed as a representative dual edge path; the set $D_\mathcal{B}$ is the set of dual edge paths (triangle strips) on $\mathcal{B}$. Fig. 2 illustrates the combined effect of ϕ and ℓ. A path homotopy representative h is illustrated in Fig. 5.

3.2 Point-Based Map Definition

To obtain a complete surface-to-surface map $\Phi : \mathcal{A} \to \mathcal{B}$, defined for every surface point, from a tuple (ϕ, h, ℓ), it remains to fix a definition of edge and face parametrizations. That is, we have to settle two questions:

- Consider an edge of $\mathcal{A}$ and its image (a geodesic path) on $\mathcal{B}$. Which point on the edge corresponds to which point on the path?
- Given a triangle on $\mathcal{A}$ and the region its image covers on $\mathcal{B}$, which point in the triangle is mapped to which point in the region?

We need to choose these definitions such that the overall map is continuous, bijective, and such that the reverse map can be represented in the same form. The latter property is not just for convenience, it enables a lossless switching of the map's representation direction; we make use of this, for instance, in the map distortion optimization process (cf. Sec. 6).

In the following, we introduce the discrete geometry background our definitions build on. In Sec. 5, we show how to computationally construct the map Φ. This includes computing the required geodesics and defining and evaluating the edge and face parametrizations.

4 PRELIMINARIES: GEODESICS & MODELS

We equip a surface with a constant curvature metric such that each triangle is a spherical, flat, or hyperbolic triangle, depending on the curvature being positive, zero, or negative. This means that the triangle's edges are geodesics under the metric. Given merely the three edge lengths of a triangle under the metric, by means of the map ℓ, the intrinsic shape of the triangle is fully defined.

To compute geodesics under the respective metric we need to, at least locally, equip the mesh with coordinates realizing this metric. To this end we need to employ constructions in planar, spherical, or hyperbolic geometry. In the planar and spherical cases this can be done quite intuitively in Euclidean space: with respect to a metric with zero curvature a surface can (locally) be embedded isometrically (thus equipped with coordinates) in the two-dimensional plane, and with respect to a metric with constant positive curvature a surface can be embedded isometrically on a sphere. Hence, in both cases an isometric embedding in $\mathbb{R}^3$ is possible. For the hyperbolic plane, however, no such isometric embedding exists.

A variety of models of two-dimensional hyperbolic geometry are available (Poincaré disk model, Poincaré half-space model, Beltrami-Klein model, hemisphere model, hyperboloid model) [Stillwell 1996]. The latter has favorable properties for our purpose: like the above spherical and flat cases, the hyperboloid model makes use of $\mathbb{R}^3$ as ambient space—just equipped with a dot product that is non-Euclidean. Furthermore, again like in the spherical and flat cases, geodesics are simple plane intersections in this model (cf. Fig.4).

4.1 Unifying Geometry Models

These close similarities allow us to perform computations in a unified manner regardless of the type of constant curvature metric, i.e., regardless of surface genus. To this end, we define the following three models for the flat, spherical, and hyperbolic case, respectively. All three are 2-manifolds in $\mathbb{R}^3$ equipped with a distance function, into which a surface of constant curvature can be embedded isometrically (relative to their respective inner product).

Plane Model. The set of points $\mathbb{E}^2 = \{\mathbf{p} \in \mathbb{R}^3 | p_z = 1\}$ forms a plane. The Euclidean distance between two points $\mathbf{a}, \mathbf{b} \in \mathbb{E}^2$ is given by $\mathrm{d}_\mathbb{E} = \|\mathbf{a} - \mathbf{b}\|_2$.

Sphere Model. The set of points $\mathbb{S}^2 = \{\mathbf{p} \in \mathbb{R}^3 | \langle \mathbf{p}, \mathbf{p} \rangle = 1\}$, where $\langle \cdot, \cdot \rangle$ is the Euclidean dot product, forms the unit sphere. The spherical distance between two points $\mathbf{a}, \mathbf{b} \in \mathbb{S}^2$ is given by $\mathrm{d}_\mathbb{S} = \arccos(\langle \mathbf{a}, \mathbf{b} \rangle)$.

Hyperboloid Model. The three-dimensional Minkowski space is $\mathbb{R}^3$ equipped with the dot product $\langle \mathbf{a}, \mathbf{b} \rangle_M = a_x b_x + a_y b_y - a_z b_z$. The set of points $\mathbb{H}^2 = \{\mathbf{p} \in \mathbb{R}^3 | \langle \mathbf{p}, \mathbf{p} \rangle_M = -1, p_z > 0\}$ forms the positive sheet of a two-sheeted hyperboloid in $\mathbb{R}^3$ and can be used as a model of the hyperbolic plane. The hyperbolic distance between two points $\mathbf{a}, \mathbf{b} \in \mathbb{H}^2$ is given by $\mathrm{d}_\mathbb{H} = \operatorname{arccosh}(-\langle \mathbf{a}, \mathbf{b} \rangle_M)$ in this model [Wilson and Leimeister 2018].

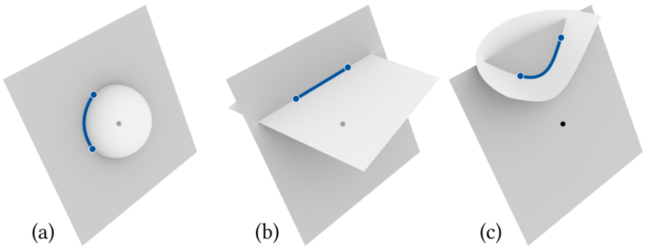


Fig. 4. Given two points (blue dots) on one of the three model manifolds $\mathbb{E}^2$, $\mathbb{S}^2$, or $\mathbb{H}^2$, a geodesic between them is easily computed as the intersection (blue curve) of a plane through these two points (blue dots) and the origin (black dot) with the model manifold.

To summarize these three models:

$$\mathbb{E}^2 = \{\mathbf{p} \in \mathbb{R}^3 \mid p_z = 1\} \qquad \mathrm{d}_{\mathbb{E}} = \|\mathbf{a} - \mathbf{b}\|_2$$

$$\mathbb{S}^2 = \{\mathbf{p} \in \mathbb{R}^3 \mid \langle \mathbf{p}, \mathbf{p} \rangle = 1\} \qquad \mathrm{d}_{\mathbb{S}} = \arccos(\langle \mathbf{a}, \mathbf{b} \rangle)$$

$$\mathbb{H}^2 = \{\mathbf{p} \in \mathbb{R}^3 \mid \langle \mathbf{p}, \mathbf{p} \rangle_M = -1, p_z > 0\} \qquad \mathrm{d}_{\mathbb{H}} = \operatorname{arccosh}(-\langle \mathbf{a}, \mathbf{b} \rangle_M)$$

The orientation of these manifolds is such that the origin $\mathbf{0} \in \mathbb{R}^3$ is on the inside. We write $\mathrm{d}(\cdot,\cdot)$ without subscript when referring to the respective distance function in the generic case.

In Sec. 5.1 we describe how a region of a triangle mesh endowed with a constant curvature metric can be embedded in the respective model manifold, locally equipping it with coordinates for the purpose of geometric constructions. A global embedding is never required.

4.2 Geometric Constructions

Computing Geodesics. A property beneficial for our purpose is the following: in each of the three models, geodesic paths can be obtained by intersecting a plane with the manifold. The geodesic path through points **a** and **b** on the manifold is the manifold's intersection curve with the plane through the origin **0**, **a**, and **b**, as illustrated in Fig. 4. Determining, as relevant in our discrete setting, which edges are crossed by a geodesic therefore amounts to simple point-plane orientation tests in 3D Euclidean space.

Geodesic Intersection Test. The question whether a point **p** on the manifold lies to the left or to the right of a geodesic through **a** and **b** is answered by checking the sign of the tetrahedral volume spanned by **0**, **a**, **b** and **p**. Testing whether two geodesic segments intersect each other reduces to four sign checks.

Projection to Manifold. We will make use of certain computations (interpolation, intersection) in ambient space ($\mathbb{R}^3$). This will result in points on chords (secant lines) of the model manifolds ($\mathbb{E}^2, \mathbb{S}^2, \mathbb{H}^2$). In the spherical and hyperbolic cases, these may lie off the manifold. In this context, we need a consistent definition, relating points $\mathbf{p} \neq \mathbf{0}$ on an ambient chord to points on the manifold. This is achieved by a central projection $\mathbf{p} \mapsto \tilde{\mathbf{p}}$, which amounts to

$$\tilde{\mathbf{p}} = \mathbf{p} \;\; (\mathbb{E}^2), \quad \tilde{\mathbf{p}} = \frac{\mathbf{p}}{\sqrt{\langle \mathbf{p}, \mathbf{p} \rangle}} \;\; (\mathbb{S}^2), \quad \tilde{\mathbf{p}} = \frac{\mathbf{p}}{\sqrt{-\langle \mathbf{p}, \mathbf{p} \rangle_M}} \;\; (\mathbb{H}^2). \qquad (1)$$

In all three cases, this projection bijectively maps an ambient chord to a geodesic path segment with the same endpoints on the manifold. Only in the spherical setting can a chord pass through a point ($\mathbf{p} = \mathbf{0}$) for which the projection is not well-defined; a case that we carefully exclude in our method. Ultimately, a direct consequence of this is that our maps are bijections along edges. In Appendix B we show that this bijectivity property extends also to the interior of triangles.

In the following we always denote point coordinates of a vertex p on the model manifold by $\tilde{\mathbf{p}}$, whereas a position in ambient space or on $\mathcal{A}$ or $\mathcal{B}$ is denoted $\mathbf{p}$.

5 MAP EVALUATION

Given a tuple (ϕ, h, ℓ), we wish to evaluate the map and its distortion at arbitrary points. To this end, as outlined in Sec. 3.2, we first need to complete the map definition to an inter-surface map Φ from $\mathcal{A}$ to $\mathcal{B}$: so far the tuple specifies (via ϕ) the source vertex positions on the target surface and (via the metric ℓ in conjunction with h) the paths of source edges on the target surface. It remains to define a homeomorphism per edge, relating (pointwise) a source edge with its path on the target surface, as well as a homeomorphism per triangle, relating it with the region enclosed by its three edges' paths' on the target surface.

For reasons of practicality and interoperability, we wish to define and compute this map Φ in a piecewise linear manner. Furthermore, we want to be able to efficiently evaluate a distortion measure $E(\Phi)$—and ultimately optimize the map with respect to it. We therefore carefully choose all constructions such that we are later able to compute derivatives of $E(\Phi)$ with respect to the map-defining data, in particular the vertex-to-surface map ϕ. Concretely, for a vertex of $\mathcal{A}$, its barycentric coordinates (α, β) in a triangle of $\mathcal{B}$ are considered as the main variables for optimization.

Overview. Our constructions can be carried out independently per triangle of mesh $\mathcal{A}$. In an implementation, efficiency can be gained by exploiting redundancies, but the per-triangle view allows for a simpler exposition in the following. For each $t_{\mathcal{A}} \in \mathcal{A}$, we perform the following steps:

- Discover the set of triangles on $\mathcal{B}$ that is (partially) covered by $\Phi(t_{\mathcal{A}})$ and embed this region on the model manifold using ℓ (Sec. 5.1).
- Embed the vertices of $t_{\mathcal{A}}$ (using ϕ) on top of this embedding on the model manifold. Determine the three geodesic edge paths between these vertices and their intersections with the underlying edges of $\mathcal{B}$. Use this information to collect the list of intersection pieces (overlay polygons) between $t_{\mathcal{A}}$ and triangles of $\mathcal{B}$ (Sec. 5.2).
- Define piecewise linear homeomorphisms per edge by equipping this combinatorial mesh overlay structure with metric information: For each edge-edge intersection, compute its parameters $\lambda, \mu \in (0, 1)$ on the surfaces of $\mathcal{A}$ and $\mathcal{B}$ respectively. For each vertex of $\mathcal{B}$ lying inside $t_{\mathcal{A}}$, compute its barycentric coordinates in $t_{\mathcal{A}}$ (Sec. 5.3).
- Obtain a piecewise linear map by optimally triangulating each overlay polygon, while taking care to maintain continuity of $E(\Phi)$ (Sec. 5.4).

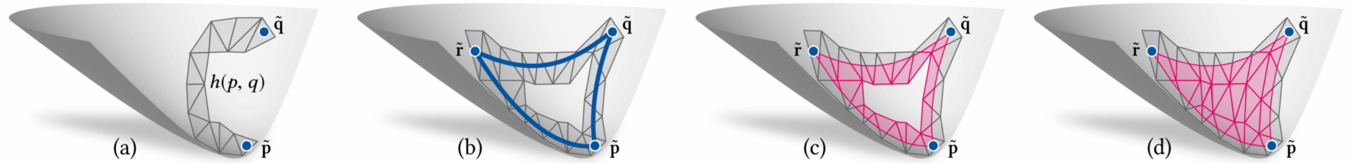


Fig. 5. Steps of our mesh overlay algorithm, which embeds a triangle (p, q, r) of mesh $\mathcal{A}$ on the model manifold (here the hyperboloid, sliced open to reveal the upper side) and intersects it with the underlying region of mesh $\mathcal{B}$. a) From an initial embedding of point $\tilde{\mathbf{p}}$, the position $\tilde{\mathbf{q}}$ is found by embedding the triangle strip $h(p, q)$; analogously for $\tilde{\mathbf{r}}$ via $h(p, r)$. b) By following the geodesic paths $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$, $(\tilde{\mathbf{q}}, \tilde{\mathbf{r}})$ and $(\tilde{\mathbf{r}}, \tilde{\mathbf{p}})$, their supporting triangle strips in mesh $\mathcal{B}$ are discovered. c) All polygonal pieces of these strips lying on the inside of (p, q, r) are collected as overlay polygons. d) Additionally, all triangles of $\mathcal{B}$ that are fully contained in (p, q, r) are discovered by a flood-fill.

5.1 From Metric to Coordinates

Our constructions rely on locally embedding small regions of the target mesh $\mathcal{B}$ on the model manifold according to a given constant-curvature metric ℓ.

Consider a flat, spherical or hyperbolic triangle $t_{\mathcal{B}} = (a, b, c)$ with edge lengths ℓ_{ab}, ℓ_{bc}, ℓ_{ca} under the metric ℓ. We embed the triangle on the model manifold by assigning coordinates $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}} \in \mathbb{E}^2, \mathbb{S}^2$ or $\mathbb{H}^2$ to its vertices that induce the prescribed edge lengths and preserve surface orientation. This embedding is unique up to the rigid transformations (orientation-preserving isometries) of the plane, the sphere, or the hyperbolic plane, respectively. As a constant-curvature metric has the same intrinsic properties at every point, our constructions will be invariant to the chosen absolute position and rotation of the embedding. For similar reasons, all local embeddings of neighboring or overlapping surface regions are mutually consistent.

To embed a first triangle, we start by positioning one of its vertices at $\tilde{\mathbf{a}} = (0, 0, 1)$ and its neighbor $\tilde{\mathbf{b}}$ with distance $\mathrm{d}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) = \ell_{ab}$ in an arbitrary direction (e.g. along the positive x-axis of the ambient space). The position of the third vertex $\mathbf{c}$ is then uniquely defined by $\mathrm{d}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}}) = \ell_{bc}$ and $\mathrm{d}(\tilde{\mathbf{c}}, \tilde{\mathbf{a}}) = \ell_{ca}$. Adjacent triangles can then be embedded incrementally by computing the position of one further vertex via its ℓ-distances to two existing vertices. We derive the necessary formulas in Appendix A.

5.2 Computing the Overlay Tessellation

Given the tuple (ϕ, h, ℓ), mapping from $\mathcal{A}$ to $\mathcal{B}$ and equipping $\mathcal{B}$ with the constant curvature metric ℓ, we compute the induced mutual tessellation of $\mathcal{A}$ and $\mathcal{B}$. It is defined by mapping the triangulation of $\mathcal{A}$ via ϕ to the surface of $\mathcal{B}$ and overlaying it with the triangulation of $\mathcal{B}$. This overlay is with respect to the metric ℓ, in which all edges are unique geodesics.

We start by computing its combinatorial structure, formed by a set of (abstract) overlay polygons $\rho \in \mathcal{P}$, each representing the overlap of one triangle of $\mathcal{A}$ with one triangle of $\mathcal{B}$. Fig. 7b illustrates all overlay polygons formed by one triangle of $\mathcal{A}$. In this figure, they are visualized geometrically on $\mathcal{B}$, but here we first of all determine them combinatorially. A vertex of ρ corresponds to either (1) a vertex of $\mathcal{A}$, (2) a vertex of $\mathcal{B}$, or (3) an intersection point between an edge of $\mathcal{A}$ and an edge of $\mathcal{B}$. Being the overlap of two (flat, spherical or hyperbolic) triangles, ρ is a convex 3-, 4-, 5-, or 6-gon in $\mathbb{E}^2, \mathbb{S}^2$ or $\mathbb{H}^2$ (cf. the planar case in [Schmidt et al. 2019]).

Overlay Algorithm. Per triangle $t_{\mathcal{A}} = (p, q, r)$ of $\mathcal{A}$, we perform the following steps (visualized in Fig. 5). We denote by $t_{\mathcal{B}}(p), t_{\mathcal{B}}(q)$ and $t_{\mathcal{B}}(r)$ the triangles of $\mathcal{B}$ that contain the image of the respective vertex of $\mathcal{A}$ (as given by ϕ).

(1) Embed $t_{\mathcal{A}}$ on the model manifold: As vertex images of $\mathcal{A}$ are defined relative to triangles of $\mathcal{B}$, we start by embedding the triangle $t_{\mathcal{B}}(p)$ on the manifold (as described in Sec. 5.1). To then embed triangles $t_{\mathcal{B}}(q)$ and $t_{\mathcal{B}}(r)$ relative to $t_{\mathcal{B}}(p)$, we rely on two triangle strips connecting each of them to $t_{\mathcal{B}}(p)$. These are provided by the homotopy map h. We incrementally add the triangle strips $h(pq)$ and $h(pr)$ to our local embedding. Now, with existing embeddings of $t_{\mathcal{B}}(p), t_{\mathcal{B}}(q)$ and $t_{\mathcal{B}}(r)$, the model coordinates $\tilde{\mathbf{p}}$, $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{r}}$ can be determined. This is done by evaluating the vertices' respective barycentric coordinates in ambient space (i.e., in chord triangles spanned by the vertices of $\mathcal{B}$ on the model manifold) and projecting the resulting points to the manifold (Eq. (1)). This is also illustrated in Fig. 6 left. We can now discard all triangle embeddings of $\mathcal{B}$ except for $t_{\mathcal{B}}(p)$.

(2) We successively compute the intersections of the geodesic $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$ with edges of $\mathcal{B}$. Starting at $t_{\mathcal{B}}(p)$, we check which of its edges intersects the geodesic $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$. We then embed the neighboring triangle of $\mathcal{B}$ (incident to the intersected edge) and continue the procedure until we reach the target $t_{\mathcal{B}}(q)$. We have now found the triangle strip of $\mathcal{B}$ supporting the geodesic $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$. Using the same procedure we also compute the strips supporting $(\tilde{\mathbf{q}}, \tilde{\mathbf{r}})$ and $(\tilde{\mathbf{r}}, \tilde{\mathbf{p}})$.

(3) We collect all overlay polygons formed by $t_{\mathcal{A}}$ and triangles of the three strips supporting $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$, $(\tilde{\mathbf{q}}, \tilde{\mathbf{r}})$ and $(\tilde{\mathbf{r}}, \tilde{\mathbf{p}})$. For each polygon vertex, we record its type ($\mathcal{A}$-vertex, $\mathcal{B}$-vertex, or $\mathcal{A}$-$\mathcal{B}$-edge-intersection) and the involved mesh elements, i.e. the triangle it lies in or the two intersecting edges.

(4) There can be triangles of $\mathcal{B}$ strictly lying within $t_{\mathcal{A}}$, i.e. not intersecting its edges. We discover these by a simple flood-fill on $\mathcal{B}$, seeded at the inner boundary of the triangle strips. All triangles of $\mathcal{B}$ thus discovered are added to the list of overlay polygons—they are 3-gons with all three vertices being $\mathcal{B}$-vertices.

The set $\mathcal{P}$ of all overlay polygons emitted in the above procedure form the facets of the mutual tessellation of $\mathcal{A}$ and $\mathcal{B}$.

5.3 Computing the Overlay Geometry

We equip the overlay tessellation with two geometric embeddings: one on the surface of $\mathcal{A}$ and one on the surface of $\mathcal{B}$. We define functions that assign to each overlay vertex v positions on both surfaces:

$$\mathbf{x}_{\mathcal{A}} : v \mapsto \mathcal{A} \subset \mathbb{R}^3,$$
$$\mathbf{x}_{\mathcal{B}} : v \mapsto \mathcal{B} \subset \mathbb{R}^3.$$

We carefully define the embedding functions $\mathbf{x}_{\mathcal{A}}$ and $\mathbf{x}_{\mathcal{B}}$ in dependence of the vertex-to-surface map ϕ. This will later allow us to compute derivatives with respect to the continuous degrees of freedom in ϕ, namely the barycentric coordinates α, β of each vertex of $\mathcal{A}$ in a triangle of $\mathcal{B}$.

Depending on the type of v ($\mathcal{A}$-vertex, $\mathcal{B}$-vertex, or $\mathcal{A}$-$\mathcal{B}$-edge-intersection) and the target of the embedding (surface $\mathcal{A}$, surface $\mathcal{B}$), different cases apply:

$\mathcal{A}$-Vertices. The embedding of an $\mathcal{A}$-vertex on $\mathcal{A}$ directly is its corresponding vertex position in $\mathcal{A}$ (analogously for $\mathcal{B}$-vertices on $\mathcal{B}$); it does not depend on ϕ and always stays fixed. Its embedding on $\mathcal{B}$ is defined directly by the vertex-to-surface map $\phi(v) = (t_{\mathcal{B}}(v), \alpha, \beta)$:

$$\mathbf{x}_{\mathcal{B}}(v) = \alpha\mathbf{a} + \beta\mathbf{b} + (1 - \alpha - \beta)\mathbf{c} \tag{2}$$

with fixed vertex positions $\mathbf{a}, \mathbf{b}, \mathbf{c}$ of the triangle $t_{\mathcal{B}}(v)$ on $\mathcal{B}$.

$\mathcal{A}$-$\mathcal{B}$-Edge-Intersections. Let v be the intersection vertex between edge $(\mathbf{p}, \mathbf{q}) \in \mathcal{A}$ and edge $(\mathbf{a}, \mathbf{b}) \in \mathcal{B}$. We define its embedding on the respective surface via parameters $\lambda_{\mathcal{A}}$ and $\lambda_{\mathcal{B}}$:

$$\begin{aligned}\mathbf{x}_{\mathcal{A}}(v) &= (1 - \lambda_{\mathcal{A}})\mathbf{p} + \lambda_{\mathcal{A}}\mathbf{q},\\ \mathbf{x}_{\mathcal{B}}(v) &= (1 - \lambda_{\mathcal{B}})\mathbf{a} + \lambda_{\mathcal{B}}\mathbf{b},\end{aligned} \tag{3}$$

To compute these parameters, we make use of a local embedding containing $\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}} \in \mathbb{E}^2, \mathbb{S}^2$ or $\mathbb{H}^2$, as computed during mesh overlay. Note that, while $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ are independent of ϕ, the positions $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ depend on ϕ via barycentric interpolation in ambient space and projection via Eq. (1) to the manifold (illustrated in Fig. 6).

Let $\tilde{\mathbf{v}}$ be the intersection point of the geodesics $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$ and $(\tilde{\mathbf{a}}, \tilde{\mathbf{b}})$ on the model manifold. Conceptually, we parametrize the curve $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$ via linear interpolation in ambient space and subsequent projection to the manifold. Under this parametrization, the parameter of the intersection point $\tilde{\mathbf{v}}$ along the geodesic $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$ has a simple expression: It can be computed by intersecting the straight line segment between $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ in $\mathbb{R}^3$ with the plane through $\mathbf{0}, \tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$:

$$(1 - \lambda_{\mathcal{A}})\tilde{\mathbf{p}} + \lambda_{\mathcal{A}}\tilde{\mathbf{q}} = s\tilde{\mathbf{a}} + t\tilde{\mathbf{b}}.$$

with $s, t \in \mathbb{R}$. We employ the symmetric construction to express $\lambda_{\mathcal{B}}$ by intersecting the line segment between $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ with the plane through $\mathbf{0}, \tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$. Applying Cramer's rule to both systems yields

$$\lambda_{\mathcal{A}} = \frac{\det[\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{p}}]}{\det[\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{p}} - \tilde{\mathbf{q}}]} \qquad \lambda_{\mathcal{B}} = \frac{\det[\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{a}}]}{\det[\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{a}} - \tilde{\mathbf{b}}]}. \tag{4}$$

$\mathcal{B}$-Vertices. Similarly, we express the embedding of a point on $\mathcal{A}$ via barycentric parameters

$$\mathbf{x}_{\mathcal{A}}(v) = \alpha\mathbf{p} + \beta\mathbf{q} + (1 - \alpha - \beta)\mathbf{r} \tag{5}$$

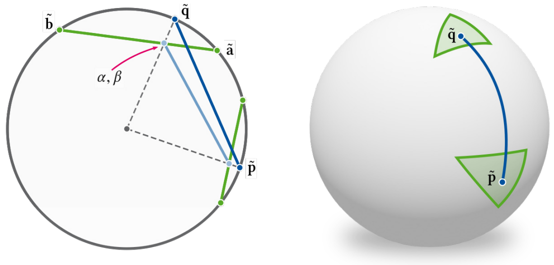


Fig. 6. Left: 2D Illustration of the edge and facet parametrizations, corresponding to the 3D setting on the right. Given a point $\mathbf{q}$ defined via barycentric coordinates α, β on a chord triangle (here: green chord), its correspondence $\tilde{\mathbf{q}}$ on the model manifold is defined via central projection. Inversely, the barycentric coordinates corresponding to a point $\tilde{\mathbf{q}}$ are obtained by projection onto the chord triangle and determining this projected point's barycentric coordinates in the planar chord triangle, cf. Eq. (7).

defined by parametrizing the corresponding flat, spherical, or hyperbolic triangle on the model manifold via barycentric interpolation in ambient space followed by central projection, cf. Fig. 6.

This means, given the position $\tilde{\mathbf{a}}$ of a vertex of $\mathcal{B}$ inside the triangle $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{r}})$ on the model manifold, we can again obtain its parameters (now α, β) via a simple construction in ambient space. We intersect the line through $\mathbf{0}$ and $\tilde{\mathbf{a}}$ with the plane spanned by $\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{r}}$:

$$s\tilde{\mathbf{a}} = \alpha\tilde{\mathbf{p}} + \beta\tilde{\mathbf{q}} + (1 - \alpha - \beta)\tilde{\mathbf{r}} \tag{6}$$

Again, Cramer's rule gives the closed-form solutions

$$\alpha = \frac{\det[\tilde{\mathbf{a}}, \tilde{\mathbf{r}}, \tilde{\mathbf{r}} - \tilde{\mathbf{q}}]}{\det[\tilde{\mathbf{a}}, \tilde{\mathbf{r}} - \tilde{\mathbf{p}}, \tilde{\mathbf{r}} - \tilde{\mathbf{q}}]} \qquad \beta = \frac{\det[\tilde{\mathbf{a}}, \tilde{\mathbf{r}} - \tilde{\mathbf{p}}, \tilde{\mathbf{r}}]}{\det[\tilde{\mathbf{a}}, \tilde{\mathbf{r}} - \tilde{\mathbf{p}}, \tilde{\mathbf{r}} - \tilde{\mathbf{q}}]}. \tag{7}$$

Via $\mathbf{x}_{\mathcal{A}}$ and $\mathbf{x}_{\mathcal{B}}$ all vertices of both $\mathcal{A}$ and $\mathcal{B}$ as well as certain points on their edges (given by linear parameters λ) now have a geometric embedding, an image, on the respective other surface. The mapping of edges can be completed in a simple piecewise linear manner, where the pieces are delineated by the intersection points (i.e., the λ-parameters associated with the edge). This is possible because each such piece falls into a single triangle in the other surface, such that the corresponding part of the edge's image (a geodesic path) is a straight line segment. In this way, the map is already fully defined on edges and vertices. What remains is to extend the map into the faces.

5.4 Piecewise-Linear Map Completion

Given embeddings $\mathbf{x}_{\mathcal{A}}, \mathbf{x}_{\mathcal{B}}$ of the overlay tessellation on both surfaces we now complete the surface-to-surface map Φ. For an overlay polygon $\rho \in \mathcal{P}$, consider its two (planar) embeddings (on $\mathcal{A}$ and on $\mathcal{B}$). These two versions of the polygon are not generally related by an affine transformation—rather, in both the spherical and the hyperbolic case, by a perspective transformation (homography). This means, in contrast to the edges, the map is not directly extendable into the faces as a linear map. A piecewise linear map, however, is of high importance, not only for downstream applications, but also to allow simple expressions for common distortion measures.

We therefore triangulate each overlay polygon and extend the map linearly per triangle, i.e., piecewise-linearly per overlay polygon. From all possible triangulations (1, 2, 5, or 14, depending on

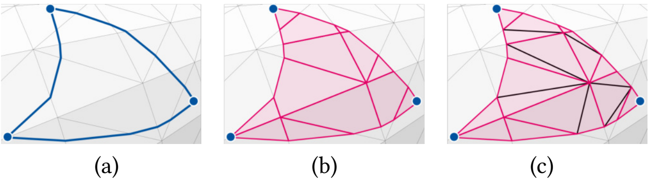


Fig. 7. a) Images of $\mathcal{A}$-edges (blue) on mesh $\mathcal{B}$ (gray), computed as geodesics under metric ℓ. Note that, conveniently, they are straight in each triangle by our definition of the metric inside the triangles. b) Overlay polygons (pink) formed by one $\mathcal{A}$-triangle with the underlying $\mathcal{B}$-triangles. c) Triangulation of the involved 4-gons and 5-gons (additional black edges)

polygon valence) we choose the optimal one, minimizing the objective function $E(\Phi)$ (defined in Sec. 6). The optimal triangulation is easily computed per polygon using dynamic programming. We will refer to the resulting set of triangles as $\mathcal{T}$. Choosing the optimal triangulation instead of an arbitrary one allows us to maintain continuity of the objective (as we will discuss in Sec. 6.4). In the genus 1 (flat) case, the triangulation step could be omitted as the two embeddings of each polygon are related by a linear transformation in this case (cf. [Schmidt et al. 2019]). The triangulation of overlay polygons is illustrated in Fig. 7.

With a mutual triangulation and its geometric embedding on mesh $\mathcal{A}$ and $\mathcal{B}$, the surface-to-surface map $\Phi : \mathcal{A} \to \mathcal{B}$ is now fully defined. It is a standard piecewise linear map, linear per triangle of the mutual triangulation.

6 OPTIMIZATION

Given a bijective inter-surface map in the representation described above, we now strive to optimize it with respect to a concrete distortion measure. During this optimization, we wish to strictly preserve bijectivity.

Due to the piecewise-linear definition of our map Φ via a mutual overlay tessellation, we are in a position to apply ideas recently described by Schmidt et al. [2019] for optimization. In particular, we can perform global second-order optimization of the map, for instance with respect to the symmetric Dirichlet energy, which has proven to be a good choice for a variety of applications.

A number of modifications and extensions is necessary though. While this previous method represents maps in a global 2D coordinate system (and thereby is restricted to disk topology), our maps are defined using local intrinsic coordinates. We describe the required modifications in the following.

6.1 Objective

With Φ defined in a piecewise-linear manner, a wide range of distortion measures based on the map Jacobian $\mathbf{J} \in \mathbb{R}^{2\times2}$ is available. Here we choose the symmetric Dirichlet energy

$$E(\Phi) = \int_{\mathcal{B}} \|\mathbf{J}\|_{\mathrm{F}}^2 \, \mathrm{d}A_{\mathcal{B}} + \int_{\mathcal{A}} \|\mathbf{J}^{-1}\|_{\mathrm{F}}^2 \, \mathrm{d}A_{\mathcal{A}} \tag{8}$$

$$= \sum_{\tau\in\mathcal{T}} \|\mathbf{J}(\tau)\|_{\mathrm{F}}^2 \, \mathrm{d}A_{\mathcal{B}}(\tau) + \|\mathbf{J}^{-1}(\tau)\|_{\mathrm{F}}^2 \, \mathrm{d}A_{\mathcal{A}}(\tau) \tag{9}$$

evaluated per linear piece $\tau \in \mathcal{T}$ with normalized area $A_{\mathcal{A}}(\tau)$ and $A_{\mathcal{B}}(\tau)$ on the respective surface [Schreiner et al. 2004].

To guarantee that Φ (given a feasible initialization) strictly stays within the class of bijective maps, we rely on the barrier character of $E(\Phi)$. Namely, it diverges to infinity as the image of a triangle of one mesh approaches a degenerate configuration on the other mesh. We extend the energy with infinity in the presence of a flipped triangle, i.e. in configurations in which (ϕ, h, ℓ) does not define an orientation-preserving map. This is detected during the mesh overlay procedure in Sec. 5.2.

The construction of the map Φ is parametrized by the vertex-to-surface map ϕ. Its continuous degrees of freedom are the barycentric coordinates α, β per vertex of $\mathcal{A}$ in a triangle of $\mathcal{B}$. In slight abuse of notation, we refer to our variable vector (the concatenation of all α, β) as $\phi \in \mathbb{R}^n$, with $n = 2|\mathcal{V}_{\mathcal{A}}|$.

Non-Convexity. The objective $E(\Phi)$ depends on the variables α, β in ϕ via composition of the following operators:

(1) barycentric evaluation in ambient space (linear),
(2) projection to model manifold (Eq. (1), non-convex),
(3) computation of intersection parameters or barycentric coordinates in other mesh (Eq. (4) and (7), non-convex),
(4) barycentric evaluation on surface (Eq. (3) and (5), linear),
(5) symmetric Dirichlet energy (non-convex).

As a composition of non-convex functions, E is also a non-convex function in ϕ. In the flat case, in which the projection operator reduces to the identity, this discussion of non-convexity matches the one given in [Schmidt et al. 2019, Sec. 5.1] for the disk topology case.

6.2 Derivatives

For each triangular piece τ (i.e., part of a polygon ρ) of the map, its Jacobian as well as the areas of its embeddings on $\mathcal{A}$ and $\mathcal{B}$ depend on the vertex positions of τ on both surfaces. These vertex positions in turn are computed from the variable vector ϕ using the formulas derived in Sec. 5.3. Each piece τ of the map—a summand in $E(\Phi)$—emerges as part of the overlap between a pair of triangles $(t_{\mathcal{A}}, t_{\mathcal{B}})$. While the triangle $t_{\mathcal{B}}$ is constant, the three vertices of $t_{\mathcal{A}}$ are parametrized via barycentric coordinates. Thus, each overlay triangle τ depends on exactly 6 entries in ϕ.

We collect all pieces of the map corresponding to the same triangle $t_{\mathcal{A}}$, i.e. all summands that depend on the same subset of 6 variables. We then compute the gradient $\mathbf{g}_{t_{\mathcal{A}}} \in \mathbb{R}^6$ and Hessian $\mathbf{H}_{t_{\mathcal{A}}} \in \mathbb{R}^{6\times6}$ independently for each $t_{\mathcal{A}}$ using automatic differentiation [Walther and Griewank 2012]. After projecting each $\mathbf{H}_{t_{\mathcal{A}}} \in \mathbb{R}^{6\times6}$ to a positive definite matrix (see [Schmidt et al. 2019, Sec. 5.1], [Teran et al. 2005, Sec. 6] for details), we assemble the global gradient $\mathbf{g}$ and sparse Hessian $\mathbf{H}$.

6.3 Switch of Representation Direction

A key feature of our approach is to not only optimize the images of vertices on the respective target surface, but to also optimize the underlying metric, which determines the images of points on edges and in triangles (cf. Fig. 2). This metric optimization is actually already present in the above formulation: Updating the vertex-to-surface map ϕ (described in Sec. 6.7) in direction $\mathcal{A}$ to $\mathcal{B}$ (moving $\mathcal{A}$ over $\mathcal{B}$), comes with an induced change of metric on $\mathcal{A}$. Hence, alternating between directions of our map representation, i.e. swapping the

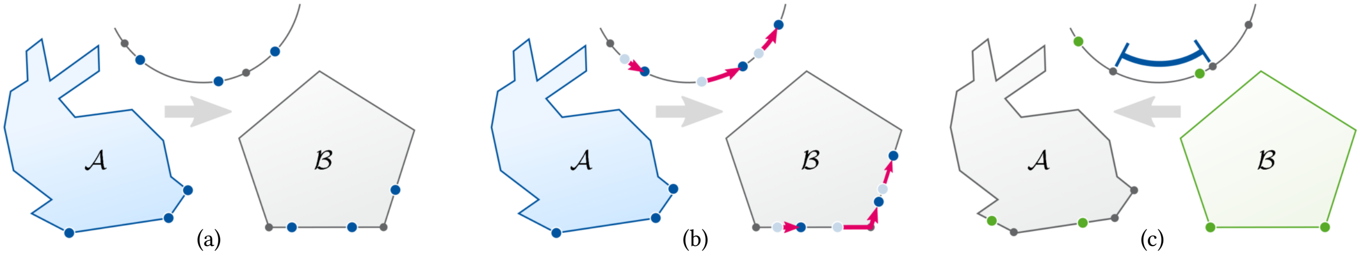


Fig. 8. 2D illustration of mapping between $\mathcal{A}$ and $\mathcal{B}$; in this case the model manifold is a circle (local embeddings visualized at the top). a) A map is represented in direction $\mathcal{A}$ to $\mathcal{B}$, by mapping vertices of $\mathcal{A}$ (blue) to points in triangles of $\mathcal{B}$. b) An iteration of our algorithm updates the vertex-to-surface map, i.e. moves the images of $\mathcal{A}$ along the surface of $\mathcal{B}$. c) Change of representation direction: using a local embedding on the model manifold, the positions of vertices of $\mathcal{B}$ on $\mathcal{A}$ are obtained and the metric lengths of the edges of $\mathcal{A}$ are measured.

roles of $\mathcal{A}$ and $\mathcal{B}$ between each iteration of optimization, modifies the constant-curvature metrics of both meshes and in this way takes advantage of all available degrees of freedom.

While throughout this paper our map representation was defined (w.l.o.g.) in direction $\mathcal{A}$ to $\mathcal{B}$, we now show how to reverse this direction. We define the operation

$$(\phi^{\triangleright}, h^{\triangleright}, \ell^{\triangleright}) \quad \mapsto \quad (\phi^{\triangleleft}, h^{\triangleleft}, \ell^{\triangleleft}),$$

where $\phi^{\triangleleft}$ is a vertex-to-surface map from $\mathcal{B}$ to $\mathcal{A}$, $h^{\triangleleft}$ assigns a homotopy representative on $\mathcal{A}$ to each edge of $\mathcal{B}$ and $\ell^{\triangleleft}$ is a discrete constant-curvature metric assigning a length to each edge of $\mathcal{A}$. After extracting the map Φ from $(\phi^{\triangleright}, h^{\triangleright}, \ell^{\triangleright})$, all information required to obtain its alternative representation $(\phi^{\triangleleft}, h^{\triangleleft}, \ell^{\triangleleft})$ is already present:

- The map $\phi^{\triangleleft}$ is fully defined by the assignment of vertices of $\mathcal{B}$ to triangles of $\mathcal{A}$ in Sec. 5.2 and their barycentric coordinates computed in Eq. (7).
- A (shortest) homotopy representative $h^{\triangleleft}$ for each edge $e_{\mathcal{B}}$ of $\mathcal{B}$ is obtained by collecting the sequence of edges of $\mathcal{A}$ intersected by $e_{\mathcal{B}}$ (Sec. 5.2) and taking their dual. The resulting dual path is the triangle strip of $\mathcal{A}$ supporting the image of $e_{\mathcal{B}}$ on $\mathcal{A}$.
- The discrete metric $\ell^{\triangleleft}$ at each edge of $\mathcal{A}$ is obtained by measuring the flat, spherical, or hyperbolic distance $\mathrm{d}(\tilde{\mathbf{p}}, \tilde{\mathbf{q}})$ between its endpoints in one of its embeddings on the model manifold (Sec. 5.2).

Note that both representations $(\phi^{\triangleright}, h^{\triangleright}, \ell^{\triangleright})$ and $(\phi^{\triangleleft}, h^{\triangleleft}, \ell^{\triangleleft})$, obtained this way, define the same map Φ. This is due to our symmetric definitions of the per-edge and per-face parametrizations in Sec. 5.3. This also implies that the change of representation direction is self-inverse, i.e., switching the direction twice yields the original map representation again. Fig. 8 illustrates the switch of representation direction after an iteration of our algorithm.

6.4 Continuity

For the optimization to have a chance to reasonably optimize with respect to $E(\Phi)$, it is important that this objective is continuous in the variables ϕ. This property is ensured by the following aspects of the described method:

- Due to our definition in Eq. (3) and (6), intersection parameters λ and vertex parameters α, β vary continuously in the variables ϕ by construction.
- When the shape of an overlay polygon undergoes a continuous change due to changes in ϕ, the quality (in terms of $E(\Phi)$) of its different triangulations changes continuously. At a point where the optimal triangulation criterion switches from one triangulation to another, both have the same quality; hence, $E(\Phi)$ varies continuously with ϕ.
- The objective contribution of each overlay polygon is proportional to its area; hence disappearing or emerging overlay polygons (when vertex images cross edges) do not cause discontinuities [Schmidt et al. 2019].
- As both directed representations, $(\phi^{\triangleright}, h^{\triangleright}, \ell^{\triangleright})$ and $(\phi^{\triangleleft}, h^{\triangleleft}, \ell^{\triangleleft})$, induce the same map Φ, switching the representation directions (as done for alternating optimization, cf. Sec. 6.3) does not cause discontinuities either.

6.5 Unique Geodesics Constraint

Our formulation relies on geodesics in a constant-curvature metric being unique per path homotopy class. While in the flat and hyperbolic case, this property holds in general, the spherical case comes with one particular exception: geodesics between antipodal points are not unique. This case is easily avoided by constraining all spherical edge lengths to be strictly shorter than π.

An additional constraint is posed by the projection to the sphere in Eq. (1). Namely, we require the projection of a sphere-inscribed chord triangle to the sphere to be orientation-preserving. This constraint is fulfilled if the plane spanned by three adjacent vertices on the sphere has an outward pointing normal, i.e. $\mathbf{n}^{\mathsf{T}}\tilde{\mathbf{p}} > 0$ for a point $\tilde{\mathbf{p}}$ on the plane. Note that satisfaction of this orientation constraint implies edge lengths shorter than π. We enforce it by adding the barrier term

$$B(\phi) = \sum_{t_{\mathcal{A}}} -\log(\mathbf{n}^{\mathsf{T}}\tilde{\mathbf{p}}), \tag{10}$$

with $\mathbf{n} \in \mathbb{R}^3, \|\mathbf{n}\| = 1$ being the oriented normal of the sphere-inscribed triangle $t_{\mathcal{A}}$, computed from the vertex positions $\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{r}}$. $B(\phi)$ diverges to infinity as the plane spanned by any three adjacent vertices on the sphere approaches the origin. We define $B(\phi) = \infty$ for $\mathbf{n}^{\mathsf{T}}\tilde{\mathbf{p}} \leq 0$.

6.6 Metric Regularization

As the map representation in constant curvature space is underdetermined, we follow the suggestions by [Schmidt et al. 2019, Sec. 6.4] and regularize the optimization towards a low-distortion metric. This has a beneficial effect on optimization performance and avoids premature convergence to unfavorable local minima (see Fig. 14).

Per triangle of $\mathcal{A}$ and $\mathcal{B}$, we penalize angle and area distortion between its original shape on the input surface and its shape under the constant curvature metric, i.e., its shape on the model manifold. For a triangle with embedding $(\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{r}})$ we compute its inner angles $\tilde{\theta}_p, \tilde{\theta}_q, \tilde{\theta}_r$ from its flat, spherical, or hyperbolic edge lengths (using the respective law of cosines), as well as its area $\tilde{A}$ via Heron's formula. We penalize deviation from the corresponding quantities $\theta_p, \theta_q, \theta_r, A$ on the input surface in a symmetric manner by

$$R_{\text{angle}}(\phi) = \sum_{\theta_i} \left(\frac{\tilde{\theta}_i}{\theta_i}\right)^2 + \left(\frac{\theta_i}{\tilde{\theta}_i}\right)^2$$

for each triangle corner i and

$$R_{\text{area}}(\phi) = \sum_{t} \left(\frac{\tilde{A}}{A}\right)^2 + \left(\frac{A}{\tilde{A}}\right)^2$$

per triangle $t \in \mathcal{A}$ or $\mathcal{B}$. These terms are weighted by ω_{angle} and ω_{area}; their choice is discussed in Section 7.2.

6.7 Optimization Algorithm

Our objective function and its derivatives exhibit similar properties to the one in [Schmidt et al. 2019]. In particular, $E(\phi)$ is a non-convex function, which is C^0 but not C^1 continuous in ϕ, i.e., in the vertex positions of $\mathcal{A}$ on $\mathcal{B}$. Further, at each point in the solution space, it requires re-computing the set of surface elements (overlay triangles) over which the distortion energy is integrated. Due to these similarities, we follow the basic optimization scheme of [Schmidt et al. 2019] and make the following adjustments to adapt it to our setting.

Derivative Smoothing and Preconditioning. To mitigate the effects of derivative discontinuities, we maintain temporally smoothed versions of the gradient and Hessian matrix and compute the Newton step based on these. Since we alternate between two optimization directions with different sets of variables, we store a separate smoothed gradient and Hessian for each direction.

In addition, we apply a squared Laplacian pre-conditioner to favor smooth and consistent vertex movements. I.e., intuitively, for each variable vertex mapped to the target surface, we penalize the deviation of its update direction from the average update of its neighbors.

Tangent Vector Transport. Both of the above techniques rely on concepts of parallel transport: The derivative history (a vector field in case of the gradient) needs to be transported from the vertex positions of a previous iteration to the current vertex positions. Similarly, update directions of neighboring vertices have to be transported into a mutual tangent space before they can be compared and averaged. In [Schmidt et al. 2019] this transport is trivially carried out due to the existence of a global flat coordinate representation. In our case, per-vertex updates are tangent vectors on the target surface, defined in local barycentric coordinate systems. We adapt both the derivative smoothing as well as the Laplacian pre-conditioning to our setting by means of parallel transport via the surfaces' Levi-Civita connection. While conceptually simple, a few details have to be considered; we give detailed implementation instructions in Sec. 1 of the supplementary material.

Update Tracing and Line Search. As a result of a Newton step, we obtain an update direction $(\mathrm{d}\alpha, \mathrm{d}\beta)^\top$ for each vertex of $\mathcal{A}$ in its current barycentric coordinate system. We trace these updates along the target surface via the same transport operator used above (see supplementary material for implementation details).

We scale each vertex update by the global step size s, determined in a line search. While in the planar case, the maximum feasible step size $s_{\max}$ can be computed in closed form [Smith and Schaefer 2015], such a feature is not trivially available in our discretization of the spherical and hyperbolic case. Instead, we perform a forward line search starting from $s = 1$ until a triangle of $\mathcal{A}$ is flipped (Sec. 5.2). We then perform a backtracking line search (multiplying s by 0.8) until the Armijo condition is fulfilled. In the line search, we compute evaluate the map Φ in order to evaluate $E(\Phi)$.

Adapting Homotopy Representatives. After obtaining for each vertex of $\mathcal{A}$ the path $(\mathbf{p}, \mathbf{p}')$ it travels on $\mathcal{B}$, we update ϕ accordingly. It remains to assign a new dual path $h'(\mathbf{p}', \mathbf{q}')$ to each edge $(\mathbf{p}, \mathbf{q})$ of $\mathcal{A}$ such that it represents the same homotopy class as $h(\mathbf{p}, \mathbf{q})$ before the update took place. This is achieved by concatenating: (1) the reversed update path $(\mathbf{p}', \mathbf{p})$ with (2) the old homotopy representative $h(\mathbf{p}, \mathbf{q})$ and (3) the update path $(\mathbf{q}, \mathbf{q}')$.

Metric Sanitization. We conclude each iteration by switching the map representation direction (Sec. 6.3). This involves obtaining a new constant-curvature metric $\ell^{\triangleleft}$ on $\mathcal{A}$ from local embeddings computed during mesh overlay in Sec. 5.2. These constructions, however, are subject to numerical inaccuracies that can accumulate over multiple iterations. If left untreated, this can cause the metric to diverge from its constant-curvature property. Since this error is small in each iteration, it can be treated by a simple sanitization step following each direction switch. In the flat and hyperbolic case, we apply one iteration of the discrete Yamabe flow [Bobenko et al. 2015] to $\ell^{\triangleleft}$, restoring its constant-curvature property. We provide implementation details of this procedure in Sec. 2 of the supplementary material. In the spherical case, a convenient solution is to track global embeddings of $\mathcal{A}$ and $\mathcal{B}$ into the unit sphere, and to perform a simple renormalization to the sphere to derive an accurate $\ell^{\triangleleft}$. This global embedding is only used for this sanitization purpose and is otherwise not required for any of our constructions.

6.8 Landmark Constraints

Vertex-wise landmark constraints, when fulfilled by the initialization, can be easily maintained throughout our optimization. To fix a vertex of $\mathcal{A}$ to its current target position on $\mathcal{B}$ under ϕ, we simply constrain its update direction to $\mathbf{0}$ in every iteration. This amounts to eliminating the variables corresponding to landmark vertices from the gradient and Hessian matrix before computing a Newton step.

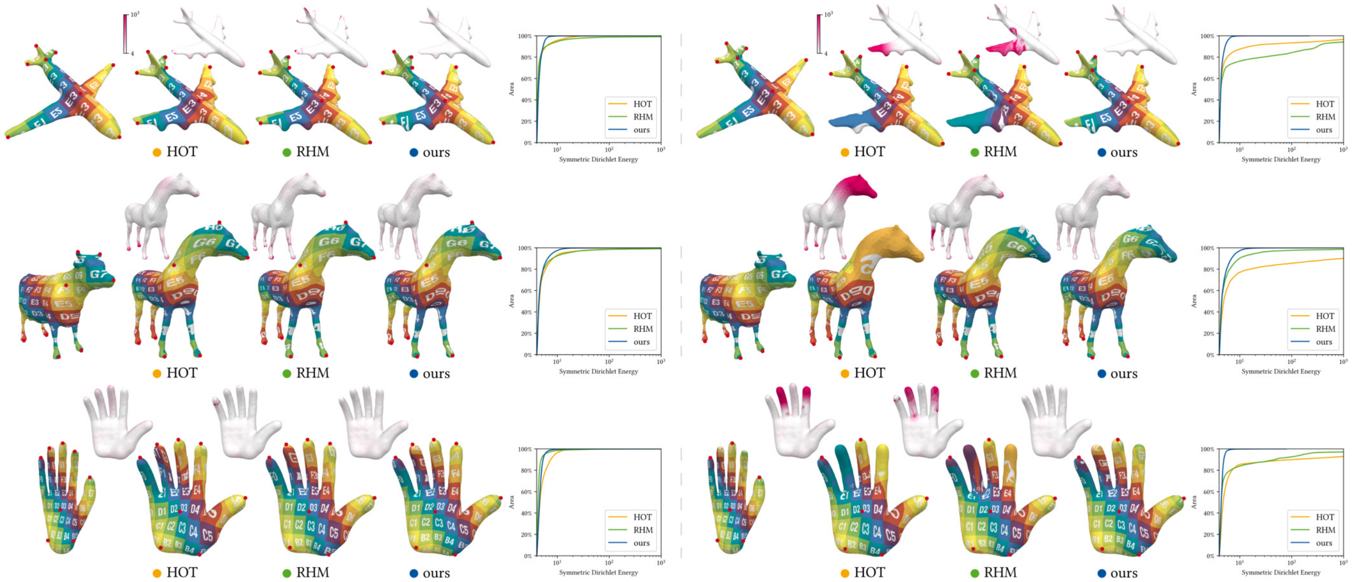


Fig. 9. We compute maps between pairs of shapes using the algorithms HOT, RHM and our method. Each map is visualized by transferring a texture from the source shape to the target shape. The pointwise symmetric Dirichlet energy is visualized as a logarithmic heat map. Cumulative distortion plots show which percentage of surface area is below a certain distortion threshold, for each of the three methods. Left: We first use a full set of landmarks (red dots). Right: We then repeat the experiment using only a subset of landmarks. While all algorithms produce fairly good results using the full set of landmarks, HOT does not align similarly curved regions in the absence of landmarks. Both RHM and our method strive to achieve such alignment (as this results in lower surface-to-surface distortion). RHM sometimes produces artifacts in regions of high curvature, e.g. the hooves and legs of the horse. This may be due to the extrinsic nature of their optimization. Our purely intrinsic method does not show such artifacts.

7 RESULTS

In the following we visualize various surface homeomorphisms optimized by our method. Due to the unified treatment of spherical, flat, and hyperbolic metrics, surfaces of arbitrary genus are easily handled by one implementation.

7.1 Initialization

A feasible initialization for the map optimization is a tuple (ϕ, h, ℓ) inducing a bijective orientation-preserving map Φ. There are multiple ways to obtain such an initialization. For example, in the spherical case any pair of orientation-preserving bijective spherical parametrizations, computed by a method of choice, implies such a tuple.

To carry out our experiments, we make use of an approach in which we first compute consistent triangular patches [Kraevoy and Sheffer 2004; Schreiner et al. 2004] on both input meshes. Given meshes $\mathcal{A}$ and $\mathcal{B}$ of genus g and a set of at least four pairs of corresponding landmark vertices we perform the following steps:

(1) Compute topologically identical path networks on $\mathcal{A}$ and $\mathcal{B}$ that form triangular patches and have the landmarks as vertices. We follow the instructions in [Schreiner et al. 2004]. Let $\mathcal{L}$ be this "macro-mesh" connectivity.
(2) Compute a discrete constant-curvature metric on $\mathcal{L}$, i.e. assign flat, spherical, hyperbolic edge lengths to $\mathcal{L}$. In the flat and hyperbolic case, obtain these edge lengths via discrete Euclidean or hyperbolic Yamabe flow [Bobenko et al. 2015] (detailed in Sec. 2 of the supplementary material). In the spherical case, we compute an orientation-preserving sphere embedding of $\mathcal{L}$ by positioning four landmarks as the vertices of a sphere-inscribed tetrahedron, followed by Tutte-embedding the remaining landmarks in its faces and edges, and projecting them onto the sphere.
(3) Independently per patch, embed its three corners as the vertices of a flat, spherical or hyperbolic triangle $\tilde{T}$ in $\mathbb{E}^2$, $\mathbb{S}^2$ or $\mathbb{H}^2$ (Sec. 5.1) with the computed edge lengths. Tutte-embed all vertices of mesh $\mathcal{A}$ (analogously for mesh $\mathcal{B}$) contained in the patch into its corresponding chord triangle (in ambient space), followed by a projection onto $\tilde{T}$ (Eq. (1)).
(4) Now, a bijective map between each macro-triangle patch on the surface of $\mathcal{A}$ or $\mathcal{B}$ and its corresponding flat, spherical, hyperbolic triangle is available. Composing such a map from a patch of $\mathcal{A}$ with the inverse of the corresponding map of $\mathcal{B}$, fully defines ϕ, h and ℓ for all triangles contained in the patch. For each vertex of $\mathcal{A}$ we save which triangle of $\mathcal{B}$ it falls into as well as its barycentric coordinates in that triangle. For each edge of $\mathcal{A}$ we detect intersections with edges of $\mathcal{B}$ to initialize h. For each edge of $\mathcal{B}$ we store its flat, spherical, or hyperbolic length on the model manifold to define ℓ.
(5) Reduce initial metric distortion (caused by jagged patch boundaries, distortion in the patch-wise Tutte embeddings, or non-optimal initial placement of landmark vertices on the model manifold): Perform the first 200 iterations of map optimization with increased metric regularization weights ($\omega_{\text{angle}} = 100$ and $\omega_{\text{area}} = 10$). In the spherical case, where global metric embeddings are available, we use global coordinates (vertex positions of $\mathcal{A}$ and $\mathcal{B}$ on the sphere, parametrized as points in $\mathbb{R}^3$, projected to $\mathbb{S}^2$) for increased efficiency in this step.

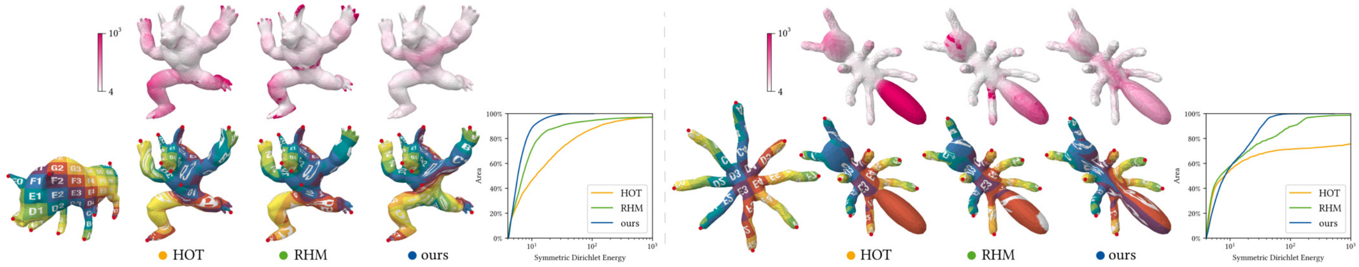


Fig. 10. We compute maps between non-isometric pairs of shapes, in which some amount of distortion is inevitable. All landmark correspondences (red dots) are kept fixed during the optimization. HOT generally produces smooth but highly distorted maps due to missing feature alignment (e.g. tail of the ant). RHM effectively eliminates distortion in some areas (e.g. body of the ant) but accumulates distortion in other areas. Our method produces a more even distribution of the inevitable distortion (note the low maximum distortion of our results in the cumulative distortion plots).

7.2 Experiments and Comparison

In our experiments, we chose weights $\omega_{\text{angle}} = 1$, $\omega_{\text{area}} = 0.001$ and performed a maximum of 600 iterations, except for the intentionally bad initializations in Fig. 13, which required 1500 iterations.

We compare the maps produced by our method to results of the state-of-the-art mapping methods HOT [Aigerman and Lipman 2016] and RHM [Ezuz et al. 2019b]. We ran all three algorithms on the same input data (meshes $\mathcal{A}, \mathcal{B}$ and a set of landmark constraints). We report the pointwise symmetric Dirichlet energy and visualize it via log-scale heat maps and cumulative distortion plots.

HOT comes with strict bijectivity guarantees via the use of two global embeddings into a hyperbolic cone orbifold. I.e. it, like our formulation, produces continuous and bijective maps (surface homeomorphisms). In contrast to our algorithm, HOT inherently relies on keeping a set of landmark constraints fixed during the optimization (as these form the singularities of the intermediate cone manifold). Further, the formulation of [Aigerman and Lipman 2016] does not include an assessment of surface-to-surface distortion. This produces low distortion maps only if similarly curved surface regions happen to align in the intermediate domain. In Fig. 9, we demonstrate that, when reducing the set of landmarks, this is not always the case.

RHM, by contrast, does optimize surface-to-surface distortion via a relaxed version of the symmetric Dirichlet energy. However, the resulting map is represented via two (not necessarily compatible) vertex-to-surface maps in both mapping directions, without a continuous notion of interpolation on edges and inside triangles. Further, by relying on a (potentially discontinuous) surface projection operator, it does not allow for strict bijectivity guarantees. We observe that while the resulting maps are well-behaved at most surface points, high distortion often concentrates in small areas, especially in regions of high curvature, such as tips of extremities (see Figures 9, 10, 11). We believe this to be due to the extrinsic aspects of the approach in combination with a discrete piecewise linear representation. Cases in which a certain amount of map distortion is inevitable (as in the highly non-isometric shape pairs in Fig. 10) are handled gracefully by our method and the distortion is distributed evenly.

Neither our method nor RHM is immune to converging to suboptimal local minima. However, we generally observe a good convergence behaviour. In Fig. 9, omitting a number of landmarks in key geometric regions does not prevent our method from converging to the desired minimum. Fig. 13 examines the sensitivity of our optimization to the initial map. While a number of different initializations converge to the same (expected) solution, the last two cases show initializations that lead to undesired local optima.

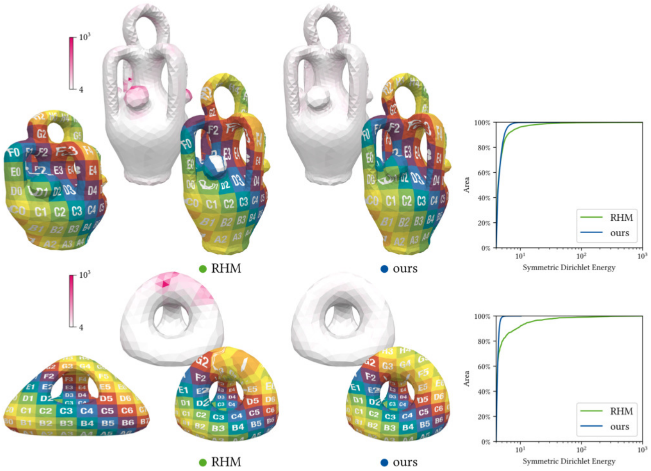


Fig. 11. We demonstrate our method on surfaces of genus 3 and 5 and compare to RHM. In both examples, landmarks are only used for initialization. While our method consistently achieves low map distortion at all surface points, RHM tends to concentrate distortion in some areas.

Figures 11 and 12 demonstrate the applicability of our methods to surfaces of varying genus. While in Figures 9 and 10, landmarks are kept fixed during the optimization, all landmarks are released after initialization in Fig. 11.

In Fig. 14 we show the relevance of metric regularization and observe that a low-distortion representation in constant curvature space benefits optimization performance and avoids small local minima.

As an example of information transfer, we map textures between multiple shapes in Fig. 12. Our method guarantees a continuous and bijective transfer. Mapping artifacts such as fold-overs or texture tearing cannot occur. This is of particular importance when the transferred data is used as an input for further processing applications, e.g. in the context of transferring integer-grid maps [Bommes et al. 2013] for subsequent quad mesh extraction. An additional property of our map definition is that it can transfer data that exhibits discontinuities across edges. This case occurs for example

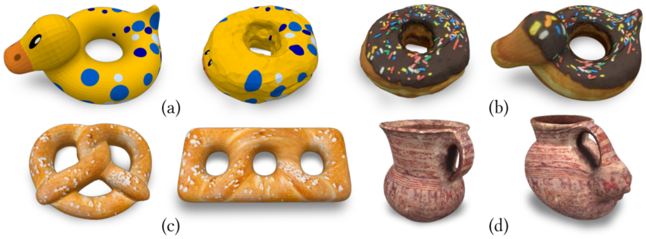


Fig. 12. Maps between different objects of genus 1 and genus 3 visualized via texture transfer and texture swapping. (a) and (b) show the same map being used in opposite directions, transferring the duck texture onto the donut and vice versa. In (c) the texture of the pretzel is mapped in a low distortion manner onto another genus 3 object, and in (d) the texture of a vase is mapped onto a different exemplar. Due to guaranteed bijectivity and continuity, no tearing artifacts or folds do occur.

when mapping textures with seams. A seam of the source mesh is faithfully represented on the target mesh via the mutual tessellation. In other words, we are able to map data defined per triangle corner instead of mere per-vertex information.

Timings. We implemented our method using automatic differentiation via [Walther and Griewank 2012]. While this enables flexible prototyping, it comes at a significant run time cost. E.g. the pig/armadillo example in Fig. 10 with $15k$ triangles takes 20s for the evaluation of gradient and Hessian per optimization iteration. This cost is not inherent to our strategy itself and can therefore most likely be reduced significantly using other forms of differentiation (which are not performed on the fly at run time). All other steps of our unoptimized implementation are comparably inexpensive. On the same example, the linear solve took less than $0.1s$ per iteration. The overlay computation, including the construction of all required model embeddings, exact intersection tests, and optimal polygon triangulation took less than $0.1s$ per evaluation.

8 LIMITATIONS & FUTURE WORK

Our current implementation allowed us to demonstrate the efficacy and effectiveness of the proposed representation and method, as presented in the previous section; it serves as a proof of concept. For practical efficiency, a relevant aspect for future work will be the investigation of techniques—on the implementation side as well as on the conceptual side—to speed up the time taken to perform map optimization. Combining the approach with some multi-resolution scheme will be an interesting direction. It could serve not only as a way to speed up the computation, but also to reach better optima via a multi-scale strategy. Depending on the application scenario (e.g., when initializations are bad, shapes are complex, or landmarks are few), this could be a major benefit, considering the non-linear nature of the optmization problem one faces in the inter-surface setting.

A further valuable direction is related to numerics. The current implementation sometimes suffers from numerical inaccuracies, for instance when tiny embedding errors accumulate along very long triangle strips (when the two surfaces have very different edge lengths) or the metric accumulates non-constant curvature over the course of long optimization sequences. We have already described some ways to sanitize this (cf. Sec. 6.7) that proved to work well in our experiments. Nevertheless, a deeper exploration of options for ensuring consistency despite inaccuracy will be interesting. An alternative could be the use of adaptive precision techniques, like in recent work on mesh processing [Hu et al. 2018].

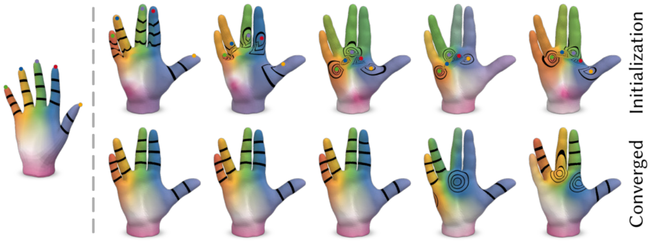


Fig. 13. We compute maps between two genus 0 hand models starting from different initializations (top row). Our optimization (bottom row) converges to the expected result if the initialization is reasonably close.

While we have already applied several modern optimization techniques and tweaks in our method, cf. Sec. 6, a variety of further significant improvement options, in particular tailored to parametrization and mapping problems, have been described in recent years [Claici et al. 2017; Kovalsky et al. 2016; Liu et al. 2018; Shtengel et al. 2017]. While these are probably not trivially applicable in our context, i.e., without a global coordinate system, it will be worthwhile exploring how ideas can be adopted.

ACKNOWLEDGMENTS

We thank the authors of the AIM@SHAPE Project and of the SHREC shape dataset. Additional meshes are courtesy of Keenan Crane (duck), as well as Jeremy E. Grayson (pretzel), Jcavalheiro96 (donut), José Diaz (crocodile, elephant) on sketchfab.com. We thank Jan Möbius for maintaining the OpenFlipper framework and Joe Jakobi for helping with an early prototype of our implementation.

This work was supported by the Excellence Initiative of the German federal and state governments (CompSE), as well as the Gottfried-Wilhelm Leibniz Programme and grant IRTG-2379 of the Deutsche Forschungsgemeinschaft (DFG).

REFERENCES

Noam Aigerman, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Spherical Orbifold Tutte Embeddings. *ACM Trans. Graph.* 36, 4 (2017), 90:1–90:13.
Noam Aigerman and Yaron Lipman. 2015. Orbifold Tutte Embeddings. *ACM Trans. Graph.* 34, 6 (2015), 190:1–190:12.
Noam Aigerman and Yaron Lipman. 2016. Hyperbolic Orbifold Tutte Embeddings. *ACM Trans. Graph.* 35, 6 (2016), 217:1–217:14.
Noam Aigerman, Roi Poranne, and Yaron Lipman. 2014. Lifted Bijections for Low Distortion Surface Mappings. *ACM Trans. Graph.* 33, 4 (2014), 69:1–69:12.
Noam Aigerman, Roi Poranne, and Yaron Lipman. 2015. Seamless Surface Mappings. *ACM Trans. Graph.* 34, 4 (2015), 72:1–72:13.
Miroslav Bacák. 2014. Computing medians and means in Hadamard spaces. *SIAM Journal on Optimization* 24, 3 (2014), 1542–1566.
Alex Baden, Keenan Crane, and Misha Kazhdan. 2018. Möbius registration. *Computer Graphics Forum* 37, 5 (2018), 211–220.
Alexander I Bobenko, Ulrich Pinkall, and Boris A Springborn. 2015. Discrete conformal maps and ideal hyperbolic polyhedra. *Geometry & Topology* 19, 4 (2015), 2155–2215.
David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.* 32, 4 (2013).
Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. 2013. Sparse Iterative Closest Point. In *Symposium on Geometry Processing*. Eurographics Association.
Sebastian Claici, Mikhail Bessmeltsev, Scott Schaefer, and Justin Solomon. 2017. Isometry-Aware Preconditioning for Mesh Parameterization. 36, 5 (2017), 37–47.

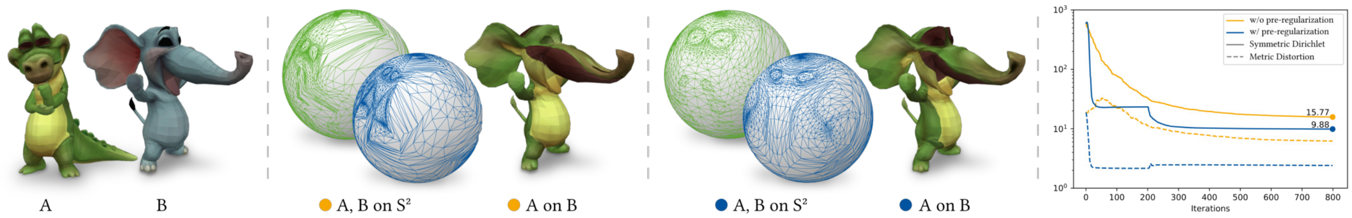


Fig. 14. We demonstrate the effect of metric regularization via two different optimization runs on the same input. The plot shows the surface-to-surface distortion (solid) as well as the metric distortion $R_{\text{angle}} + R_{\text{area}}$ (dashed) over the course of the optimization. (left, orange) With low regularization weights ($\omega_{\text{angle}} = 1$, $\omega_{\text{area}} = 0.001$) the metric distortion remains high, i.e. the embeddings of both $\mathcal{A}$ and $\mathcal{B}$ on the model manifold are quite distorted (see the $\mathbb{S}^2$ insets); the optimization progresses slowly and converges to an undesired local minimum. (right, blue) Performing an initial metric regularization phase (200 iterations with $\omega_{\text{angle}} = 100$, $\omega_{\text{area}} = 10$), as described in Sec. 7.1 (5), steers the optimization towards a low-distortion representation in constant curvature space. As a result, the subsequent surface-to-surface distortion optimization shows better performance and converges to a more desirable optimum.

Danielle Ezuz, Behrend Heeren, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. 2019a. Elastic Correspondence between Triangle Meshes. 38, 2 (2019).

Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. 2019b. Reversible harmonic maps between discrete surfaces. *ACM Trans. Graph.* 38, 2 (2019), 1–12.

Hershel M. Farkas and Irwin Kra. 1992. *Uniformization.* Springer New York, 166–256.

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4 (2018).

Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J. Guibas. 2008. Non-rigid Registration Under Isometric Deformations. In *Proceedings of the Symposium on Geometry Processing (SGP '08).* 1449–1457.

Takashi Kanai, Hiromasa Suzuki, and Fumihiko Kimura. 1997. 3D geometric metamorphosis based on harmonic map. In *Computer Graphics and Applications, 1997. Proceedings., The Fifth Pacific Conference on.* IEEE, 97–104.

Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. 2011. Blended intrinsic maps. *ACM Trans. Graph.* 30, 4 (2011), 1–12.

Shahar Z. Kovalsky, Meirav Galun, and Yaron Lipman. 2016. Accelerated Quadratic Proxy for Geometric Optimization. *ACM Trans. Graph.* 35, 4 (2016), 134:1–134:11.

Vladislav Kraevoy and Alla Sheffer. 2004. Cross-parameterization and Compatible Remeshing of 3D Models. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH '04).* 861–869.

Xin Li, Yunfan Bao, Xiaohu Guo, Miao Jin, Xianfeng Gu, and Hong Qin. 2008. Globally optimal surface mapping for surfaces with arbitrary topology. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 805–819.

Yaron Lipman and Thomas Funkhouser. 2009. Möbius voting for surface correspondence. *ACM Trans. Graph.* 28, 3 (2009), 72.

Nathan Litke, Marc Droske, Martin Rumpf, and Peter Schröder. 2005. An Image Processing Approach to Surface Matching. In *Symposium on Geometry Processing*, Vol. 255.

Ligang Liu, Chunyang Ye, Ruiqi Ni, and Xiao-Ming Fu. 2018. Progressive Parameterizations. *ACM Trans. Graph.* 37, 4 (2018), 41:1–41:12.

Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. 2017. Variance-minimizing transport plans for inter-surface mapping. *ACM Trans. Graph.* 36, 4 (2017), 39.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.* 31, 4 (2012), 30.

Daniele Panozzo, Ilya Baran, Olga Diamanti, and Olga Sorkine-Hornung. 2013. Weighted Averages on Surfaces. *ACM Trans. Graph.* 32, 4 (2013), 60:1–60:12.

Xavier Pennec. 2006. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *J. Mathematical Imaging and Vision* 25, 1 (2006), 127.

Emanuele Rodolà, Michael Möller, and Daniel Cremers. 2015. Point-wise Map Recovery and Refinement from Functional Correspondence. In *VMV.*

Raif M. Rustamov. 2010. Barycentric Coordinates on Surfaces. *Computer Graphics Forum* 29, 5 (2010), 1507–1516.

Patrick Schmidt, Janis Born, Marcel Campen, and Leif Kobbelt. 2019. Distortion-Minimizing Injective Maps Between Surfaces. *ACM Trans. Graph.* 38, 6 (2019).

John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. 2004. Inter-surface Mapping. *ACM Trans. Graph.* 23, 3 (2004), 870–877.

Andrei Sharf, Marina Blumenkrants, Ariel Shamir, and Daniel Cohen-Or. 2006. SnapPaste: An Interactive Technique for Easy Mesh Composition. *Vis. Comput.* 22, 9 (2006), 835–844.

Anna Shtengel, Roi Poranne, Olga Sorkine-Hornung, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Geometric Optimization via Composite Majorization. *ACM Trans. Graph.* 36, 4 (2017), 38:1–38:11.

Jason Smith and Scott Schaefer. 2015. Bijective Parameterization with Free Boundaries. *ACM Trans. Graph.* 34, 4 (2015), 70:1–70:9.

Justin Solomon, Andy Nguyen, Adrian Butscher, Mirela Ben-Chen, and Leonidas Guibas. 2012. Soft maps between surfaces. 31, 5 (2012), 1617–1626.

J. Stillwell. 1996. *Sources of Hyperbolic Geometry.* American Mathematical Society.

Gary K. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank Langbein, Yonghuai Liu, A. David Marshall, Ralph Martin, Xianfang Sun, and Paul Rosin. 2013. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics* 19, 7 (2013), 1199–1217.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '05).* 181–190.

Julien Tierny, Joel Daniels, Luis G. Nonato, Valerio Pascucci, and Claudio T. Silva. 2011. Inspired Quadrangulation. *Comput. Aided Des.* 43, 11 (2011), 1516–1526.

Andrea Walther and Andreas Griewank. 2012. Getting Started with ADOL-C. In *Combinatorial Scientific Computing*, Uwe Naumann and Olaf Schenk (Eds.). CRC Press, Taylor & Francis Group, 181–202.

Benjamin Wilson and Matthias Leimeister. 2018. Gradient descent in hyperbolic space. *arXiv preprint arXiv:1805.08207* (2018).

Huai-Yu Wu, Chunhong Pan, Qing Yang, and Songde Ma. 2007. Consistent correspondence between arbitrary manifold surfaces. In *ICCV 2007.* IEEE, 1–8.

Y. Yang, X. Fu, S. Chai, S. Xiao, and L. Liu. 2018. Volume-Enhanced Compatible Remeshing of 3D Models. *IEEE Transactions on Visualization and Computer Graphics* (2018).

Lei Zhang, Ligang Liu, Zhongping Ji, and Guojin Wang. 2006. Manifold parameterization. In *Advances in Computer Graphics.* Springer, 160–171.

A EMBEDDING FROM METRIC

To equip a local surface region with coordinates w.r.t. the metric ℓ, i.e. to embed it in $\mathbb{E}^2$, $\mathbb{S}^2$ or $\mathbb{H}^2$, we perform the following steps: In all three models, we choose to position the first vertex of the first triangle at $\tilde{\mathbf{a}} = (0, 0, 1)^\top$ and the second vertex $\tilde{\mathbf{b}}$ with distance ℓ_{ab} in an arbitrary direction, e.g.

$$\tilde{\mathbf{b}} = \begin{cases} (\ell_{ab}, 0, 1)^\top & \mathbb{E}^2 \\ (\sin \ell_{ab}, 0, \cos \ell_{ab})^\top & \mathbb{S}^2 \\ (\sqrt{(\cosh^2 \ell_{ab} - 1)}, 0, \cosh \ell_{ab})^\top & \mathbb{H}^2 \end{cases}$$

Given two points $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ on one of the model manifolds $\mathbb{E}^2$, $\mathbb{S}^2$ or $\mathbb{H}^2$, we now want to embed a third point $\tilde{\mathbf{c}}$ with known distances ℓ_{ac} and ℓ_{bc} to the given points and such that the triangle $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}$ is oriented counter-clockwise. In all three cases we can compute $\tilde{\mathbf{c}}$ via constructions in the Euclidean ambient space.

Flat. Given are $\tilde{\mathbf{a}}, \tilde{\mathbf{b}} \in \mathbb{E}^2$ and $\ell_{ac} = \|\tilde{\mathbf{c}} - \tilde{\mathbf{a}}\|_2$, $\ell_{bc} = \|\tilde{\mathbf{c}} - \tilde{\mathbf{b}}\|_2$, $\ell_{ab} = \|\tilde{\mathbf{b}} - \tilde{\mathbf{a}}\|_2$. Both candidate solutions for $\tilde{\mathbf{c}} \in \mathbb{E}^2$ lie on a line perpendicular to the line connecting $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$. Let $\mathbf{p}_0 + t\mathbf{n}$ be this line with $\mathbf{n} = \text{Rot}_{90}(\tilde{\mathbf{b}} - \tilde{\mathbf{a}})$ and $\mathbf{p}_0 = (1 - s)\tilde{\mathbf{a}} + s\tilde{\mathbf{b}}$. Then s and t are:

$$s = \frac{\ell_{ac}^2 - \ell_{bc}^2 + \ell ab^2}{2\ell_{ab}^2} \qquad t = \pm\frac{\sqrt{\ell_{ac}^2 - s^2\ell_{ab}^2}}{\ell_{ab}}.$$

Choosing the positive solution for t gives the counter-clockwise oriented triangle $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}$.

Spherical & Hyperbolic. Given are $\tilde{\mathbf{a}}, \tilde{\mathbf{b}} \in \mathbb{S}^2$ (or $\mathbb{H}^2$) and $\ell_{ac} = \mathrm{d}(\tilde{\mathbf{a}}, \tilde{\mathbf{c}})$, $\ell_{bc} = \mathrm{d}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}})$, with d being the spherical (or hyperbolic) distance function, respectively.

Around the given point $\tilde{\mathbf{a}}$, the iso-curves of $\mathrm{d}(\tilde{\mathbf{a}}, \cdot)$ are defined by the intersection of the model manifold (unit sphere or hyperboloid) with a tangent plane at $\tilde{\mathbf{a}}$, shifted in negative normal direction. Let $\mathbf{n}_a^\top \mathbf{p} = d_a$ be this plane, with $\mathbf{n}_a = [a_x, a_y, a_z]^\top$ ($\mathbf{n}_a = [-a_x, -a_y, a_z]^\top$), then $d_a = \cos(\ell_{ac})$ ($d_a = \cosh(\ell_{ac})$, proof below).

The points with given distances ℓ_{ac} and ℓ_{bc} to $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ lie on the intersection line of the planes $\mathbf{n}_a^\top \mathbf{p} = d_a$ and $\mathbf{n}_b^\top \mathbf{p} = d_b$. We parametrize this line by $\mathbf{p}_0 + t\mathbf{n}$. Since $\mathbf{n}$ lies in both planes we can choose $\mathbf{n} = \mathbf{n}_a \times \mathbf{n}_b$ and a point $\mathbf{p}_0$ on the line can be obtained as:

$$\mathbf{p}_0 = \frac{d_b}{\mathbf{n}_b^\top \mathbf{n}'_a}\mathbf{n}'_a + \frac{d_a}{\mathbf{n}_a^\top \mathbf{n}'_b}\mathbf{n}'_b$$

with $\mathbf{n}'_a = \mathbf{n} \times \mathbf{n}_a$ and $\mathbf{n}'_b = \mathbf{n} \times \mathbf{n}_b$. Both candidates for $\tilde{\mathbf{c}}$ are then obtained as the intersection points of this line with the sphere $\langle \mathbf{p}, \mathbf{p} \rangle = 1$ (hyperboloid $\langle \mathbf{p}, \mathbf{p} \rangle_M = -1$), i.e. via the solutions of the quadratic equation $c_2 t^2 + c_1 t + c_0 = 0$ with coefficients

$$\begin{aligned} c_2 &= \langle \mathbf{n}, \mathbf{n} \rangle, & c_1 &= 0, & c_0 &= \langle \mathbf{p}_0, \mathbf{p}_0 \rangle - 1 \quad (\mathbb{S}^2) \\ c_2 &= \langle \mathbf{n}, \mathbf{n} \rangle_M, & c_1 &= 2\langle \mathbf{n}, \mathbf{p}_0 \rangle_M, & c_0 &= \langle \mathbf{p}_0, \mathbf{p}_0 \rangle_M + 1 \quad (\mathbb{H}^2) \end{aligned}$$

which are $t = \pm(-c_1 + \sqrt{c_1^2 - 4c_2c_0})/(2c_2)$. Choosing the positive solution for t gives the counter-clockwise oriented spherical (hyperbolic) triangle $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}$.

In the spherical case this construction is limited to $\ell_{ab}, \ell_{bc}, \ell_{ca} < \pi$. We ensure that this constraint is always fulfilled (cf. Sec. 6.5).

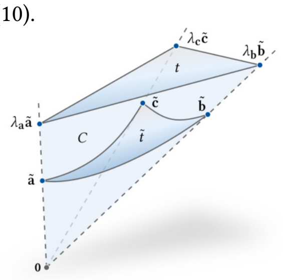
The iso-curves of $d_{\mathbb{H}}$ are shifted tangent plane intersections. For $\tilde{\mathbf{a}} \in \mathbb{H}^2$, the ℓ_{ac}-iso-curve of the hyperbolic distance function $d_{\mathbb{H}}$ is the set of points $\tilde{\mathbf{c}} \in \mathbb{H}^2$ for which $\ell_{ac} = \operatorname{arccosh}(-\langle \tilde{\mathbf{a}}, \tilde{\mathbf{c}} \rangle_M)$. We can rewrite $-\langle \tilde{\mathbf{a}}, \tilde{\mathbf{c}} \rangle_M$ as $[-a_x, -a_y, a_z]^\top \tilde{\mathbf{c}} = \mathbf{n}_a^\top \tilde{\mathbf{c}} = \cosh(\ell_{ac})$. The vector $\mathbf{n}_a$ is a normal of the implicit surface $h(\mathbf{a}) = a_x^2 + a_y^2 - a_z^2 + 1$ at point $\mathbf{a}$ since $[\frac{\partial h}{\partial a_x}, \frac{\partial h}{\partial a_y}, \frac{\partial h}{\partial a_z}] = -\mathbf{n}_a$.

B PROJECTION PROPERTIES

In this section we show that the projections between ambient space $\mathbb{R}^3$ and model manifold, as employed in our constructions, define a continuous, bijective, and orientation-preserving (c.b.o.) map.

Let $\tilde{t}$ be a flat/spherical/hyperbolic triangle with positive area spanned by (finite) vertices $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}} \in \mathbb{E}^2/\mathbb{S}^2/\mathbb{H}^2$, such that the triangle lies within one open halfspace about the origin of $\mathbb{R}^3$. Note that for the flat and hyperbolic case this is always true (as $\mathbb{E}^2$ and $\mathbb{H}^2$ are entirely contained in such a halfspace), whereas for the spherical case it is ensured by the barrier term (10).

Let $C = \{\mathrm{ray}(\mathbf{0}, \tilde{\mathbf{p}}) : \tilde{\mathbf{p}} \in \tilde{t}\} \subset \mathbb{R}^3$ be the cone of central rays through points of $\tilde{t}$ (see inset for the hyperbolic case). Due to the halfspace containment of $\tilde{t}$, C is a convex cone. Since $\tilde{t}$ has positive area, C is non-degenerate. Note that each ray C pierces exactly one such point, as the model manifolds, and therefore $\tilde{t}$, and entirely backfacing wrt. the origin. As each edge of $\tilde{t}$ is a geodesic, i.e. the intersection of a plane through the origin with the model manifold, C is a triangular cone with the edges $\mathrm{ray}(\mathbf{0}, \tilde{\mathbf{a}}), \mathrm{ray}(\mathbf{0}, \tilde{\mathbf{b}}), \mathrm{ray}(\mathbf{0}, \tilde{\mathbf{c}})$.



Let t be a flat triangle in $\mathbb{R}^3$ with vertices on these rays: $\lambda_\mathbf{a}\tilde{\mathbf{a}}$, $\lambda_\mathbf{b}\tilde{\mathbf{b}}$ $\lambda_\mathbf{c}\tilde{\mathbf{c}}$ where $\lambda_\mathbf{a}, \lambda_\mathbf{b}, \lambda_\mathbf{c} > 0$. The triangle $\tilde{t}$ is the image of t under central projection to the model manifold $\mathbb{E}^2/\mathbb{S}^2/\mathbb{H}^2$; vice versa, t is the image of $\tilde{t}$ under central projection to the supporting plane of t.

Since both t and $\tilde{t}$ form a base of C, i.e. each ray in C pierces t and $\tilde{t}$ exactly once, the central projection between t and $\tilde{t}$ is continuous and bijective. It is also orientation preserving, as both t and $\tilde{t}$ are backfacing w.r.t. the origin. Hence, the projection is c.b.o.

This theorem holds under the above preconditions that $\tilde{t}$ (1) has positive area and (2) is contained within one halfspace about the origin. In the following we show that the maps generated by our initialization satisfy these preconditions and that the optimization maintains them.

B.1 Map Initialization

Per macro-triangle of the macro-mesh (cf. Sec. 7.1), its initial constant curvature metric defines a triangle $\tilde{T}$ with positive area. The assumption that its vertices are initially contained within one halfspace about the origin is fulfilled by construction in the spherical case, cf. Sec. 7.1, step (2).

The central projection from the corresponding chord triangle T onto $\tilde{T}$ is c.b.o. The individual triangles t of $\mathcal{A}$ and $\mathcal{B}$, Tutte-embedded in a c.b.o. manner in T, therefore project onto conforming triangles $\tilde{t}$ with positive area in $\tilde{T}$, i.e., on the model manifold. Hence, after initialization, $\mathcal{A}$ and $\mathcal{B}$ are equipped with non-degenerate constant curvature metrics.

B.2 Map Optimization

By initialization, each model triangle $\tilde{t}_\mathcal{B}$ implied by the constant curvature metric has positive area. Therefore, the map from each $\tilde{t}_\mathcal{B}$ to its chord triangle $t_\mathcal{B}$ is c.b.o. For each triangle of $\mathcal{A}$: Let $t_\mathcal{A}$ be the flat triangle in $\mathbb{R}^3$ spanned by its three vertices that (via ϕ) lie in $t_\mathcal{B}^a$, $t_\mathcal{B}^b$, $t_\mathcal{B}^c$. Let $\tilde{t}_A$ be its image under central projection. As asserted by our initialization, $\tilde{t}_A$ has positive area, hence, this projection is c.b.o. The combined map $t_\mathcal{A} \to \mathbb{E}^2/\mathbb{S}^2/\mathbb{H}^2 \to t_\mathcal{B}$ (restricted to $\tilde{t}_\mathcal{A} \cap \tilde{t}_\mathcal{B}$, i.e., to an overlay polygon ρ) therefore is c.b.o. as well.

Relocating a vertex of $t_\mathcal{A}$ in $\mathbb{R}^3$ during an optimization iteration maintains the c.b.o. property as long as the above preconditions are preserved, i.e., as long as $\tilde{t}_\mathcal{A}$ maintains positive area and stays within an open halfspace about the origin of $\mathbb{R}^3$. Both conditions are preserved explicitly via barrier terms, Eq. (8) and Eq. (10), respectively.

Switching the direction of map representation (cf. Sec. 6.3) maintains the c.b.o property, because it does not change the metric, i.e., it preserves every triangle image $\tilde{t}_\mathcal{A}$ or $\tilde{t}_\mathcal{B}$ on $\mathbb{E}^2/\mathbb{S}^2/\mathbb{H}^2$. It therefore in particular preserves positivity and halfspace containment.

The c.b.o. nature of the map per overlay polygon ρ trivially carries over to restrictions, in particular to the triangles of a triangulation of ρ (cf. Sec. 5.4). The linearization of the map, as detailed in Sec. 5.4, is a continuous bijection that, due to preserving the map at the triangle's vertices, is orientation-preserving as well.