Designing triangle meshes with controlled roughness

VICTOR CEBALLOS INZA, KAUST, Saudi Arabia
PANAGIOTIS FYKOURAS, BMW Group, Germany
FLORIAN RIST, KAUST, Saudi Arabia and TU Wien, Austria
DANIEL HÄSEKER, BMW Group, Germany
MAJID HOJJAT, BMW Group, Germany
CHRISTIAN MÜLLER, TU Wien, Austria
HELMUT POTTMANN, TU Wien, Austria and KAUST, Saudi Arabia

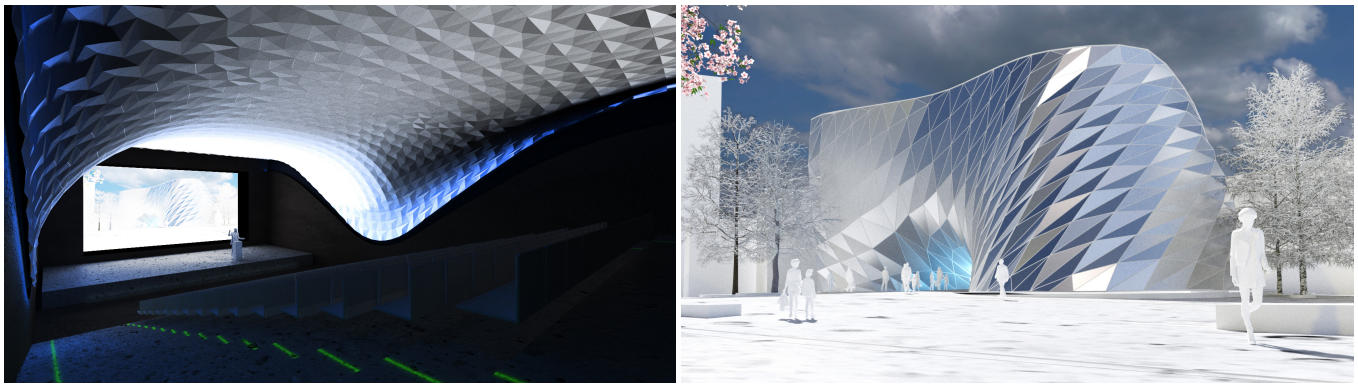


Fig. 1. *Triangle meshes with controlled roughness as a design element in architectural applications.* The ceiling of the auditorium has been computed from a triangulated reference surface by an automatic roughening algorithm, which creates a structured visual appearance and enables the designer to control the dihedral angles under orientation constraints. The façade is a triangle mesh whose vertices lie on a given smooth reference surface. The mesh has been created by a remeshing algorithm that locally aligns the mesh edges such that the desired level of roughness is achieved, with a smooth appearance towards the entrance, and rougher towards the right-hand side. The alignment of mesh edges depends on the local curvature behavior and is based on our thorough analysis of the relation between the local positioning of vertex stars and the visual appearance of a mesh.

Motivated by the emergence of rough surfaces in various areas of design, we address the computational design of triangle meshes with controlled roughness. Our focus lies on small levels of roughness. There, roughness or smoothness mainly arises through the local positioning of the mesh edges and faces with respect to the curvature behavior of the reference surface. The analysis of this interaction between curvature and roughness is simplified by a 2D dual diagram and its generation within so-called isotropic geometry, which may be seen as a structure-preserving simplification of Euclidean geometry. Isotropic dihedral angles of the mesh are close to the Euclidean angles and appear as Euclidean edge lengths in the dual diagram, which also serves as a tool for visualization and interactive local design. We present a computational framework that includes appearance-aware remeshing, optimization-based automatic roughening, and control of dihedral angles.

CCS Concepts: • **Computing methodologies** → **Shape modeling**; *Optimization algorithms.*

Additional Key Words and Phrases: discrete differential geometry, smoothness, triangle mesh, computational design, mesh optimization

ACM Reference Format:
Victor Ceballos Inza, Panagiotis Fykouras, Florian Rist, Daniel Häseker, Majid Hojjat, Christian Müller, and Helmut Pottmann. 2024. Designing triangle meshes with controlled roughness. *ACM Trans. Graph.* 43, 6, Article 166 (December 2024), 20 pages. https://doi.org/10.1145/3687940

Authors' Contact Information: Victor Ceballos Inza, victor.ceballosinza@kaust.edu.sa, KAUST, Saudi Arabia; Panagiotis Fykouras, panagiotis.fykouras@bmw.de, BMW Group, Germany; Florian Rist, florian.rist@kaust.edu.sa, KAUST, Saudi Arabia and TU Wien, Austria; Daniel Häseker, daniel.haeseker@bmw.de, BMW Group, Germany; Majid Hojjat, majid.hojjat@bmw.de, BMW Group, Germany; Christian Müller, cmueller@geometrie.tuwien.ac.at, TU Wien, Austria; Helmut Pottmann, helmut.pottmann@kaust.edu.sa, TU Wien, Austria and KAUST, Saudi Arabia.

ACM 1557-7368/2024/12-ART166
https://doi.org/10.1145/3687940

1 Introduction

Computational design of surfaces is ubiquitous across industrial design, with notable exemplars in architecture and the automotive industry. In the latter, smooth surfaces with added design aesthetics (i.e. *class A* surfaces) are commonplace, partly for aerodynamics concerns and due to established tradition. Nevertheless, newer, modern designs are introducing alternative elements, such as prominent feature lines, and faceted finishes (see Fig. 2), in particular for interior design. As new digital design approaches go beyond the traditional choice of spline surfaces in CAD systems, they facilitate the construction of alternatives to perfectly smooth surfaces.

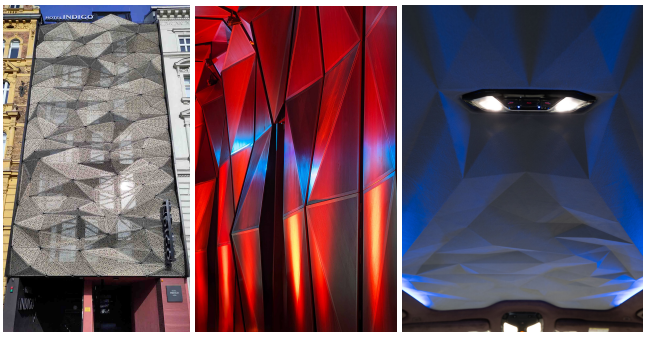


Fig. 2. *Motivation for controlled roughness design tools*: there is an interest in rough surfaces across areas of design, from architecture and interior design, to automotive design. *Left*: Façade of the Hotel Indigo in Vienna, Austria. *Middle*: Multi-function theatre at the KAFD Conference Center in Riyadh, Saudi Arabia. *Right*: Interior roof of the recently launched BMW XM.

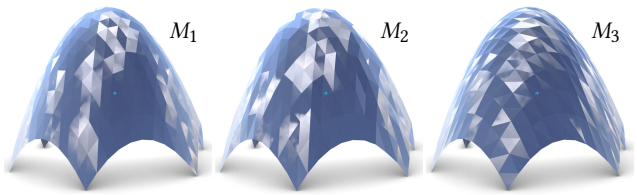


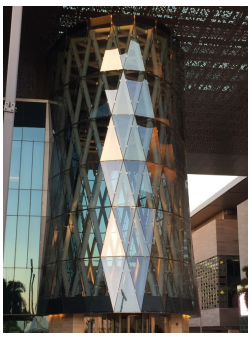
Fig. 4. Roughness achieved over a 2D mesh may disappear when mapping it to a curved 3D surface. The reflection of the cloudy sky (provided in the Appendix) on the surface helps visualize its visual smoothness or roughness. *Left:* A smooth mesh with vertices inscribed in an elliptic paraboloid has roughness measure $\mathcal{R}(M_1) = 0.070$, according to Sec. 2.1 and Eqn. (1). *Middle:* The mesh is roughened by mild vertex displacements, $\mathcal{R}(M_2) = 0.113$. The visual appearance of the mesh does not change much. *Right:* The vertices of the mesh stay inscribed in the elliptic paraboloid, but a different triangulation already exhibits a certain level of roughness, $\mathcal{R}(M_3) = 0.168$.

It is not surprising that now designers start to play with visual effects achievable via a certain level of roughness in the designed surfaces. A natural choice of surface representations which allow us to achieve roughness are meshes with planar faces and in particular triangle meshes. The design of meshes depends on their application. So far, the focus in research has been on mesh quality for finite element computations, on approximation error to a given smooth surface while keeping the mesh size within limits, on smoothing meshes that originally arise from noisy data points, and more recently on visual smoothness [Pellis et al. 2019].

We are not aware of any research dealing with *controlled roughness in triangle meshes*. When talking about roughness, one has to distinguish between two quite different scenarios. The first concerns *meshes with a strong level of roughness*. Such surfaces appear like a 3D texture mapped onto a smooth surface (see Fig. 3). Computationally, one can map a given triangle mesh S with a standard parameterization technique onto a 2D mesh S^*, design a rough mesh by adding third coordinates ζ_i to its vertices v_i^* and obtain a roughened 3D mesh S^r via vertex displacement, $v_i^r = v_i + \zeta_i n_i$, along vertex normals of S. For significant deviations from the reference mesh, roughness will be kept.

A more subtle topic is *meshes with a moderate level of roughness*. In the 3D texture approach, small deviations may completely disappear due to the curvature of the reference surface. As an example, take a convex surface. Each vertex can be relocated within a certain domain by keeping convexity, which is a form of visual smoothness [Pellis et al. 2019]. In other words, achieving some roughness can require quite strong displacements of mesh vertices. On the other hand, another triangle mesh with vertices on the same smooth reference surface may already exhibit roughness (see Fig. 4).

Meshes with a lower level of roughness may have their vertices on a smooth underlying design surface, but the triangulation is chosen so that small variations in triangle normals appear which are not present in the underlying smooth surface. An example is seen in the inset, where two families of edges in a triangulation lie on rulings of a hyperboloid. Here, the choice of triangulation has most likely been based on fabrication considerations, but the slight visual roughness adds an interesting aesthetic quality to the design.



The present paper addresses the computational design of triangle meshes with controlled roughness over given reference surfaces. While we provide algorithms for achieving strong and mild levels of roughness, the focus is on the latter. In all cases we are interested in some type of regularity. This distinguishes our work from an obvious way of achieving a rough appearance, namely by random perturbation of mesh vertices (Fig. 5). While this approach may be useful in certain applications, we do not see it as a challenging research problem and therefore focus on more structured designs.

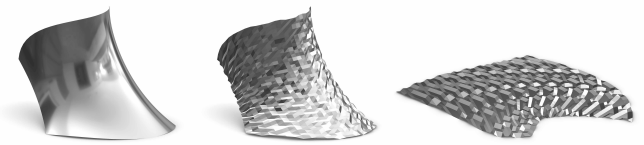


Fig. 3. A strong level of roughness can be achieved via texture mapping. A negatively-curved reference surface (*left*) is meshed, and subsequently roughened (*middle*) by displacing vertices along vertex normals, according to a roughening pattern designed on a parameterization of the mesh (*right*).

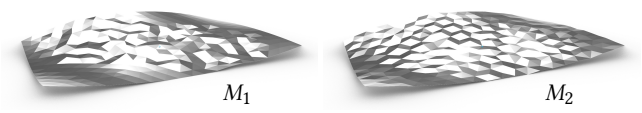


Fig. 5. Controlled roughness is inherently different from random perturbations. *Left:* A mesh is roughened by randomly displacing vertices along vertex normals, $\mathcal{R}(M_1) = 0.122$. *Right*: The same mesh is roughened with our method, using comparable displacements, $\mathcal{R}(M_2) = 0.163$.

1.1 Related work

Mesh smoothing, also known as mesh denoising, filtering or fairing, is a core topic of geometry processing and thus received a lot of attention. We just mention a few contributions ([Taubin 1995; Desbrun et al. 1999; Fleishman et al. 2003; Nealen et al. 2006]) and point to the possibility of applying reverse smoothing operations to achieve some type of roughness [Vallet and Lévy 2008]. More importantly, as many of these techniques move vertices based on local approximations (e.g., moving least squares [Levin 1998]), the result of smoothing may still be a mesh with mild roughness in our sense, even if dominant mesh polylines get smoothed as well. In other words, mesh smoothing applied to a mesh with subtle roughness will not remove roughness, as the directions of edges in vertex stars, in relation to the local curvature behavior, will not change much, but contributes essentially to the visual appearance.

Vertex stars whose Gauss images are free of self-intersections play a key role in smoothness concepts for polyhedral surfaces [Jiang et al. 2016; Pellis et al. 2019; Günther et al. 2020]. It has been shown that principal meshes exhibit the best visual smoothness of polyhedral surfaces [Pellis et al. 2019], if one measures smoothness by an energy proportional to Eqn. (1), that is the sum of absolute values of dihedral angles, weighted by the respective edge lengths. The focus in that paper is on negatively curved areas, but we will see that non-smoothness is also appearing in areas of positive curvature. We will later use a concept from graphic statics to visualize smoothness and roughness in meshes. Thus, work by Orden et al. [2004] on pseudo-triangulations whose reciprocal diagrams have no self-intersections, can also be seen as a contribution to the smoothness of meshes.

Roughness in meshes can be a result of their generation, for example in a folding process. Thus, rough meshes appear in various contributions to origami and kirigami, but there the existence of a planar unfolding is the dominant property rather than the control of roughness. We point to a survey on origami and kirigami approaches [Callens and Zadpoor 2018], Miura-origami-like representations of non-planar shapes [Dudte et al. 2016], corrugated surfaces resulting from quad-based kirigami [Jiang et al. 2020], the computation of folding patterns to produce an origami model of a given shape [Tachi 2010; Demaine and Tachi 2017], to a monograph on folding algorithms [Demaine and O'Rourke 2007] and surfaces with curved folding patterns that leads to a wrinkled appearance [Demaine et al. 2015; Kraft et al. 2023]. Roughness of a different type is present in crumpled paper [Ben Amar and Pomeau 1997]. Tymms et al. [2020] aggregate a custom metric into their optimization objective to achieve visual similarity of height fields under under a set of different lighting conditions.

Some level of un-smoothness in meshes also appeared in a study of meshes with planar faces that are arranged as in semi-regular patterns in the plane. Most works, such as [Li et al. 2014], focus on computing meshes with planar faces that follow mesh aesthetics imposed by fairness, vertex valence regularity, and face shape regularity. While certain types of patterns can adapt to a given reference surface by shape changes in faces, others, for example tri-hex patterns, keep their appearance at the price of a certain level of roughness [Jiang et al. 2015b]. A similar effect has been observed when attempting to compute meshes with repetitive faces for architectural applications [Fu et al. 2010; Singh and Schaefer 2010]. An extreme level of element repetition is achievable with very rough surfaces [Huard et al. 2015].

1.2 Contributions and overview

We present a framework for the computational design of triangle meshes with controlled roughness. Its main features are briefly outlined in Section 2.

The visual appearance of a triangle mesh T depends on the dihedral angles at its edges. Instead of using the Euclidean Gauss image to visualize these angles, we propose a related, but simpler method that is rooted in isotropic geometry. Its basics are described in Section 3.1. The isotropic viewpoint naturally associates with T a combinatorially dual 2D diagram T^* whose edge lengths are a good approximation of Euclidean dihedral angles (Section 3.2) and which we use for visualization of roughness and as a tool for design and analysis. The embedding into isotropic geometry is effectively used in Section 3.3 for studying the relationship between the visual appearance of a triangle mesh and the position of its vertex stars relative to the local curvature behavior.

Section 4.1 uses the geometric analysis to develop an algorithm that uses remeshing as a tool to lay out a triangle mesh with the desired level of roughness on a given surface. In Section 4.2, we provide details on the manipulation tool based on the dual diagram. An optimization framework for the design of roughness in meshes whose vertices are not confined to the reference surface is presented in Section 4.3.

In Section 5 we add further details on the implementation and present results including the verification of their visual appearance by physical models. We conclude with a few pointers towards directions for future research.

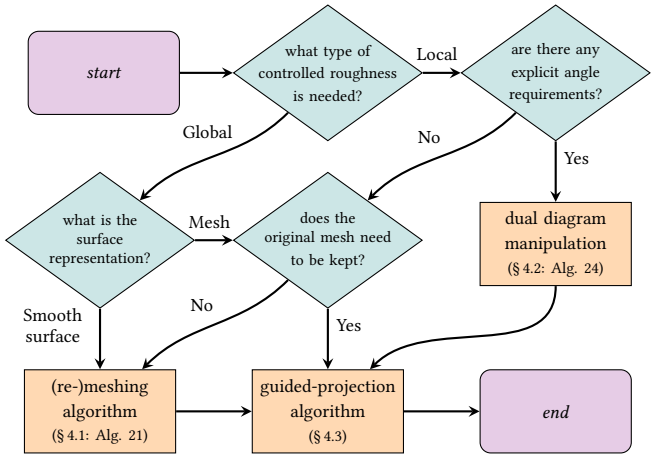


Fig. 6. *Framework for mesh design with controlled roughness.* Our local and global design tools are the key components of our proposed design framework, illustrated by this flowchart. Processes are followed by numerical optimization in order to enforce design constraints with high accuracy. Standalone guided-projection optimization can also provide control over mesh roughness.

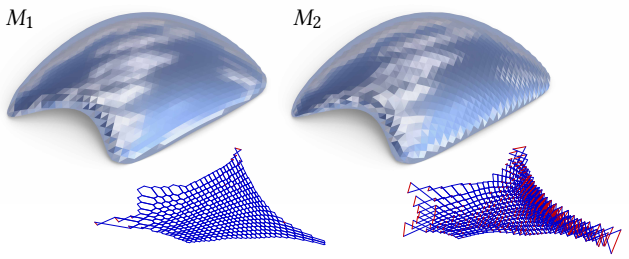


Fig. 7. *Left*: An architectural roof surface is originally meshed so that it is visually smooth, $\mathcal{R}(M_1) = 0.062$. *Right*: Using our design tools, we can remesh the surface so that its appearance subtly varies from smooth to rough, $\mathcal{R}(M_2) = 0.106$. The corresponding (partial) dual diagrams clearly characterize the roughness of the meshes. Different edge colors indicate angles of different signs.

2 Framework for mesh design with controlled roughness

We begin by providing an intuitive description of *controlled roughness* on a polyhedral mesh, in opposition to smoothness. Consider a triangle mesh as a discrete approximation of an underlying smooth, continuous surface. The smoother the triangle mesh, the better it approximates the surface. This is often characterized through the *dihedral angles* of the mesh: smaller dihedral angles correspond to smoother meshes, with the limit of increasingly finer triangulations leading to vanishing angles.

It is clear that there is no analogous continuum for rougher meshes. In general, dihedral angles can be increased by variably lifting mesh vertices off the underlying surface. While such a modified mesh does appear roughened, it may no longer be a *proper* representation of the reference surface, meaning that the neighbourhood of a vertex of the mesh does not resemble the shape of the neighbourhood of the corresponding point on the smooth surface.

We are therefore interested in *controlled roughness*, where we design rough representations of surfaces. We do this with two different approaches: first, by designing an algorithm that moderately increases dihedral angles, subject to restrictions. Secondly, by adapting the triangulation of meshes depending on the surface curvature and the required level of roughness.

The connotation of controlled roughness is two-fold: since our motivation spawns from computational design, we are also interested in *controlling* controlled roughness. To this end, we developed design tools for guiding global roughness for overall aesthetics, as well as tools for local manipulations for finer adjustments.

2.1 Measuring controlled roughness

Despite not having a proper metric on controlled roughness, we can nevertheless assign a measure of roughness to a mesh. Given a polyhedral mesh $M = (V, E, F)$, with edge set E, where edge $e \in E$ has length ℓ_e and dihedral angle α_e, we define the normalized *roughness measure* $\mathcal{R}(M)$ as

$$\mathcal{R}(M) = \frac{1}{\sum_{e\in E} \ell_e} \sum_{e\in E} \ell_e |\alpha_e|. \tag{1}$$

This measure is a slight modification of energy functionals commonly used in the mesh smoothing literature, see e.g. [Pellis et al. 2019]. It may not be used as an energy functional on roughness, yet it is still useful as a qualitative descriptor for roughness, especially when comparing different polyhedral representations of the same underlying surface.

2.2 Visualizing controlled roughness

From an aesthetic consideration, roughness is mainly a property of the visual appearance of the surface. Thus it is best revealed by the reflection patterns on shiny surfaces. This motivates the style of rendering across the figures, with flat shading and reflective materials. We often use reflections from a cloudy sky environment map (see e.g. Fig. 4, Fig. 7 or Fig. 15). Notably, we are able to capture the visual appearance of a fabricated rough model (see Fig. 27 and Fig. 31).

Since we control roughness by adjusting dihedral angles, our design tools need to build an intuition for the behaviour of local changes. Visualizing dihedral angles in 3D Euclidean space can be based on the Gauss image, see e.g. [Pellis et al. 2019]. We use a simpler dual diagram of the mesh. It is related to the Gauss image, but easier to compute and analyze, as introduced later in Sec. 3.2.

Given a point of interest on a mesh with a slightly rough triangulation, we compute a locally adapted Cartesian frame that is aligned with the mesh normal. In this frame, we find the dual diagram of a neighbourhood of the point. The dihedral angles between faces in the mesh are shown as lengths of segments between points on the diagram (see Sec. 3.1). Combinatorially regular vertex stars are shown as hexagons.

Depending on the curvature of the reference surface at that point, roughness is characterized by the shape of such hexagons (see Sec. 3.3). Therefore, we can use the dual diagram as a proxy for visualizing roughness (see Fig. 8). Moreover, changes in the dual diagram lead to changes in roughness; this is the basis for our design tool discussed in Sec. 2.3.

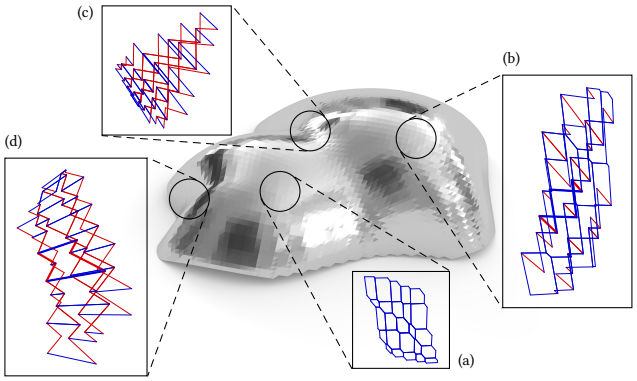


Fig. 8. Our visualization tool can help easily identify the roughness of a region. Insets (a)-(b) show diagrams typical for regions of positive Gaussian curvature with (b) rougher than (a), whereas insets (c)-(d) show diagrams, characteristic of regions of negative Gaussian curvature, with (c) rougher than (d). The interpretation of diagrams is discussed in Section 3.3.

If the reference surface at the point is planar, or has very small curvature, the dual diagram might be degenerate or become crowded with overlapping hexagons in a small region. In such a case, our visualization tool allows lifting the area of interest onto a rotational paraboloid of increasing curvature. In the local frame, this amounts to moving each vertex (x_i, y_i, z_i) to $(x_i, y_i, z_i + k(x_i^2 + y_i^2))$. The effect of applying this transformation in 3D Euclidean space translates to spreading out the dual diagram (see Fig. 9). For a small enough curvature, i.e. small k, dihedral angles change only slightly and the diagram retains its characteristic features.

2.3 The dual diagram as an interactive local design tool

Using the dual diagram as an interactive tool is an application of the geometric theory (see Sec. 3) within an optimization framework. Again, the dual diagram visualizes dihedral angles at edges of a mesh as lengths of segments. Given explicit angle requirements, it is easy to scale these segments as needed. Duality also provides a means to map back these requirements onto the mesh, which can later be optimized for refinement. This tool is illustrated in Fig. 10. Details regarding the algorithm and implementation are provided in Sec. 4.2 and summarized in Alg. 24.

2.4 Global design tools for controlled roughness

As shown in Fig. 4, and later elaborated on in Sec. 3, the visual appearance of a surface heavily depends on its discretization, and in particular, its triangulation. A triangle mesh with vertices on a smooth underlying surface may appear smooth, while another mesh with the same vertices but different triangles can have a very rough appearance. It stands to reason that an interactive tool to design controlled roughness is based on laying out a mesh on a surface where edges align with certain directions. We introduce our roughening algorithm (Alg. 21) in Sec. 4.1.

This originally global design tool can also be used to control roughness locally, by keeping edge directions fixed in the areas where the existing rough appearance is to be kept. In this manner, you can design surfaces with varying appearance from smooth to rough (see Fig. 7). The flowchart in Fig. 6 shows how our design tools can be applied in different design input scenarios.

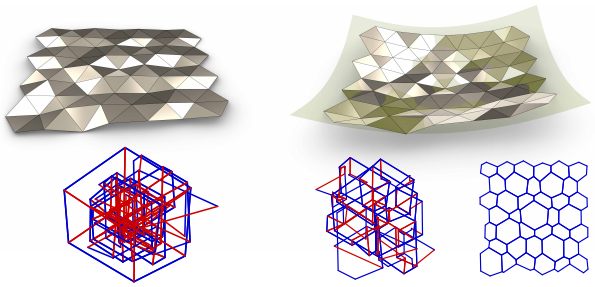


Fig. 9. Our visualization tool can also display dual diagrams for areas of small curvature. *Left:* A rough mesh where the dual diagrams of the vertex stars overlap and are not very informative. *Right:* Lifting onto a slightly curved paraboloid (yellow), the dual diagrams spread out and are easier to interpret. A highly curved paraboloid removes roughness, as seen in the dual diagram in the bottom right, confirming the loss of mild roughness by a simple 3D texture mapping approach (cf. Figs. 3 and 4).

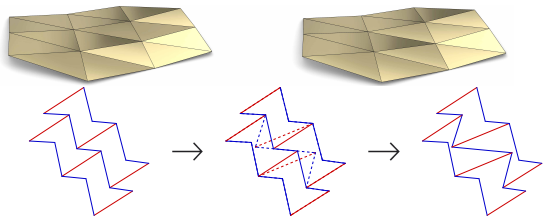


Fig. 10. Illustration of the dual diagram manipulation tool. The dual diagram of the neighborhood of a given edge is shown on the left, to which a transformation is applied, in the middle. The resulting changes in the dual diagram on the right are mapped back onto the neighborhood.

3 Geometric concepts

In this section, we investigate the relation between the visual appearance of a triangulated surface T and the curvature of a smooth underlying reference surface S. We focus on *small* dihedral angles, since their appearance as visually smooth or rough is subtle and strongly depends on the curvature of S.

We use a locally adapted simple non-Euclidean geometry, the so-called *isotropic geometry* (see Sec. 3.1), in which dihedral angles are easier to compute than in Euclidean geometry and which do not differ much from the Euclidean angles near the area of interest. As shown in Sec. 3.2, this geometry naturally leads to a 2D dual diagram D in which isotropic dihedral angles on the original surface T appear as edge lengths in D. It will turn out that the dual diagram is a projection of the classical Gauss image and in our context easier to deal with than the Gauss image.

This dual diagram supports us in studying the connection between the visual appearance (level of roughness) of vertex stars in a triangulation and the positioning of the vertex star w.r.t. the local curvature behavior (see Fig. 15 and Sec. 3.3).

3.1 Locally adapted isotropic geometry

We consider a point O of the reference surface S and its neighbourhood. We first compute a locally adapted Cartesian (x, y, z) coordinate system, by aligning the z-axis with the surface normal. Depending on the nature of the given reference surface the normal is estimated with a standard averaging/fitting method. The origin may be taken as barycenter of the neighbourhood, but its choice will turn out to have no influence on the dual diagram (Def. 3.6).

We associate with the (x, y, z) system a so called *isotropic geometry* I^3. It can be seen as a simplified version of Euclidean geometry, which has been systematically developed by K. Strubecker (see e.g., [Strubecker 1941, 1942] and the monograph [Sachs 1990]). For a short introduction in English, we refer to [Pottmann and Liu 2007].

Isotropic metric and motions. The metric in isotropic 3-space I^3 is degenerate. It is the Euclidean metric after "forgetting" the z-component:

Definition 3.1. *The* isotropic distance (i-distance) *between two points* $A = (a_1, a_2, a_3)$ *and* $B = (b_1, b_2, b_3)$ *is defined as*

$$d_i(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}. \tag{2}$$

To simplify its description, we call the orthogonal projection $A' = (a_1, a_2, 0)$ of a point $A = (a_1, a_2, a_3)$ onto the xy-plane $\Pi\colon z = 0$

the *top view*. Thus, the i-distance between two points is measured as Euclidean distance in the top view (see Fig. 11 left).

Points with the same top view lie on the same z-parallel line and are called *parallel*, an unusual notion that will be justified later (under Eqn. (6)); z-parallel lines are called *isotropic lines*. The i-distance of parallel points is zero. The isotropic angle between two non-isotropic straight lines equals the Euclidean angle in the top view (Fig. 11, left). Very important for us is the notion of angles between planes.

Definition 3.2. *The* isotropic angle between two planes $P\colon z = p_1x + p_2y + p_3$ *and* $Q\colon z = q_1x + q_2y + q_3$ *is defined as*

$$a_i(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}. \tag{3}$$

Geometrically, the isotropic angle between two planes P, Q is the distance between the intersection points P^*, Q^* of the plane $z = 0$ with the corresponding normals through $H = (0, 0, 1)$ (Fig. 11, left).

Just as Euclidean geometry studies properties which are invariant under Euclidean motions (i.e., congruence transformations), isotropic geometry addresses invariance w.r.t. *isotropic motions*. These are affine transformations which keep the i-distance invariant. Isotropic motions are affine maps which appear in the top view as Euclidean 2D motions (cf. [Sachs 1990]). The z-coordinate transforms by adding a linear function in x, y to the z-coordinates. Consequently, *isotropic motions* are given by equations of the form

$$\begin{aligned} x' &= x\cos\alpha - y\sin\alpha + a,\\ y' &= x\sin\alpha + y\cos\alpha + b,\\ z' &= z + c_1x + c_2y + c. \end{aligned} \tag{4}$$

Example 3.3. *The isotropic motion* (4) *with* $\alpha = 0$, $c = \frac{1}{2}(\kappa_1 a^2 + \kappa_2 b^2)$, $c_1 = \kappa_1 a$, $c_2 = \kappa_2 b$, *leaves the paraboloid with equation*

$$2z = \kappa_1 x^2 + \kappa_2 y^2$$

invariant. It maps the origin $(0, 0, 0)$ *to* $(a, b, \frac{1}{2}(\kappa_1 a^2 + \kappa_2 b^2))$.

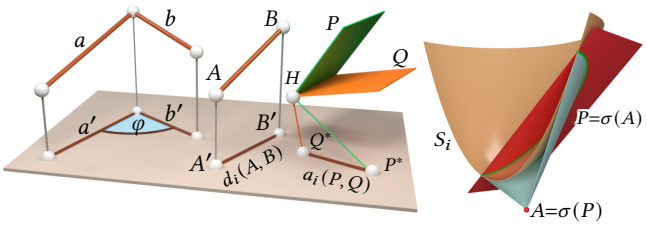


Fig. 11. *Left:* Distances and angles in isotropic space I^3. The isotropic distance $d_i(A, B)$ between two points $A, B \in I^3$ is measured as Euclidean distance between their top views A', B'. Similarly, the isotropic angle between two lines a, b is measured as Euclidean angle between their top views a', b'. The isotropic angle between two planes P, Q is the distance between the intersection points P^*, Q^* of the plane $z = 0$ with the corresponding normals through $H = (0, 0, 1)$. *Right:* The polarity σ on the unit sphere of parabolic type S_i. A plane P (red) is mapped to its pole $\sigma(P)$ and conversely, a point A is mapped to its polar plane $\sigma(A)$. The tangential cone with vertex A touches S_i along the intersection of S_i with $\sigma(A)$ (for A outside S_i).

Isotropic spheres. In isotropic space I^3 there are two types of spheres. A sphere of *cylindrical type* consists of points with constant i-distance to a fixed point, which is a cylinder with a circle as top view. The spheres of *parabolic type*, which have constant isotropic Gaussian curvature and constant isotropic mean curvature, arise from uniform scaling and translation of the *parabolic unit sphere* [Sachs 1990],

$$S_i\colon 2z = x^2 + y^2, \tag{5}$$

which is a rotational paraboloid from the Euclidean perspective.

Metric duality. The geometry in I^3 possesses a so called *metric duality* which does not exist in Euclidean space. At first, this *metric duality* σ is a bijective map between points and non-vertical planes:

$$\sigma\colon (u, v, w) \longleftrightarrow z = ux + vy - w. \tag{6}$$

Geometrically, this duality is the polarity σ with respect to the parabolic unit sphere S_i (see Fig. 11, right).

Parallel planes $z = ux + vy - w_i$ with $i = 1, 2$ correspond to parallel points (u, v, w_i) (hence the notion of parallel points). Applying σ twice is the identity. Eqn. (2) and Eqn. (3) immediately imply the following lemma.

Lemma 3.4 ([Sachs 1990]). *The metric duality* σ *maps two planes* P, Q *with angle* $a_i(P, Q)$ *to two points* $\sigma(P) = (p_1, p_2, -p_3)$, $\sigma(Q) = (q_1, q_2, -q_3)$ *with i-distance* $d_i(\sigma(P), \sigma(Q))$, *where*

$$d_i(\sigma(P), \sigma(Q)) = a_i(P, Q),$$

and hence the name metric duality.

The metric duality preserves metric quantities even between different types of objects.

Gauss maps. In Euclidean geometry, the Gauss image $\gamma(P)$ of an oriented plane P with unit normal vector $\mathbf{n}$ is the point on the Euclidean unit sphere S^2 with coordinate vector $\mathbf{n}$. The tangent plane of S^2 at $\mathbf{n}$ is parallel to P. Hence, γ maps an oriented plane P to a point of S^2 which has a parallel oriented tangent plane.

The isotropic Gauss map γ_i (*i-Gauss map*) maps a plane P to that point $\gamma_i(P)$ of the parabolic unit sphere S_i (Eqn. (5)) whose tangent plane is parallel to P (which in contrast to the Euclidean case is unique even without an orientation of the plane).

Lemma 3.5. *The i-Gauss image of a plane* P *with equation* $z = p_1x + p_2y + p_3$ *is the contact point* $\gamma_i(P) = (p_1, p_2, \frac{1}{2}(p_1^2 + p_2^2))$ *of the parallel tangent plane. Furthermore, the top view of the i-Gauss image* $\gamma_i(P)$ *and the top view of the dual image point* $\sigma(P)$ *of a plane* P *are identical.*

A proof can be found in the Appendix.

Definition 3.6. *We denote this common top view of* $\sigma(P)$ *and* $\gamma_i(P)$, *which has coordinate vector* $\mathbf{p}^* = (p_1, p_2, 0)$, *by* P^*. *For a polyhedral surface* T *the combinatorially dual mesh in the xy-plane with vertices* P^* *for all faces* $P \in T$ *is called* dual diagram.

We will elaborate more on the dual diagram in Sec. 3.2. We are mostly interested in the map $P \mapsto P^*$, since the Euclidean distance between points P^*, Q^* equals the i-angle between planes P, Q (which we want to be able to control).

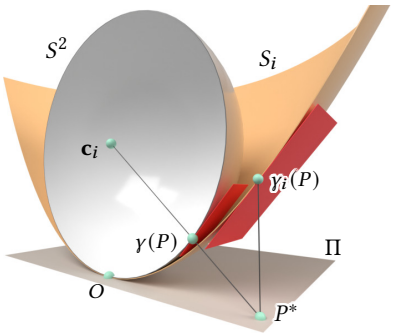


Fig. 12. The isotropic unit sphere S_i is in third-order contact with a Euclidean unit sphere (center $\mathbf{c}_i$) at its Euclidean vertex O. The diagram point $P^* \in \Pi$ of a plane P arises from the Euclidean Gauss image $\gamma(P)$ of P via projection from the sphere center $\mathbf{c}_i$ onto plane Π.

Let us consider a point O on a smooth surface S. We center our coordinate system at O such that the xy-plane coincides with the tangent plane. The parabolic unit sphere S_i is now in tangential contact with S. The Euclidean unit sphere S^2, centered at $\mathbf{c}_i = (0, 0, 1)$, is in third order contact with S_i at O (see Fig. 12).

Let us take S^2 and S_i as carriers of Euclidean and isotropic Gauss images of our surface S. Consequently, the two Gauss maps γ and γ_i do not differ a lot in the vicinity of O.

The normal vector $(p_1, p_2, -1)$ of P equals $\mathbf{p}^* - \mathbf{c}_i$ (see Fig. 12), showing that P^* *arises from* $\gamma(P)$ *by central projection from the sphere center* $\mathbf{c}_i$ *onto the plane* Π (which is called *gnomonic projection*).

The Euclidean angle between two planes P and Q equals the Euclidean arc length of the geodesic arc connecting the two points $\gamma(P), \gamma(Q) \in S^2$. The gnomonic projection maps $\gamma(P), \gamma(Q)$ to the straight line segment P^*Q^*. The distance between diagram points P^*, Q^* is the isotropic angle $a_i(P, Q)$ between planes P, Q (see also Fig. 11 left). Clearly, distances are distorted, but not much in the vicinity of O. We summarize these properties in the following lemma, which is proved in the Appendix.

Lemma 3.7. *The Euclidean distance d in the dual diagram approximates the Euclidean angle φ between tangent planes of S^2 in the vicinity of O up to order $O(\varphi^2)$. Similarly, if S_i approximates a polyhedral surface T, then the edge lengths in the dual diagram approximate the Euclidean dihedral angles in the vicinity of O up to second order.*

3.2 Dual diagram of a polyhedral surface

In this section, we study the dual diagram (Def. 3.6). It is the top view of the dual image (which is identical with the top view of the i-Gauss image; cf. Lemma 3.5) of the tangent planes of a surface S and of the face planes of a polyhedral surface T. The dual diagram is helpful for geometric analysis, visualization and interactive editing.

Most of our discussion is on triangle meshes T, but at some places quad meshes and hex meshes with planar faces, which we call *PQ meshes* and *Phex meshes*, respectively, appear.

The polarity σ maps the intersection line $l = P \cap Q$ of two planes P, Q to the line connecting the image points of the planes, $\sigma(P), \sigma(Q)$ —that is, $\sigma(l) = \sigma(P) \vee \sigma(Q)$. We look again at the top views. Normal vectors of the planes P, Q are $\mathbf{n}_P = (p_1, p_2, -1)$ and $\mathbf{n}_Q = (q_1, q_2, -1)$ and thus

$$\mathbf{l} = \mathbf{n}_P \times \mathbf{n}_Q = (q_2 - p_2, p_1 - q_1, p_1 q_2 - p_2 q_1)$$

is a direction vector of l. Its top view is obviously orthogonal to the top view $(p_1 - q_1, p_2 - q_2)$ of $\sigma(l)$. Consequently, *lines which correspond in σ have orthogonal top views* (see Fig. 13). One may rotate the dual image to obtain parallel top views (see below).

Application of the duality σ to T results in a combinatorially dual polyhedral surface $\sigma(T)$. Assuming regular combinatorics, duality maps triangle meshes to Phex meshes and vice versa, while PQ meshes are mapped to PQ meshes. Allowing combinatorial singularities, an irregular face corresponds to an irregular vertex star and vice versa.

Given a polyhedral surface T with top view T', we are mostly interested in the dual diagram (cf. Def. 3.6), i.e., the top view T^* of the dual surface $\sigma(T)$. The meshes T' and T^* are a pair of combinatorially dual meshes in the xy-plane. Corresponding edges are orthogonal (Fig. 13); the length of an edge in T^* is equal to the dihedral i-angle of the corresponding edge in T (Lemma 3.4). Since $\sigma(\sigma(T)) = T$, the relation between the diagrams is symmetric, i.e., T' is also the dual diagram to T^* (top view of $\sigma(T)$).

For better visualization, we may rotate the diagram T^* by 90 degrees about the origin and call it T^r. Then corresponding edges in T' and T^r are parallel and we have a pair of so-called *reciprocal parallel meshes (diagrams)*. For a plane $P\colon z = p_1 x + p_2 y + p_3$, the corresponding points P^*, P^r in the two diagrams are $\mathbf{p}^* = (p_1, p_2, 0)$ and $\mathbf{p}^r = (-p_2, p_1, 0)$, respectively.

The present setting allows us also to visualize and define a *sign of the dihedral angles*. Let us consider two faces f_P, f_Q in planes P, Q meeting at an edge l and let further ν_P and ν_Q denote the unit normal vectors pointing from l to the interior of the faces f_P and f_Q, respectively (see Fig. 14). Then, we may call the angle *positive* (the edge *convex*) if the vector $\nu_P + \nu_Q$ has a positive z-coordinate and *negative* (or *concave*) otherwise.

We may orient the normal vector $\mathbf{n}$ of the top view l' of l so that it points from f'_P to f'_Q. Then, the orientation of the vector $\mathbf{q}^* - \mathbf{p}^*$ of the dual diagram D agrees with that of $\mathbf{n}$ or $-\mathbf{n}$. If $\mathbf{n}$ is a unit vector, we have

$$\mathbf{q}^* - \mathbf{p}^* = a_s(P, Q)\mathbf{n},$$

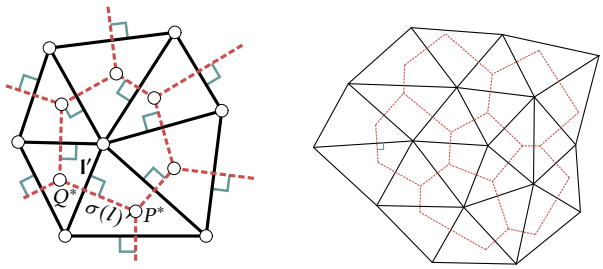


Fig. 13. *Left:* The dual diagram of a vertex star of a triangle mesh T. Two neighboring face planes P, Q are mapped to their dual image points $\sigma(P), \sigma(Q)$. Their top view P^*, Q^* constitute the vertices of the dual diagram (dashed red). The top views l' of the edges l of the triangle mesh T are orthogonal to the top view $\sigma(l)'$ of the edges $\sigma(l)$ of the dual mesh. *Right:* Illustration of the top view and the dual diagram of triangle mesh T with singularity. The combinatorics of the dual diagram is dual to the combinatorics of original mesh T.

where $a_s(P,Q)$ is the *signed dihedral i-angle*. Using the rotated diagram, we orient l' so that f'_P lies on the left side of l'. Then this orientation agrees with that of the edge vector $\mathbf{q}^r - \mathbf{p}^r$ for $a_s(P,Q) > 0$ and is the opposite for $a_s(P,Q) < 0$.

Remark 3.8. The pairs of reciprocal diagrams (T', T^r) or (T', T^*) appear in 2D graphic statics as form and force diagram [Cremona 1890]. The edge vectors in T^r are the forces acting in the corresponding edges in T'. The surface T is the polyhedral stress surface of the 2D system in equilibrium. A positive dihedral i-angle characterizes an edge in compression, a negative angle belongs to a tension force. The present spatial setting including the polarity with respect to a paraboloid has already been used by Maxwell [1870] and recently received increasing interest in computational structural design [Block and Ochsendorf 2007; McRobie 2017; Millar et al. 2022].

If we keep the top view T' but change the dihedral angles, the dual diagram T^* or T^r transforms to a parallel diagram, i.e., corresponding edges in the old and new diagram are parallel.

Note also that points of a surface S correspond to tangent planes of the dual surface $\sigma(S)$. Therefore, if T is inscribed to S (its vertices lie on S), the dual surface $\sigma(T)$ is circumscribed to $\sigma(S)$ (its face planes are tangent to $\sigma(S)$).

3.3 Relation between curvature and visual appearance

Even if the vertices of a triangle mesh T lie on a smooth reference surface S, the mesh T may have a rough visual appearance. This depends on the way in which its vertex stars are positioned with respect to the local curvature elements (principal frames plus principal curvatures). See Fig. 15 for an example where we project regular triangle meshes onto paraboloids.

Again, we are interested in small dihedral angles and want to be close to the Euclidean setting. Therefore, we align our coordinate system such that the origin is placed at a point O whose neighborhood we are interested in and such that $z = 0$ is the tangent plane of the given smooth surface S at O. After aligning the coordinate axes with the principal directions, the second order Taylor approximation

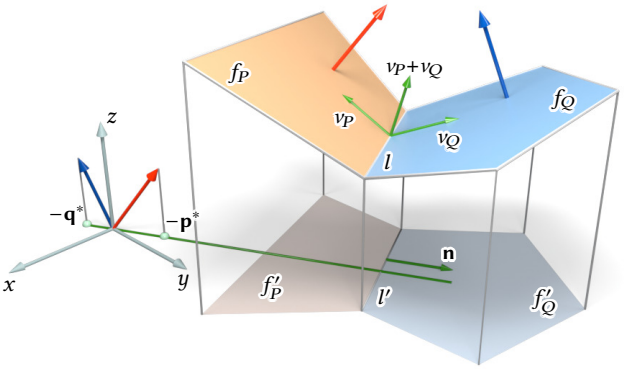


Fig. 14. Dihedral angles between two faces of a polyhedral surface in I^3 can be oriented. If the sum $v_P + v_Q$ of the two edge-normal vectors pointing to the interior of their respective faces has positive z-coordinate, we call the angle positive. The edge vector $\mathbf{q}^* - \mathbf{p}^*$ of the dual diagram is orthogonal to the top view l' of the edge l and it is the signed angle $a_s(P,Q)$ times the unit normal vector $\mathbf{n}$ pointing from f'_P to f'_Q.

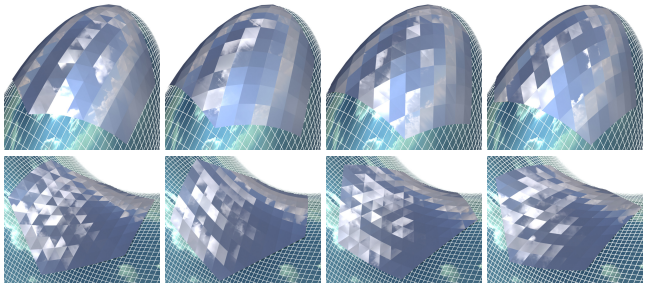


Fig. 15. Regular triangle meshes projected to two paraboloids with equations $z = 4x^2 + y^2$ (*top row*) and $z = x^2 - y^2$ (*bottom row*). The original triangle meshes have been placed in the plane $z = 0$ and rotated incrementally by 15 degrees around the z-axis.

of S in that adapted coordinate system is the *osculating paraboloid* with equation

$$S\colon 2z = \kappa_1 x^2 + \kappa_2 y^2. \tag{7}$$

with κ_1, κ_2 as Euclidean principal curvatures. Euclidean and isotropic normal curvatures agree at O.

Our goal is to analyze the behavior of vertex stars in triangle meshes of regular combinatorics that are formed by three families of fair polylines. The key for that is a study of the behavior of triangle meshes T with vertices on S, whose *top views are regular triangular lattices*, defined by two independent basis vectors $\mathbf{a}, \mathbf{b}$. We call T an *affine-regular triangulation of* S. We will show how the choice of lattice is related to visual smoothness or roughness of T.

Lemma 3.9. *Any two vertex stars in an affine-regular triangulation T^* of a paraboloid S in Eqn.* (7) *are related by an isotropic motion which appears in the top view as translation. The dual diagram T^* is a hex mesh whose faces are related by translations.*

We give a proof in the Appendix. A rough surface T belongs to hexagons with long edges since high roughness means big dihedral angles and therefore long edges in the dual diagram.

By Lemma 3.9, we only need to investigate a single *vertex star* V_O of T, with O as central vertex. Our interest is in the corresponding hexagon V_O^*. Its edges lie in the intersection lines of the plane $z = 0$ (σ image of O) and the σ image planes of the six remaining vertices of V_O. Let $\mathbf{v} = (\xi, \eta, \frac{1}{2}(\kappa_1\xi^2 + \kappa_2\eta^2))$ be a point on the osculating paraboloid. By Eqn. (6), its image plane under σ is $\sigma(\mathbf{v})\colon z = \xi x + \eta y - \frac{1}{2}(\kappa_1\xi^2 + \kappa_2\eta^2)$. Its intersection line $L(\mathbf{v})^*$ in $z = 0$ has the equation

$$L(\mathbf{v})^*\colon \xi x + \eta y = \frac{1}{2}(\kappa_1\xi^2 + \kappa_2\eta^2). \tag{8}$$

Using polar coordinates (r, ϕ) by substituting $(r\cos\phi, r\sin\phi)$ for (ξ, η), we obtain

$$L(\mathbf{v})^*\colon x\cos\phi + y\sin\phi = \frac{r}{2}(\kappa_1\cos^2\phi + \kappa_2\sin^2\phi).$$

Let us recall Euler's formula for the normal curvature in direction ϕ (see, e.g., [do Carmo 1976, p. 145] and [Sachs 1990, p.173])

$$\kappa_n(\phi) = \kappa_1\cos^2\phi + \kappa_2\sin^2\phi,$$

which is identical for isotropic and Euclidean normal curvatures of the paraboloid at the origin. We finally obtain the simple result:

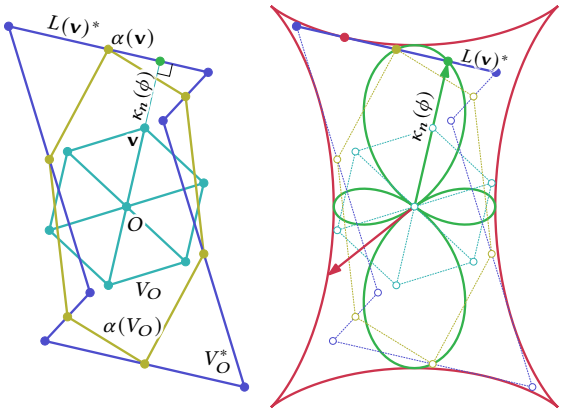


Fig. 16. Relation between the top view of a vertex star and its dual hexagon at a negatively curved point. *Left:* Given is a vertex star V_O with central vertex O and 1-ring vertices $\mathbf{v}$. The edge $L(\mathbf{v})^*$ of the dual hexagon V_O^* is orthogonal to the corresponding edge from O to $\mathbf{v}$. The distance from O to $L(\mathbf{v})^*$ is $\kappa_n(\phi)$. The affine map $\alpha : (\xi, \eta) \mapsto \frac{1}{2}(\kappa_1\xi, \kappa_2\eta)$ maps the 1-ring vertices $\mathbf{v}$ to a hexagon whose vertices $\alpha(\mathbf{v})$ pass through the edges of the dual hexagon. *Right:* The radial diagram of the normal curvature (green) with polar coordinates $(\phi, \kappa_n(\phi))$ is an algebraic curve of order 6. The dual edges $L(\mathbf{v})^*$ also envelope an algebraic curve of order 6 (red).

Lemma 3.10. *The edge in the dual diagram being dual to the edge $O\mathbf{v}$ where $\mathbf{v} = (r\cos\phi, r\sin\phi, \frac{r^2}{2}(\kappa_1\cos^2\phi + \kappa_2 r\sin^2\phi))$, is contained in the line*

$$L(\mathbf{v})^* : x\cos\phi + y\sin\phi = \frac{r\kappa_n(\phi)}{2}. \tag{9}$$

Therefore also $L(\mathbf{v})^*$ is orthogonal to $\mathbf{e}' = (\cos\phi, \sin\phi)$, i.e., orthogonal to the top view of the edge $O\mathbf{v}$.

Orienting the line $L(\mathbf{v})^*$ with the normal $\mathbf{e}'$, the origin has signed distance $r\kappa_n(\phi)/2$ to it (see Fig. 16). Up to scaling with factor $r/2$, related to the scale r appearing in the vertex star, the orthogonal base point of potential diagram edges $L(\mathbf{v})^*$ lie on the curve with polar coordinate representation $r = \kappa_n(\phi)$ (the *pedal curve* of $L(\mathbf{v})^*$). This curve should not be confused with the Dupin indicatrix which is the curve with polar representation $r = 1/\sqrt{|\kappa_n(\phi)|}$. Our pedal curve is still a rather simple rational curve of order 6, with equation $(x^2+y^2)^3 = (\kappa_1 x^2 + \kappa_2 y^2)^2$ (for $\kappa_1 = \kappa_2$ it is a circle). This geometric meaning of edges in the dual face V_O^* of a vertex star is very helpful for subsequent considerations.

Eqn. (8) implies another simple construction of V_O^* from the top view V_O' of the vertex star given by the following lemma.

Lemma 3.11. *The edge $L(\mathbf{v})^*$ corresponding to a vertex $\mathbf{v}$ with top view $\mathbf{v}' = (\xi, \eta)$ is orthogonal to (ξ, η) and passes through $\alpha(\mathbf{v}')$ where α is the affine map given by $\alpha : (\xi, \eta) \mapsto \frac{1}{2}(\kappa_1\xi, \kappa_2\eta)$ (see Fig. 16).*

For the inverse construction we start from a line $L(\mathbf{v})^* : x\cos\phi + y\sin\phi = h$ and obtain the top view of the corresponding vertex via

$$(\xi, \eta) = \frac{2h}{\kappa_n(\phi)}(\cos\phi, \sin\phi). \tag{10}$$

Before proceeding with the discussion, where the sign of Gaussian curvature $K = \kappa_1\kappa_2$ will play an important role, we mention two useful facts which we prove in the Appendix.

Lemma 3.12. *If an affine-regular triangulation of a paraboloid is transformed under an affine map α that maps isotropic lines to isotropic lines, the dual diagram is also transformed by an affine map α^*.*

Affine-regular vertex stars are symmetric w.r.t. the z-axis, thus invariant w.r.t. α with $(a_1, a_2, a_3) = (-1, -1, 1)$, and thus the *dual hexagons are centrally symmetric* due to $\alpha^* : (x, y) \mapsto (-x, -y)$.

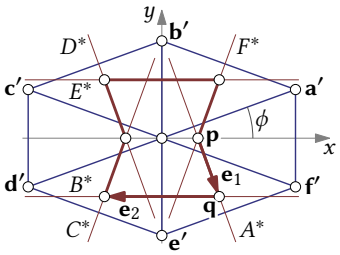
If one is only interested in the essential types of diagram shapes for a specific sign of the Gauss curvature, it is by Lemma 3.12 enough to consider the three special cases, where (κ_1, κ_2) equals $(1, 1)$, $(1, -1)$ and $(1, 0)$.

We call a vertex star with central vertex O *convex*, if it lies on a convex pyramid with tip O. While in geometry we often extend a cone or pyramid to both sides of the tip O, here we are not doing that. The following Lemma is proved in the Appendix.

Lemma 3.13. *A convex vertex star V_O is associated with a* convex diagram V_O^*.

However, a convex hexagon V_O^* does not imply a convex vertex star V_O, as the vertices do not have to lie on a convex pyramid; the vertices could be on different sides of the pyramid tip O (see Fig. 17).

3.3.1 Dual diagrams and isotropic dihedral angles for principal symmetric vertex stars. We now discuss vertex stars V_O which are symmetric w.r.t. the principal curvature directions, i.e., symmetric w.r.t. the planes $x = 0$ and $y = 0$. For the vertex star to be symmetric in that sense two vertices of the boundary hexagon must lie in a symmetry plane; w.l.o.g., we choose $\mathbf{b}$ and $\mathbf{e}$ in $x = 0$ and thus consider the top view of the boundary hexagon of V_O defined by the point $\mathbf{a}' = (r\cos\phi, r\sin\phi)$. The top views $\mathbf{c}', \mathbf{f}', \mathbf{d}'$ are immediately determined by the symmetry (see inset) and $\mathbf{b}' = \mathbf{a}' + \mathbf{c}'$ as well as $\mathbf{e}'$ by the translation invariance of the lattice. Consequently, $\mathbf{c}' = (-r\cos\phi, r\sin\phi)$ and $\mathbf{b}' = \mathbf{a}' + \mathbf{c}' = (0, 2r\sin\phi)$. The corresponding edges $L(\mathbf{a})^*, L(\mathbf{c})^*, L(\mathbf{b})^*$ in the dual diagram V_O^* are denoted by A^*, C^*, B^* (see inset). Due to symmetry, the three edges already determine V_O^*, and it is sufficient to know vertices $\mathbf{p} = A^* \cap (y = 0)$ and $\mathbf{q} = A^* \cap B^*$,



$$\mathbf{p} = \left(\frac{r\kappa_n(\phi)}{2\cos\phi}, 0\right), \quad \mathbf{q} = \left(\frac{r(\kappa_1\cos^2\phi - \kappa_2\sin^2\phi)}{2\cos\phi}, r\kappa_2\sin\phi\right).$$

The two base vectors in the symmetric hexagon V_O^* are therefore $\mathbf{e}_1 = \mathbf{q} - \mathbf{p}$ and $\mathbf{e}_2$ where

$$\mathbf{e}_1 = r\kappa_2\tan\phi(-\sin\phi, \cos\phi),\ \mathbf{e}_2 = \left(\frac{r(-\kappa_1\cos^2\phi + \kappa_2\sin^2\phi)}{\cos\phi}, 0\right).$$

The lengths of $\mathbf{e}_1$ and $\mathbf{e}_2$ are the i-dihedral angles at corresponding edges:

Lemma 3.14. *The signed i-dihedral angles α, β in the vertex star along edges $O\mathbf{a}$ and $O\mathbf{b}$ are*

$$\alpha = r\kappa_2\tan\phi, \quad \beta = \frac{r(\kappa_1\cos^2\phi - \kappa_2\sin^2\phi)}{\cos\phi}. \tag{11}$$

Equal dihedral angles. A particular case is distinguished by equal absolute dihedral angles $|\alpha| = |\beta|$. By equating (11), the case $\alpha = \beta$, yields a quadratic equation for $\sin\phi$ with two solutions

$$\sin\phi_1 = -1, \quad \sin\phi_2 = \frac{\kappa_1}{\kappa_1 + \kappa_2}. \tag{12}$$

In that case the sign of the dihedral angles in the vertex star do not change and is therefore considered smooth in comparison to changing signs in the case $\alpha = -\beta$ which yields

$$\sin\phi_1 = 1, \quad \sin\phi_2 = -\frac{\kappa_1}{\kappa_1 + \kappa_2}. \tag{13}$$

The solution $\sin\phi_1 = 1$ is degenerate whereas the latter may be a valid solution, depending on the value of κ_1/κ_2. We will discuss the arising cases below.

More general principal symmetric vertex stars. The search for equal angles in a vertex star may be done by starting from a principal symmetric hexagon V_O^* and reconstruction of V_O based on Eqn. (10). Apart from the fact that one may obtain a self-intersection in the vertex star V_O, it may not project onto an affinely regular hexagon (diagonals parallel to edges). In our discussion above, this requires to scale the vector $\mathbf{b}'$ to $\sigma\mathbf{b}'$. This changes the isotropic dihedral angles α, β as follows,

$$\alpha = \sigma r \kappa_2 \tan\phi, \quad \beta = \frac{r(\kappa_1 \cos^2\phi + (1-2\sigma)\kappa_2 \sin^2\phi)}{\cos\phi}.$$

The top views of these generalized principal symmetric vertex stars can also tile the plane, and yield a triangulation with a pattern that may be preferred for an application with controlled roughness as a design element (see Fig. 17). Even if the generating vertex star has $|\alpha| = |\beta|$, the overall mesh does not have exactly this property, since there is another type of vertex stars, as shown in Fig. 17. The arising patterns in meshes with mild roughness may be interesting for applications.

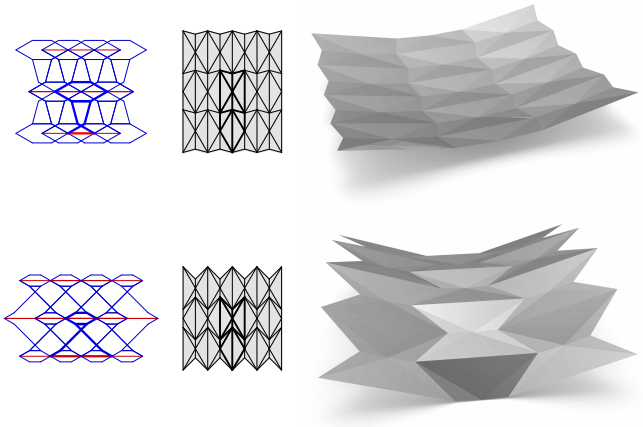


Fig. 17. Triangulations of an elliptic paraboloid (*top*), and a hyperbolic paraboloid (*bottom*), which possess two types of vertex stars. One of them is principal symmetric and appears in the dual diagram (*left*) as a convex hexagon with equal edge lengths. The top views (*middle*) show that the corresponding vertex stars are no longer affine-regular. The mild roughness of the meshes is visualized by the self-intersections in the dual diagrams.

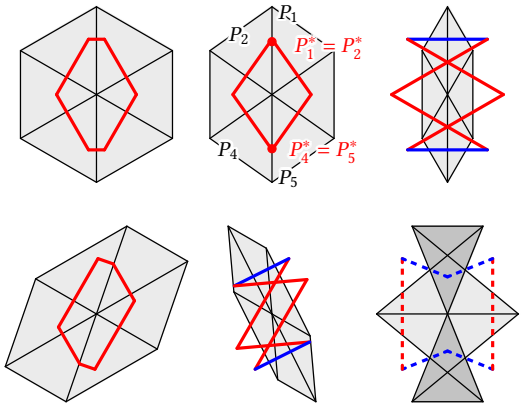


Fig. 18. Vertex stars (shaded) and their dual polygons for an elliptic surface point. Different edge colors indicate angles of different signs. The top row shows principal symmetric stars; the one in the middle with a dual parallelogram is at the border between convexity and non-convexity to $\phi = \operatorname{atan}\sqrt{\kappa_1/\kappa_2}$ and has coplanar face planes $P_1 = P_2$ and $P_4 = P_5$. Non-intersecting, non-convex hexagons can only arise from invalid vertex stars (bottom right).

We are now ready to discuss the possible shapes of diagrams and roughness of vertex stars depending on the sign of Gaussian curvature K. A summary of findings is provided in Sec. 3.3.5.

3.3.2 Elliptic paraboloid ($K > 0$). We can assume $\kappa_1 > 0, \kappa_2 > 0$. Up to affine transformations the shape of V_O^* is even determined by choosing $\kappa_1 = \kappa_2 = 1$, meaning S equals the rotational paraboloid S_i. Lemma 3.11 implies that the edge $L(\mathbf{v}_i)^*$ of V_O^* corresponding to the boundary vertex $\mathbf{v}_i$ of V_O lies in the bisecting line of $O\mathbf{v}_i$. The arising types are shown in Fig. 18. Bowtie-shaped centrally symmetric hexagons V_O^* (without self intersection) cannot arise. In that case, the original vertex star V_O would be self-intersecting (see Fig. 18, bottom right, and Proposition 3.16).

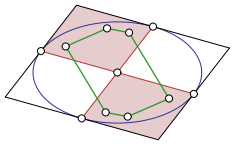
Remark 3.15. The bisector property is well-known in Computational Geometry (see, e.g., [Aurenhammer et al. 2013]): To compute the Voronoi diagram of a set of points $\mathbf{v}_i' \in \mathbb{R}^2$, $i = 1, \ldots, N$, we could project them vertically to the rotational paraboloid to obtain points $\mathbf{v}_i \in S_i$. The tangent planes at $\mathbf{v}_i$, which are the polar planes $\delta(\mathbf{v}_i)$, create the faces of a circumscribed polyhedral surface $\delta(T)$ whose top view is the Voronoi diagram of the point set. The triangle mesh T is the convex hull of the points $\mathbf{v}_i \in S_i$, and its top view equals the associated Delaunay triangulation of the points $\mathbf{v}_i'$.

We now turn to general elliptic paraboloids ($\kappa_1 \neq \kappa_2$). Only convex vertex stars, equivalently convex hexagons V_O^*, can be considered visually smooth. Roughness requires different signs of dihedral angles. For the principal symmetric case, Eqn. (11) shows that α is always positive as we assume $0 < \phi < \pi/2$. Therefore the sign of β is crucial. The limit where convexity turns into non-convexity is $\beta = 0$ or in terms of ϕ convexity changes at $\tan\phi = \sqrt{\kappa_1/\kappa_2}$ (see Fig. 18, middle top). This particular angle ϕ determines so called *characteristic directions* $O\mathbf{a}'$, $O\mathbf{c}'$. These are the conjugate directions which are also principal symmetric. Recall that conjugate directions

(x_1, y_1) and (x_2, y_2) satisfy $\kappa_1 x_1 x_2 + \kappa_2 y_1 y_2 = 0$. Now the dihedral angle along the edge defined by $O\mathbf{b}$ is zero. The triangle mesh in that case is actually a quad mesh with planar faces where one family of diagonals in the quads are added.

This is just a special case of the fact that conjugate directions define planar quads (parallelograms) on the paraboloid. It does not require principal symmetry, and thus the following simple fact characterizes roughness.

A vertex star is rough if the top views of its boundary vertices lie in the interior of a wedge bounded by conjugate directions (see inset: red lines are a pair of conjugate lines; the conic is the Dupin indicatrix).



Roughness with exactly equal absolute values of angles α, β implies $\alpha = -\beta$ and therefore by Eqn. (13) we must have $\sin\phi_2 < 0$, which violates our assumption of $0 < \phi < \pi/2$.

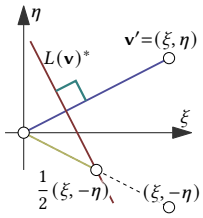
While we cannot get exactly equal absolute values of angles α, β for the 'roughness case' $\alpha\beta < 0$, it is still possible to control the level of roughness. For example, we may fix the ratio $\rho = \beta/\alpha$ of dihedral angles, which amounts to a quadratic equation for $\sin\phi$ with the solutions

$$(\sin\phi)_{1,2} = \frac{-\rho\kappa_2 \pm \sqrt{4\kappa_1^2 + 4\kappa_1\kappa_2 + \rho^2\kappa_2^2}}{2(\kappa_1 + \kappa_2)}.$$

Equal dihedral angles $\alpha = \beta$ in a mesh T with a smooth appearance are possible with ϕ_2 in Eqn. (12); see Fig. 32.

3.3.3 Hyperbolic paraboloid ($K < 0$). We let $\kappa_1 > 0, \kappa_2 < 0$. We may follow along the same path as for the elliptic type and first consider the special case $\kappa_1 = -\kappa_2 = 1$ of a hyperbolic paraboloid with orthogonal rulings in the top view.

The construction of V_O^* from the top view of the vertex star follows from Lemma 3.11. The top view $\mathbf{v}' = (\xi, \eta)$ of a boundary vertex $\mathbf{v}$ of V_O is scaled from O with factor $1/2$ and reflected at the ξ-axis. The diagram edge line $L(\mathbf{v})^*$ passes through the resulting point $\frac{1}{2}(\xi, -\eta)$ and is orthogonal to (ξ, η). Resulting shapes are shown in Fig. 19.



Hexagons V_O^* cannot be convex. For a vertex star with Euclidean regular top view, this follows from the shape of the envelope of possible edges of V_O^*, which turns to more general cases via Lemma 3.12; cf. also Prop. 3.16. We can obtain parallelograms as degenerate hexagons where two opposite edges degenerate to points. Again this case belongs to conjugate directions and quad meshes from parallelograms where one family of diagonals is added to make it formally a triangle mesh. This yields the fact that quad meshes with planar faces are visually smoother than any triangle mesh with non-vanishing dihedral angles [Pellis et al. 2019]. Note that there are still different signs in dihedral angles of a vertex star in these quad meshes.

Returning to proper triangle meshes and to the principal symmetric case for now. Eqn. (11) show that $\alpha < 0$ and $\beta > 0$. The smoothest

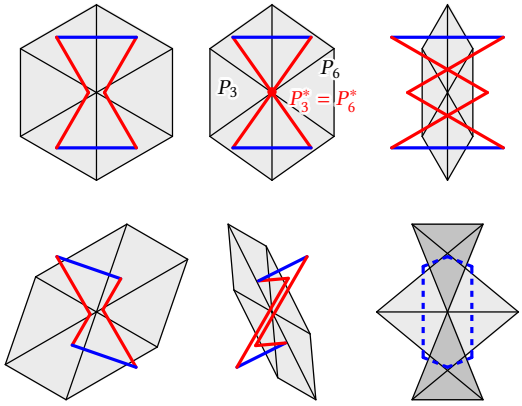
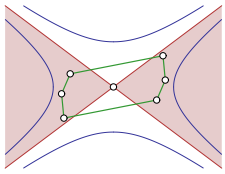


Fig. 19. Vertex stars (shaded) and their dual hexagons for a hyperbolic surface point. Different edge colors indicate angles of different signs. The top row shows principal symmetric stars. The star in the middle has four edges in asymptotic directions ($\phi = \operatorname{atan}\sqrt{|\kappa_1/\kappa_2|}$) and thus the marked triangles are coplanar ($P_3 = P_6$) and two vertices of the dual hexagon agree. Non-intersecting, convex hexagons can only arise from invalid vertex stars (*bottom right*).

appearance happens for hexagons V_O^* with a bowtie shape. The limit of a bowtie consists of two triangles. It happens when line A^* in the notation of Sec. 3.3.1 passes through O, i.e., by Eqn. (9), for $\kappa_n(\phi) = 0$. This characterizes an asymptotic direction. Edges in asymptotic direction lie in $z = 0$. A centrally symmetric vertex star V_O with four edges in asymptotic direction has two faces in $z = 0$ and V_O^* consists of two triangles meeting at O (Fig. 19, middle top). Principal symmetry is not needed for that.

This yields the following known fact (see [Jiang et al. 2015a; Pellis et al. 2019]): *A vertex star V_O at a hyperbolic point O is rough if the top views of its boundary vertices lie in the interior of a wedge bounded by the asymptotic directions.*



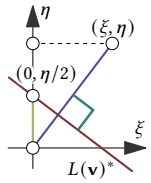
Due to the different signs of dihedral angles in a principal symmetric vertex star, $|\alpha| = |\beta|$ requires $\alpha = -\beta$ and thus Eqn. (13) applies. Using the curvature ratio $k = \kappa_1/\kappa_2$, it reads

$$\sin\phi_2 = -\frac{1}{1+k}.$$

Thus, we obtain a real ϕ_2 only for $k < -2$ (see Fig. 32).

3.3.4 Parabolic cylinder ($K = 0$). We let $\kappa_1 = 0$, $\kappa_2 > 0$. Up to scaling, the shapes of vertex stars and dual hexagons are the same as for $\kappa_2 = 1$, which we assume in the following. The construction of V_O^* from the top view of the vertex star follows from Lemma 3.11.

First, project the top view $\mathbf{v}' = (\xi, \eta)$ of a boundary vertex $\mathbf{v}$ of V_O parallel to the rulings or ξ-axis onto $\xi = 0$ to obtain $(0, \eta)$. Then, scale from O with factor $1/2$ yielding $(0, \eta/2)$. Now $L(\mathbf{v})^*$ passes through that point and is orthogonal to (ξ, η). Resulting shapes are shown in Fig. 20.



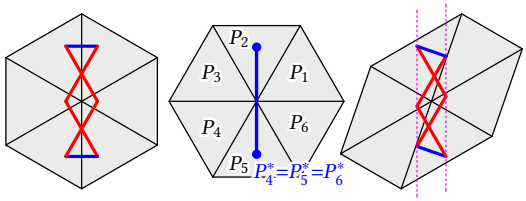


Fig. 20. Vertex stars (shaded) and their dual polygons for a parabolic surface point. Different edge colors indicate angles of different signs. The middle star has the six faces in two planes ($P_1 = P_2 = P_3$ and $P_4 = P_5 = P_6$) that intersect at a ruling of the parabolic cylinder. Thus the dual diagram is a straight line segment. In the non-degenerate cases, the dual hexagons have their vertices on two parallel lines which are orthogonal to the direction of vanishing principal curvature.

Only the very special case where two edges of a vertex star lie in a ruling through its vertex O yields a convex star. Its faces lie in two planes (Fig. 20, top right). These stars have been used by Stein et al. [2018] for the computation of triangle meshes representing developable surfaces. An entire triangulation with this type of vertex star has one family of dominant mesh polylines which lie in rulings. Omitting edges with vanishing dihedral angle, these are just prismatic surfaces. Among those are principal symmetric meshes which belong to $\kappa_2 = 0$ (thus $\alpha = 0$) in our discussion above.

Other cases exhibit some roughness. This is also seen in the different signs of α and β in the non-trivial principal symmetric vertex stars ($\kappa_1 = 0$),

$$\alpha = r\kappa_2 \tan\phi > 0, \quad \beta = -r\kappa_2 \frac{\sin^2\phi}{\cos\phi} < 0,$$

where obviously $\alpha = -\beta$ is not possible.

3.3.5 Summarizing view onto all types. Looking at the types of vertex stars in Fig. 18-20, one comes to the following observation, which we prove in the Appendix.

Proposition 3.16. *The dual polygon V_O^* to an affine-regular vertex star V_O with vertices on a paraboloid S and without coplanar faces is a centrally symmetric hexagon V_O^* whose vertices lie on an ellipse, a hyperbola or a pair of parallel lines, depending on whether S is (i) an elliptic paraboloid, (ii) a hyperbolic paraboloid or (iii) a parabolic cylinder, respectively. In the elliptic case (i), all vertices of V_O^* lie on the boundary of their convex hull. In the hyperbolic case (ii), two vertices lie inside the convex hull (parallelogram) of V_O^*. In the parabolic case (iii), the vertices lie on parallel lines that are orthogonal to the direction of vanishing principal curvature.*

For principal symmetric stars, a critical value is given by

$$C = \frac{|\kappa_1|\cos^2\phi - |\kappa_2|\sin^2\phi}{\cos\phi}. \tag{14}$$

At the border between smoothness and roughness we have $C = 0$ ($\phi = \operatorname{atan}\sqrt{|\kappa_1/\kappa_2|}$). $C < 0$ characterizes roughness, where $r|C|/2$ is the smallest distance of a vertex in the self-intersecting dual diagram from the principal edge (y-axis); by Prop. 3.16, this is a lower bound for the distance of the other vertices from the principal edge. For $\phi \to \pi/2$, C tends to ∞. $C > 0$ belongs to visual smoothness. Given C, ϕ follows from

$$\cos\phi = \frac{C + \sqrt{C^2 + 4(|\kappa_1| + |\kappa_2|)|\kappa_2|}}{2(|\kappa_1| + |\kappa_2|)}. \tag{15}$$

While $\cos\phi > 0$ is always fulfilled, $\cos\phi < 1$ happens for $C < |\kappa_1|$. The upper bound $C = |\kappa_1|$ is reached for $\phi = 0$ and characterizes a vertex star of valence 4 with edges in principal direction. The resulting mesh is a principal mesh, known as smoothest polyhedral surface representation [Pellis et al. 2019].

Affine-regular stars are good approximations of vertex stars in triangle meshes of regular combinatorics whose three families of main polylines resemble smooth curves. We have shown that the dual diagram is close to the Euclidean Gauss image (see Fig. 12). Therefore, our results are applicable to the subtle design of controlled roughness in meshes that otherwise have a smooth appearance of the edge pattern, as shown next.

4 Algorithms for mesh design with controlled roughness

Here we introduce several algorithms inspired by the geometric theory we developed in the previous section. First, we present an algorithm that uses remeshing as a tool to lay out a mesh with principal symmetric vertex stars with the desired level of roughness. Secondly, we elaborate on the manipulation of dual diagrams for fine-tuning dihedral angles at specific edges. We also provide details on our optimization framework to enforce design constraints.

Input: mesh $M = (V, E, F)$ or surface $S : U \subseteq \mathbb{R}^2 \to \mathbb{R}^3$
desired roughness level $\Upsilon : U \to \mathbb{R}_{\geq 0}$

1 $P \leftarrow \{\ v \mid v \in V\ \}$ or
$P \leftarrow \{\ S(u,v) \mid (u,v) \in U' \subseteq U\ \}$
2 $\kappa_1(p), \kappa_2(p), \mathbf{d}_1(p), \mathbf{d}_2(p) \leftarrow$ jet-fitting$(p),\ p \in P$
3 $\psi_*(p) \leftarrow \operatorname{atan}\left(\sqrt{|\kappa_1 \,/\, \kappa_2|}\right),\ |\kappa_1| \leq |\kappa_2|,\ p \in P$
4 $\psi(p) \leftarrow \Upsilon(p)\,\frac{\pi}{2} + (1 - \Upsilon(p))\,\psi_*(p),\ p \in P$
5 $\mathbf{a}(p) \leftarrow$ rotate$(\mathbf{d}_1(p), \psi(p), p, \mathbf{n}(p)),\ p \in P$
i.e. rotate $\mathbf{d}_1(p)$ by $\psi(p)$ about $\mathbf{n}(p)$
6 $\mathbf{c}(p) \leftarrow$ reflect$(\mathbf{a}(p), \mathbf{d}_1(p), p, \mathbf{n}(p)),\ p \in P$
i.e. reflect $\mathbf{a}(p)$ across $\mathbf{d}_1(p)$
7 $\mathbf{b}(p) \leftarrow$ bisector$(\mathbf{a}(p), \mathbf{c}(p)),\ p \in P$
8 frame field $\mathcal{F} \leftarrow \{\ (\mathbf{a}(p), \mathbf{c}(p), -\mathbf{a}(p), -\mathbf{c}(p)) \mid p \in P\ \}$
9 $\mathcal{X} \leftarrow$ frame-field-deformation$(\mathcal{F})$
10 $\mathcal{X} \leftarrow$ cross-field-interpolation$(\mathcal{X})$
11 $\mathcal{X} \leftarrow$ comb-and-cut$(\mathcal{X})$
12 $\Omega = \{\ (u,v)\ \} \subseteq \mathbb{R}^2 \leftarrow$ miq-global-parameterization$(P, \mathcal{X})$
13 $Q \leftarrow$ quad-mesh-extraction(P, Ω)
14 $T \leftarrow$ triangulate$(Q, \{\ \mathbf{b}(p) \mid p \in P\ \})$
15 $T \leftarrow$ optimize(T, P, Υ)

Return: a triangle mesh T with controlled roughness

Alg. 21. We can use principal symmetric vector fields to lay out a mesh on a surface that is already adapted to roughness design constraints. Details on the optimization routine (line 15) are provided in Sec. 4.3.

4.1 Remeshing as a design tool for controlled roughness

Drawing on the results from Sec. 3.3, we know that a vertex star of a mesh at an *elliptic* (resp. *hyperbolic*) point of the underlying surface is rough if the top views of its boundary vertices lie in the interior of a wedge bounded by *conjugate* (resp. *asymptotic*) directions.

Our algorithm begins by sampling a smooth surface at regularly spaced locations, P. If given a polyhedral mesh instead, we consider the mesh vertices as the sample points. We compute principal curvature values κ_1, κ_2 and directions $\mathbf{d}_1, \mathbf{d}_2$ by estimating a locally best-fit quadric at each of these points $p \in P$, according to the jet-fitting method of [Cazals and Pouget 2005].

We select principal direction $\mathbf{d}_1$ such that $|\kappa_1| \leq |\kappa_2|$ and denote by ψ the angle between an edge in a mesh vertex star and $\mathbf{d}_1$. Smoothness changes into roughness at $\tan(\psi_*) = \sqrt{|\kappa_1 \,/\, \kappa_2|}$ (see Sec. 3.3).

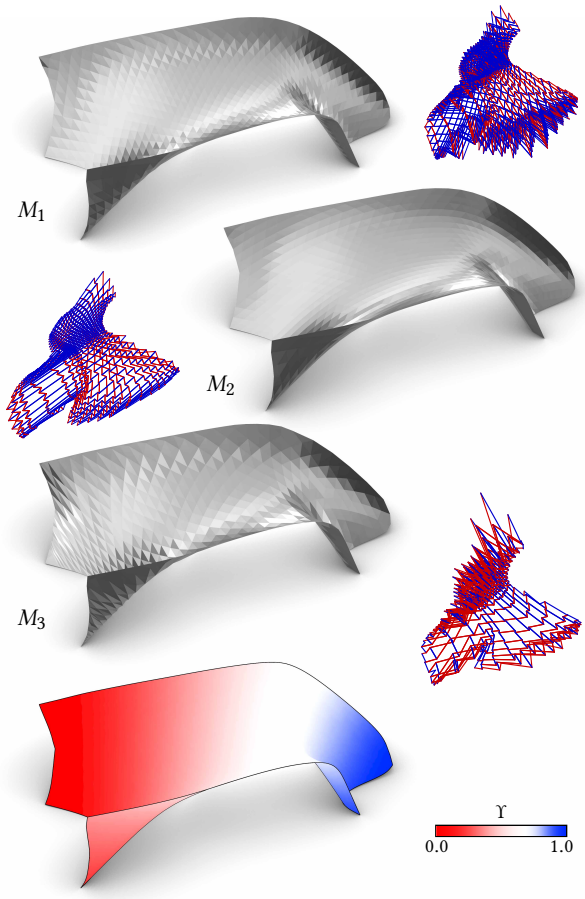


Fig. 22. A surface designed by Zaha Hadid for the Nuragic and Contemporary Art Museum in Cagliari, Italy, is remeshed following Alg. 21 with $\Upsilon = 0.5$, $\psi = \frac{\pi}{4} + 0.5\,\psi_*$, $\mathcal{R}(M_1) = 0.092$ (*top*). Inserting the opposite diagonal in the intermediate quad mesh results in a smoother mesh, $\mathcal{R}(M_2) = 0.046$ (*middle*), confirming the wedge conditions from Sec. 3.3. By varying Υ across the surface, we can achieve an appearance that subtly varies from smooth to rough, even in negatively-curved areas, $\mathcal{R}(M_3) = 0.135$ (*bottom*). Scalar field Υ is provided as a colormap over the surface.

The interval $\psi \in [\psi_*, \pi/2]$ defines the bounds for the wedge of admissible (rough) vertex stars which have two edges in direction $\mathbf{d}_2$. Our choice of $\mathbf{d}_1$ allows us to work in the wider of the two wedges spanned by the critical directions, since $\psi_* \leq \pi/4$. The wedge spans the complete plane at parabolic points, where $\psi_* = 0$.

We construct principal symmetric vector fields at an angle ψ from $\mathbf{d}_1$. The angle ψ is free to vary over the input surface, and represents the main method of roughness control by the user. We found the most intuitive input to be a non-negative scalar field Υ over a parameterization of the surface (see Fig. 22). Alternatively, a field $C(p)$ for the critical value (14) can steer roughness/smoothness and obtain the corresponding angle $\psi(p)$ from Eqn. (15), as illustrated in Fig. 23.

We use the frame field interpolation algorithm from [Panozzo et al. 2014] to perform an as-rigid-as-possible deformation into orthogonal vector fields (or *cross field*), which can then be used to compute a global parametrization via mixed-integer quadrangulation [Bommes et al. 2009]. The resulting integer-grid map is used to extract a quad meshing of the surface using the method of [Ebke et al. 2013].

The quad mesh is triangulated by inserting, for each face, the diagonal that bisects the original principal symmetric vectors. This is the diagonal that is most closely aligned with the principal curvature direction $\mathbf{d}_2$. Notably, inserting the other set of diagonals results in a comparably smoother mesh (see Fig. 22).

Optionally, the resulting triangle mesh is finally optimized to fine-tune the roughness control according to the user constraints. This algorithm is summarized in Alg. 21 and details on the optimization routine are provided in Sec. 4.3.

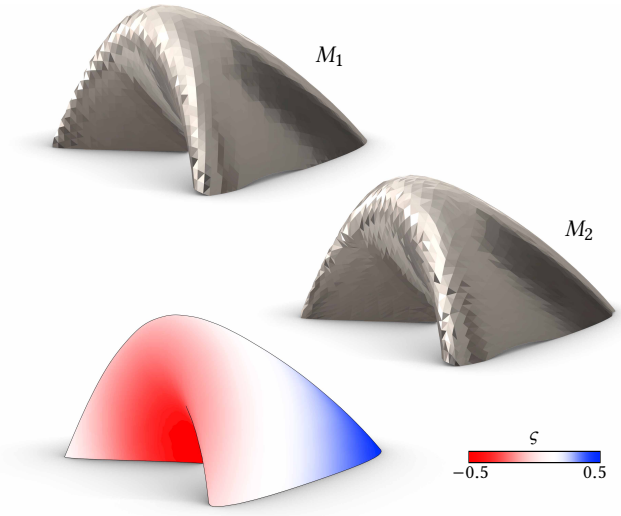


Fig. 23. A smooth mesh, $\mathcal{R}(M_1) = 0.056$ (*top*) is roughened slightly, resulting in a mesh with $\mathcal{R}(M_2) = 0.066$ (*middle*). We design the controlled roughness following Alg. 21, but instead of using a scalar field Υ, we compute $\psi(p)$ in line 4 via the inverse cosine of Eqn. (15). The field of critical values $C(p)$ is computed as $C(p) = \varsigma(p)|\kappa_1|$, with $\varsigma(p) \in (-0.5, 0.5)$ (see colormap, *bottom*).

Input: (possibly rough) mesh $M = (V, E, F)$ to be adjusted
edge $e \in E$ with dihedral angle to be modified
new dihedral angle ϑ for edge e
radius r of the area of influence for local changes

1 $\mathcal{N}_e \leftarrow \{ f \mid f \cap e \neq \emptyset, f \in F \} \subseteq F$
2 $\mathcal{N}^r \leftarrow \{ f \mid \mathrm{dist}(f, e) \leq r, f \in F \} \subseteq F$
3 $\mathbf{n}_e \leftarrow \sum_{f \in \mathcal{N}_e} \mathbf{n}_f \; / \; \| \sum_{f \in \mathcal{N}_e} \mathbf{n}_f \|$
4 $C \leftarrow$ local coordinate system with z-axis aligned with $\mathbf{n}_e$
5 $\Sigma \leftarrow \sigma(\mathcal{N}_e)$, dual image of $\mathcal{N}_e$ in coordinate system C
6 $\mathcal{N}_e^* \leftarrow$ dual diagram of $\mathcal{N}_e$ in coordinate system C
7 $M' \leftarrow M$
8 **while** *user interaction* **do**
9 $\quad \Sigma' \leftarrow$ apply user deformations on $\mathbf{e}^* \in \mathcal{N}_e^*$
10 $\quad \mathcal{N}_e' \leftarrow \sigma(\Sigma')$
11 $\quad M' \leftarrow \mathrm{optimize}(M', e, \vartheta, \mathcal{N}_e', \mathcal{N}^r)$
12 **end**

Return: mesh M' with local changes in controlled roughness

Alg. 24. The visualization provided by the dual diagram of a mesh can double as an interactive design tool for the local adjustment of dihedral angles. Details on the optimization routine (line 11) are provided in Sec. 4.3.

4.2 Interactive manipulation of the dual diagram

We discuss the algorithmic details of the manipulation tool introduced in Sec. 2.3. Given a (possibly rough) polyhedral surface $M = (V, E, F)$, the user wants to modify the dihedral angle α_e at edge $e \in E$ so that it meets certain requirement, i.e. $\alpha_e = \vartheta$.

First, we extract the set of faces $\mathcal{N}_e \subseteq F$ that share a vertex with edge e. We compute a locally adapted Cartesian coordinate system C, with z-axis aligned with the average normal $\mathbf{n}_e \parallel \sum_{f \in \mathcal{N}_e} \mathbf{n}_f$. The dual diagram of $\mathcal{N}_e$ in coordinate system C is presented to the user, where they can manipulate the segment $\mathbf{e}^*$ corresponding to edge $e = f_l \cap f_r$, where $f_l, f_r \in F$. In practice, manipulating the edge $\mathbf{e}^*$ in the dual diagram, instead of the faces f_l, f_r in 3D Euclidean space, removes unintuitive degrees of freedom in the relative orientation of the faces, which can be recovered afterwards by applying rigid motions on M.

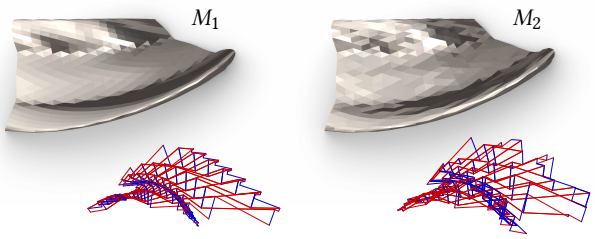


Fig. 25. Our automatic roughening optimization is very flexible despite its simplicity. Different combinations of the scaling factors λ_e, $\lambda_f^{(1)}$, and $\lambda_f^{(2)}$ yield very distinct controlled roughness. Here, a smooth mesh with $\mathcal{R}(M_1) = 0.099$ (*left*), is roughened with $\lambda_e = 1.01$ and $\lambda_f^{(1)} = 0.1 - \lambda_f^{(2)}$, resulting in $\mathcal{R}(M_2) = 0.114$ (*right*).

The user deformations of $\mathbf{e}^*$ are mapped back onto M. Clearly, any changes in the mesh are confined to $\mathcal{N}_e$. The user has the option to specify a neighbourhood $\mathcal{N}$, with $\mathcal{N}_e \subseteq \mathcal{N}$, where the changes in dihedral angles are smoothed out. The mesh M is then optimized in order to satisfy these angle requirements with high accuracy. This algorithm is summarized in Alg. 24.

4.3 Optimization algorithm

We have mainly dealt so far with the subtle topic of designing triangle meshes that already exhibit the desired level of mild roughness. These meshes have their vertices on given reference surfaces. We can also introduce and control roughness by allowing for small deviations of vertices from the reference surfaces. To that end, we present a flexible optimization framework that can be used both for automatic roughening and for fine control of dihedral angles under orientation constraints.

In the following discussion, we are given an initial triangle mesh $M^0 = (V^0, E, F)$ either smooth or resulting e.g. from the remeshing algorithm of Sec. 4.1. We aim to design a rougher mesh $M = (V, E, F)$. We can handle arbitrary mesh combinatorics, but we assume that we have a consistent orientation of faces.

Automatic roughening. Roughening a given triangulation is not necessarily a simple task, especially if the mesh M^0 is not combinatorially regular. An effective automatic way is inspired by the fact that wrinkles appear if the surface area of a mesh is enlarged while enforcing proximity to a reference surface. We achieve this by increasing the edge lengths in the starting mesh M^0. Within the optimization, this is enforced by the following constraint on mesh edges $e = v_i v_j \in E$:

$$c_{\mathrm{rough}}(e) := \|v_i - v_j\|^2 - \lambda_e \|v_i^0 - v_j^0\|^2 = 0.$$

Here $v_i^0 \in M^0$ denotes the initial location of vertex $v_i \in M$. The scaling factors $\lambda_e \geq 1$ for edges $e \in E$ can be used to drive the control of roughness in the result. They may be taken constant for uniform roughening, or may be defined as a scalar-valued function on M^0 (see Fig. 26).

To achieve a wrinkling effect, we must keep proximity to the initial mesh M^0. We compute the barycenter $b_f = \sum_{v \in f} v \; / \; |f|$ of each face $f \in F$, and its closest point projection b^* on the initial

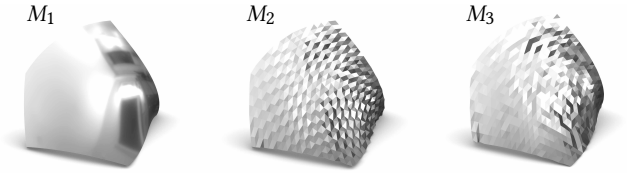


Fig. 26. Roughening of a smooth triangle mesh with regular combinatorics and $\mathcal{R}(M_1) = 0.064$ (*left*) can be achieved purely through optimization. Different proximity terms to the original mesh achieve visually different roughness. Minimizing point-to-point distances (i.e., $c_{\mathrm{ref},2}$) leads to a more structured roughness (*middle*), $\mathcal{R}(M_2) = 0.366$, than minimizing tangent-plane distances (i.e., $c_{\mathrm{ref},1}$) (*right*), $\mathcal{R}(M_3) = 0.156$. A linearly varying scaling factor λ_e across the mesh results in a subtle transition from a smooth to a rough appearance.

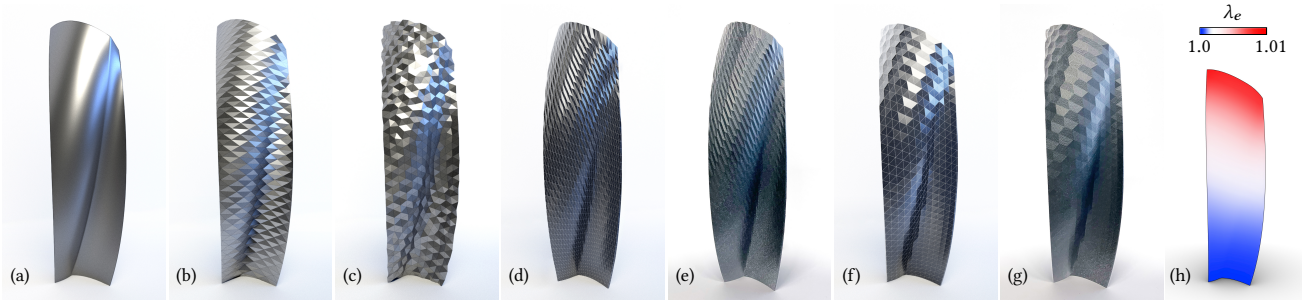


Fig. 27. *Comparing different roughening algorithms.* (a) A smooth mesh of a surface, with $\mathcal{R}(M_a) = 0.016$. (b) The surface is roughened using Alg. 21, resulting in $\mathcal{R}(M_b) = 0.340$. (c) A mesh with irregular combinatorics is roughened using our automatic roughening algorithm, $\mathcal{R}(M_c) = 0.317$. The original triangulation has a notable impact on the visual appearance of controlled roughness: (d) and (f) show two different regular triangulations of the same surface, optimized following *automatic roughening*, and with $\mathcal{R}(M_d) = 0.378$ and $\mathcal{R}(M_f) = 0.249$. Corresponding 3D-printed models are shown in (e) and (g). The same scalar factors λ_e were used in (c), (d) and (f), and are shown in (h).

mesh M^0, which falls within some face $f(b^*)$ of M^0. We approximate the distance field of M^0 by the distances of b_f to the tangent planes at b^*. Therefore, proximity is imposed via the constraint:

$$c_{\text{ref},1}(f) := \lambda_f^{(1)} \langle b_f - b^*,\ \mathbf{n}^0_{f(b^*)} \rangle = 0,$$

where $\mathbf{n}^0_{f(b^*)}$ denotes the initial unit normal at $f(b^*) \in F$. The closest point projections b^* and corresponding faces $f(b^*)$ are recomputed after each iteration of the optimization.

The scaling factors $\lambda_f^{(1)} \geq 0$ for faces $f \in F$ can also be used to control roughness. In general, greater values enforce more proximity, and thus greater roughness when combined with the previous constraints (see Fig. 25).

We can combine this with, or completely replace it with, point-to-point distances that restrict the deviation of the barycenter b_f from its original position b_f^0:

$$c_{\text{ref},2}(f) := \lambda_f^{(2)} \|b_f - b_f^0\|^2 = 0.$$

The different behavior of the two types of barycenter proximity constraints is illustrated in Fig. 26.

Constraints on dihedral angles. After remeshing or manipulation of the dual diagram, or in combination with automatic roughening, one may want to prescribe dihedral angles for selected edges. We also have to consider the orientation of an angle. Using terminology from origami, we classify each edge $e \in E$ as a mountain edge or a valley edge.

Given explicit dihedral angle requirements α_e at (possibly sparse) edges $e = f_l \cap f_r \in E$, these can be enforced in the optimization with the constraint:

$$c_{\text{angle},0}(e) := \langle \mathbf{n}_{f_l}, \mathbf{n}_{f_r} \rangle - \cos(\alpha_e) = 0,$$

where $\mathbf{n}_f$ denotes the unit normal of face $f \in F$. These normal vectors are implicitly constrained by

$$c_{\text{norm},0}(f) = \langle \mathbf{n}_f, \mathbf{n}_f \rangle - 1 = 0, \quad c_{\text{norm},i}(f) = \langle \mathbf{n}_f, v_i - v_{i+1} \rangle = 0,$$

where $f = v_i v_{i+1} \cdots \in F$. Normal vectors are recomputed after each iteration of the optimization.

Alternatively, a roughening design may allow for dihedral angles within an admissible range $[\alpha_e^{\min}, \alpha_e^{\max}]$ for edge $e \in E$. These bounds may also be specified as deviation intervals from initial dihedral angles α_e^0. In this scenario, we prescribe inequality constraints $c(x) \geq 0$, which we then turn into equality constraints $c(x) - \delta^2 = 0$ with dummy variables δ. Since we only deal with rather small dihedral angles, certainly within $[0, \pi/2]$, the resulting constraints on edges $e = f_l \cap f_r \in E$ are expressed by

$$c_{\text{angle},1}(e) := \langle \mathbf{n}_{f_l}, \mathbf{n}_{f_r} \rangle - \cos(\alpha_e^{\min}) + \delta_{e,1}^2 = 0,$$
$$c_{\text{angle},2}(e) := \langle \mathbf{n}_{f_l}, \mathbf{n}_{f_r} \rangle - \cos(\alpha_e^{\max}) - \delta_{e,2}^2 = 0.$$

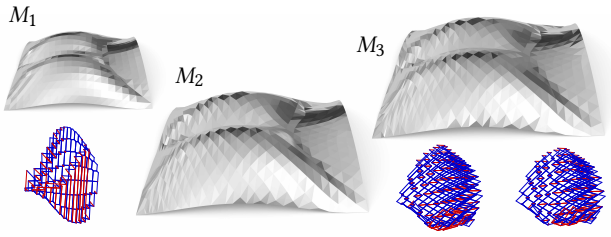


Fig. 28. A smooth mesh, $\mathcal{R}(M_1) = 0.099$ (*left*), is first roughened using our algorithm based on remeshing, $\mathcal{R}(M_2) = 0.153$ (*middle*). Subsequently, the roughness level is controlled by optimizing with $\alpha_e^{\max} = 0.8\,\alpha_e^0$, while preserving the feature lines, $\mathcal{R}(M_3) = 0.137$ (*right*). The resulting change in dihedral angles is visualized in the dual diagrams by corresponding edges of shorter lengths.

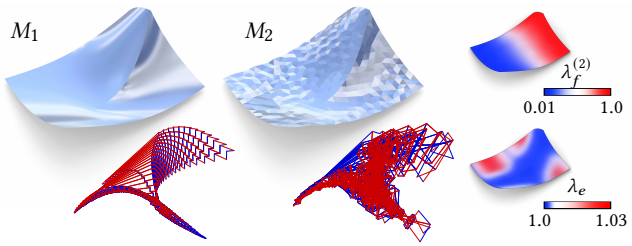


Fig. 29. Our roughening algorithms can preserve feature lines in the original mesh. Here, a smooth mesh, $\mathcal{R}(M_1) = 0.031$ (*left*), is roughened using the automatic roughening approach, $\mathcal{R}(M_2) = 0.127$ (*middle*), by letting $\lambda_e \to 1$ near the feature line (see colormaps, *right*).

Table 1. *Optimization Details.* For selected examples, we give the number of vertices, edges, and faces, of the mesh, as well as input scalars and weights used in the optimization. Each example uses only the constraints whose corresponding weight values are provided. We also indicate the energy of the objective function at convergence $\mathcal{E}$, as well as the number of iterations, and the total time (in milliseconds) used for optimization.

Fig.	|V|	|E|	|F|	λ_e	$\lambda_f^{(1)}$	$\lambda_f^{(2)}$	$w_{\text{ref},1}$	$w_{\text{ref},2}$	$w_{\text{angle},0}$	$w_{\text{angle},k}$	w_{norm}	w_{orient}	$\mathcal{E}$	t (ms)	#it
1, *left*	1689	4877	3198	1.03	[.01, .1]	[.01, .1]	1.0	1.0	50.0	200.0	1.0	0.7	$2.38 \cdot 10^1$	$4.6 \cdot 10^3$	61
5, *right*	348	976	629	[1.0, 1.06]		1.0		.9					1.33	36	5
26, *middle*	1148	3307	2160	[1.0, 1.05]	1.0		.1						7.02	502	18
26, *right*	1148	3307	2160	[1.0, 1.05]		1.0		.9					$2.66 \cdot 10^1$	180	6
25, *right*	1326	3871	2545	1.01	[.01, .1]	[.01, .1]	1.0	1.0		100.0	1.0	5.0	$2.97 \cdot 10^1$	328	14
27 (c)	549	1518	979	[1.0, 1.01]		1.0		.9					$6.02 \cdot 10^{-1}$	105	7
27 (d)	560	1573	1014	[1.0, 1.01]		1.0		.9					$3.00 \cdot 10^{-1}$	172	8
27 (f)	560	1573	1014	[1.0, 1.01]		1.0		.9					$1.88 \cdot 10^{-1}$	195	6
28	476	1328	853						100.0	100.0	1.0	5.0	$1.09 \cdot 10^{-5}$	273	23
29, *middle*	517	1470	954	[1.0, 1.03]		[.01, 1.0]		1.0					2.47	99	11
30, *middle*	2116	6085	3970	[1.0, 1.02]	1.0	1.0	.1	.6	50.0	100.0	1.0	1.0	$1.07 \cdot 10^1$	$6.4 \cdot 10^3$	38
31	340	947	608							100.0	1.0	1.0	$4.02 \cdot 10^{-6}$	846	52

Our topic also requires the consideration of oriented dihedral edge angles (see Fig. 14). Since $\cos(\alpha_e) = \cos(-\alpha_e)$, we do not work with signs of angles, but express the correct behavior with respect to a consistent orientation of face normal vectors in M. Simplifying the approach from Fig. 14, we consider oriented edges (*half-edges*) $\vec{e} = f_l \cap f_r$ with f_l to the left and f_r to the right of $\vec{e}$. We compute half-edge vectors $\mathbf{u}_{\vec{e}} = b_{f_l} - m_e$ going from the mid point $m_e = (v_i + v_j)/2$ of the corresponding edge $e = v_i v_j$ to the barycenter b_{f_l} of adjacent face $f_l \in F$. We constrain the inner product of the half-edge vectors with the average normal vector $\mathbf{n}_{f_l} + \mathbf{n}_{f_r}$ at $e \in E$:

$$c_{\text{orient}}(\vec{e}) := \langle \mathbf{n}_{f_l} + \mathbf{n}_{f_r}, \mathbf{u}_{\vec{e}} \rangle - \sigma_e \delta^2_{\vec{e},s} = 0.$$

Here, $\sigma_e = 1$ for a valley edge (as shown in Fig. 14) and $\sigma_e = -1$ for a mountain edge. This constraint ensures that we preserve the type of edges during optimization. The type of an edge $e \in E$ is initially computed; edges with $\alpha_e^0 < \tau$ are assumed to have no orientation, as adjacent faces f_l, f_r are nearly planar, and are subsequently skipped. Here, τ is a user-set tolerance, typically smaller than half a degree and chosen based on visual considerations during the design process.

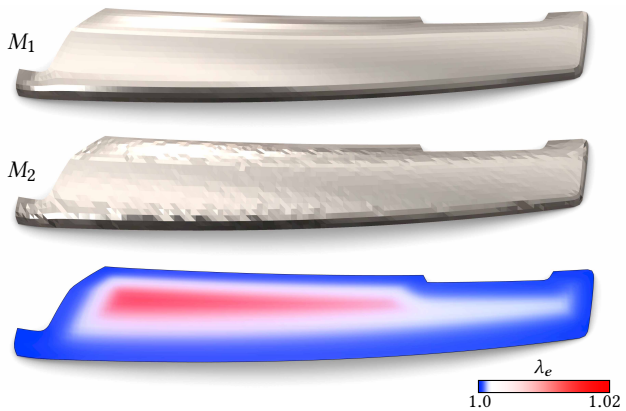


Fig. 30. The orientation of dihedral angles σ_e can be determined by mapping a 3D texture onto a smooth surface, $\mathcal{R}(M_1) = 0.033$ (*top*). Here, a car dashboard is later optimized while combining automatic roughening with constraints on dihedral angles, $\mathcal{R}(M_2) = 0.053$ (*middle*). Corresponding scalars λ_e are shown via color mapping (*bottom*).

Optimization. Automatic roughening can be applied as a standalone algorithm. In such an application, the required user inputs are the initial mesh $M^0 = (V^0, E, F)$ together with scalars λ_e, $\lambda_f^{(1)}$, and $\lambda_f^{(2)}$. The variables in the optimization are the vertices $v \in V$. These vertices can be directly initialized from the initial mesh, $v_i = v_i^0$, or initialized via 3D texture mapping (see Fig. 31). We define energy functionals via weighted sums of squares of the constraints:

$$\mathcal{E}_{\text{rough}} = \sum\nolimits_{e \in E} c_{\text{rough}}(e)^2, \quad \mathcal{E}_{\text{ref}} = \sum\nolimits_k w_{\text{ref},k} \sum\nolimits_{f \in F} c_{\text{ref},k}(f)^2.$$

The optimization minimizes the objective function $\mathcal{E} = \mathcal{E}_{\text{rough}} + \mathcal{E}_{\text{ref}}$. The weights $w_{\text{ref},1}$ and $w_{\text{ref},2}$ need be chosen according to the particular application (see Table 1).

In combination with automatic roughening, constraints on dihedral angles are often imposed on all edges $e \in E$, $E' = E$. If the optimization is run after the remeshing algorithm of Sec. 4.1, then the set of active edges $E' \subseteq E$ is computed by proximity to the sample points $p \in P$. The scalar field Υ is used to adjust the target dihedral angles α_e. Similarly, if the optimization is run after the interactive manipulation of a dual diagram, the set of active edges $E' \subseteq E$ is taken from the neighborhoods $\mathcal{N}'_e$ and $\mathcal{N}^r$. The angle ϑ enters as the target angle for edge $e \in E$, i.e., $\alpha_e = \vartheta$. The variables in these optimizations are the vertices $v \in V$, a unit normal vector $\mathbf{n}_f$ for each face $f \in F'$ adjacent to some $e \in E'$, auxiliary variables $\delta_{e,1}$ and/or $\delta_{e,2}$ for $e \in E'$ (if α_e is specified by bounds) and auxiliary variable $\delta_{\vec{e},s}$ for each half-edge $\vec{e}$. The corresponding energy functionals

$$\mathcal{E}_{\text{angle},k} = \sum\nolimits_{e \in E'} c_{\text{angle},k}(e)^2, \quad \mathcal{E}_{\text{norm}} = \sum\nolimits_k \sum\nolimits_{f \in F'} c_{\text{norm},k}(f)^2,$$
$$\mathcal{E}_{\text{orient}} = \sum\nolimits_{f \in F'} \sum\nolimits_{\vec{e} \in f} c_{\text{orient}}(\vec{e})^2,$$

are weighted and summed into the objective function:

$$\mathcal{E} = \sum\nolimits_k w_{\text{angle},k}\, \mathcal{E}_{\text{angle},k} + w_{\text{norm}}\, \mathcal{E}_{\text{norm}} + w_{\text{orient}}\, \mathcal{E}_{\text{orient}}$$

for minimization. Again, the weights $w_{\text{angle},k}$, w_{norm} and w_{orient} depend on the particular application (see Table 1). This objective function may be combined with automatic roughening into a single minimization routine.

5 Results and discussion

Implementation. All of our computational design tools were implemented as *Grasshopper* components. Grasshopper is a visual programming environment for procedural geometry generation, embedded within the *Rhinoceros3D* computer-aided design application, and widely used across industrial design.

We used McNeel's openNURBS open-source toolkit for geometric manipulations. We adapted the frame field interpolation algorithm provided in *libigl* [Jacobson et al. 2018], and used the constrained mixed-integer solver CoMISo to compute global parametrizations [Bommes et al. 2010]. Quad mesh extraction was performed with the help of OpenMesh [Botsch et al. 2002] and *libQEx*.

We developed our own unmanaged dynamic-link library from Intel® oneAPI Math Kernel static library, and a NET wrapper to interface from C#. This was used for efficient sparse matrix manipulations. Similarly, we used Intel®'s oneMKL pardiso for solving large sparse symmetric linear systems. The optimization procedures introduced in Sec. 4.3 were solved following the guided-projection algorithm of [Tang et al. 2014]. This is equivalent to using a Levenberg-Marquardt method according to [Madsen et al. 2004], with at most quadratic constraints.

Numerical optimization. Throughout the optimizations, we use a damping parameter in the Levenberg-Marquardt algorithm of 10^{-6}. The input meshes are normalized to unit average edge length for stability. Table 1 provides statistics on the size of optimization problems, the choice of weights, and the time needed. Times are given in milliseconds (ms) and refer to an Intel® Xeon® W-2225 CPU @ 4.10GHz, running 64-bit Windows® 10. Our implementation is configured to use 64 parallel openMP threads, but could be tuned depending on the application.

The objective functions from Sec. 4.3 are not zero-residual functionals. Thus, we impose a termination criterion based on the order of magnitude of the decrease in the total energy of the objective function. The optimization is terminated when the total energy decrease, on iteration, does not exceed at least one order of magnitude smaller than the order of magnitude of the total energy at the current iteration.

In practice, whenever the $\mathcal{E}_{\text{rough}}$ energy term is used, the optimization reaches the termination criterion after 10 to 20 iterations (e.g. Fig. 26 and Fig. 27). When used in combination with $\mathcal{E}_{\text{angle},k}$ terms, the optimization requires 3x to 4x the number of iterations (e.g. Fig. 31).

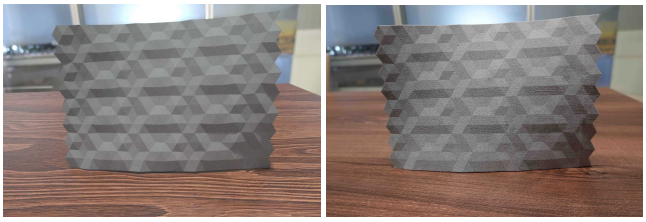


Fig. 31. A smooth surface was roughened via vertex displacements along vertex normals, $v_i^r = v_i + 0.1\, n_i$, following a regular 3D texture. The resulting mesh had a maximum dihedral angle $\max_e \alpha_e = 16.4°$. It was subsequently optimized with $\alpha_e^{\max} = 10°$ for all $e \in E$. The texture used in the initialization is clearly visible in the result, $\mathcal{R} = 0.075$. Remarkably, our rendering (*left*) closely captures the visual appearance of the fabricated model (*right*).

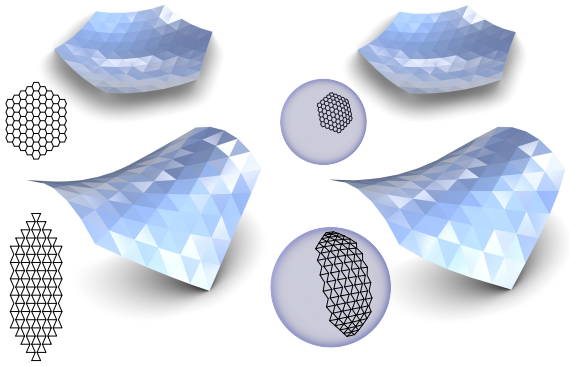


Fig. 32. Turning constant isotropic dihedral angles into constant Euclidean angles. Using Eqn. (12) and Eqn. (13), triangle meshes with constant isotropic dihedral angles on paraboloids are computed (*left*), seen in constant Euclidean edge lengths in the dual diagrams. Optimization towards constant Euclidean angles (*right*) succeeds and is confirmed by constant spherical edge lengths in the Gauss images.

Fabrication. The models in Fig. 27 (e) and (g), and Fig. 31 (right), were fabricated using a Multi Jet Fusion (MJF) 3D printer with Nylon PA12 as material. All models have a thickness of 1.4mm, and the average normal vertex displacement is 0.15mm.

Validation. As explained in Sec. 2, roughness is a property of the visual appearance of a surface. There is no absolute measure for the ideal controlled roughness, and consequently, the objective functional in the automatic roughening optimization will not reach a zero residual. However, our normalized roughness measure $\mathcal{R}$ from Sec. 2.1 proves to be a consistent indicator when comparing different triangulations of the same underlying surface (see Fig. 4-7 and Fig. 22-30). The dual diagrams also both correctly illustrate the local changes in dihedral angles (see Fig. 10 and Fig. 28) and characterize the roughness across a surface (see Fig. 7 and Fig. 22). When replicating lighting conditions, the visual appearance of the fabricated models closely resembles our renderings (see Fig. 27 and Fig. 31).

Euclidean angles. Isotropic geometry not only considerably simplified the local analysis of the behavior of vertex stars depending on curvature. It is also useful to obtain initial guesses for otherwise difficult Euclidean problems. An example for that is shown in Fig. 32, where constant Euclidean dihedral angles on a mesh with smooth appearance is achieved by this idea of isotropic initialization. Similarly, Fig. 33 shows an example with two unique Euclidean dihedral angles: one constant Euclidean dihedral angle for mountain edges, and a different constant Euclidean dihedral angle for valley edges. Constant dihedral angles are a type of element repetition in architectural structures. Here, it concerns the supports of flat panels along beams that are aligned with mesh edges.

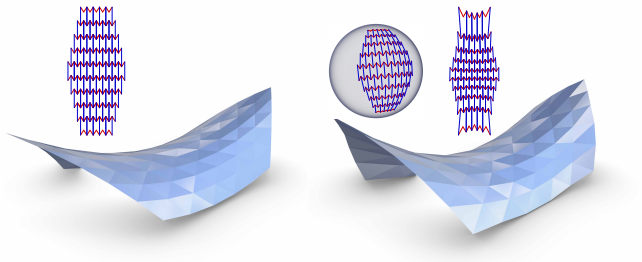


Fig. 33. We may also use Sec. 3.3 to compute triangle meshes with different constant isotropic dihedral angles on edges of different type, resulting in a constant ratio ρ (*left*). Optimization achieves a constant ratio of constant Euclidean angles (*right*), shown by the spherical edge lengths in the Gauss image. The initially constant Euclidean edge lengths in the dual diagram change more the further from the saddle point O on the paraboloid.

Limitations. Due to the novelty of the topic we had to limit our scope. Apart from that, a technical limitation is that we based remeshing on principal symmetric vector fields. Our remeshing algorithm works best with at most isolated singularities. This is a result of using an off-the-shelf remesher, such as [Bommes et al. 2009]. Our contribution focuses on *guiding* such remeshings to achieve controlled roughness, but it would benefit from a more stable and advanced remesher. For instance, our current remeshing pipeline does not guarantee the preservation of feature lines. This can be achieved to a certain degree by aligning vector fields —that is, given a feature line direction $\mathbf{u}(p)$, by setting $\psi(p)$ in line 4 of Alg. 21 to the angle between $\mathbf{u}(p)$ and $\mathbf{d}_1(p)$ (see Fig. 28). Alternatively, preservation of feature lines *can* be enforced in *automatic roughening* by excluding relevant vertices from the optimization (see Fig. 29).

Our contributions focus on *design* tools, and the proposed roughness control mechanisms might not generate the desired controlled roughness across geometries in the wild. Similarly, various further tools which one may want to have in a design environment, e.g. related to an enumeration of specific patterns, are probably less interesting for a research paper. Many of them could be created in a 3D texture mapping approach as in Fig. 3.

Future work. The presented methodology can form the basis for further research. This includes mesh design away from principal symmetry and the study of other polyhedral surfaces (meshes with planar faces). Albeit only hinted at, Sec. 2 already contains theory applicable to Phex meshes, as metric dual objects to triangle meshes. Also note that our optimization framework in Sec. 4.3 works for meshes with planar faces with an arbitrary number of vertices.

Even more interesting may be the design of controlled roughness in polyhedral patterns formed by different types of faces, generalizing the ones with smooth appearance in [Jiang et al. 2015b]. Moreover, the inclusion of lighting conditions and appearance from typical viewpoints could enter the design stage.

Acknowledgments

This work was supported by the Austrian Science Fund via grants I~4868 (SFB-Transregio programme Discretization in geometry and dynamics), and F77 (SFB grant Advanced Computational Design). V. Ceballos and F. Rist were supported by KAUST baseline funding.

References

Franz Aurenhammer, Rolf Klein, and Der-Tsai Lee. 2013. *Voronoi diagrams and Delaunay triangulations.* World Scientific Publishing Co. Pte. Ltd., Singapore.

Martine Ben Amar and Yves Pomeau. 1997. Crumpled paper. In *Proceedings: Mathematical, Physical and Engineering Sciences*, Vol. 453. The Royal Society, London, UK, 729–755.

Phillipe Block and John Ochsendorf. 2007. Thrust Network Analysis: A New Methodology For Three-Dimensional Equilibrium. *Journal of the International Association for Shell and Spatial Structures* 48, 3 (2007), 167–173.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Transactions on Graphics* 28, 3 (2009), 77:1–77:10.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2010. Practical Mixed-Integer Optimization for Geometry Processing. In *Proceedings of the 7th International Conference on Curves and Surfaces.* Springer, Berlin-Heidelberg, Germany, 193–206.

Mario Botsch, Stephan Steinberg, Stephan Bischoff, and Leif Kobbelt. 2002. OpenMesh – A Generic and Efficient Polygon Mesh Data Structure. In *Proceedings of the 1st OpenSG Symposium.*

Sebastien Callens and Amir Zadpoor. 2018. From flat sheets to curved geometries: Origami and kirigami approaches. *Materials Today* 21, 3 (2018), 241–264.

Frédéric Cazals and Marc Pouget. 2005. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design* 22, 2 (2005), 121–146.

Luigi Cremona. 1890. *Graphical statics: two treatises on the graphical calculus and reciprocal figures in graphical statics.* Clarendon Press, Oxford, UK.

Erik Demaine, Martin Demaine, Duks Koschitz, and Tomohiro Tachi. 2015. A review on curved creases in art, design and mathematics. *Symmetry: Culture and Science* 26, 2 (2015), 145–161.

Erik D. Demaine and Joseph O'Rourke. 2007. *Geometric folding algorithms: linkages, origami, polyhedra.* Cambridge University Press, New York, NY, USA.

Erik D. Demaine and Tomohiro Tachi. 2017. Origamizer: A practical algorithm for folding any polyhedron. In *Proceedings of the 33rd International Symposium on Computational Geometry.* Dagstuhl Publishing, Saarbrücken, Germany, 1–15.

Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques.* Association for Computing Machinery, New York, NY, USA, 317––324.

Manfredo P. do Carmo. 1976. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, NJ, USA.

Levi H. Dudte, Etienne Vouga, Tomohiro Tachi, and Lakshminarayanan Mahadevan. 2016. Programming curvature using origami tessellations. *Nature Materials* 15, 5 (2016), 583.

Hans-Christian Ebke, David Bommes, Marcel Campen, and Leif Kobbelt. 2013. QEx: Robust Quad Mesh Extraction. *ACM Transactions on Graphics* 32, 6 (2013), 168:1–168:10.

Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. 2003. Bilateral mesh denoising. *ACM Transactions on Graphics* 22, 3 (2003), 950––953.

Chi-Wing Fu, Chi-Fu Lai, Ying He, and Daniel Cohen-Or. 2010. K-set tilable surfaces. *ACM Transactions on Graphics* 29, 4 (2010), 44:1–44:6.

Felix Günther, Caigui Jiang, and Helmut Pottmann. 2020. Smooth polyhedral surfaces. *Advances in Mathematics* 363 (2020), 107004.

Mathieu Huard, Michael Eigensatz, and Philippe Bompas. 2015. Planar Panelization with Extreme Repetition. In *Advances in Architectural Geometry 2014.* Springer, Cham, Switzerland, 259–279.

Alec Jacobson, Daniele Panozzo, et al. 2018. *libigl:* A simple C++ geometry processing library.

Caigui Jiang, Felix Günther, Johannes Wallner, and Helmut Pottmann. 2016. Measuring and controlling fairness of triangulations. In *Advances in Architectural Geometry 2016.* vdf Hochschulverlag AG, Zürich, Switzerland, 24–39.

Caigui Jiang, Florian Rist, Helmut Pottmann, and Johannes Wallner. 2020. Freeform quad-based kirigami. *ACM Transactions on Graphics* 39, 6 (2020), 209:1–209:11.

Caigui Jiang, Chengcheng Tang, Marko Tomičí, Johannes Wallner, and Helmut Pottmann. 2015a. Interactive Modeling of Architectural Freeform Structures – Combining Geometry with Fabrication and Statics. In *Advances in Architectural Geometry 2014.* Springer, Cham, Switzerland, 95–108.

Caigui Jiang, Chengcheng Tang, Amir Vaxman, Peter Wonka, and Helmut Pottmann. 2015b. Polyhedral patterns. *ACM Transactions on Graphics* 34, 6 (2015), 172:1–172:12.

Robby Kraft, Rupert Maleczek, Klara Mundilova, and Tomohiro Tachi. 2023. From Quad Filling to Wrinkled Surfaces. In *Advances in Architectural Geometry 2023.* De Gruyter, Berlin, Germany, 327–338.

David Levin. 1998. The approximation power of moving least-squares. *Math. Comp.* 67, 224 (1998), 1517––1531.

Yufei Li, Yang Liu, and Wenqiang Wang. 2014. Planar Hexagonal Meshing for Architecture. *IEEE Transactions on Visualization and Computer Graphics* 21, 1 (2014), 95–106.

Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. 2004. *Methods for non-linear least squares problems.* Technical University of Denmark, Denmark.

James Clerk Maxwell. 1870. On Reciprocal Figures, Frames, and Diagrams of Forces. *Earth and Environmental Science Transactions of The Royal Society of Edinburgh* 26, 1 (1870), 1–40.
Allan McRobie. 2017. The geometry of structural equilibrium. *Royal Society Open Science* 4, 3 (2017), 160759.
Cameron Millar, Toby Mitchell, Arek Mazurek, Ashpica Chhabra, Alessandro Beghini, Jeanne N. Clelland, Allan McRobie, and William F. Baker. 2022. On designing plane-faced funicular gridshells. *International Journal of Space Structures* 38, 1 (2022), 40–63.
Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. 2006. Laplacian mesh optimization. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*. Association for Computing Machinery, New York, NY, USA, 381–389.
David Orden, Günter Rote, Francisco Santos, Brigitte Servatius, Herman Servatius, and Walter Whiteley. 2004. Non-Crossing Frameworks with Non-Crossing Reciprocals. *Discrete & Computational Geometry* 32, 4 (2004), 567–600.
Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. 2014. Frame fields: anisotropic and non-orthogonal cross fields. *ACM Transactions on Graphics* 33, 4 (2014), 134:1–134:11.
Davide Pellis, Martin Kilian, Felix Dellinger, Johannes Wallner, and Helmut Pottmann. 2019. Visual smoothness of polyhedral surfaces. *ACM Transactions on Graphics* 38, 4 (2019), 31:1–31:11.
Helmut Pottmann and Yang Liu. 2007. Discrete Surfaces in Isotropic Geometry. In *Proceedings of the 12th IMA International Conference on Mathematics of Surfaces*, Vol. 4647. Springer, Berlin-Heidelberg, Germany, 341–363.
Hans Sachs. 1990. *Isotrope Geometrie des Raumes*. Vieweg, Wiesbaden, Germany.
Mayank Singh and Scott Schaefer. 2010. Triangle surfaces with discrete equivalence classes. *ACM Transactions on Graphics* 29, 4 (2010), 46:1–46:7.
Oded Stein, Eitan Grinspun, and Keenan Crane. 2018. Developability of triangle meshes. *ACM Transactions on Graphics* 37, 4 (2018), 77:1–14.
Karl Strubecker. 1941. Differentialgeometrie des isotropen Raumes I: Theorie der Raumkurven. In *Sitzungsberichte der Akademie der Wissenschaften in Wien*, Vol. 150. Hölder, Vienna, Austria, 1–53.
Karl Strubecker. 1942. Differentialgeometrie des isotropen Raumes III: Flächentheorie. *Mathematische Zeitschrift* 48, 1 (1942), 369–427.
Tomohiro Tachi. 2010. Origamizing Polyhedral Surfaces. *IEEE Transactions on Visualization and Computer Graphics* 16, 2 (2010), 298–311.
Chengcheng Tang, Xiang Sun, Alexandra Gomes, Johannes Wallner, and Helmut Pottmann. 2014. Form-finding with polyhedral meshes made simple. *ACM Transactions on Graphics* 33, 4 (2014), 70:1–70:9.
Gabriel Taubin. 1995. A signal processing approach to fair surface design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. Association for Computing Machinery, New York, NY, USA, 351––358.
Chelsea Tymms, Siqi Wang, and Denis Zorin. 2020. Appearance-preserving tactile optimization. *ACM Transactions on Graphics* 39, 6 (2020), 1–16.
Bruno Vallet and Bruno Lévy. 2008. Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum* 27, 2 (2008), 251–260.

Appendix

This appendix contains the environment map of a cloudy sky used to render several of the figures, as well as a selection of proofs for lemmas and propositions presented in Sec. 3.

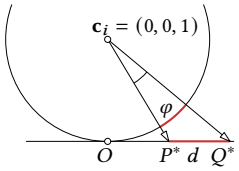
Fig. 34. Environment map of a cloudy sky, used to render Fig. 4, Fig. 7, Fig. 15, Fig. 29, Fig. 32 and Fig. 33.

Proof of Lemma 3.5. The unit sphere S_i is the zero set of the function $F(x,y,z) = \frac{1}{2}(x^2+y^2) - z$. The normal vectors are parallel to $\nabla F = (x, y, -1)$. Since a normal vector of P is $(p_1, p_2, -1)$ the top view of the contact point must be (p_1, p_2). The definition of metric duality (6) implies $\sigma(P) = (p_1, p_2, -p_3)$. Hence, the top views of $\sigma(P)$ and $\gamma_i(P)$ are the same. □

Proof of Lemma 3.7. Consider the following asymptotic analysis: close to O, the projection vectors $\mathbf{n}_P := \mathbf{p}^* - \mathbf{c}_i$ become vertical, implying that $\lim_{\mathbf{n}_P \to \text{vertical}} \|\mathbf{n}_P\| = 1$. Furthermore, the law of cosines also implies $d^2 = \|\mathbf{n}_P\|^2 + \|\mathbf{n}_Q\|^2 - 2\|\mathbf{n}_P\|\|\mathbf{n}_Q\| \cos\varphi$. Thus

$$\lim_{\mathbf{n}_P,\mathbf{n}_Q \to \text{vert.}} \cos\varphi = \lim_{\mathbf{n}_P,\mathbf{n}_Q \to \text{vert.}} \frac{\|\mathbf{n}_P\|^2 + \|\mathbf{n}_Q\|^2}{2\|\mathbf{n}_P\|\|\mathbf{n}_Q\|} - \frac{d^2}{2\|\mathbf{n}_P\|\|\mathbf{n}_Q\|}$$
$$= \frac{1+1}{2\cdot 1\cdot 1} - \frac{d^2}{2\cdot 1\cdot 1} = 1 - \frac{d^2}{2}.$$

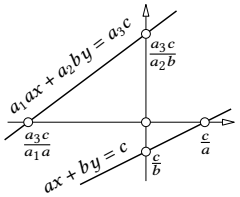
Moreover, the Taylor series expansion of the cosine can be expressed as $\cos\varphi = 1 - \frac{\varphi^2}{2} + \frac{\varphi^4}{24} - \cdots$, which implies that, in the vicinity of O, the projection vectors become vertical, and therefore $1 - \frac{\varphi^2}{2} + \frac{\varphi^4}{24} - \cdots$ becomes $1 - \frac{d^2}{2}$. Consequently, the distance between P^* and Q^* in the dual diagram, $d = \|P^* - Q^*\|$, approximates the Euclidean angle φ of the planes P and Q very well in the vicinity of O. □



Proof of Lemma 3.9. The lattice is invariant under translations by vectors of the form $\lambda\mathbf{a} + \mu\mathbf{b}$ with $\lambda, \mu \in \mathbb{Z}$. By Example 3.3 there is always an isotropic motion that leaves the paraboloid S invariant and which appears in the top view as translation by any given vector. Such isotropic motions to a translation vector $\lambda\mathbf{a} + \mu\mathbf{b}$ map T as a whole onto itself, and thus any two vertex stars are isotropic congruent.

T is composed of three families of polylines in vertical planes. Metric duality σ maps lines with the same top view to parallel lines. Thus, $\sigma(T)$ is formed by strips of hexagonal planar faces that are attached to each other along parallel edges. This shows that any two hexagons of T^* are related by a translation. Hence, metric duality maps congruent vertex stars to congruent hexagonal faces. □

Proof of Lemma 3.12. An affine map α which maps isotropic lines to isotropic lines, appears in the top view as affine map α'. It is sufficient to consider O fixed. By singular value decomposition of the matrix A' of the linear part of α', we see that α is the product of isotropic motions and independent scaling $(x, y, z) \mapsto (a_1 x, a_2 y, a_3 z)$. It is therefore sufficient to consider only α of this form. Then, by Eqn. (8), the edges of V_O^* corresponding to $\alpha(\mathbf{v})$ are



$$L(\sigma(\mathbf{v}))^* \colon a_1\xi x + a_2\eta y = a_3\frac{1}{2}(\kappa_1\xi^2 + \kappa_2\eta^2).$$

This line arises from the original line $L(\mathbf{v})^*$ via the planar affine map $\alpha^* \colon (x, y) \mapsto (\frac{a_3}{a_1}x, \frac{a_3}{a_2}y)$ (whose action is illustrated above). Thus the polygon V_O^* gets transformed with the affine map α^*. □

Proof of Lemma 3.13. Parallelly translating the Euclidean normals to the faces of V_O through $(0, 0, 1)$ results in the edges of a convex pyramid, whose intersection with the plane $z = 0$ forms the convex diagram V_O^*. □

Proof of Proposition 3.16. The six vertices of a centrally symmetric hexagon always lie on an ellipse, hyperbola, or two parallel lines. Thus, the first part of the claim is equivalent to the second part on the convex hull. We consider a vertex star V_O with central vertex at the origin and six one-ring vertices with top views $\pm\mathbf{a}$, $\pm\mathbf{b}$ and $\pm(\mathbf{a}+\mathbf{b})$. We take $\mathbf{a}$ in a direction with non-vanishing normal curvature. Hence, it is different from its conjugate direction $\bar{\mathbf{a}}$. Then, there is an affine map α to which Lemma 3.12 applies and which maps $\mathbf{a}, \bar{\mathbf{a}}$ into principal directions, and thus we can assume $\mathbf{a} = (1, 0)$, $\mathbf{b} = (b, c)$ with $c \neq 0$ and $\kappa_1 \neq 0$. Using Eqn. (8), intersecting adjacent edge lines and scaling the result with factor $c/2$, we get the following four vertices of the dual hexagon V_O^*,

$$\pm(\kappa_1 c, \kappa_1(b + b^2) + \kappa_2 c^2), \quad \pm(-\kappa_1 c, \kappa_1(b + b^2) + \kappa_2 c^2),$$

which form a rectangle R. The remaining two vertices are $\pm\mathbf{v}$, with

$$\mathbf{v} = (-\kappa_1 c(2b + 1), \kappa_1(b + b^2) - \kappa_2 c^2).$$

For a parabolic cylinder to $\kappa_2 = 0$, all six vertices lie on the parallel lines $y = \pm\kappa_1(b + b^2)$, orthogonal to the y-axis, i.e., the ruling of the parabolic cylinder. These lines agree with the y-axis exactly for $b = 0$ or $b = -1$, which characterizes vertex stars with two edges in the y-axis. Since we applied an affine map α, the dual diagram was transformed by an affine map as well. To fully prove orthogonality of the parallel lines to the direction of vanishing principal curvature, one needs to work with a more general $\mathbf{a}$. This confirms the claim, but the discussion of the remaining cases is simpler with $\mathbf{a} = (1, 0)$. In the other two cases, both with $\kappa_2 \neq 0$, we have to discuss the location of $\mathbf{v}$ w.r.t. the rectangle R. The regions where (i) all six vertices lie on their convex hull and (ii) two of them are inside the convex hull are bounded by the edge lines of R, as shown in the inset. One can easily be convinced that (i) happens exactly for $\kappa_1\kappa_2 > 0$ and (ii) exactly for $\kappa_1\kappa_2 < 0$. □