Closed-Form Construction of Voronoi Diagrams with Star-Shaped Metrics

HAOYANG ZHOU, ETH Zürich, Switzerland
LOGAN NUMEROW, ETH Zürich, Switzerland
STELIAN COROS, ETH Zürich, Switzerland
BERNHARD THOMASZEWSKI, ETH Zürich, Switzerland

Fig. 1. Our closed-form construction enables restricted star-shaped distance Voronoi diagrams on 3D surfaces. *Left:* metric with six-fold rotational symmetry. *Middle:* interpolation between star-shaped and Euclidean distance metric. *Right:* metric with threefold rotational symmetry. Diagrams under each example show the metric polygons defining each star-shaped distance function.

Cellular patterns, from planar ornaments to architectural surfaces and mechanical metamaterials, blend aesthetics with functionality. Homogeneous patterns like isohedral tilings offer simplicity and symmetry but lack flexibility, particularly for heterogeneous designs. They cannot smoothly interpolate between tilings or adapt to double-curved surfaces without distortion. Voronoi diagrams provide a more adaptable patterning solution. They can be generalized to star-shaped metrics, enabling diverse cell shapes and continuous grading by interpolating metric parameters. Martínez et al. [2019] explored this idea in 2D using a rasterization-based algorithm to create compelling patterns. However, this discrete approach precludes gradient-based optimization, limiting control over pattern quality. We introduce a novel, closed-form, fully differentiable formulation for Voronoi diagrams with piecewise linear star-shaped metrics, enabling optimization of site positions and metric parameters to meet aesthetic and functional goals. It naturally extends to arbitrary dimensions, including curved 3D surfaces. For improved on-surface patterning, we propose a per-sector parameterization of star-shaped metrics, ensuring uniform cell shapes in non-regular neighborhoods. We demonstrate our approach by generating diverse patterns, from homogeneous to continuously graded designs, with applications in decorative surfaces and metamaterials.

CCS Concepts: • **Computing methodologies** → **Shape modeling**; *Physical simulation.*

Additional Key Words and Phrases: Voronoi Diagram, Tiling, Pattern, Differentiable Simulation, Metamaterial

ACM Reference Format:
Haoyang Zhou, Logan Numerow, Stelian Coros, and Bernhard Thomaszewski. 2025. Closed-Form Construction of Voronoi Diagrams with Star-Shaped Metrics. *ACM Trans. Graph.* 44, 6, Article 236 (December 2025), 13 pages. https://doi.org/10.1145/3763296

Authors' Contact Information: Haoyang Zhou, ETH Zürich, Zürich, Switzerland, haoyang.zhou@ai.ethz.ch; Logan Numerow, ETH Zürich, Zürich, Switzerland, logan.numerow@inf.ethz.ch; Stelian Coros, ETH Zürich, Zürich, Switzerland, scoros@inf.ethz.ch; Bernhard Thomaszewski, ETH Zürich, Zürich, Switzerland, bthomasz@ethz.ch.

ACM 1557-7368/2025/12-ART236
https://doi.org/10.1145/3763296

1 Introduction

From planar ornaments such as wallpaper to architectural surfaces and mechanical metamaterials, cellular patterns combine aesthetics and function in many applications. Homogeneous patterns such as those from the family of isohedral tilings appeal through their simplicity and symmetry [Grunbaum and Shephard 1989; Kaplan 2009]. The high regularity of plane tilings, however, comes at the cost of low flexibility in particular regarding heterogeneous patterns.

For example, smooth interpolations between different tilings do generally not exist. Moreover, mapping planar tilings onto double curved surfaces is not possible without tile distortions.

The space of patterns induced by Voronoi diagrams is far more flexible in these regards. Given a set of sites, the Voronoi diagram is defined as the set of points that are equidistant from their closest sites. Chew and Dyrsdale [1985] expand the space of Voronoi diagrams by exploring generalized distance functions with convex level sets. Martínez et al. [2019] introduce generalizations to non-convex star-shaped level sets, opening a rich space of possible cell shapes. Moreover, the patterns can be continuously graded by simply interpolating distance metric parameters across the design. Martínez et al. [2019] demonstrate the potential of this concept in two dimensions using a rasterization-based algorithm that determines cell boundaries (*i.e.* the diagram) by advancing fronts in a time-discrete manner. While this approach can produce visually compelling patterns, the inherently discrete nature of the rasterization approach precludes gradient-based design optimization.

In this work, we present a novel closed-form construction of Voronoi diagrams with star-shaped metrics. We replace the raterization-based algorithm by Martínez et al. [2019] with an explicit, fully differentiable construction, enabling optimization within a continuous and compact parameterization of dense cellular patterns. At the core of our approach lies the concept of generalized cones that partition the space around Voronoi sites and divide Voronoi edges into linear segments. The patterns are determined by site locations and metric parameters, but finding inputs that lead to well-shaped cells is challenging, especially on curved surfaces. Our method naturally generalizes to arbitrary dimensions, and we demonstrate the extension of a restricted 3D version to generate patterns on arbitrary surface meshes. A full 3D version and its application to volumetric metamaterials is described in [Numerow et al. 2025]. To maximize pattern quality in the on-surface setting, we furthermore propose a per-sector parameterization of star-shaped metrics that is able to generate uniform cell shapes for non-regular neighborhoods. We demonstrate the versatility of our method by generating a broad and diverse set of patterns, including homogeneous and continuously graded patterns. We show applications to decorative patterning of 3D surfaces and metamaterials optimized to achieve desired macromechanical properties.

2 Related Work

Pattern Generation. Generating patterns on 3D surfaces is a problem that has received much attention from the graphics community. The spectrum of existing methods is large, ranging from texture mapping [Lefebvre and Hoppe 2006; Neyret and Cani 1999; Turk 2001] and procedural geometry synthesis [Bhat et al. 2004; Brodersen et al. 2008; Porumbescu et al. 2005; Zhou et al. 2006] to specialized methods for generating scale [Landreneau and Schaefer 2010], curve [Michel and Boubekeur 2023], and stripe [Knöppel et al. 2015] patterns. A more recent stream of works focuses on creating patterns that combine aesthetic and functional goals [Chen et al. 2017, 2016; Martínez et al. 2015; Schumacher et al. 2016; Zehnder et al. 2016]. Our work shares the general goal of designing patterns in controllable, flexible, and functional manners. However, we focus on smoothly varying cellular patterns induced by Voronoi diagrams with non-standard metrics.

Irregular Tilings and Packings on Surfaces. A line of pattern generation methods extends irregular tilings [Glassner 1998] and packings [Bennell and Oliveira 2009] from 2D planes to manifold surfaces. Decorative and mosaic approaches distribute irregular tiles for aesthetic surface coverings, often guided by optimization or Voronoi partitions [Chen et al. 2017; Fanni et al. 2022; Hu et al. 2016; Liu et al. 2020]. Procedural and mesostructural methods employ tile-based primitives to generate non-periodic, connected structures on meshes and volumes [Fu and Leung 2005; Michel and Boubekeur 2023]. In the architectural and fabrication domain, irregular tilings are used in panelization and pattern optimization under manufacturing constraints [Du et al. 2023; Eigensatz et al. 2010]. Building on this body of work, we propose to extend Voronoi tessellations with non-Euclidean metrics, enabling tile patterns that combine aesthetics with functional control.

Metamaterial Design. Beyond purely decorative purposes, patterns and tilings find functional applications in many fields. One such example is the design of flexible metamaterials which, through precisely architected microstructures, can achieve a broad range of macromechanical properties [Bertoldi et al. 2017]. The spectrum includes materials with lattice- [Panetta et al. 2017, 2015], voxel- [Schumacher et al. 2015; Zhu et al. 2017], and foam-like [Martínez et al. 2016, 2017] structures, as well as planar microstructures [Li et al. 2023; Tozoni et al. 2020; Zhang et al. 2023] and sheet-like materials [Leimer and Musialski 2020; Martínez et al. 2019; Montes Maestre et al. 2024; Schumacher et al. 2018].

The above mentioned works target periodic materials that are generated by repeated tiling of a single unit cell. Another class of methods takes a global approach for generating structured materials. Examples include tube-like microstructures whose channels follow given direction fields [Tricard et al. 2020] as well as bi-material distributions [Montes et al. 2023] or pneumatic channels [Panetta et al. 2021; Ren et al. 2024] in the form of stripe patterns. While these works provide specialized solutions for different types of materials, we address the fundamental problem of designing heterogeneous cellular patterns.

Of particular relevance to our work are metamaterials based on Voronoi diagrams with isotropic [Martínez et al. 2016] or orthotropic [Martínez et al. 2017] elastic behavior. In this context, the work of Martínez et al. [2019], who study planar Voronoi Diagrams with star-shaped metrics, shares the most similarities with ours—we likewise use star-shaped metrics for extended control over cell shapes and achieve smoothly varying patterns through metric interpolation. However, whereas Martínez et al. construct their diagrams using a discrete rasterization approach, we propose a closed-form algorithm that produces accurate and fully differentiable representations of cell boundaries. We furthermore introduce a generalization to the three-dimensional setting and show how this formulation can be used to generate diagrams on 3D surfaces represented by triangle meshes.

Voronoi Diagrams. Their flexibility with respect to topology changes makes Voronoi diagrams an attractive representation for many applications, including simulation of fluids and foams [Busaryev et al. 2012; de Goes et al. 2015; Hong et al. 2008; Numerow et al. 2024] as well as structural design and optimization [Feng et al. 2022; Lumpe et al. 2022; Zhang et al. 2017]. Among the many variations, Centroidal Voronoi tesselations (CVT) are the most widely used class [Alliez et al. 2005; Lévy and Liu 2010]. Many different methods exist for generating such diagrams, ranging from the classical algorithm by Lloyd [1982] to gradient-based and quasi-Newton methods [Liu et al. 2009; Wang et al. 2015; Yan et al. 2009a, 2011]. Our method is arguably most similar to the recent work by Numerow et al. [2024] which provides analytical derivatives of vertex positions with respect to site locations. However, while Numerow et al. focus on standard metrics, we introduce the first closed-form construction of Voronoi diagrams with non-standard, star-shaped metrics.

Many of the aforementioned methods and formulations are generic in the sense that they will, in principle, work in spaces of arbitrary dimension. A special case, however, are methods that target the construction of Voronoi diagrams on discrete surfaces embedded in 3D-space [Alliez et al. 2005; Li et al. 2024; Liu et al. 2010]. Our formulation directly extends to this embedded setting, enabling us to generate on-surface diagrams through strategic site placement and metric alignment in the ambient space.

3 Method

In this section, we introduce the concepts of Voronoi diagrams and star-shaped metrics before detailing our closed-form construction algorithm, along with various important considerations for generating high-quality star-shaped distance Voronoi diagrams in 3D.

3.1 Generalized Voronoi Diagrams and Star-Shaped Metrics

The Voronoi cell $\mathcal{R}_i$ for a given site c_i can be defined as

$$\mathcal{R}_i = \{\mathbf{p} : d_i(\mathbf{p}) < d_j(\mathbf{p}) \ \ \forall j \neq i\}\,, \tag{1}$$

where $d_i(\mathbf{p}) = \|\mathbf{p} - \mathbf{c}_i\|$ is the Euclidean distance between point $\mathbf{p}$ and site $\mathbf{c}_i$. Using alternative distance metrics $d_i(\mathbf{p})$ to determine the Voronoi cell leads to so-called *generalized Voronoi diagrams*.

Martínez et al. [2019] define the *star-shaped distance* to a site $\mathbf{c}_i$ using a corresponding star-shaped set $\mathcal{M}_i \subset \mathbb{R}^n$ containing the origin. $\mathcal{M}_i$ is *star-shaped* if for all points $\mathbf{p} \in \partial\mathcal{M}_i$ on the set boundary, the line segment connecting $\mathbf{p}$ to the origin is fully contained in the interior of $\mathcal{M}_i$. $\mathcal{M}_i$ is analogous to the unit ball for Euclidean distance; a point $\mathbf{p} \in (\mathbf{c}_i + \partial\mathcal{M}_i)$ has unit distance to the site. Likewise, a point $\mathbf{p} \in (\mathbf{c}_i + \alpha\partial\mathcal{M}_i)$ has distance $d_i(\mathbf{p}) = \alpha$ to the site.

3.2 Closed-Form Construction Overview

Polygonal or polyhedral metric sets $\mathcal{M}_i$ give rise to piecewise linear star-shaped distances. We provide in this paper a closed-form construction of Voronoi diagrams using piecewise linear star-shaped distances, both in 2D and on surfaces in 3D space. The inputs to the algorithm are:

(a) The position of each Voronoi site $\mathbf{c}_i$.
(b) The corresponding *metric polytope* $\partial\mathcal{M}_i$, defined by its n_i *metric vertices* $\{\mathbf{m}_i^j : j = 0 \cdots n_i - 1\}$ and, if in three dimensions, their triangulation. The metric polytope must be star-shaped.
(c) The domain, which is either a 2D polygonal domain or a surface mesh in 3D space. Thin triangles, holes, non-manifold and disconnected meshes are permitted. Collinear triangles are not.

Our closed-form construction hinges on the observation that the space around a site $\mathbf{c}_i$ can be partitioned into generalized *cones* $\mathcal{K}_i^j$ (see Figure 2). Cone $\mathcal{K}_i^j$ is the union of all rays extending from $\mathbf{c}_i$ and passing through $\mathbf{c}_i + \mathcal{F}_i^j$, where $\mathcal{F}_i^j$ is a face of the metric polytope $\mathcal{M}_i$. The star-shaped distance within cone $\mathcal{K}_i^j$ is

$$d_i^j(\mathbf{p}) = \frac{\mathbf{n}_i^j \cdot (\mathbf{p} - \mathbf{c}_i)}{b_i^j}\,, \tag{2}$$

where $\mathbf{n}_i^j \cdot \mathbf{x} + b_i^j = 0$ defines the hyperplane containing $\mathcal{F}_i^j$. The critical observation is that the bisector (Figure 2) of sites $\mathbf{c}_0$ and $\mathbf{c}_1$ restricted to the intersection of cones $\mathcal{K}_0^j \cap \mathcal{K}_1^k$ is also linear, which follows from Eqn. 2 by setting $d_0^j(\mathbf{x}) = d_1^k(\mathbf{x})$,

$$\left(b_1^k \mathbf{n}_0^j - b_0^j \mathbf{n}_1^k\right) \cdot \mathbf{x} + \left(b_0^j \mathbf{n}_1^k \cdot \mathbf{c}_1 - b_1^k \mathbf{n}_0^j \cdot \mathbf{c}_0\right) = 0\,. \tag{3}$$

The complete bisector of $\mathbf{c}_0$ and $\mathbf{c}_1$ is piecewise linear, with vertices at transitions between cones where the star-shaped distance is itself only C^0-continuous.

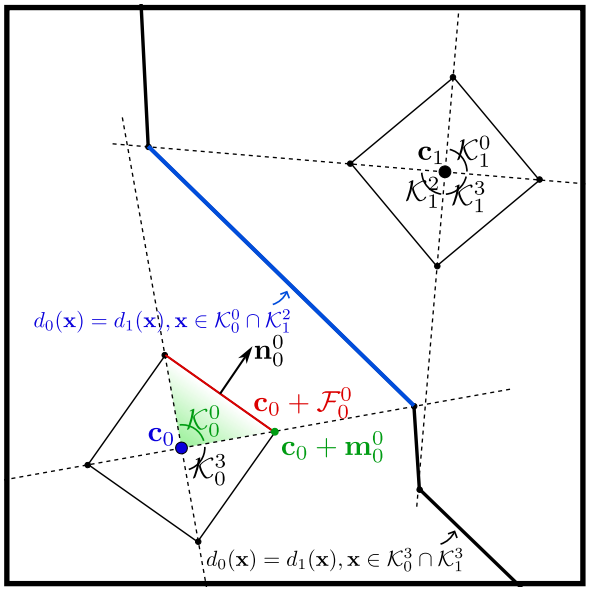


Fig. 2. Voronoi diagram of two sites with quadrilateral metric polygons and the partitioning of space around each site into 2D cones. The bisector is piecewise linear with corners only at cone transitions. Colors are for label clarity.

The bisector is, by definition, equidistant to sites $\mathbf{c}_0$ and $\mathbf{c}_1$, and exists in the completed Voronoi diagram only where it is closest to sites $\mathbf{c}_0$ and $\mathbf{c}_1$ among all sites. Consequently, segments of the bisector where a third site $\mathbf{c}_2$ is closer, such that $d_2(\mathbf{x}) < d_0(\mathbf{x}) =$

$d_1(\mathbf{x})$, are excluded from the diagram. If no segment of the bisector remains, $\mathbf{c}_0$ and $\mathbf{c}_1$ are not adjacent in the Voronoi diagram. The following sections 3.3 and 3.4 present an efficient algorithm for constructing star-shaped distance Voronoi diagrams in both 2D and 3D settings.

3.3 Voronoi Vertices

The geometry of a Voronoi diagram is defined by its *Voronoi vertices* and the edges connecting them. This section describes the method for computing Voronoi vertices and their derivatives.

Any Voronoi vertex in m dimensions can be computed as the intersection of m hyperplanes, each of which is a Voronoi bisector, a separator of two cones, or a facet of the bounding geometry. Voronoi bisectors are given by Eqn. 3. A cone separator in 2D is the line containing $\mathbf{c}_i$ and a metric point $\mathbf{c}_i + \mathbf{m}_i^j$. For example, in Figure 2, the cone separator $\mathcal{K}_0^0\mathcal{K}_0^3$ contains points $\mathbf{c}_0$ and $\mathbf{c}_0 + \mathbf{m}_0^0$. In 3D, a cone separator is the plane containing $\mathbf{c}_i$ and the two points of a metric edge $\mathbf{c}_i + \mathbf{m}_i^j$ and $\mathbf{c}_i + \mathbf{m}_i^k$.

After determining the system of line or plane equations which determine a vertex position, we solve it in closed-form. This is equivalent to inverting a 2×2 or 3×3 matrix, for which the closed-form solution is readily available. Our implementation obtains the expressions for vertex positions, and their derivatives with respect to site positions and metric vertices, using a computer algebra system.

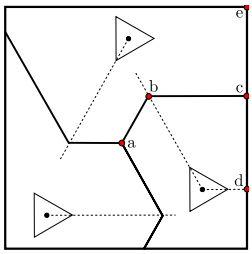


Fig. 3. Voronoi diagram with labeled vertex types: a) Bisector Junction, b) Cone Transition, c) Boundary Bisector, d) Boundary Cone Transition, e) Boundary Corner.

We define here the five types of Voronoi vertices for the 2D and surface-restricted settings. Each vertex is labeled with a unique identifier; a tuple of inputs determining its position. Elements in parentheses are used only in 3D.

Bisector Junction Vertex. Junction of three Voronoi cells corresponding to sites $\mathbf{c}_i$, $\mathbf{c}_j$, $\mathbf{c}_k$ as found in standard Voronoi diagrams. Identified by $[\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{K}_k^r, (\mathcal{B}_\ell)]$, where $\mathcal{K}$ are cones from each site and $\mathcal{B}_\ell$ is the surface mesh facet that contains the vertex. Computed as the intersection of bisectors $\mathcal{K}_i^p\mathcal{K}_j^q$ and $\mathcal{K}_i^p\mathcal{K}_k^r$, and (in 3D) the plane containing boundary facet $\mathcal{B}_\ell$.

Cone Transition Vertex. Midpoint on bisector of $\mathbf{c}_i$ and $\mathbf{c}_j$ which occurs at the transition between two cones $\mathcal{K}_i^p$ and $\mathcal{K}_i^q$ of site $\mathbf{c}_i$, identified by $[\mathcal{K}_i^p, \mathcal{K}_i^q, \mathcal{K}_j^r, (\mathcal{B}_k)]$. Computed as the intersection of the cone separator $\mathcal{K}_i^p\mathcal{K}_i^q$, bisector $\mathcal{K}_i^p\mathcal{K}_j^r$, and (in 3D) the plane containing boundary facet $\mathcal{B}_k$.

Boundary Bisector Vertex. Intersection of bisector $\mathcal{K}_i^p\mathcal{K}_j^q$ with a boundary edge $\mathscr{E}_k$, identified by $[\mathcal{K}_i^p, \mathcal{K}_j^q, \mathscr{E}_k]$.

Boundary Cone Transition Vertex. Point on boundary edge $\mathscr{E}_k$ at the transition between two cones $\mathcal{K}_i^p\mathcal{K}_i^q$ of site $\mathbf{c}_i$, identified by $[\mathcal{K}_i^p, \mathcal{K}_i^q, \mathscr{E}_k]$. These are not true vertices in the Voronoi diagram, as they join collinear boundary segments, but are used in the construction algorithm.

Boundary Corner Vertex. Vertex of the bounding geometry, trivially identified by the vertex itself $[\mathbf{b}_i]$.

To show that this set of vertex types is complete, we use the fact that each Voronoi vertex in 2D is the intersection of two lines. These lines can be Voronoi bisectors (V), cone separators (M) or boundary edges (B). As shown in Table 1, each possible combination of two line types results in a different vertex type, except the intersection of two cone separators. This is because cone separators, unlike Voronoi bisectors and boundary edges, do not generate edges in the Voronoi diagram.

Table 1. Vertex type resulting from each possible combination of intersecting lines. No vertex results from the intersection of two cone separators (M), as cone separators are not Voronoi edges in our formulation.

V	M	B	Vertex Type
2			Bisector Junction
1	1		Cone Transition
1		1	Boundary Bisector
	2		-
	1	1	Boundary Cone Transition
		2	Boundary Corner

The same vertex types are present in the surface-restricted case, as can be understood from the fact that this construction decomposes into per-mesh-triangle 2D constructions. More concretely, the two generators of a vertex (bisectors, cone separators and/or boundary facets) are now planes and intersect to form lines; vertices are the intersections of these lines with an additional mesh facet.

Differentiability. Because the positions of Voronoi vertices completely define the geometry of the Voronoi diagram, differentiability of the construction is equivalent to differentiability of the vertex positions with respect to the algorithm inputs. While the derivatives of a given Voronoi vertex can always be computed, the construction is generally not differentiable across topology changes, where certain vertices disappear from the diagram and are replaced by new ones. However, some quantities defined on the Voronoi diagram, such as cell volumes, are differentiable even across topology changes; when this is the case, their derivatives can be computed in the standard way from the derivatives of vertex positions.

3.4 Detailed Algorithm

Our algorithm builds the Voronoi diagram incrementally by traversing known Voronoi edges. We maintain a queue of *traces*, representing Voronoi vertex-edge pairs where the opposite endpoint of the edge is unknown. For each trace, we search for the opposite endpoint of the Voronoi edge, and initialize new traces from this vertex if it had not been previously visited. Given appropriate initialization, this algorithm terminates when the entire domain has been traversed and the Voronoi diagram is complete. The processes for tracing boundary edges and Voronoi bisectors are illustrated in Figures 4 and 5 respectively, and described in detail in the following paragraphs.

Initialization. For a 2D Voronoi diagram restricted to a polygonal domain, the complete bounding polygon is guaranteed to exist in the diagram. We therefore initialize the tracing queue with a trace from a boundary vertex along a boundary edge. For 3D surfaces, all edges in the surface mesh are treated as boundary edges. Therefore, for the purposes of the construction algorithm, all mesh vertices and edges are added to the Voronoi diagram. They are later removed in post-processing, except for outer edges of non-closed surface meshes. The tracing queue is initialized with boundary traces along all mesh edges, ensuring that the resulting Voronoi diagram covers the entire mesh even in case of disjoint mesh topology.

Tracing. A trace of a boundary edge is identified by $\{\mathcal{K}, \ell, \mathcal{V}\}$, where $\mathcal{K}$ is the cone (of the nearest site) containing the start vertex identified by $\mathcal{V}$, and ℓ is the boundary edge. Similarly, a trace of a Voronoi bisector edge is identified by $\{\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{V}, (\mathcal{B})\}$. When a trace $\mathcal{T}$ is removed from the queue, it is processed as follows. Let $\mathbf{p}$ represent the position of the start vertex $\mathcal{V}$ of the trace. The edge being traced can be defined as $\mathbf{p} + t\mathbf{d}$, $t > 0$, for a suitable direction vector $\mathbf{d}$. We search over a list of candidate endpoints for the trace and find the endpoint $\mathcal{V}'$ which minimizes t, $t > 0$. We add vertex $\mathcal{V}'$ and edge $\mathcal{V}\mathcal{V}'$ to the Voronoi diagram. Candidate endpoints for a boundary trace $\mathcal{T} = \{\mathcal{K}_i^p, \ell, \mathcal{V}\}$ are:

(a) An end of the boundary edge (a boundary corner vertex). In this case, add a trace $\{\mathcal{K}_i^p, \ell', \mathcal{V}'\}$ to the queue for each boundary edge $\ell' \neq \ell$ adjacent to $\mathcal{V}'$.
(b) The intersection of the edge with cone boundary $\mathcal{K}_i^p\mathcal{K}_i^q$ for any adjacent cone $\mathcal{K}_i^q$ (a boundary cone transition vertex $[\mathcal{K}_i^p, \mathcal{K}_i^q, \ell]$). Add trace $\{\mathcal{K}_i^q, \ell, \mathcal{V}'\}$.
(c) The intersection of the edge with bisector $\mathcal{K}_i^p\mathcal{K}_j^q$ for any cone from another site $\mathcal{K}_j^q$ (a boundary bisector vertex $[\mathcal{K}_i^p, \mathcal{K}_j^q, \ell]$). Add traces $\{\mathcal{K}_j^q, \ell, \mathcal{V}'\}$ and $\{\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{V}'\}$.

Candidate endpoints for a bisector trace $\mathcal{T} = \{\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{V}\}$ are:

(a) The intersection of the edge with a boundary edge ℓ (a boundary bisector vertex $[\mathcal{K}_i^p, \mathcal{K}_j^q, \ell]$). Add traces $\{\mathcal{K}_i^p, \ell, \mathcal{V}'\}$ and $\{\mathcal{K}_j^q, \ell, \mathcal{V}'\}$.
(b) The intersection of the edge with cone boundary $\mathcal{K}_i^p\mathcal{K}_i^r$ (a cone transition vertex $[\mathcal{K}_i^p, \mathcal{K}_i^r, \mathcal{K}_j^q]$). Add trace $\{\mathcal{K}_i^r, \mathcal{K}_j^q, \mathcal{V}'\}$.
(c) The intersection of the edge with cone boundary $\mathcal{K}_j^q\mathcal{K}_j^r$ (a cone transition vertex $[\mathcal{K}_j^q, \mathcal{K}_j^r, \mathcal{K}_i^p]$). Add trace $\{\mathcal{K}_i^p, \mathcal{K}_j^r, \mathcal{V}'\}$.
(d) The intersection of the edge with bisector $\mathcal{K}_i^p\mathcal{K}_k^r$ (a bisector junction vertex $[\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{K}_k^r]$). Add traces $\{\mathcal{K}_i^p, \mathcal{K}_k^r, \mathcal{V}'\}$ and $\{\mathcal{K}_j^q, \mathcal{K}_k^r, \mathcal{V}'\}$. Note that testing for bisectors $\mathcal{K}_j^q\mathcal{K}_k^r$ would be redundant here, as they would intersect the current trace at the same bisector junction vertex.

The overall number of traces (equivalently, Voronoi edges) is proportional to the number of sites and the degree of their metric polygons. Cases (c) for boundary tracing and (d) for bisector tracing require iterating over all other sites and their respective cones. The overall runtime of the Voronoi diagram construction is therefore quadratic with respect to the number of sites and metric polygon vertices. In practice, we use bounding volume hierarchies of cones to reduce the set of candidate endpoints in the bottleneck cases.

The tracing process is identical in 2D and 3D, except for the fact that bisectors in 3D are specific to a mesh facet, which is included in trace and non-boundary vertex identifiers. For a boundary trace $\mathcal{T} = \{\mathcal{K}_i^p, \ell, \mathcal{V}\}$, only bisectors within the mesh facets adjacent to ℓ are candidate endpoints. For a bisector trace $\mathcal{T} = \{\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{V}, \mathcal{B}\}$, only boundary edges adjacent to $\mathcal{B}$ and bisectors within $\mathcal{B}$ are candidate endpoints.

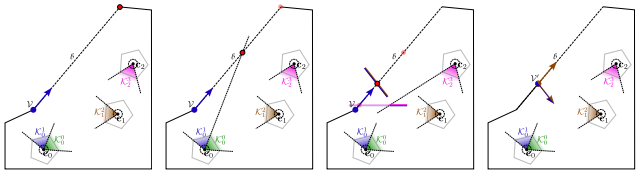


Fig. 4. Tracing a boundary edge to find the next Voronoi vertex. From left to right: a) Initialize trace $\{\mathcal{K}_0^1, \ell, \mathcal{V}\}$. Default endpoint is the end of the boundary edge. b) Closer endpoint found at cone boundary $\mathcal{K}_0^1\mathcal{K}_0^0$. c) Bisector $\mathcal{K}_0^1\mathcal{K}_2^3$ endpoint rejected because the intersection with the current trace is outside cone $\mathcal{K}_2^3$. Bisector $\mathcal{K}_0^1\mathcal{K}_1^2$ endpoint is the new closest endpoint. d) New boundary bisector vertex $\mathcal{V}' = [\mathcal{K}_0^1, \mathcal{K}_1^2, \ell]$ confirmed. Further boundary trace $\{\mathcal{K}_1^2, \ell, \mathcal{V}'\}$ and new bisector trace $\{\mathcal{K}_0^1, \mathcal{K}_1^2, \mathcal{V}'\}$ added.

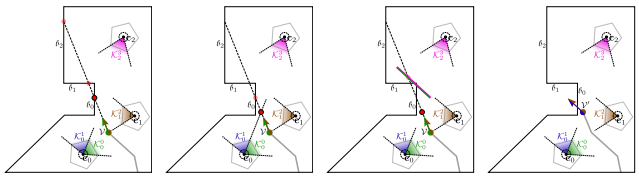


Fig. 5. Tracing a Voronoi bisector to find the next Voronoi vertex. From left to right: a) Initialize trace $\{\mathcal{K}_0^0, \mathcal{K}_1^2, \mathcal{V}\}$. Intersection with boundary edge ℓ_0 accepted as endpoint. Other boundary intersections are farther and therefore rejected. b) Cone boundary $\mathcal{K}_0^0\mathcal{K}_0^1$ accepted as new endpoint. c) Bisector $\mathcal{K}_0^0\mathcal{K}_2^3$ endpoint rejected, farther than current endpoint. d) New cone transition vertex $\mathcal{V}' = [\mathcal{K}_0^0, \mathcal{K}_0^1, \mathcal{K}_1^2]$ confirmed. New bisector trace $\{\mathcal{K}_0^1, \mathcal{K}_1^2, \mathcal{V}'\}$ added.

Edge Direction. The direction vector $\mathbf{d}$ for a trace is either along the boundary edge (for boundary traces) or along the bisector line (Eqn. 3, for bisector traces), but this determines the direction only up to a change in sign. In cases where the trace begins at a boundary corner vertex or at any cone transition vertex, the sign is determined by the requirement to point along the boundary edge or into the cone. Similarly, a bisector trace from a boundary bisector vertex must point into the domain (or the corresponding mesh triangle, in 3D). Otherwise, we can choose between the two opposing directions by considering the cones which determine a trace and its start vertex.

For a boundary trace $\mathcal{T} = \{\mathcal{K}_i^p, \ell, \mathcal{V}\}$ from boundary bisector vertex $\mathcal{V} = [\mathcal{K}_i^p, \mathcal{K}_j^q, \ell]$, the correct direction satisfies

$$\frac{\mathbf{d} \cdot \mathbf{n}_i^p}{|b_i^p|} < \frac{\mathbf{d} \cdot \mathbf{n}_j^q}{|b_j^q|}. \tag{4}$$

with $\mathbf{n}$ and b defined as in Eqn. 3. Similarly, for a bisector trace $\mathcal{T} = \{\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{V}\}$ from bisector junction vertex $\mathcal{V} = [\mathcal{K}_i^p, \mathcal{K}_j^q, \mathcal{K}_k^r]$,

the correct direction satisfies

$$\frac{\mathbf{d}\cdot\mathbf{n}_i^p}{|b_i^p|} = \frac{\mathbf{d}\cdot\mathbf{n}_j^q}{|b_j^q|} < \frac{\mathbf{d}\cdot\mathbf{n}_k^r}{|b_k^r|}. \tag{5}$$

3.5 Removing Disconnected Components

Voronoi diagrams using star-shaped distances have the property that cells may consist of multiple disconnected components, often resulting in unique and aesthetic tilings. However, structures intended for fabrication must be without disconnected curves, and designers may otherwise wish to ensure connected cells. Martínez et al. [2019] use a growth model to avoid disconnected components, though their approach does not lend itself to our closed-form construction.

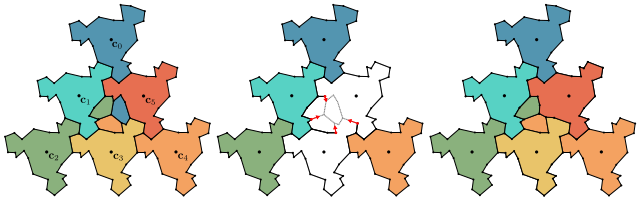


Fig. 6. Illustration of our procedure for disconnected component removal. *Left:* A star-shaped distance Voronoi diagram containing disconnected components from $\mathbf{c}_0$, $\mathbf{c}_2$ and $\mathbf{c}_4$ in the center. *Middle:* We remove the disconnected component from $\mathbf{c}_0$ and create new traces (red arrows) along all incoming edges. From here, we apply the tracing algorithm from section 3.4, including only sites $\mathbf{c}_2$, $\mathbf{c}_3$, $\mathbf{c}_4$ and $\mathbf{c}_5$, which were adjacent to the removed component. *Right:* Reconstructed Voronoi diagram after removing the disconnected component. The process repeats until no disconnected components remain.

We propose instead an exact algorithm for eliminating disconnected components from an existing Voronoi diagram. As illustrated in Figure 6, our method removes disconnected components (*i.e.* components that do not contain their sites) one by one, reconstructing the Voronoi diagram locally after each removal. This is done by applying the tracing algorithm from section 3.4 along the edges which previously connected to the removed component. However, during tracing we only consider bisectors between sites whose cells shared an edge with the removed component. The removal of one disconnected component may generate one or more new disconnected components, even multiple from the same cell, but the recursive removal of disconnected components is guaranteed to terminate.

Proof. Consider a disconnected component from cell $\mathcal{R}_n$, neighboring n other cells $\mathcal{R}_0 \dots \mathcal{R}_{n-1}$. Removing this component may result in a new disconnected component from, *e.g.*, cell $\mathcal{R}_{n-1}$. At most, this new disconnected component has $n-1$ neighboring cells $\mathcal{R}_0 \dots \mathcal{R}_{n-2}$. Therefore, any disconnected component resulting from the removal of a previous component has strictly fewer neighbors than the removed one, and the recursive removal of disconnected components terminates after at most n steps. To prove termination, it remains to be shown only that the number of disconnected components is bounded. As discussed in section 3.3, each Voronoi vertex is uniquely defined by a combination of three cones and boundary edges. There is a finite number of such combinations, and therefore a finite number of possible Voronoi vertices. Each cell component is a loop of Voronoi vertices, and a graph with finite vertices has a finite number of cycles. Therefore, the number of possible (disconnected) components is finite, and the process terminates.

In practice, it is rare that any new disconnected components are created during removal, and the process is much faster than the initial construction of the diagram. Figure 7 shows the result of removing disconnected components from a tangled Voronoi diagram. It should be noted that the resulting tessellation is not a true Voronoi diagram with respect to the star-shaped distances. However, all vertices belong to the types presented in Section 3.3, and the construction remains differentiable.

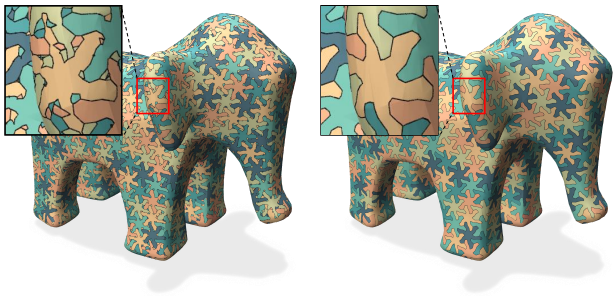


Fig. 7. Removing disconnected components on an elephant. While this tiling is particularly tangled, surface-restricted star-shaped distance Voronoi diagrams typically have a small number of disconnected components.

3.6 3D Lattice and Metric Construction

Extruded Metric Polyhedra. Generating quasi-uniform tilings on 3D surfaces requires careful construction of the metric polytopes. Firstly, while any star-shaped polyhedron is a valid metric polytope, most will not result in uniform tilings when the 3D Voronoi diagram is restricted to a surface mesh. For this reason, we use *extruded metrics*, *i.e.*, 2D star polygons extruded in the normal direction of the surface mesh, for all 3D results. Figure 8 illustrates this concept, and demonstrates that extruded metrics achieve more uniform tilings compared to polyhedral metrics with fixed orientation.

Fig. 8. Comparison of a fixed-orientation tetrahedral metric (left) with a surface-aligned triangular extruded metric (right). For each, a visualization of site locations and metric polyhedra is shown along with the resulting Voronoi diagram. The extruded metric enables consistent shape of Voronoi cells around the cylinder, where cells in the fixed metric example become distorted due to misalignment.

3D Lattice Generation. Most regular tilings, including triangular and square lattices, cannot exist without singularities on most 3D surfaces. To see this, we consider the Euler characteristic $\chi = V - E + F$, where V, E and F are respectively the numbers of vertices, edges and faces in a tessellation. The triangular lattice has $E = 3V$ and $F = 2V$, while the square lattice has $E = 2V$ and $F = V$. This leads to a requirement of $\chi = 0$ for both lattice types, which is true only for genus-1 surfaces such as the torus. Spheres and other genus-0 surfaces have $\chi = 2$, making these regular tilings impossible without singularities.

To obtain nearly-regular site distributions, we utilize the method of Knöppel *et al.* [2015] (chosen arbitrarily; there are likely many suitable methods) for generating stripe patterns on surfaces. The intersection of two stripe patterns yields a point set determining the sites of the Voronoi diagram. Intersecting the stripe pattern generated from a smoothest direction field [Knöppel et al. 2013] with one generated from the same direction field rotated by $\frac{\pi}{2}$ yields a rectangular lattice; rotating by $\frac{\pi}{3}$ gives a triangular lattice. We additionally use the smoothest direction field to initialize the orientation of extruded metric polyhedra on the surface.

In addition to topological singularities, irregularities in site placements generally translate into tile shape variations that can degrade the visual quality of the tiling. We find that triangular lattice site distributions can be further improved by optimizing to a restricted centroidal Voronoi tessellation using the method of Yan *et al.* [2009b]. The resulting improvement is demonstrated in Figure 10. This optimization is performed on the standard Voronoi diagram with Euclidean distance, and works because CVT optimization tends to produce triangular lattices.

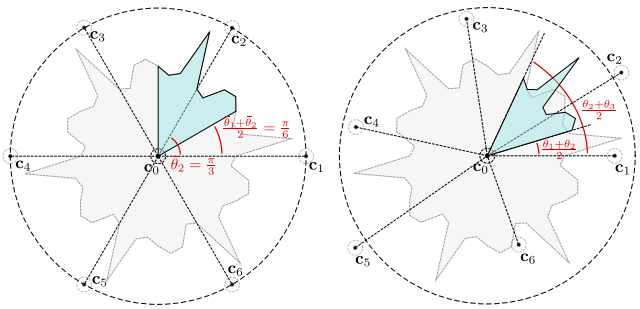


Fig. 9. Alignment of metric polygon sector to neighboring sites. *Left:* sector of original metric polygon, assuming a triangular lattice. *Right:* sector adjusted based on locations of neighboring sites.

Metric Alignment. The patterns generated by star-shaped distance Voronoi diagrams depend heavily on the alignment of metric polygon features with neighboring sites; rotating the metric polygons while maintaining site positions often results in completely different tilings. Because of this, we must take care to ensure that metric features are well-aligned despite lattice imperfections on surfaces. Consider a site $\mathbf{c}_0$ with six lattice neighbors $\mathbf{c}_1 \dots \mathbf{c}_6$ in counterclockwise order and original metric polygon $r_0(\theta)$ defined as a function of angle θ. θ is initially mapped to an angle in the tangent plane counterclockwise from $\mathbf{c}_1$. Let $\theta_1 \coloneqq 0$ and $\theta_k, k = 2 \dots 6$ represent the angle of $\mathbf{c}_k$. In a regular triangular lattice we have $\theta_k = \bar{\theta}_k \coloneqq \frac{\pi}{3}(k-1)$. To align the appropriate *sector* (Figure 9) of the metric with the appropriate neighbor, we take each sector $r_0(\theta), \bar{\theta}_{k-\frac{1}{2}} < \theta < \bar{\theta}_{k+\frac{1}{2}}$, and map it to the interval $\theta_{k-\frac{1}{2}} < \theta < \theta_{k+\frac{1}{2}}$ to obtain a modified metric polygon $r'_0(\theta)$, where $\theta_{k\pm\frac{1}{2}} \coloneqq (\theta_k + \theta_{k\tilde{\pm}1})/2$ and $\tilde{\pm}$ denotes addition or subtraction modulo $n = 6$. Figure 10 shows the resulting improvement to the Voronoi diagram.

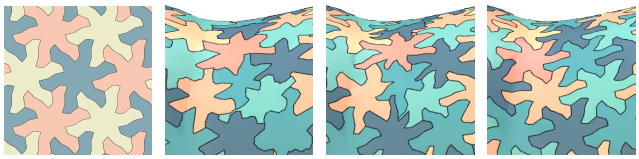


Fig. 10. Metric alignment techniques significantly improve surface-restricted Voronoi diagrams. From left to right: regular pattern in 2D; result using extruded metric on surface without alignment; result after metric alignment; result after CVT optimization and metric alignment.

Lattice Irregularity. For metric polygons with six-fold rotational symmetry and sites arranged in a triangular lattice, the sector of the metric polygon pointing towards each neighboring vertex is identical. Here, handling lattice singularities is straightforward; a site with $n \neq 6$ neighbors can be given a modified metric with n-fold rotational symmetry composed of the same sector.

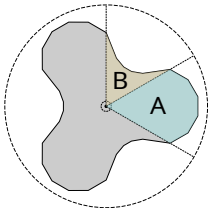


Fig. 11. 3-fold rotational metric polygon with sector labels.

However, metrics with distinct sectors pointing toward different neighbors require more careful treatment. Consider a metric with 3-fold rotational symmetry producing a ball-and-socket pattern as shown in Figure 11. The metric polygon can be divided into A (ball) and B (socket) sectors. To maintain the character of the tessellation throughout, if site $\mathbf{c}_i$ has metric sector A pointing to neighbor $\mathbf{c}_j$, $\mathbf{c}_j$ should have metric sector B pointing to $\mathbf{c}_i$. Additionally, adjacent sectors of the same metric polygon should be opposite. These constraints cannot be satisfied in general, *e.g.* sites with an odd number of neighbors must have at least one pair of adjacent sectors of the same type. To satisfy the constraints as well as possible for a given site distribution, we solve the mixed integer programming problem,

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} \left(c_1 \sum \mathbb{1}_{y_{ij}=y_{ji}} + c_2 \sum \mathbb{1}_{y_i^k = y_i^{k\tilde{+}1}} \right), y \in [\mathrm{A}, \mathrm{B}] \,, \quad (6)$$

to assign optimal A/B labels. Here, y_{ij} is the binary label assigned to the sector of $\mathcal{M}_i$ pointing towards $\mathbf{c}_j$, and y_i^k denotes the label assigned to the sector of $\mathcal{M}_i$ pointing towards the kth neighbor of $\mathbf{c}_i$, in counterclockwise order. Constants c_1 and c_2 can be tuned to visually balance the constraint violations.

3.7 Implementation

Our method is implemented in C++ using the CGAL library [2024]. We use the Eigen library [Guennebaud et al. 2010] for matrix and vector operations, the CHOLMOD solver [Chen et al. 2008] for linear systems, Gurobi [2024] for mixed integer programming, and the

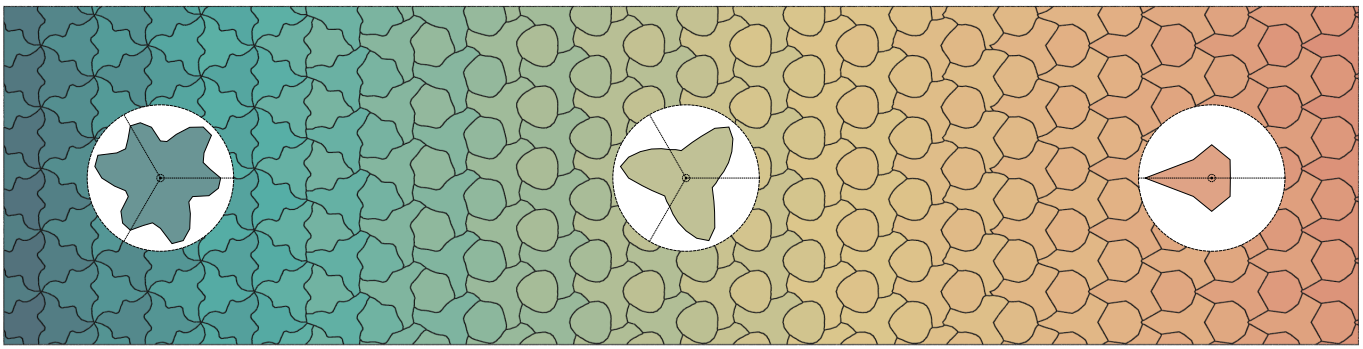


Fig. 12. Graded star-shaped distance Voronoi diagram on a 2D honeycomb lattice. The continuous grading is achieved by linear interpolation between the three metric polygons shown in the inset circles. Voronoi edges are piecewise linear and positions of all vertices are computed in closed form.

NLopt library [Johnson 2007] for the Method of Moving Asymptotes. Geometry processing libraries Geometry Central [Sharp et al. 2019a] and Geogram [Lévy 2015] are used for stripe patterns and centroidal Voronoi tessellations, respectively. Finally, we use Polyscope [Sharp et al. 2019b] for visualization and rendering of figures. Our code can be found at https://github.com/zhou-haoyang/ssmvd.

4 Results

4.1 2D Interpolated Tilings

We demonstrate the effectiveness of our closed-form construction algorithm in generating a range of homogeneous as well as graded tilings using star-shaped distance Voronoi diagrams. The two-dimensional design in Figure 12 shows that, similar to the approach of Martínez et al. [2019], our method can interpolate between star-shaped distance metrics to generate continuously-graded patterns. Though sometimes smooth in appearance, the Voronoi edges are piecewise linear with the positions of all vertices computed in closed form.

4.2 3D Tilings on Surfaces

Our approach extends naturally to higher dimensions, which we demonstrate using surface-restricted 3D star-shaped distance Voronoi diagrams. Figure 13 shows three examples of vases patterned using surface-restricted Voronoi diagrams. Two of these examples exhibit tilings with disconnected components, highlighting the depth of the design space enabled by star-shaped distance Voronoi diagrams. Because this surface topology permits site placement in a regular lattice, we are able to utilize different lattice types and metrics with different types of rotational and reflection symmetry.

Table 2 shows restricted star-shaped distance Voronoi diagrams on more general surface meshes for a variety of metrics. For these examples, we use the lattice generation and metric alignment techniques described in Section 3.6 and remove disconnected components as discussed in Section 3.5. For the examples in the second row, using a metric with 3-fold rotational symmetry, we modify the metric polyhedra to limit the effects of lattice irregularity by solving the mixed-integer program in Eqn. 6. Some visual artifacts remain due to lattice irregularity, particularly in the 3-fold rotational symmetry examples, and due to the difference in surface orientation between neighboring sites on high-curvature features such as bunny ears.

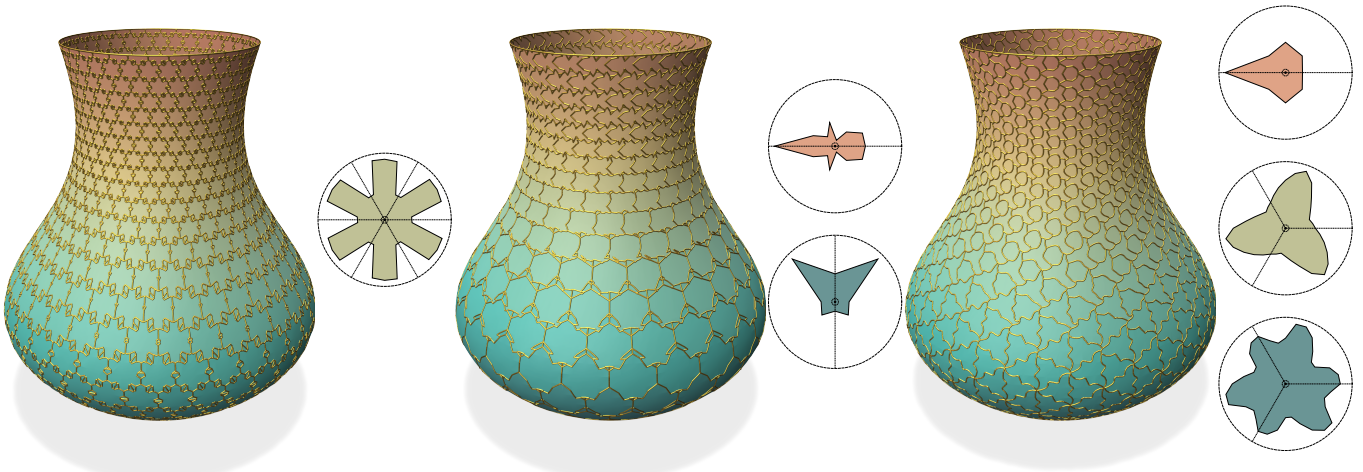


Fig. 13. Star-shaped distance Voronoi diagrams on vases in 3D. These designs exhibit disconnected components (*left*, *middle*) and interpolation between different star-shaped distance metrics (*middle*, *right*).

Table 2. Star-shaped distance Voronoi diagrams on general 3D surfaces. To obtain high-quality tilings, we apply the lattice generation and metric alignment techniques described in Section 3.6 and remove disconnected components as discussed in Section 3.5. *2nd row:* metric with 3-fold rotational symmetry, requiring optimization to align metric sectors. *4th row:* interpolation between star-shaped and Euclidean distance metrics.

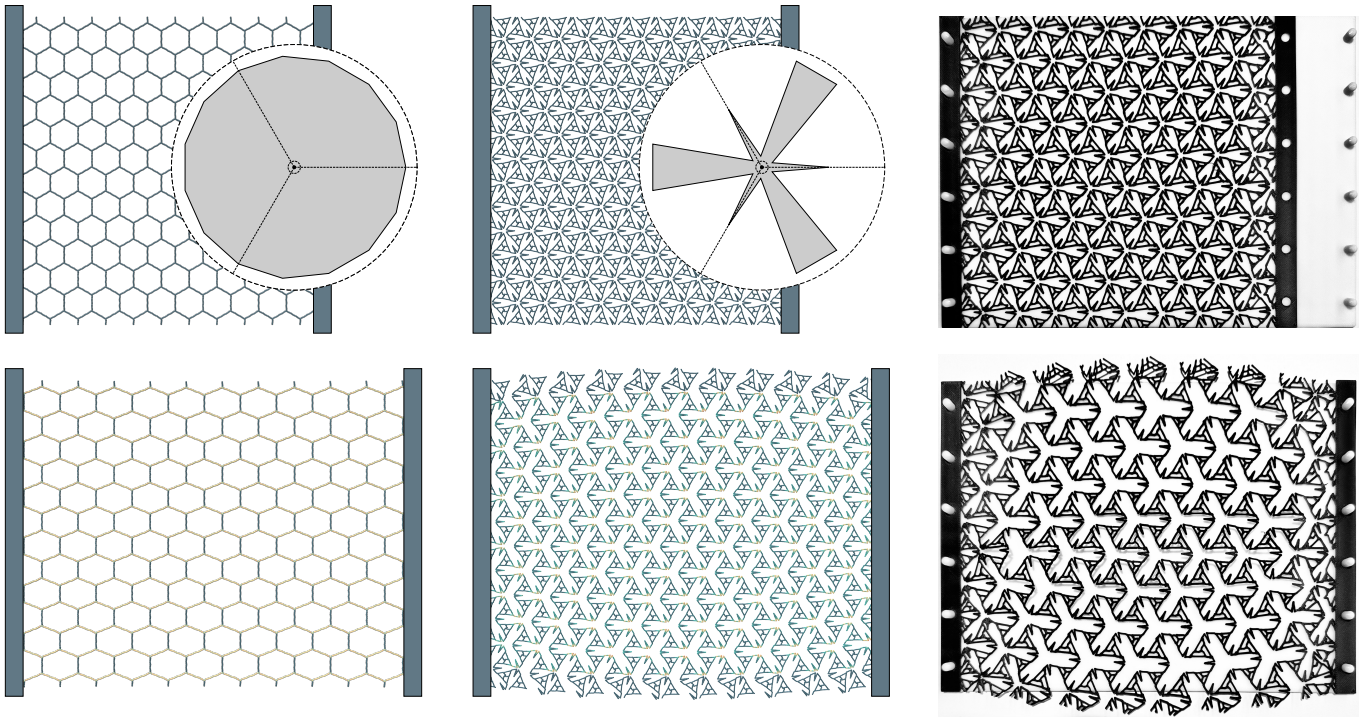
4.3 Mechanical Optimization for Structured Sheet Materials

Our differentiable construction enables gradient-based optimization of tilings with respect to high-level objectives. We demonstrate this approach by designing a set of mechanical metamaterials based on star-shaped distance Voronoi diagrams. Specifically, we consider structured sheet materials obtained by generating a rod network on the edges of a Voronoi diagram.

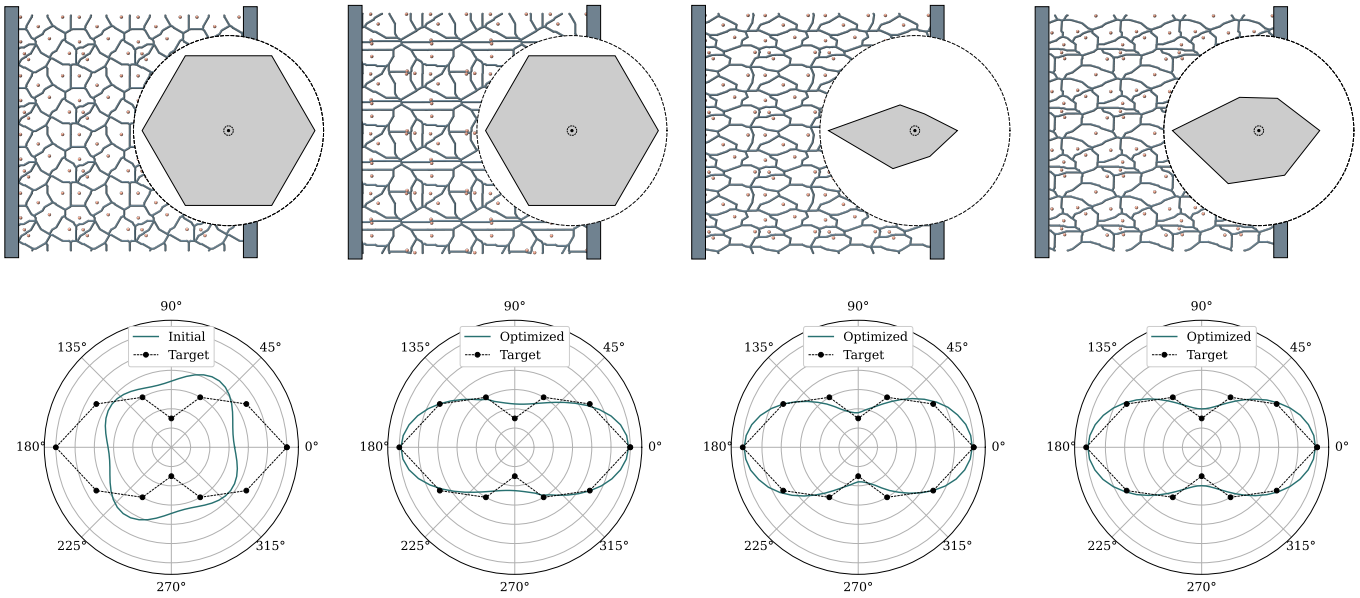
Given a rod network whose rest state matches a given Voronoi diagram, we apply a constant uniaxial stretching force to the material and find the deformed equilibrium state. Appendix A provides the discrete rod energy model used in this simulation. We compute stiffness from deformation along the stretch direction and Poisson ratio from deformation in the perpendicular direction. Following the equilibrium-constrained optimization procedure of Numerow *et al.* [2024], we optimize for a desired Poisson ratio, or for a desired stiffness profile over a range of stretch directions.

In the first experiment, shown in Figure 14a, we optimize for maximum negative Poisson ratio, using a 3-fold rotational metric and sites in a triangular lattice. Site locations are fixed and we optimize over parameters of the metric polygon. In a second example (Figure 14b), we optimize for the target stiffness profile shown in the polar plot, using a metric without enforced symmetries. Here, we simultaneously optimize for the metric polygon along with site positions within a square unit cell.

Finding the deformed equilibrium state requires solving an optimization problem to minimize the system's energy, which we achieve using Newton's method. For the equilibrium-constrained optimization of the metric polygon, we use the Globally-Convergent Method of Moving Asymptotes [Svanberg 2002].



(a) Optimizing the metric polygon for maximum negative Poisson ratio. *Left:* initial pattern with $\nu = 0.332$. *Middle:* optimized pattern with $\nu = -0.392$. The stretched equilibrium state of the optimized pattern exhibits outward bulging characteristic of auxetic materials. *Right:* 3D printed sample of the pattern.



(b) Optimizing the metric polygon and site locations for target stiffness profile. *1st column:* initial pattern, with polar plot comparing initial and target profiles. *2nd & 3rd columns:* independent optimizations of site locations and metric parameters respectively. *4th column:* pattern optimized with both sites and metric.

Fig. 14. Optimization of 2D structured sheet metamaterials using star-shaped distance Voronoi diagrams. Our closed-form construction provides derivatives of the Voronoi diagram which enable gradient-based optimization.

4.4 Performance

Runtime statistics for results of section 4.2 are collected in Table 3. Experiments were conducted on a Mac mini desktop computer with 64 GB RAM and 64-bit ARM CPU with 10 4.5 GHz cores and 4 2.6 GHz cores.

Table 3. Runtime statistics for experiments.

	# Sites	# Metric Polytope Vertices	# Surface Mesh Faces	# Voronoi Vertices	Time [ms] Build	Time [ms] DC Removal
Sphere 1	162	60	5120	18478	1025.27	61.63
Sphere 2	258	60	5120	22437	1096.26	N/A
Sphere 3	162	96	5120	22161	619.00	N/A
Sphere 4	642	60	5120	50134	5014.04	0.98
Bunny 1	746	60	10004	73118	21544.01	53.68
Bunny 2	748	60	10004	61900	9551.52	3.89
Bunny 3	746	96	10004	90356	12418.42	3.30
Bunny 4	829	60	10004	69696	10723.66	1.83
Elephant 1	696	60	10624	69146	16316.61	54.42
Elephant 2	874	60	10624	70829	12701.17	0.94
Elephant 3	696	96	10624	84483	10935.16	1.77
Elephant 4	696	60	10624	59044	6464.25	1.09
Fertility 1	576	60	111816	141181	42593.66	159.19
Fertility 2	681	60	111816	127575	27756.94	3.09
Fertility 3	576	96	111816	145695	24277.49	6.11
Fertility 4	691	60	111816	133313	23397.89	5.46
Vase 1	525	112	9800	76305	16677.30	N/A
Vase 2	840	60	9800	97304	20267.38	N/A
Vase 3	1260	144	9800	179496	82291.99	N/A

5 Conclusions

We presented an exact algorithm for the construction of star-shaped distance Voronoi diagrams. Unlike the rasterized construction of Martínez et al. [2019], our formulation extends to restricted Voronoi diagrams on 3D surfaces and allows gradient-based design optimization. Combined with the implicit parameterization provided by star-shaped distance Voronoi diagrams, this approach enables a rich design landscape over dense cellular patterns of arbitrary topologies. We demonstrated these advantages using a range of designs on various 3D models, and through optimization of structured sheet materials for specific mechanical properties.

5.1 Limitations & Future Work

In our current implementation, numerical issues can arise when sites are not in a general position, *e.g.* when sites are arranged in a regular lattice. However, a small random perturbation of site positions (we use 10^{-5} times lattice spacing) prevented this entirely in our experiments. Further implementation work is required to handle corner cases, *e.g.* junctions where more than three Voronoi edges coincide.

Results of patterning on surfaces could be further improved using remeshing algorithms which minimize singularities, *e.g.* [Surazhsky and Gotsman 2003]. Furthermore, our dependence on centroidal Voronoi tessellations for site placement precludes the use of non-triangular lattice types (*e.g.* rectangular, honeycomb) on general surfaces, which limits the space of possible tilings. An intrinsic formulation for surface meshes, based on geodesic rather than Euclidean distances, would reduce visual artifacts in areas of high curvature.

While our mechanical optimization experiments are limited to 2D structured sheet materials, the approach could be extended to designing structural networks on surfaces [Zehnder et al. 2016].

Finally, exploring aesthetics-driven optimization objectives, such as target shapes for Voronoi cells, is another worthwhile direction for future work. Discontinuities arising from topological changes in the Voronoi network, which occur more frequently for star-shaped metrics than for smoothly varying distances, make this problem particularly challenging.

Acknowledgments

We are grateful to the anonymous reviewers for their valuable feedback. This work was supported by the Swiss National Science Foundation through SNF project grant 200021_200644.

References

Pierre Alliez, Éric Colin De Verdière, Olivier Devillers, and Martin Isenburg. 2005. Centroidal Voronoi diagrams for isotropic surface remeshing. *Graphical models* 67, 3 (2005), 204–231.

J A Bennell and J F Oliveira. 2009. A tutorial in irregular shape packing problems. *Journal of the Operational Research Society* 60, 1 (May 2009), S93–S105. doi:10.1057/jors.2008.169

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Trans. Graph.* 29, 4, Article 116 (July 2010), 10 pages. doi:10.1145/1778765.1778853

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) *(SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 63, 12 pages. doi:10.1145/1399504.1360662

Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin Van Hecke. 2017. Flexible mechanical metamaterials. *Nature Reviews Materials* 2, 11 (2017), 1–11.

Pravin Bhat, Stephen Ingram, and Greg Turk. 2004. Geometric texture synthesis by example. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (Nice, France) *(SGP '04)*. Association for Computing Machinery, New York, NY, USA, 41–44. doi:10.1145/1057432.1057437

Anders Brodersen, Ken Museth, Serban Porumbescu, and Brian Budge. 2008. Geometric Texturing Using Level Sets. *IEEE Transactions on Visualization and Computer Graphics* 14, 2 (March 2008), 277–288. doi:10.1109/TVCG.2007.70408

Oleksiy Busaryev, Tamal K. Dey, Huamin Wang, and Zhong Ren. 2012. Animating Bubble Interactions in a Liquid Foam. *ACM Trans. Graph.* 31, 4, Article 63 (jul 2012), 8 pages. doi:10.1145/2185520.2185559

Weikai Chen, Yuexin Ma, Sylvain Lefebvre, Shiqing Xin, Jonàs Martínez, and wenping wang. 2017. Fabricable tile decors. *ACM Trans. Graph.* 36, 6, Article 175 (Nov. 2017), 15 pages. doi:10.1145/3130800.3130817

Weikai Chen, Xiaolong Zhang, Shiqing Xin, Yang Xia, Sylvain Lefebvre, and Wenping Wang. 2016. Synthesis of filigrees for digital fabrication. *ACM Trans. Graph.* 35, 4, Article 98 (July 2016), 13 pages. doi:10.1145/2897824.2925911

Yanqing Chen, Timothy A Davis, William W Hager, and Sivasankaran Rajamanickam. 2008. Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)* 35, 3 (2008), 1–14.

L. Paul Chew and Robert L. (Scot) Dyrsdale. 1985. Voronoi diagrams based on convex distance functions. In *Proceedings of the First Annual Symposium on Computational Geometry* (Baltimore, Maryland, USA) *(SCG '85)*. Association for Computing Machinery, New York, NY, USA, 235–244. doi:10.1145/323233.323264

Fernando de Goes, Corentin Wallez, Jin Huang, Dmitry Pavlov, and Mathieu Desbrun. 2015. Power Particles: An Incompressible Fluid Solver Based on Power Diagrams. *ACM Trans. Graph.* 34, 4, Article 50 (jul 2015), 11 pages. doi:10.1145/2766901

Chaoyu Du, Tom Van Mele, and Philippe Block. 2023. Tile patterning on free-form surfaces that reduces tile cutting. *Proceedings of IASS Annual Symposia* 2023, 21 (2023), 1–10. https://www.ingentaconnect.com/content/iass/piass/2023/00002023/00000021/art00001

Michael Eigensatz, Mario Deuss, Alexander Schiftner, Martin Kilian, Niloy J. Mitra, Helmut Pottmann, and Mark Pauly. 2010. Case Studies in Cost-Optimized Paneling of Architectural Freeform Surfaces. In *Advances in Architectural Geometry 2010*, Cristiano Ceccato, Lars Hesselgren, Mark Pauly, Helmut Pottmann, and Johannes Wallner (Eds.). Springer Vienna, Vienna, 49–72.

Filippo Andrea Fanni, Fabio Pellacini, Riccardo Scateni, and Andrea Giachetti. 2022. PAVEL: Decorative Patterns with Packed Volumetric Elements. *ACM Trans. Graph.* 41, 2, Article 19 (Jan. 2022), 15 pages. doi:10.1145/3502802

Fan Feng, Shiying Xiong, Ziyue Liu, Zangyueyang Xian, Yuqing Zhou, Hiroki Kobayashi, Atsushi Kawamoto, Tsuyoshi Nomura, and Bo Zhu. 2022. Cellular topology optimization on differentiable Voronoi diagrams. *Internat. J. Numer. Methods Engrg.* 124, 1 (sep 2022), 282–304. doi:10.1002/nme.7121

Chi-Wing Fu and Man-Kang Leung. 2005. Texture tiling on arbitrary topological surfaces using wang tiles. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques* (Konstanz, Germany) *(EGSR '05)*. Eurographics Association, Goslar, DEU, 99–104.

Andrew Glassner. 1998. Aperiodic Tiling . *IEEE Computer Graphics and Applications* 18, 03 (May 1998), 83–90. doi:10.1109/38.674976

Branko Grunbaum and Geoffrey Colin Shephard. 1989. *Tilings and patterns: An introduction*. WH Freeman and Company.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Gurobi Optimization, LLC. 2024. Gurobi Optimizer Reference Manual. https://www.gurobi.com

Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. 2008. Bubbles alive. *ACM Trans. Graph.* 27, 3 (aug 2008), 1–4. doi:10.1145/1360612.1360647

Wenchao Hu, Zhonggui Chen, Hao Pan, Yizhou Yu, Eitan Grinspun, and Wenping Wang. 2016. Surface Mosaic Synthesis with Irregular Tiles. *IEEE Transactions on Visualization and Computer Graphics* 22, 3 (2016), 1302–1313. doi:10.1109/TVCG.2015.2498620

Steven G. Johnson. 2007. The NLopt nonlinear-optimization package. https://github.com/stevengj/nlopt.

Craig Kaplan. 2009. *Introductory Tiling Theory for Computer Graphics* (1st ed.). Morgan and Claypool Publishers.

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2013. Globally optimal direction fields. *ACM Transactions on Graphics (ToG)* 32, 4 (2013), 1–10.

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2015. Stripe Patterns on Surfaces. *ACM Trans. Graph.* 34 (2015). Issue 4.

Eric Landreneau and Scott Schaefer. 2010. Scales and scale-like structures. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 1653–1660.

Sylvain Lefebvre and Hugues Hoppe. 2006. Appearance-space texture synthesis. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 541–548.

Kurt Leimer and Przemyslaw Musialski. 2020. Reduced-Order Simulation of Flexible Meta-Materials. *Proceedings - SCF 2020: ACM Symposium on Computational Fabrication*. doi:10.1145/3424630.3425411

Bruno Lévy. 2015. Geogram. https://github.com/BrunoLevy/geogram.

Bruno Lévy and Yang Liu. 2010. L p centroidal voronoi tessellation and its applications. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 1–11.

Yue Li, Stelian Coros, and Bernhard Thomaszewski. 2023. Neural Metamaterial Networks for Nonlinear Material Design. *ACM Trans. Graph.* 42, 6, Article 186 (dec 2023), 13 pages. doi:10.1145/3618325

Yue Li, Logan Numerow, Bernhard Thomaszewski, and Stelian Coros. 2024. Differentiable Geodesic Distance for Intrinsic Minimization on Triangle Meshes. *ACM Trans. Graph.* 43, 4, Article 91 (July 2024), 14 pages. doi:10.1145/3658122

Xiaokang Liu, Lin Lu, Andrei Sharf, Xin Yan, Dani Lischinski, and Changhe Tu. 2020. Fabricable dihedral Escher tessellations. *Computer-Aided Design* 127 (2020), 102853. doi:10.1016/j.cad.2020.102853

Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. 2009. On Centroidal Voronoi Tessellation—Energy Smoothness and Fast Computation. *ACM Trans. Graph.* 28, 4, Article 101 (sep 2009), 17 pages. doi:10.1145/1559755.1559758

Yong-Jin Liu, Zhanqing Chen, and Kai Tang. 2010. Construction of iso-contours, bisectors, and Voronoi diagrams on triangulated surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (2010), 1502–1517.

S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137. doi:10.1109/TIT.1982.1056489

Thomas S Lumpe, Michael Tao, Kristina Shea, and David I W Levin. 2022. Computational design and fabrication of active 3D-printed multi-state structures for shape morphing. *Smart Materials and Structures* 32, 1 (dec 2022), 015008. doi:10.1088/1361-665X/aca5d6

Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. 2016. Procedural voronoi foams for additive manufacturing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.

Jonàs Martínez, Jérémie Dumas, Sylvain Lefebvre, and Li-Yi Wei. 2015. Structure and appearance optimization for controllable shape design. *ACM Trans. Graph.* 34, 6, Article 229 (Nov. 2015), 11 pages. doi:10.1145/2816795.2818101

Jonàs Martínez, Mélina Skouras, Christian Schumacher, Samuel Hornus, Sylvain Lefebvre, and Bernhard Thomaszewski. 2019. Star-shaped metrics for mechanical metamaterial design. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–13.

Jonàs Martínez, Haichuan Song, Jérémie Dumas, and Sylvain Lefebvre. 2017. Orthotropic k-nearest foams for additive manufacturing. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.

Élie Michel and Tamy Boubekeur. 2023. MesoGen: Designing Procedural On-Surface Stranded Mesostructures. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 50, 10 pages. doi:10.1145/3588432.3591496

Juan Montes, Yinwei Du, Ronan Hinchet, Stelian Coros, and Bernhard Thomaszewski. 2023. Differentiable Stripe Patterns for Inverse Design of Structured Surfaces. *ACM Transactions on Graphics (TOG)* (2023).

Juan Sebastian Montes Maestre, Yinwei Du, Ronan Hinchet, Stelian Coros, and Bernhard Thomaszewski. 2024. FlexScale: Modeling and Characterization of Flexible Scaled Sheets. *ACM Trans. Graph.* 43, 4, Article 86 (July 2024), 14 pages. doi:10.1145/3658175

Fabrice Neyret and Marie-Paule Cani. 1999. Pattern-based texturing revisited. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 235–242.

Logan Numerow, Stelian Coros, and Bernhard Thomaszewski. 2025. Star-Shaped Distance Voronoi Diagrams for 3D Metamaterial Design. In *SIGGRAPH Asia 2025 Conference Papers* (Hong Kong, Hong Kong) *(SA '25)*. Association for Computing Machinery, New York, NY, USA.

Logan Numerow, Yue Li, Stelian Coros, and Bernhard Thomaszewski. 2024. Differentiable Voronoi Diagrams for Simulation of Cell-Based Mechanical Systems. *ACM Trans. Graph.* 43, 4, Article 69 (July 2024), 11 pages. doi:10.1145/3658152

Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. 2021. Computational inverse design of surface-based inflatables. *ACM Trans. Graph.* 40, 4, Article 40 (July 2021), 14 pages. doi:10.1145/3450626.3459789

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case stress relief for microstructures. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–16.

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–12.

Serban D. Porumbescu, Brian Budge, Louis Feng, and Kenneth I. Joy. 2005. Shell maps. *ACM Trans. Graph.* 24, 3 (July 2005), 626–633. doi:10.1145/1073204.1073239

Yingying Ren, Julian Panetta, Seiichi Suzuki, Uday Kusupati, Florin Isvoranu, and Mark Pauly. 2024. Computational Homogenization for Inverse Design of Surface-based Inflatables. *ACM Trans. Graph.* 43, 4, Article 87 (July 2024), 18 pages. doi:10.1145/3658125

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics (Tog)* 34, 4 (2015), 1–13.

Christian Schumacher, Steve Marschner, Markus Gross, and Bernhard Thomaszewski. 2018. Mechanical characterization of structured sheet materials. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–15.

Christian Schumacher, Bernhard Thomaszewski, and Markus Gross. 2016. Stenciling: Designing Structurally-Sound Surfaces with Decorative Patterns. *Computer Graphics Forum* 35, 5 (2016), 101–110. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12967 doi:10.1111/cgf.12967

Nicholas Sharp et al. 2019b. Polyscope. www.polyscope.run.

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).

Vitaly Surazhsky and Craig Gotsman. 2003. Explicit Surface Remeshing. In *Eurographics Symposium on Geometry Processing*, Leif Kobbelt, Peter Schroeder, and Hugues Hoppe (Eds.). The Eurographics Association. doi:/10.2312/SGP/SGP03/020-030

Krister Svanberg. 2002. A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations. *SIAM Journal on Optimization* 12, 2 (2002), 555–573. doi:10.1137/S1052623499362822

The CGAL Project. 2024. *CGAL User and Reference Manual* (6.0.1 ed.). CGAL Editorial Board. https://doc.cgal.org/6.0.1/Manual/packages.html

Davi Colli Tozoni, Jérémie Dumas, Zhongshi Jiang, Julian Panetta, Daniele Panozzo, and Denis Zorin. 2020. A low-parametric rhombic microstructure family for irregular lattices. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 101–1.

Thibault Tricard, Vincent Tavernier, Cédric Zanni, Jonàs Martínez, Pierre-Alexandre Hugron, Fabrice Neyret, and Sylvain Lefebvre. 2020. Freely orientable microstructures for designing deformable 3D prints. *ACM Trans. Graph.* 39, 6 (2020), 211–1.

Greg Turk. 2001. Texture synthesis on surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 347–354.

Xiaoning Wang, Xiang Ying, Yong-Jin Liu, Shi-Qing Xin, Wenping Wang, Xianfeng Gu, Wolfgang Mueller-Wittig, and Ying He. 2015. Intrinsic computation of centroidal Voronoi tessellation (CVT) on meshes. *Computer-Aided Design* 58 (2015), 51–61.

Dong-Ming Yan, Bruno Levy, Yang Liu, Feng Sun, and Wenping Wang. 2009a. Isotropic Remeshing with Fast and Exact Computation of Restricted Voronoi Diagram. *Comput. Graph. Forum* 28 (07 2009), 1445–1454. doi:10.1111/j.1467-8659.2009.01521.x

Dong-Ming Yan, Bruno Levy, Yang Liu, Feng Sun, and Wenping Wang. 2009b. Isotropic Remeshing with Fast and Exact Computation of Restricted Voronoi Diagram. *Comput. Graph. Forum* 28 (07 2009), 1445–1454. doi:10.1111/j.1467-8659.2009.01521.x

Dong-Ming Yan, Wenping Wang, Bruno Levy, and Yang Liu. 2011. Efficient Computation of Clipped Voronoi Diagram for Mesh Generation. *Computer-Aided Design* 45 (09 2011). doi:10.1016/j.cad.2011.09.004

Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2016. Designing structurally-sound ornamental curve networks. *ACM Trans. Graph.* 35, 4, Article 99 (July 2016), 10 pages. doi:10.1145/2897824.2925888

Xiaoting Zhang, Guoxin Fang, Chengkai Dai, Jouke Verlinden, Jun Wu, Emily Whiting, and Charlie C.L. Wang. 2017. Thermal-Comfort Design of Personalized Casts. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 243–254. doi:10.1145/3126594.3126600

Zhan Zhang, Christopher Brandt, Jean Jouve, Yue Wang, Tian Chen, Mark Pauly, and Julian Panetta. 2023. Computational Design of Flexible Planar Microstructures. *ACM Trans. Graph.* 42, 6, Article 185 (dec 2023), 16 pages. doi:10.1145/3618396

Kun Zhou, Xin Huang, Xi Wang, Yiying Tong, Mathieu Desbrun, Baining Guo, and Heung-Yeung Shum. 2006. Mesh quilting for geometric texture synthesis. In *ACM SIGGRAPH 2006 Papers* (Boston, Massachusetts) *(SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 690–697. doi:10.1145/1179352.1141942

Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Trans. Graph.* 36, 4, Article 120b (jul 2017), 16 pages. doi:10.1145/3072959.3095815

A Structured Sheet Material Model

In the experiments of Section 4.3, we simulate rod networks with rest state rods and joints defined by the vertices and edges of a star-shaped distance Voronoi diagram. An additional n nodes are inserted along each rod, uniformly spaced between the endpoints, to enable bending. We adapt our simulation model from discrete Kirchhoff rods [Bergou et al. 2010, 2008], which defines the elastic energy $E(\mathbf{x}, \bar{\mathbf{x}})$ of a rod network with nodes $\mathbf{x}$ and rest state $\bar{\mathbf{x}}$.

For a rod connecting nodes $\mathbf{x}_i$ and $\mathbf{x}_j$, let $\mathbf{e}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ and $\bar{\mathbf{e}}_{ij} = \bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i$. The total elastic energy of the system is given by $E(\mathbf{x}, \bar{\mathbf{x}}) = E_{\text{stretching}}(\mathbf{x}, \bar{\mathbf{x}}) + E_{\text{bending}}(\mathbf{x}, \bar{\mathbf{x}}) + E_{\text{joints}}(\mathbf{x}, \bar{\mathbf{x}})$. Here,

$$E_{\text{stretching}}(\mathbf{x}, \bar{\mathbf{x}}) = \sum_{\text{edges } \mathbf{e}} \frac{1}{2} k_{\text{s}} \left(\frac{|\mathbf{e}|}{|\bar{\mathbf{e}}|} - 1 \right)^2 |\bar{\mathbf{e}}|, \tag{7}$$

$$E_{\text{bending}}(\mathbf{x}, \bar{\mathbf{x}}) = \sum \frac{1}{|\mathbf{e}_{ij}| + |\mathbf{e}_{jk}|} k_{\text{b}} \left(\frac{2|\mathbf{e}_{ij} \times \mathbf{e}_{jk}|}{|\mathbf{e}_{ij}||\mathbf{e}_{jk}| + \mathbf{e}_{ij} \cdot \mathbf{e}_{jk}} \right)^2 \tag{8}$$

where $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ are consecutive nodes along a rod, and

$$E_{\text{joint}}(\mathbf{x}, \bar{\mathbf{x}}) = \sum \frac{1}{2} k_{\text{j}} \left| \langle \mathbf{e}_{ij}, \mathbf{e}_{ik} \rangle - \langle \bar{\mathbf{e}}_{ij}, \bar{\mathbf{e}}_{ik} \rangle \right|^2. \tag{9}$$

where $\mathbf{x}_i$ is a multi-rod joint and $\mathbf{x}_j$, $\mathbf{x}_k$ ($j \neq k$) are nodes adjacent to the joint. $\langle \mathbf{e}_{ij}, \mathbf{e}_{ik} \rangle$ denotes the angle between two edges.

In the presence of an external force F_{ext}, the net force in the system is given by

$$F = -\frac{\partial E}{\partial \mathbf{x}} + F_{\text{ext}}, \tag{10}$$

and the equilibrium state $\mathbf{x}^*$ satisfying $F = 0$ is obtained by solving the optimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \; E(\mathbf{x}, \bar{\mathbf{x}}) - F_{\text{ext}} \cdot \mathbf{x}. \tag{11}$$