Seamless Parametrization in Penner Coordinates

RYAN CAPOUELLEZ, New York University, USA
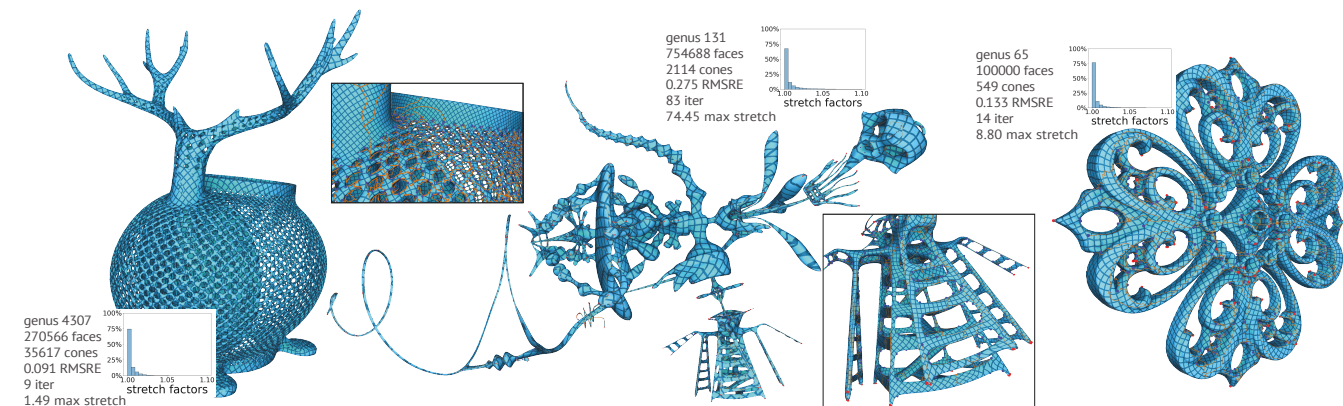DENIS ZORIN, New York University, USA

Fig. 1. Our algorithm is demonstrated to work reliably on a broad range of challenging models, with high geometric complexity and high genus. Three examples shown above: the highest-genus model from Thingi10k dataset [Zhou and Jacobson 2016], "yeahright" model courtesy of Keenan Crane, and the Filigree model from the dataset of Myles et al. [2014].

We introduce a conceptually simple and efficient algorithm for seamless parametrization, a key element in constructing quad layouts and texture charts on surfaces. More specifically, we consider the construction of parametrizations with prescribed *holonomy signatures* i.e., a set of angles at singularities, and rotations along homology loops, preserving which is essential for constructing parametrizations following an input field, as well as for user control of the parametrization structure. Our algorithm performs exceptionally well on a large dataset based on Thingi10k [Zhou and Jacobson 2016], (16156 meshes) as well as on a challenging smaller dataset of [Myles et al. 2014], converging, on average, in 9 iterations. Although the algorithm lacks a formal mathematical guarantee, presented empirical evidence and the connections between convex optimization and closely related algorithms, suggest that a similar formulation can be found for this algorithm in the future.

CCS Concepts: • **Computing methodologies** → **Mesh models**; **Mesh geometry models**.

Additional Key Words and Phrases: Parametrization, seamless, discrete metrics, cone metrics, conformal, intrinsic triangulation, Penner coordinates

Authors' addresses: Ryan Capouellez, rjc8237@nyu.edu, New York University, USA; Denis Zorin, dzorin@cs.nyu.edu, New York University, USA.

1 INTRODUCTION

Seamless parametrization is an important starting point for a widely used family of algorithms for constructing quad layouts. It can be applied in any context that requires an atlas of texture charts on the surface with complete freedom for choosing the cuts with no or minimal transition artifacts. Informally, a global parametrization i.e., a locally injective map from a cut surface to the plane, is seamless, if the parametric lines along u and v directions continue smoothly across the cut, and the parametric lengths of cut edges match.

A geometrically natural way to define a seamless parametrization is as a metric on the mesh (i.e., an assignment of *lengths* to edges) with angle constraints. A map to the plane corresponds to the angle constraint that at almost all vertices the sum of angles is 2π. The vertices where the sum is not 2π (*cones*) are necessary due to topological reasons for all surfaces of genus $g \neq 1$. *Seamless* parametrizations require additional conditions on sums of angles at cones and more generally along all dual loops: these need to be multiples of $\pi/2$ if added with appropriate signs (Section 5). There is a basis of $n_c + 2g$ loops, n_c loops around cone vertices and $2g$ homology basis loops, that completely determines the angles on all dual loops.

Following Campen and Zorin [2017b], we refer to this collection of $n_c + 2g$ angles of the form $k_i\pi/2$ as *holonomy signature*. One can think about this as defining the coarse topology of a seamless parameterization: e.g., if a parameterization is used to obtain a quadrangulation by tracing u and v lines, then cones become extraordinary vertices of valence different from 4, and holonomy angles on non-contractible loops define how quads are linked together along closed loops. As this qualitative behavior is a defining quality of a parametrization, its control is critical: e.g., changing one holonomy

angle on a loop may increase the distortion considerably, and make alignment with natural directions on the surface impossible.

In existing methods, holonomy signature is often determined partially or in full from an optimized cross-field, an optimization process for cone placement, or user input (placing cones). To have complete freedom of manipulating parametrization topology it is natural to ask the question on the constraints that need to be observed, specifically

> *For which holonomy signatures, seamless parametrizations with corresponding topology exist?* [Shen et al. 2022]

Under mild assumptions, this question was answered in Shen et al. [2022]. There is a seamless parametrization of a refinement of an input mesh for *any* holonomy signature (e.g., implied by a cross-field) under a simple condition: it is sufficient to have at least one cone with angle $3\pi/2$ or $5\pi/2$ (the actual condition is even weaker), unless the signature has *exactly* two cones ($n_c = 2$) with angles $3\pi/2$ and $5\pi/2$ respectively.

Shen et al. [2022] also describes an algorithm which, with an important caveat of numerical precision limitations, produces a seamless parametrization for any valid input holonomy signature. While in principle one can take any input cross-field and produce a parametrization with matching topology, the algorithm is complex and time-consuming, as we discuss in Section 2 in greater detail. It requires a combinatorial search for suitable loops, and extreme refinement at intermediate stages that may fail at standard floating point precisions.

Contributions. We propose a conceptually simple (effectively a Newton iteration on a set of equations), and efficient algorithm that we demonstrate to succeed on a version of Thingi10k dataset, for shapes of widely varying geometric and topological complexity, up to 800,000 vertices and genus over 4,000, as well as on the smaller dataset from Myles et al. [2014] that includes highly challenging crossfields. It typically obtains accurate constraint satisfaction and low distortion in just a few iterations.

A critical element of the algorithm following Capouellez and Zorin [2023], is working in the space of Penner coordinates, which establish a one-to-one correspondence between metrics on meshes with a given set of vertices, and assignment of real numbers to edges, which reduce to logarithmic lengths if an intrinsic Delaunay condition is satisfied. If it is not satisfied, Penner coordinates correspond to valid metrics on different connectivity with the same vertices, obtained by a simple flip algorithm from the original.

While we do not present a mathematical proof of its correctness, we outline the mathematical reasons why (under suitable limitations on the prescribed angles) we can expect the algorithm to succeed.

Aside from solving the seamless parametrization problem, the algorithm can be easily modified to solve other types of problems, e.g., the less restricted problem of parametrization with prescribed cones, computing a similarity map with a given holonomy structure, or adding additional angle-based constraints.

2 RELATED WORK

There is a wealth of work on various types of parametrization and related geometric problems. We briefly review the most closely related work; a more complete review can be found in surveys, e.g., Naitsat et al. [2021] and Fu et al. [2021]). Seamless parametrization in particular is a starting point for many quad layout algorithms, e.g., Bommes et al. [2009]; Campen et al. [2015]; Lyon et al. [2021] and many others. A range of general methods that can be applied both for seamless and other parametrizations, assume a feasible starting point already satisfying all constraints, is obtained by another method, e.g., Liu et al. [2018]; Rabinovich et al. [2017]; Schüller et al. [2013].

Intrinsic methods. Our method belongs to the category of *intrinsic* methods, working with variables intrinsic to the metric, as opposed to representing maps to the plane directly. Many intrinsic methods were proposed [Ben-Chen et al. 2008; Kharevych et al. 2006; Sheffer and de Sturler 2001; Springborn et al. 2008]. Among these, the most closely related to our work is the approach to discrete conformal maps described in Springborn et al. [2008] and extended in Campen et al. [2021] and Gillespie et al. [2021] to methods providing guarantees of correctness, based on the mathematical foundations built in Gu et al. [2018a,b]; Springborn [2020]. Angle constraints are natural for intrinsic methods; however, so far, no intrinsic method has been proposed for a full set of holonomy constraints: critically, there are not enough degrees of freedom in discrete conformal maps to satisfy all seamlessness constraints.

Another important extension, which considers *similarity maps* extending conformal maps, is Campen and Zorin [2017b]. While this method provides greater flexibility and thus making it possible to satisfy constraints on the loops, the resulting maps do not correspond to a metric.

Most recently, Capouellez and Zorin [2023], extended these approaches to general metric optimization, which is the closest work to ours. Crucially, it still relies on conformal maps to enforce constraints, and has the same limitation on the number of constraints that can be enforced. We discuss the similarities and differences in Section 6 in more detail.

Seamless parametrization constructions. Many methods for seamless parametrization were based on parametric plane coordinates. While most aim to preserve cones and input field topology, i.e., holonomy signature, in many cases, this goal is not stated explicitly.

To the best of our knowledge, Shen et al. [2022] is the only work presenting a method handling general seamless constraints with theoretical guarantees; however, the algorithm has many complex stages, and first constructs an extremely distorted parametrization that needs to be optimized at considerable expense. Less complete solutions with guarantees include Campen et al. [2019] and Zhou et al. [2020] which also involve a highly distorted parametrization stage, and do not provide control over loop holonomy angles. Another work with partial control of holonomy is Levi [2023].

The majority of methods are used for seamless parametrization, starting with Kälberer et al. [2007]; Tong et al. [2006], often in the context of quadrangulation applications. These methods often do not guarantee injectivity or finding a solution. For example, Bommes

et al. [2009], which is a foundation for many quadrangulation methods, uses a heuristic change of weights in an optimization problem to reduce the chance of foldovers. Other methods, e.g., Bommes et al. [2013]; Bright et al. [2017]; Campen et al. [2015]; Hefetz et al. [2019] use various types of convexified injectivity constraints [Lipman 2012], but there is no guarantee that a solution can be obtained; as shown in Myles et al. [2014], these methods do not find a valid solution in many cases. An alternative approach is to construct a T-mesh partition of the surface that does not necessarily correspond to a valid seamless parametrization, e.g., by tracing a cross-field, and then modify it by inserting or merging singularities [Lyon et al. 2021; Myles et al. 2014]. In comparison, our method produces a result without failures on a large dataset, with numerical difficulties only for extremely low mesh quality.

Cross-fields. As most common holonomy angles for parametrization are determined by a cross-field or a frame field [Vaxman et al. 2016], we briefly mention important work on field generation: Bommes et al. [2009]; Crane et al. [2010]; Farchi and Ben-Chen [2018]; Li et al. [2006]; Ray et al. [2009, 2008]; many of these offer control over the fields' turning numbers. Moreover, as demonstrated in Ray et al. [2008], a metric field can be obtained for *any* set of turning numbers, the field equivalent of the holonomy signature for parametrizations. In contrast, for parametrizations there are some exceptions, e.g., signatures with exactly two cones with angles $3\pi/2$ and $5\pi/2$.

3 OVERVIEW

We start with a precise formulation of a "naive" algorithm (that, in general, does not work) to make the problem formulation exact, and explain the approach in the simplest form.

Let M be a triangular mesh with N_v vertices, N_f faces, and N_e edges. For a vertex i and incident triangle T, let α_i^T be the angle of T at vertex i. Consider $2g$ dual loops L_j, $j = 1 \ldots 2g$, i.e., chains of triangles T_m^j, with two sequential triangles sharing an edge, and the first and last triangles sharing an edge. Each triangle has exactly one edge on the boundary of the triangle; let α_m^j be the angle opposite this edge. This is the angle between two internal edges e_{m-1}^j and e_m^j of the loop, where $m - 1$ is modulo loop length n_j. The notation is illustrated in Figure 2.

The *vertex angles* are defined as $\sum_{T \ni i} \alpha_i^T$, where the summation is over all triangles incident at vertex i. The loop *holonomy angles* are defined as $\sum_{m=1}^{n_j} d_m^j \alpha_m^j$ where d_m^j is 1, if the rotation from e_{m-1}^j to e_m^j is counterclockwise, and -1 if it is clockwise. This sum is equal to the discrete geodesic curvature of the loop, as each signed angle measures the rotation between two dual edges.

The *holonomy signature* is an assignment k_i^v, $i = 1 \ldots N_v$, and k_j^ℓ, $j = 1 \ldots 2g$ of integers to vertices and loops, corresponding to angles $k_i^v \pi/2$ and $k_j^\ell \pi/2$, and satisfying the discrete Gauss-Bonnet theorem, i.e., the sum of all vertex angles should be equal to $2\pi(2 - 2g)$

For an edge e, let ℓ_e denote the length of the edge, and $\lambda_e = 2 \log \ell_e$ the (scaled) logarithmic edge length. Clearly, the angles can be computed from edge lengths: $\alpha_i^T = \alpha_i^T(\lambda)$, where $\lambda \in \mathbb{R}^{N_e}$ is a vector of the logarithmic edge lengths.

The seamless metric is defined as a (nonunique) solution with respect to λ of the following constrained system of $N_v + 2g - 1$

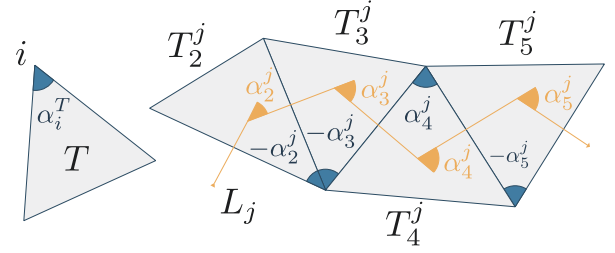equations in N_e variables (one vertex constraint is redundant due to Gauss-Bonnet):

Fig. 2. Holonomy angle notation. The signed loop holonomy angles $d_m^j \alpha_m^j$ measure the rotation between dual edges.

$$\sum_{T \ni i} \alpha_i^T(\lambda) = \frac{k_i^v \pi}{2}, \ i = 1 \ldots N_v - 1,$$
$$\sum_{m=1}^{n_j} d_m^j \alpha_m^j(\lambda) = \frac{k_j^\ell \pi}{2}, \ j = 1 \ldots 2g. \tag{1}$$

This is a nonlinear, underdetermined system of equations in variables λ. Let $\alpha(\lambda)$ be the vector of triangle angles, and let C be the $(N_v + 2g - 1) \times (3N_f)$ matrix of equations above, which are linear in α, and Θ the vector of holonomy signature angles. Then in vector form, the system is $C\alpha(\lambda) = \Theta$.

The naive algorithm for obtaining a feasible seamless parametrization is to solve this system of equations using Newton's method (it is not as naive as it sounds, as we explain below). As the system is underdetermined, we can solve it using the *extended* Newton method. Let $F(\lambda) = C\alpha(\lambda) - \Theta$ with $(N_v + 2g - 1) \times N_e$ Jacobian matrix $\nabla F = C\nabla_\lambda \alpha$,

---

Algorithm 1: Naive Newton algorithm.

1 **Function** FEASIBLESEAMLESS(λ^0, Θ)**:**
2     $\lambda \leftarrow \lambda^0$
3   **while not** CONVERGED(λ, Θ) **do**
4         $\nabla F \leftarrow C\nabla_\lambda \alpha$
5         $L \leftarrow \nabla F \nabla F^T$
6   Solve $L\mu = -F$
7         $d \leftarrow \nabla F^T \mu$
8         $\beta \leftarrow$ LINESEARCH(λ, d)
9         $\lambda \leftarrow \lambda + \beta d$.
10   **return** λ

We note that the gradient $\nabla_\lambda \alpha$ is closely related to the well-known cotangent matrix.

It is important that the algorithm is initialized with λ^0, the original edge lengths. If the constraints were linear, then the pseudoinverse solve would minimize the norm $\|\lambda - \lambda^0\|^2$, which is a measure of isometry used in Capouellez and Zorin [2023]. In the nonlinear setting, the change in the norm is minimized at each step; this is not equivalent to minimizing the norm subject to linear constraints, but, as we will see, it is a useful approximation.

However, clearly, there is no *a priori* reason to believe that the algorithm can find a solution to the constraint system: the solution may not even exist on fixed connectivity. In the next sections, we make changes to the algorithm which ensures that it converges in the case of seamless parametrization with only vertex constraints and provide empirical evidence and mathematical considerations that it can also handle the additional loop constraints.

Connection to discrete conformal maps. We further note that the algorithm is not as naive as it may first appear. With two changes, (1) eliminating the constraints on loops, and (2) reducing variables λ to $N_v - 1$ vertex logarithmic scale factors u via $\lambda_e = \lambda_e^0 + u_i + u_j$, where vertices i and j are the endpoints of edge e, the $N_v - 1$ *vertex* constraints happen to be the gradient of a convex function with respect to scale factors u. As a consequence, solving this restricted version of the constraint system by the Newton method is efficient and robust, and globally converges to a solution *if one exists*, as demonstrated in Campen et al. [2021]; Gillespie et al. [2021]. The critical step for making the conformal algorithm provably convergent was to enlarge the space by allowing connectivity changes, which we do next for our algorithm.

4 BACKGROUND: PENNER COORDINATES

We briefly summarize the idea and use of Penner coordinates. While the type of problems we consider (seamless parametrization with prescribed cones) does not necessarily have a solution for an arbitrary input mesh connectivity, it turns out that it can be solved if the connectivity is allowed to change, and it is sufficient to change it in a restrictive way: specifically, the vertices remain the same, but the connectivity may change through edge flips. However, performing optimization on varying mesh connectivity is difficult, as the variables and equations typically are connectivity-dependent. *Penner coordinates* provide a way to parameterize, with coherent variables, *all* metrics defined on *all* mesh connectivities sharing the same vertex set.

We start with an assignment (M, ℓ) of edge lengths ℓ to the edges of a mesh M, satisfying the triangle inequality. We will also use (M, λ) to denote an equivalent assignment of logarithmic lengths. (We use logarithmic edge lengths to eliminate a positivity constraint, and as we will see these are particularly natural for our algorithm.)

If we consider each triangle as flat, this assignment defines a *cone metric* on the mesh, with nonzero curvature only at vertices. If we are allowed to change connectivity, then there are many ways to describe the same metric: if we perform an intrinsic edge flip (Figure 3), the metric does not change, but we get new connectivity and new edge lengths (M', ℓ').

We can convert a description of a metric in terms of edge lengths to a (nearly) uniquely defined one if we require that the mesh M is *Delaunay*. In other words, given a pair (M, ℓ), we can produce a pair $(\tilde{M}, \tilde{\ell}) = \text{Del}(M, \ell)$ such that each edge e of $\tilde{M}$ satisfies the intrinsic Delaunay condition $\alpha_i + \alpha_j \leq \pi$, where α_i, α_j are the triangle angles opposite edge e. In terms of edge lengths, the Delaunay condition for edge e is equivalent to a simple rational expression:

$$\frac{\ell(a)^2 + \ell(b)^2 - \ell(e)^2}{2\ell(a)\ell(b)} + \frac{\ell(c)^2 + \ell(d)^2 - \ell(e)^2}{2\ell(c)\ell(d)} \geq 0 \qquad (2)$$
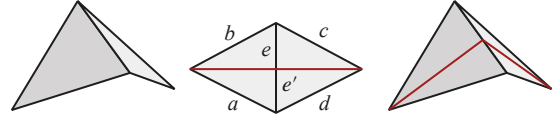
Fig. 3. Intrinsic flip. Week's flip algorithm uses Ptolemy formula for $\ell(e')$ which coincides with the Euclidean length if the angles opposite to e' sum up to π.
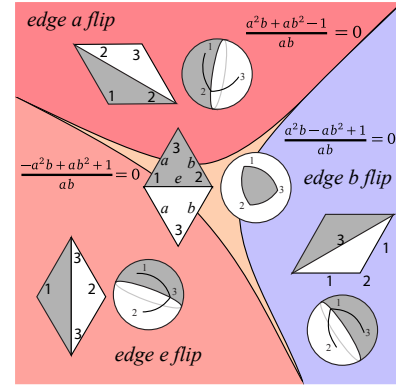

Fig. 4. Penner cell decomposition of cone metrics with 3 vertices and 3 edges (Figure from Capouellez and Zorin [2023])

$\text{Del}(M, \ell)$ can be computed using the standard intrinsic Delaunay flip algorithm, which repeatedly flips every edge that does not satisfy this condition until none are left.

If we fix the connectivity M, then there is a set of choices of cone metrics for which M is Delaunay. This set of metrics is called a *Penner cell* of M, which we denote $\mathcal{P}(M)$. For each cell, we have local coordinates for metrics, which are simply the edge lengths ℓ.

Penner cells cover the whole space of metrics with a fixed set of vertices and surface genus (Figure 4). Two adjacent cells differ by a single flip at an edge e. Since the sum of angles opposite e is equal to π at the boundary between the cells, this flip corresponds to a simple transformation for the lengths, defined by the Ptolemy formula. Removing e and inserting the flipped edge e' in a pair of adjacent triangles with external edges $\ell_a, \ell_b, \ell_c, \ell_d$ corresponds to the edge length update $\ell'(e') = \frac{\ell(a)\ell(c) + \ell(b)\ell(d)}{\ell(e)}$, and $\ell'(f) = \ell(f)$ for all edges $f \neq e$. These formulas define a transition from lengths with respect to connectivities M and M' of two adjacent cells. We denote this transition map $\tau(\ell)$.

The idea of Penner coordinates is to extend the length coordinates ℓ on one, arbitrarily picked, cell $\mathcal{P}(M_0)$ to the whole space of metrics, by using the formulas above as transition maps. Note that the formula can be applied to an assignment of positive numbers to the edges whether these correspond to actual lengths (i.e., satisfy triangle inequality) or not. Then for a chain of flips of edges $e_1, ..., e_n$ connecting two connectivities M and M_0, we can define the transition map $\tau(M, M_0) : \mathbb{R}_+^{N_e} \rightarrow \mathbb{R}_+^{N_e}$ as the composition $\tau_n \circ \tau_{n-1} \circ \cdots \circ \tau_1$ of the transition maps for the individual flips. These transition maps are smooth and well-defined, i.e., they do not

depend on the sequence of flips used to construct the map [Penner 1987].

More formally, given a connectivity M_0 and a metric (M, ℓ) the Penner coordinates of (M, ℓ) are defined as follows:

Definition 4.1. [Capouellez and Zorin 2023] Penner coordinates for a cone metric with length coordinates (M, ℓ) in Penner cell $\mathcal{P}(M)$, with respect to M_0 is a vector $P_{M_0}(M, \ell)$ of positive numbers in $\mathbb{R}_+^{N_e}$ defined as $P_{M_0}(M, \ell) = \tau(M, M_0)(\ell)$. i.e., simply the coordinate change from $\mathcal{P}(M)$ to $\mathcal{P}(M_0)$ by a composition of Ptolemy formulas.

The key features of Penner coordinates that we need for our algorithm are:

- For a fixed mesh M_0, any choice of logarithmic edge length assignments defines a metric. Its canonical representation $(\tilde{M}, \tilde{\lambda})$ can be obtained by the Week's flip algorithm, which is identical to the standard Delaunay flip algorithm, but with length updates based on the Ptolemy formula. It is important to note that the Delaunay criterion expressed in terms of lengths is well-defined for Penner coordinates, i.e., does not require triangle inequality, and Week's algorithm is guaranteed to produce a Delaunay mesh, i.e., at termination, Penner coordinates become lengths.
- Conversely, for any metric (M, λ) its logarithmic Penner coordinates (M_0, λ^0) can be obtained by finding a flip sequence connecting M and M_0 and applying Ptolemy transformations to λ at ever flip.

In this way, (logarithmic) Penner coordinates establish a one-to-one correspondence between the space of metrics on a mesh and $\mathbb{R}^{N_e}$.

The idea of extending Algorithm 1 to work on arbitrary meshes is to apply it to Penner coordinates on a mesh, which allows one to optimize in a larger space where, e.g., solutions for conformal maps are known to exist.

5 COMPLETE ALGORITHM

Next, we describe the modified version of Algorithm 1, now extended to solving the system over the whole space of metrics. Compared to the naive version, the most significant change to the algorithm is that the optimization is performed in Penner coordinates, i.e., variables assigned to the edges of the initial connectivity, which may not satisfy the triangle inequality.

To compute the angles that are needed both for constraints and constraint Jacobians in the algorithm, we simply apply the flip algorithm with Ptolemy length changes, to find the connectivity $\tilde{M}$ which is Delaunay with respect to the updated coordinates $\tilde{\lambda}$. As $\tilde{M}$ is Delaunay w.r.t. $\tilde{\lambda}$, the triangle inequality is satisfied, and angles can be computed. Our constraint function F in Penner coordinates thus becomes

$$F(\lambda) = C\alpha(\text{Del}(M_0, \lambda)) - \Theta = 0 \qquad (3)$$

The rest of the algorithm is largely unchanged, but some additional work is also necessary to update the angle Jacobian $\nabla_\lambda \alpha$ and the constraint matrix C in Penner coordinates, which we now delineate.
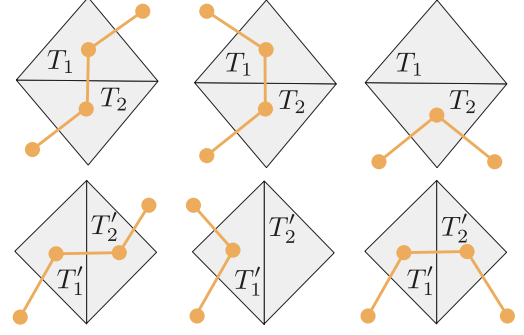

Fig. 5. Dual loop update for a flip.

Jacobian update. In order to compute

$$\nabla_\lambda \alpha = \nabla_{\tilde{\lambda}} \alpha \cdot \nabla_\lambda \text{Del},$$

we need to compute the Jacobian $\nabla_\lambda \text{Del}$ of the transition map to the Delaunay connectivity $\tilde{M}$. We use an incremental update of the Jacobian matrix as in Capouellez and Zorin [2023]. Specifically, the function DIFFPTOLEMY(M, ℓ, e) computes an update matrix corresponding to the Jacobian of the transition map corresponding to the change of coordinates resulting from flipping the edge e, with incident edges a, b, c, d as in Figure 3. Define the standard shear

$$t = \frac{\ell_a \ell_c}{\ell_b \ell_d}.$$

The matrix of derivatives of the transition map with respect to λ, is an identity matrix, except the rows corresponding to e, which is zero except the subrow corresponding to edges e, a, b, c, d, which has the form:

$$D_{e,[e,a,b,c,d]} = \left[-2, \frac{2t}{1+t}, \frac{2}{1+t}, \frac{2t}{1+t}, \frac{2}{1+t}, \right]$$

By the chain rule, $\nabla_\lambda \text{Del}$ is simply the product of the matrices corresponding to the sequence of flips determined by the flip algorithm.

Constraint update. Another complication that needs to be addressed is the update of the constraints themselves, which must be formulated in the Delaunay connectivity $\tilde{M}$ where we can compute angles. The update of the vertex angle constraints is simple: the angles incident at the vertex in $\tilde{M}$ are used for the constraint; the update for the loops is more complicated.

During each intrinsic flip and for each dual loop L_j, we find a dual loop on the flipped mesh M' that is topologically equivalent to the dual loop on the original mesh M. More formally, we find a loop L'_j that is homotopic to the original loop L_j, in the sense that the closed loop of dual edges defined by L_j can be continuously deformed to that of L'_j, without passing through cones. Since the intrinsic metric is flat away from cones, the holonomy of these two dual loops with respect to the metric (which is unchanged by the intrinsic flip) will be the same.

If L_j does not intersect the pair of adjacent triangles T_1, T_2 where a flip occurs, then there is nothing to do and the loop is left unchanged. If a dual loop does contain T_1 or T_2, we locally modify it

so that it enters and exits the flipped triangle pair T_1', T_2' across the same edges as in the original mesh. Three representative cases are illustrated in Figure 5. The constraint matrix is then recomputed for the new triangulation; the function UPDATECONSTRAINTS returns the updated matrix.

Line search. We use a backtracking line search. Since our extended Newton method does not correspond to a (known) convex energy, we cannot use a sufficient decrease condition, so instead we use the following conditions for the step size β:

(1) The norm of the constraint vector does not increase,

$$\|F(\lambda + \beta d)\| \leq \|F(\lambda)\|$$

(2) The direction of the constraint vector does not reverse,

$$F(\lambda) \cdot F(\lambda + \beta d) \geq 0$$

The first condition ensures that we make global progress in each iteration toward satisfying our constraints, and the latter condition ensures that our descent direction remains a descent direction at the end of the line step.

Algorithm 2: Seamless parametrization algorithm.

Input : triangle mesh $M = (V, E, F)$, closed, manifold,
edge lengths $\ell^0 = e^{\lambda^0/2}$ satisfying triangle inequality,
target angles $\hat{\Theta} > 0$ respecting Gauss-Bonnet at vertices, and on a basis of dual loops.

Output : triangle mesh $\tilde{M} = (V, \tilde{E}, \tilde{F})$,
edge lengths $e^{\tilde{\lambda}/2}$ satisfying triangle inequality,
with angles $\max_\Theta \|\Theta - \hat{\Theta}\| \leq \epsilon_c$.

1 **Function** FEASIBLESEAMLESS($M, \lambda^0, \hat{\Theta}$):
2     $\lambda \leftarrow \lambda^0$
3   **while** **not**CONVERGED(M, λ) **do**
4        $\tilde{M}, \tilde{\lambda}, D, C \leftarrow$ DIFFMAKEDELAUNAY(M, λ, C)
5        $\alpha, \nabla_{\tilde{\lambda}}\alpha \leftarrow$ COMPUTEANGLESANDGRADIENT($\tilde{M}, \tilde{\lambda}$)
6        $\nabla F \leftarrow C \nabla_{\tilde{\lambda}}\alpha\, D$
7        $L \leftarrow \nabla F \nabla F^T$
8   Solve $L\mu = -F$
9        $d \leftarrow \nabla F^T \mu$
10       $\beta \leftarrow$ LINESEARCH(λ, d)
11       $\lambda \leftarrow \lambda + \beta d$
12   **return**MAKEDELAUNAY(M, λ)

13 **Function** DIFFMAKEDELAUNAY(M, λ, C):
14     $\tilde{M}, \tilde{\lambda} \leftarrow M, \lambda$
15     $D \leftarrow$ Id
16     $Q \leftarrow \{e | \text{NONDELAUNAY}(M, \lambda, e)\}$
17   **while** $Q \neq \emptyset$ **do**
18   remove e from Q
19        $\tilde{M}, \tilde{\lambda} \leftarrow$ PTOLEMYFLIP($\tilde{M}, \tilde{\lambda}, e$)
20        $D \leftarrow$ DIFFPTOLEMY($\tilde{M}, \tilde{\lambda}, e$) $\cdot D$
21        $C \leftarrow$ UPDATECONSTRAINTS($\tilde{M}, \tilde{\lambda}, C, e$)
22   **return** $\tilde{M}, \tilde{\lambda}, D, C$

Postprocessing: overlay meshes. Once the final lengths on the mesh are computed, these need to be mapped to a refinement of the original connectivity (the overlay mesh). We use the approach of Capouellez and Zorin [2023] to do this without changes. Briefly, the intersection points of the original edges and flipped edges are tracked through the flips. As the Penner coordinates in intermediate configurations do not necessarily satisfy the triangle inequality, hyperbolic geometry considerations are used to assign edge coordinates to these vertices. At the end of the process, lengths are assigned to the edges of the overlay mesh, and it is mapped to the plane using a layout process similar to Springborn et al. [2008]. The process is fast and robust. Whenever possible, inserted vertices are removed, resulting only in a small percentage increase in mesh size relative to the original [Capouellez and Zorin 2023].

Preprocessing: intrinsic improvement. Since our method is fundamentally intrinsic, it is amenable to intrinsic preprocessing to improve the initial triangle quality without modifying the original geometry. Intrinsic Delaunay refinement [Sharp et al. 2019b], which inserts vertices in triangles with small angles to improve triangle quality while maintaining the Delaunay property with edge flips, is a particularly natural choice. Such vertex insertion is guaranteed to produce meshes with a minimum triangle angle of up to $30°$ at almost all vertices; however, thin needle-like features (see inset) with a very small total angle at a vertex cannot be improved by such intrinsic refinement. We use the implementation provided by Sharp et al. [2019a] with vertex insertions sufficient to produce triangle angles all above some threshold $\alpha_{\min}$ (away from needle-like features).

A simpler complimentary approach for intrinsic preprocessing is to interpolate between the Penner coordinates λ^0 of the original metric and perfectly regular Penner coordinates $\lambda = 0$, which corresponds to a metric of completely equilateral triangles with a uniform angle of $60°$. In other words, rather than initializing our method with $\lambda \leftarrow \lambda^0$, we can use $\lambda \leftarrow \beta\lambda^0$ for any $\beta \in [0, 1]$. Such interpolation can achieve arbitrarily regular triangles, but there is a trade-off between the initial triangle quality and the final geometric distortion as $\lambda \approx 0$ may initialize the optimization with a metric already far from λ^0.

We use a line search $\lambda^0 = \beta^n\lambda$ with $\beta = 0.9$ until $\min \alpha(\lambda^0) \geq \alpha_{\min}$. We also recenter λ^0, which corresponds to a global scaling with no impact on α, after interpolation so that the average Penner coordinate does not change, i.e., $\frac{1}{N_e}\sum_{e \in E} \lambda_e^0 = \frac{1}{N_e}\sum_{e \in E} \lambda_e$. This allows us to partially reduce the geometric distortion of the interpolation without compromising triangle quality.

6 MATHEMATICAL CONSIDERATIONS

The empirical performance of the algorithm shown in Section 7 provides strong evidence that the algorithm is likely to be related to a convex optimization problem: it is highly unlikely that Newton method in a vast majority of cases converges in a near-optimal number of iterations, and in all tested cases converges to a solution (with some caveats for extremely poor quality meshes). At the same time, three closely related problems do correspond to convex problems:

(1) metric optimization of convex objectives described in Capouellez and Zorin [2023] subject to angle constraints at vertices, but not on dual loops, (2) the conformal mapping based on scale factors described in Campen et al. [2021]; Gillespie et al. [2021], and (3) the similarity mapping based on a scale-factor 1-form ψ described in Campen and Zorin [2017b], with details on the convex functional provided in Campen and Zorin [2017a].
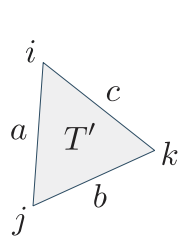
Comparison to metric optimization. Our algorithm is very close to the projected gradient algorithm of Capouellez and Zorin [2023], but is different in critical respects. The most important difference is the *absence of conformal projection*. The algorithm of Capouellez and Zorin [2023] performs a conformal projection at every state to enforce constraints, which requires an inner loop conformal solve, and more importantly limits the supported constraints to the ones conformal maps support (i.e., not full seamless constraints).

The second major difference is the absence of explicitly optimized energy. While this limits the type of parametrizations the method can produce, at the same time, this has a substantial impact on the algorithm's performance. Note that the method described in Capouellez and Zorin [2023] is inherently a first-order method, with linear convergence. To apply a second-order Newton method in the setting of constrained optimization, would require second derivatives of the constraints. However, these are known to be discontinuous. At the same time, our method is a Newton method with quadratic convergence, and only *first* derivatives of constraints are needed.

Comparison to conformal mapping. As described in Section 3, conformal methods, e.g., Campen et al. [2021]; Gillespie et al. [2021], operate in a reduced subspace of Penner coordinates spanned by logarithmic scale factors $u \in \mathbb{R}^{N_v}$. They minimize the convex function

$$E(u) = \sum_{T \in \tilde{M}} \left(2f(\tilde{\lambda}_a, \tilde{\lambda}_b, \tilde{\lambda}_c) - \pi(u_i + u_j + u_k) \right) + \Theta_V^\mathsf{T} u$$

where a, b, c and i, j, k are the edges and vertices of triangle T' respectively (see inset), and Θ_V are the vertex angle constraints.

As in our method, $(\tilde{M}, \tilde{\lambda}) = \mathrm{Del}(M, \lambda)$, where $\lambda_a = \lambda_a^0 + u_i + u_j$. f is a per-triangle function involving Milnor's Lobachevsky function [Springborn et al. 2008, Eq. (8)] with the important property that

$$\frac{\partial f}{\partial \tilde{\lambda}_a} = \alpha_k^T,$$

and the gradient of $E(u)$ is precisely the vertex constraints.

The iteration of our algorithm is very similar to the Newton iteration used to solve for conformal maps in Gillespie et al. [2021] and Campen et al. [2021]. There are several important differences: most significantly, as we use N_e length variables, rather than N_v vertex scale factors, we can introduce $V + 2g - 1$ holonomy constraints, with degrees of freedom to spare. As the optimization problem in the conformal case is fully constrained, there is no additional objective optimized directly or indirectly. In our case, each iteration optimizes the deviation from the metric at the previous step. For this reason,

as in Capouellez and Zorin [2023], we evaluate the descent direction and the Jacobian of constraints in Penner coordinates, i.e., with respect to the initial mesh connectivity.

Comparison to similarity mapping. Similarity methods, e.g., [Campen and Zorin 2017b], use the energy

$$E(\psi) = \sum_{T \in M} g(\psi_i^T, \psi_j^T, \psi_k^T) - \Theta^\mathsf{T} P^+ \psi$$

where ψ_i^T is the value of the closed one-form ψ on the edge of triangle T opposite vertex i, and P^+ is the pseodoinverse of a matrix formed by coefficient vectors for a basis of closed one-forms. Like f, the function g is a triangle function that depends on Milnor's Lobachevsky function with the corresponding property that

$$\frac{\partial g}{\partial \psi_i^T} = \alpha_i^T.$$

Similarity map problems were also considered in a similar context in Rivin [1994]. While these methods were developed in a fixed connectivity setting and a fully rigorous analysis using Penner coordinates is still absent in the literature, the theoretical extension is straightforward.

The mathematical approach used successfully to obtain variational principles for seamless constraints do not directly apply in the metric setting since the similarity methods produce a discrete scaling one-form that generally will not be exact and thus not integrable on the surface. However, we note that, intuitively, there are more than enough of degrees of freedom (N_e) in the metric λ to satisfy $(N_v - 1) + 2g$ constraints, and the $N_v - 1$ vertex constraints can be satisfied using the conformal degrees of freedom alone.

Invalid signatures. Unlike in the similarity setting, in the metric setting there are holonomy signatures that theoretically cannot be satisfied by any seamless parameterization. The known invalid signatures are:

- any signature with exactly two cones of angles $3\pi/2$ and $5\pi/2$, and
- a signature with no cones but nontrivial dual loop holonomy angles, i.e., $k_j^\ell \neq 0$ on simple loops.

The first case is considered in [Izmestiev et al. 2013; Jucovič and Trenkler 1973] and the second follows from the classification of flat metrics on a torus which all correspond to periodic tilings of the plane by parallelograms, and have trivial holonomy.

By the discrete Gauss-Bonnet theorem, these particular invalid cone prescriptions are only possible on genus 1 surfaces. They can also be interpreted in our metric setting. For instance, signatures with no cones correspond to flat tori, and the holonomy of any dual loop on a flat torus is necessarily trivial, so the loop holonomy constraints are redundant with the vertex angle constraints.

Verifying that our algorithm produces a valid solution whenever one exists, or identifying cases when it does not work is an important future direction.

7 EVALUATION

We evaluated our method on 94 closed meshes with challenging fields provided in the dataset of Myles et al. [2014]. Our method produces a parameterization satisfying both vertex and loop holonomy
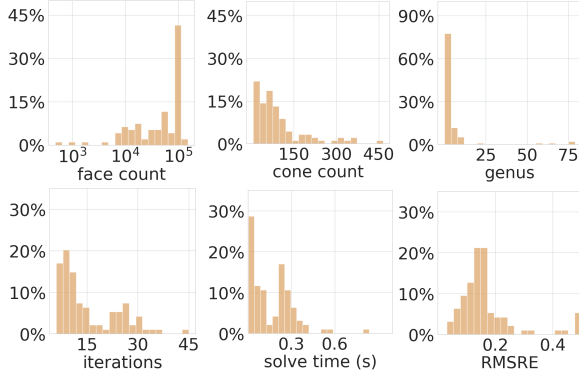
Fig. 6. Top: Distributions of mesh face counts, cross-field cone counts, and surface genus for the dataset of Myles et al. [2014]. Bottom: Distributions of iteration counts, average linear solve times, and RMSRE errors for our method. Outliers are aggregated in the rightmost bin.

Fig. 7. Modified Thingi10k dataset and result statistics. 548 meshes have genus above 20, with a maximum value of 4307.

constraints without exception in under fifty iterations, with the main bottleneck being a sparse linear solve for the descent direction, and the resulting parametrizations exhibit low geometric distortion. Example parametrizations are provided in Figure 11. Intrinsic preprocessing is only necessary for three meshes with poor triangle quality, and the final geometric distortion is qualitatively low. We also measure the distortion quantitatively in terms of Penner coordinates using the Root Mean Squared Relative Error

$$RMSRE(\ell, \ell^0) = \left(\sum_{e \in E} \frac{1}{|E|} \left(\frac{\ell_e - \ell_e^0}{\ell_e^0} \right)^2 \right)^{1/2},$$

where ℓ^0 are the lengths of the original embedding metric. Distributions of the dataset constraints and the performance of our method are provided in Figure 6.

Topological robustness. To evaluate the reliability of our method on surface meshes with more varied and extreme topology, we also tested our method on a dataset derived from Thingi10k [Zhou and Jacobson 2016]. As our focus in this experiment is topological robustness, i.e., the ability of the algorithm to handle high genus joined with complex geometry, as well as complex fields, we used the remeshed version of the dataset included with Hu et al. [2018], which includes almost all meshes from the original dataset, but with triangle quality improved to have worst inradius to circumradius ratio typically above 10^{-4}, in contrast to the original dataset containing many models with numerically degenerate triangles. As a significant fraction of meshes produced by Tetwild are non-manifold, but not in a fundamentally difficult way (i.e., a union of manifold 3D domains attached at edges and/or vertices), we separated all non-manifold meshes into closed manifold components, and split all meshes with at most 10 components into separate meshes, and selected meshes with nontrivial genus. Smooth cross-fields and corresponding holonomy angle constraints were computed for this dataset, and meshes for which the resulting vertex angle constraints were theoretically unsatisfiable were discarded (3-5 torus topology [Shen et al. 2022], 0 cone angles, and tori without cones but with nontrivial holonomy). In total, we obtained 16156 meshes by this
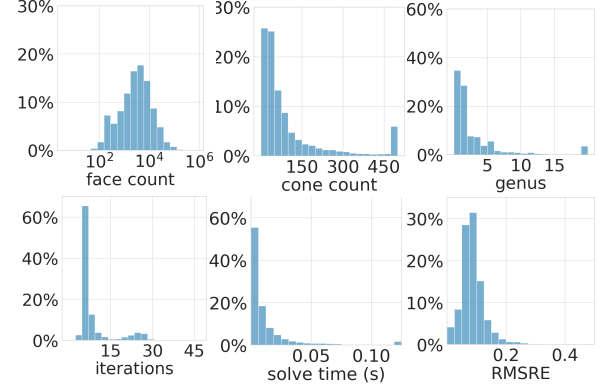
procedure. Statistics for this dataset and our method's performance on it are shown in Figure 7. This dataset contains extremely high genus meshes that impose proportionately many loop holonomy angle constraints, but our method quickly produces parametrizations satisfying these constraints with low distortion. The output and performance of our method on some of the highest genus meshes in this dataset are presented in Figure 10.

Intrinsic preprocessing. We evaluated both intrinsic preprocessing approaches, i.e., refinement and interpolation, on the closed manifold meshes of the original Thingi10k dataset [Zhou and Jacobson 2016]. After discarding meshes with degenerate triangles (with angles or lengths less than 10^{-10}), we obtained a dataset of 5342 connected meshes with generally poor triangle quality. By splitting the disconnected meshes into separate manifold components, we obtained a larger dataset of 27180 meshes. In order to test on this full dataset, we applied simple heuristics (e.g., adding random cone pairs) to minimally modify any theoretically unsatisfiable constraints and produce a valid holonomy signature.

Unlike the extrinsic Tetwild remeshing, neither intrinsic method individually succeeded on the dataset for any choice of $\alpha_{\min}$. However, *either* intrinsic refinement or interpolation succeeded on the *full* dataset of 27180 meshes with some $\alpha_{\min}$ within 500 iterations. That is, we parameterize any closed nondegenerate mesh in the original Thingi10k dataset using only intrinsic methods for preprocessing.

In Figure 8, we show the percentage of the 5342 connected models that failed to converge in 100 iterations with a timeout of 1 hour for increasing values of $\alpha_{\min}$. Note that for larger values of $\alpha_{\min}$ the number of failures with interpolation actually starts to increase. This degradation in performance is not surprising as the change in initial triangle angles $\alpha(\beta\lambda^0)$ resulting from small values of β may increase the initial angle constraint error $F(\beta\lambda^0)$, e.g., at high valence vertices. More iterations may thus be required to satisfy the angle constraints, and poorly conditioned triangles can arise in later iterations. We also note that the intrinsic refinement increases the mesh size and thus increases the per-iteration cost of our method, although we found that the increase in per-iteration time is often well compensated by a decrease in the total number of iterations.
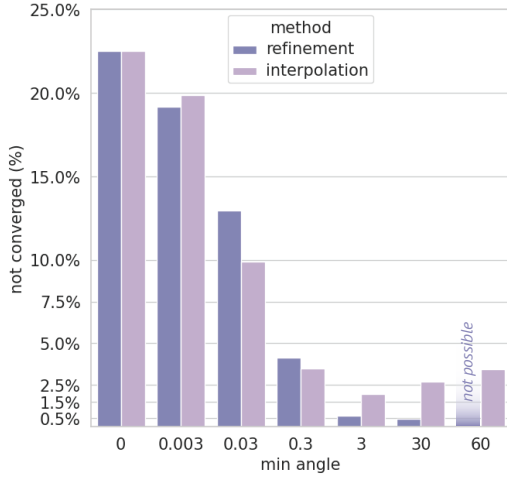
Fig. 8. Convergence results on connected, nondegenerate meshes in the original Thingi10k dataset with intrinsic preprocessing. Refinement and interpolation are performed until some minimum triangle angle threshold $\alpha_{\min}$ is satisfied (away from needle-like features for refinement).
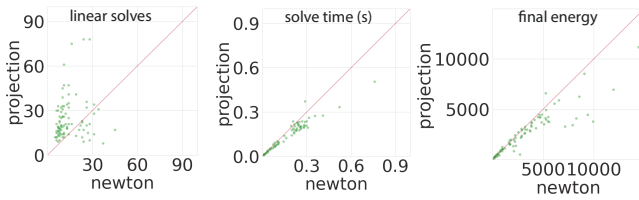

Fig. 9. Comparison of the performance of our Newton method with the projected gradient descent method of Capouellez and Zorin [2023] on the dataset from Myles et al. [2014]. We plot (a) the number of linear solves required by our method to converge to $\max_\Theta \|\Theta - \hat{\Theta}\| \leq 10^{-12}$, against (b) the number of linear solves in projected gradient descent necessary to achieve the same geometric distortion $||\lambda - \lambda^0||_2$. We also compare the average time for each linear solve and the final energies after running the metric isometry optimization to convergence.

Metric optimality. As stated above, our algorithm minimizes the norm $||\lambda - \lambda^0||_2$ at each Newton iteration, but, since our constraints are nonlinear, our final solution only approximately minimizes this norm. In contrast, Capouellez and Zorin [2023] explicitly optimizes the isometry energy $||\lambda - \lambda^0||_2^2$ while satisfying the same *vertex* angle constraints as our method; however, this method provides no control over *loop* holonomy angle constraints. In Figure 9, we compare the performance of this alternative approach and our method *with only vertex angle constraints*. Our method produces a comparable amount of geometric distortion while on average requiring fewer linear solves.

Symmetric Dirichlet post-processing. Once an initial seamless parameterization is obtained, it can be robustly optimized further to improve isometry using the symmetric Dirichlet energy by Newton's method with linear constraints to preserve seamlessness. We optimized our parameterizations of the dataset from Myles et al. [2014]; examples of parameterizations before and after optimization

are demonstrated in Figure 13. Unlike conformal methods, which can produce extreme geometric distortion and consequently numerical instability, our method produces initial parameterizations that are already approximately isometric and thus amenable to such optimization, and in most cases such post-processing is not needed.

Arbitrary loop holonomy angle constraints. While we primarily focus on parameterizations satisfying angle constraints arising from frame or cross-fields, our method also supports arbitrary holonomy angle constraints, including angle constraints that are not integral multiples of $\pi/2$. We demonstrate the effect of increasing the constraint for a single dual loop on the resulting parameterization in Figure 12. While the parameterization quality suffers from this extreme geometry distorting constraint, our method is still able to robustly produce a valid solution.

Robustness compared to state-of-the-art UV and hybrid methods. Our method succeeded on the closed meshes of the dataset introduced in Myles et al. [2014] and used in a few papers, with original connectivity and fields as input, as well as on 100% of 16,147 mesh dataset described above. We focus on methods that solve the seamless parametrization problem, i.e., aim to satisfy both angle and loop constraints.

The method of Myles et al. [2014] succeeds in producing a seamless parametrization (along with T-mesh partition) for all shapes, but for four, all closed, needs to add cones, and for 39 required T-mesh modification, which may result in changes in holonomy on loops. Three other methods evaluated in that paper are: original MIQ [Bommes et al. 2009], the IGM method [Bommes et al. 2013] (did not find a solution in 25% of cases), and MIQ combined with the convexified bijectivity constraints [Lipman 2012] (did not find a solution in 17%). Analyzing the data, we also observe that while the remaining parameterizations are always bijective, they often introduce integer-index cones, i.e., do not preserve holonomy signature.

More recent work includes Campen and Zorin [2017a] which succeeds on the whole dataset of Myles et al. [2014], but produces only similarity maps, not seamless maps. Most importantly, the recent method Shen et al. [2022], while having theoretical guarantees, fails on the highest genus models (6% of nontrivial genus models, highest genus around 100) in Myles et al. [2014] due to the extreme distortion of the intermediate maps it generates, and requiring expensive path rerouting and optimization steps in the end. In contrast, our method succeeds on the whole set (to the best of our knowledge, the first one, without using cone insertion or cone or holonomy angle modification).

8 CONCLUDING REMARKS

The main limitation of our algorithm that we have observed so far is sensitivity to extremely bad mesh quality, which is somewhat higher than for conformal maps. At the same time, its performance on a large dataset, along with theoretical considerations suggests that there are fundamental reasons for its convergence behavior, in particular, that can be connected to convex function optimization.

We identify several promising directions for future research: (1) placing the algorithm on a solid theoretical basis, and fully describing constraints on the holonomy signature for which it does not produce a seamless parametrization. (2) While the extension to handling boundaries is straightforward, following the doubling process described in Campen et al. [2021], the extension to constraints on sharp features (e.g., requiring features to be straight and axis aligned in parametric domain) requires more work. Techniques for doing this in angle variables were developed in Myles and Zorin [2013]. (3) The algorithm produces feasible seamless parametrizations but optimizes a specific energy only indirectly. (4) Due to its simplicity, it is amenable to domain decomposition and hierarchical extensions, to scale it to large meshes.

REFERENCES

Mirela Ben-Chen, Craig Gotsman, and Guy Bunin. 2008. Conformal Flattening by Curvature Prescription and Metric Scaling. *Computer Graphics Forum* 27, 2 (2008).

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), 98:1–98:12.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), 77.

Alon Bright, Edward Chien, and Ofir Weber. 2017. Harmonic Global Parametrization with Rational Holonomy. *ACM Trans.Graph.*36, 4 (2017).

Marcel Campen, David Bommes, and Leif Kobbelt. 2015. Quantized global parametrization. *ACM Trans. Graph.* 34, 6 (2015), 192.

Marcel Campen, Ryan Capouellez, Hanxiao Shen, Leyi Zhu, Daniele Panozzo, and Denis Zorin. 2021. Efficient and Robust Discrete Conformal Equivalence with Boundary. *ACM Trans. Graph.* 40, 6, Article 261 (dec 2021), 16 pages. https://doi.org/10.1145/3478513.3480557

Marcel Campen, Hanxiao Shen, Jiaran Zhou, and Denis Zorin. 2019. Seamless Parametrization with Arbitrary Cones for Arbitrary Genus. *ACM Trans. Graph.* 39, 1 (2019).

Marcel Campen and Denis Zorin. 2017a. *On Discrete Conformal Seamless Similarity Maps*. arXiv:1705.02422 [cs.GR]

Marcel Campen and Denis Zorin. 2017b. Similarity Maps and Field-Guided T-Splines: a Perfect Couple. *ACM Trans. Graph.* 36, 4 (2017).

Ryan Capouellez and Denis Zorin. 2023. Metric Optimization in Penner Coordinates. *ACM Trans. Graph.* 42, 6, Article 234 (dec 2023), 19 pages. https://doi.org/10.1145/3618394

Keenan Crane, Mathieu Desbrun, and Peter Schröder. 2010. Trivial Connections on Discrete Surfaces. *Computer Graphics Forum* 29, 5 (2010), 1525–1533.

Nahum Farchi and Mirela Ben-Chen. 2018. Integer-only cross field computation. *ACM Trans. Graph.* 37, 4, Article 91 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201375

Xiao-Ming Fu, Jian-Ping Su, Zheng-Yu Zhao, Qing Fang, Chunyang Ye, and Ligang Liu. 2021. Inversion-free geometric mapping construction: A survey. *Computational Visual Media* 7, 3 (2021), 289–318.

Mark Gillespie, Boris Springborn, and Keenan Crane. 2021. Discrete Conformal Equivalence of Polyhedral Surfaces. *ACM Trans. Graph.* 40, 4, Article 103 (jul 2021), 20 pages. https://doi.org/10.1145/3450626.3459763

Xianfeng Gu, Ren Guo, Feng Luo, Jian Sun, and Tianqi Wu. 2018a. A discrete uniformization theorem for polyhedral surfaces II. *Journal of Differential Geometry* 109, 3 (2018), 431–466.

Xianfeng Gu, Feng Luo, Jian Sun, and Tianqi Wu. 2018b. A discrete uniformization theorem for polyhedral surfaces. *Journal of Differential Geometry* 109, 2 (2018), 223–256.

Eden Fedida Hefetz, Edward Chien, and Ofir Weber. 2019. A Subspace Method for Fast Locally Injective Harmonic Mapping. *Computer Graphics Forum* 38, 2 (2019), 105–119.

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), 60–1.

Ivan Izmestiev, Robert B Kusner, Günter Rote, Boris Springborn, and John M Sullivan. 2013. There is no triangulation of the torus with vertex degrees 5, 6,..., 6, 7 and related results: Geometric proofs for combinatorial theorems. *Geometriae Dedicata* 166 (2013), 15–29.

Ernest Jucovič and Marián Trenkler. 1973. A theorem on the structure of cell–decompositions of orientable 2–manifolds. *Mathematika* 20, 01 (1973), 63–82.

F. Kälberer, M. Nieser, and K. Polthier. 2007. QuadCover: Surface Parameterization using Branched Coverings. *Computer Graphics Forum* 26, 3 (2007), 375–384.

Liliya Kharevych, Boris Springborn, and Peter Schröder. 2006. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.* 25, 2 (2006), 412–438.

Zohar Levi. 2023. Seamless Parametrization with Cone and Partial Loop Control. *ACM Transactions on Graphics* (2023).

W. Li, B. Vallet, N. Ray, and B. Levy. 2006. Representing Higher-Order Singularities in Vector Fields on Piecewise Linear Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 1315–1322.

Yaron Lipman. 2012. Bounded Distortion Mapping Spaces for Triangular Meshes. *ACM Trans. Graph.* 31, 4 (2012), 108:1–108:13.

Ligang Liu, Chunyang Ye, Ruiqi Ni, and Xiao-Ming Fu. 2018. Progressive parameterizations. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Max Lyon, Marcel Campen, and Leif Kobbelt. 2021. Quad layouts via constrained t-mesh quantization. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 305–314.

Ashish Myles, Nico Pietroni, and Denis Zorin. 2014. Robust Field-aligned Global Parametrization. *ACM Trans. Graph.* 33, 4 (2014), 135:1–135:14.

Ashish Myles and Denis Zorin. 2013. Controlled-distortion constrained global parametrization. *ACM Transactions on Graphics* 32, 4 (2013), 105.

Alexander Naitsat, Gregory Naitzat, and Yehoshua Y Zeevi. 2021. On Inversion-Free Mapping and Distortion Minimization. *Journal of Mathematical Imaging and Vision*

(2021), 1–36.

Robert C Penner. 1987. The decorated Teichmüller space of punctured surfaces. *Communications in Mathematical Physics* 113, 2 (1987), 299–339.

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Trans. Graph.* 36, 2 (2017), 16:1–16:16.

Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Levy. 2009. Geometry-Aware Direction Field Processing. *ACM Trans. Graph.* 29, 1 (2009).

Nicolas Ray, Bruno Vallet, Wan Chiu Li, and Bruno Lévy. 2008. N-Symmetry Direction Field Design. *ACM Trans. Graph.* 27, 2 (2008).

Igor Rivin. 1994. Euclidean structures on simplicial surfaces and hyperbolic volume. *Annals of mathematics* 139, 3 (1994), 553–580.

Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. 2013. Locally Injective Mappings. *Computer Graphics Forum* 32, 5 (2013), 125–135.

Nicholas Sharp, Keenan Crane, et al. 2019a. GeometryCentral: A modern C++ library of data structures and algorithms for geometry processing. https://geometry-central.net/. (2019).

Nicholas Sharp, Yousuf Soliman, and Keenan Crane. 2019b. Navigating intrinsic triangulations. *ACM Transactions on Graphics* 38, 4 (2019), 1–16.

Alla Sheffer and Eric de Sturler. 2001. Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with computers* 17, 3 (2001), 326–337.

Hanxiao Shen, Leyi Zhu, Ryan Capouellez, Daniele Panozzo, Marcel Campen, and Denis Zorin. 2022. Which cross fields can be quadrangulated? global parameterization from prescribed holonomy signatures. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–12.

Boris Springborn. 2020. Ideal Hyperbolic Polyhedra and Discrete Uniformization. *Discrete & Computational Geometry* 64, 1 (2020), 63–108.

Boris Springborn, Peter Schröder, and Ulrich Pinkall. 2008. Conformal Equivalence of Triangle Meshes. *ACM Transactions on Graphics* 27, 3 (2008), 1–11.

Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. 2006. Designing quadrangulations with discrete harmonic forms. *Symposium on Geometry Processing* (2006), 201–210.

Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional Field Synthesis, Design, and Processing. *Comp. Graph. Forum* 35, 2 (2016).

Jiaran Zhou, Changhe Tu, Denis Zorin, and Marcel Campen. 2020. Combinatorial construction of seamless parameter domains. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 179–190.

Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).
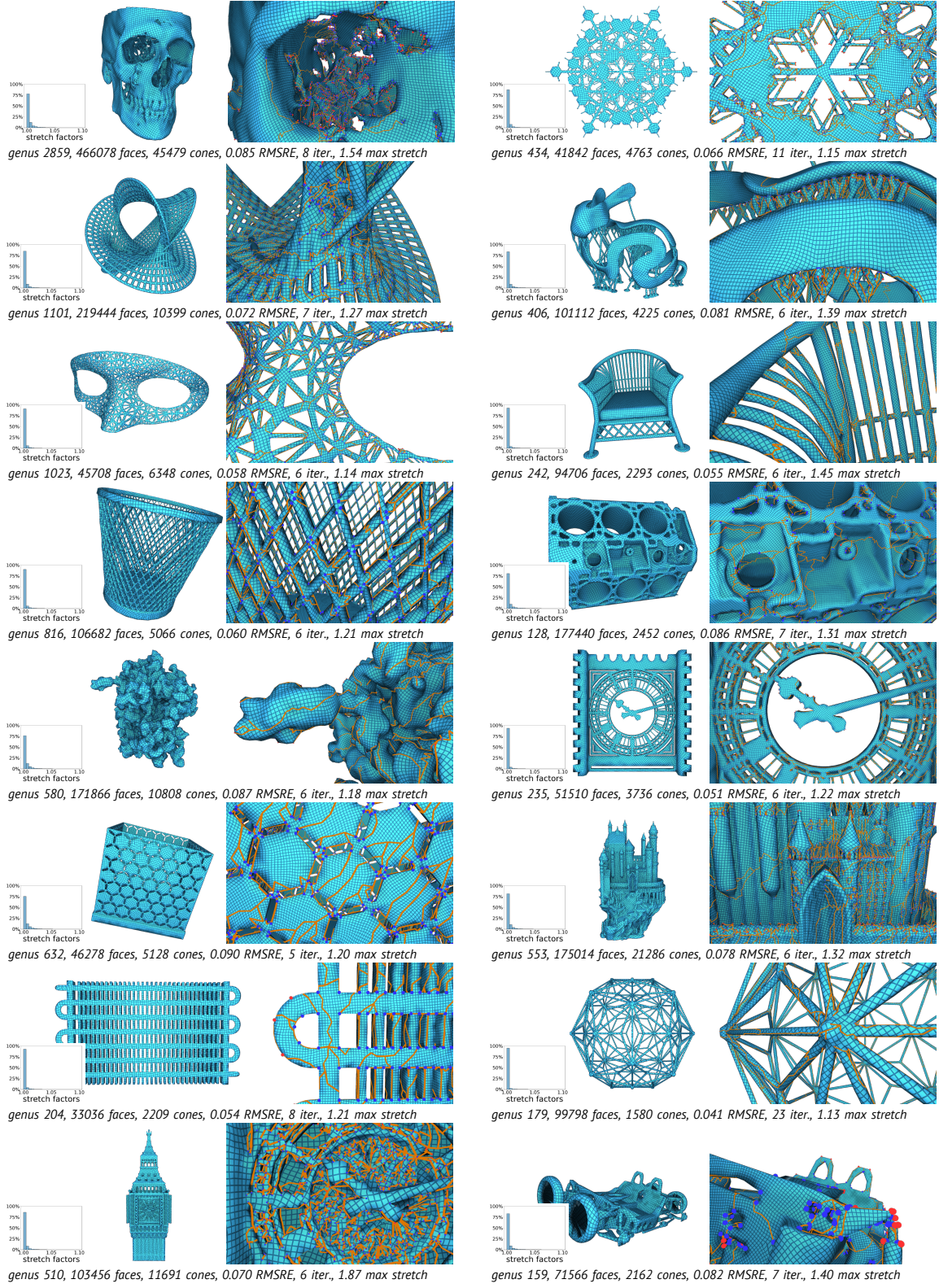
genus 2859, 466078 faces, 45479 cones, 0.085 RMSRE, 8 iter., 1.54 max stretch

genus 434, 41842 faces, 4763 cones, 0.066 RMSRE, 11 iter., 1.15 max stretch

genus 1101, 219444 faces, 10399 cones, 0.072 RMSRE, 7 iter., 1.27 max stretch

genus 406, 101112 faces, 4225 cones, 0.081 RMSRE, 6 iter., 1.39 max stretch

genus 1023, 45708 faces, 6348 cones, 0.058 RMSRE, 6 iter., 1.14 max stretch

genus 242, 94706 faces, 2293 cones, 0.055 RMSRE, 6 iter., 1.45 max stretch

genus 816, 106682 faces, 5066 cones, 0.060 RMSRE, 6 iter., 1.21 max stretch

genus 128, 177440 faces, 2452 cones, 0.086 RMSRE, 7 iter., 1.31 max stretch

genus 580, 171866 faces, 10808 cones, 0.087 RMSRE, 6 iter., 1.18 max stretch

genus 235, 51510 faces, 3736 cones, 0.051 RMSRE, 6 iter., 1.22 max stretch

genus 632, 46278 faces, 5128 cones, 0.090 RMSRE, 5 iter., 1.20 max stretch

genus 553, 175014 faces, 21286 cones, 0.078 RMSRE, 6 iter., 1.32 max stretch

genus 204, 33036 faces, 2209 cones, 0.054 RMSRE, 8 iter., 1.21 max stretch

genus 179, 99798 faces, 1580 cones, 0.041 RMSRE, 23 iter., 1.13 max stretch

genus 510, 103456 faces, 11691 cones, 0.070 RMSRE, 6 iter., 1.87 max stretch

genus 159, 71566 faces, 2162 cones, 0.082 RMSRE, 7 iter., 1.40 max stretch

Fig. 10. Parametrizations produced by our method for the modified Thingi10k dataset with per-edge symmetric stretch factors and mesh statistics.

genus 6
30658 faces
131 cones
0.133 RMSRE
9 iter
2.91 max stretch

genus 0
105022 faces
325 cones
0.122 RMSRE
32 iter
3.00 max stretch

genus 57
100000 faces
1369 cones
0.143 RMSRE
12 iter
3.08 max stretch

genus 7
100000 faces
52 cones
0.061 RMSRE
7 iter
1.37 max stretch

genus 5
82332 faces
70 cones
0.159 RMSRE
12 iter
3.29 max stretch

genus 0
14454 faces
42 cones
0.071 RMSRE
6 iter
1.15 max stretch

genus 4
50000 faces
112 cones
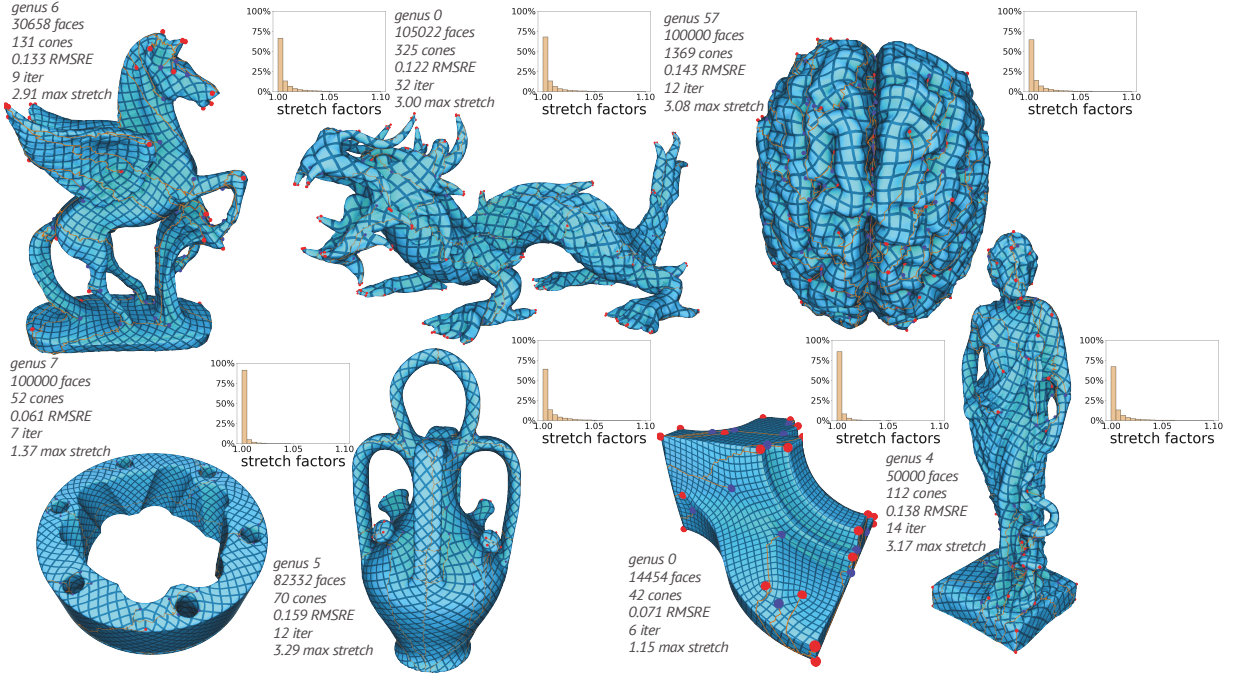0.138 RMSRE
14 iter
3.17 max stretch

Fig. 11. Parametrizations produced by our method for meshes from the dataset of [Myles et al. 2014] with histograms of per-edge symmetric stretch factors.

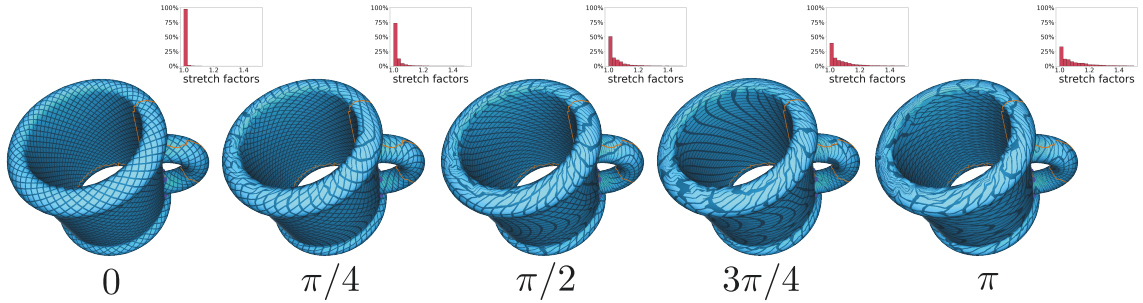0 $\pi/4$ $\pi/2$ $3\pi/4$ π

Fig. 12. Parametrizations produced by our method with an increasing holonomy angle constraint for a single dual loop. We increase the angle constraint from an initial value arising from a cross field in increments of $\pi/4$.
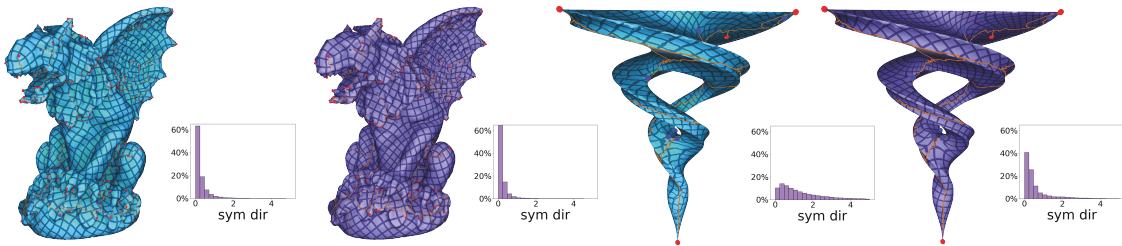

Fig. 13. Comparison of quality of parametrizations before (blue) and after (purple) symmetric Dirichlet optimization with seamless constraints. Histograms show the distribution of per face symmetric Dirichlet values. Left: our parameterization already has low geometric distortion, and is largely unchanged by further optimization. Right: intrinsic metric interpolation was used to improve the initial triangle quality, and our parameterization is distorted, but symmetric Dirichlet optimization is able to produce a visually pleasing result.