

# Rectifying Strip Patterns

BOLUN WANG, KAUST, Saudi Arabia  
HUI WANG, KAUST, Saudi Arabia  
EIKE SCHLING, The University of Hong Kong, China  
HELMUT POTTMANN, KAUST, Saudi Arabia

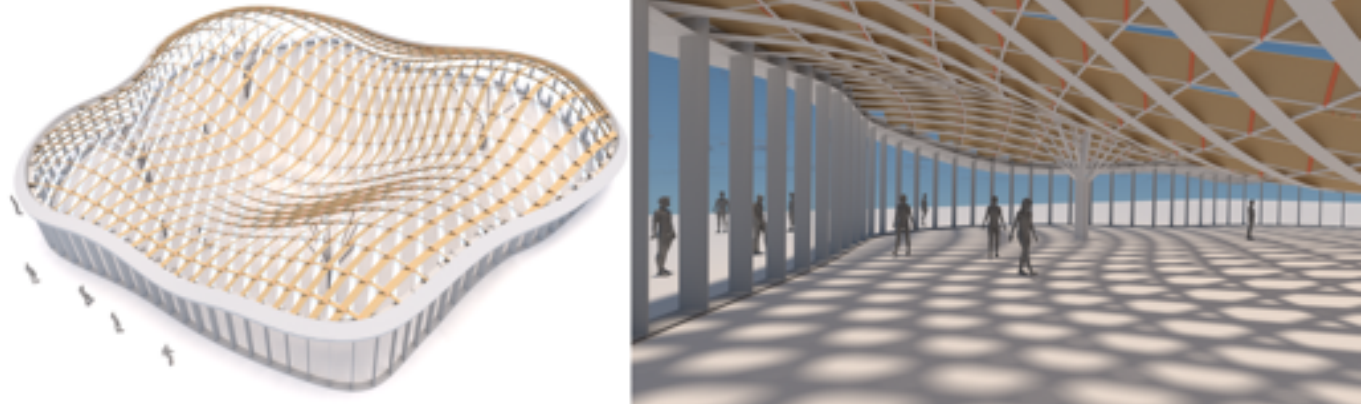


Fig. 1. This gridshell is a rectifying strip pattern, computed with our method. It can be built by bending straight flat strips of material along a reference surface  $S$ , while guiding their pattern and relative orientation. Here, three families of strips are arranged in a Kagome pattern. The two upper families have a constant incline angle towards  $S$  of  $45^\circ$  and  $30^\circ$  respectively, acting as structural lamellas and controlling natural lighting and ventilation. The bottom strips (white) lie tangential to  $S$  following geodesic (shortest) paths, triangulating the grid, connecting to the midpoints of lamellas, and acting as stiffening elements. The evaluation of this example is shown in Tables 2, 3, and 4.

Straight flat strips of inextensible material can be bent into curved strips aligned with arbitrary space curves. The large shape variety of these so-called rectifying strips makes them candidates for shape modeling, especially in applications such as architecture where simple elements are preferred for the fabrication of complex shapes. In this paper, we provide computational tools for the design of shapes from rectifying strips. They can form various patterns and fulfill constraints which are required for specific applications such as gridshells or shading systems. The methodology is based on discrete models of rectifying strips, a discrete level-set formulation and optimization-based constrained mesh design and editing. We also analyze the geometry at nodes and present remarkable quadrilateral arrangements of rectifying strips with torsion-free nodes.

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

Authors' addresses: Bolun Wang, KAUST, Saudi Arabia, bolun.wang@kaust.edu.sa; Hui Wang, KAUST, Saudi Arabia, hui.wang.1@kaust.edu.sa; Eike Schling, The University of Hong Kong, China, schling@hku.hk; Helmut Pottmann, KAUST, Saudi Arabia, helmut.pottmann@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2023 Association for Computing Machinery.  
0730-0301/2023/9-ART \$15.00  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Additional Key Words and Phrases: architectural geometry, computational design, computational fabrication, gridshell, shading system, pseudo-geodesic

## ACM Reference Format:

Bolun Wang, Hui Wang, Eike Schling, and Helmut Pottmann. 2023. Rectifying Strip Patterns. *ACM Trans. Graph.* 1, 1 (September 2023), 18 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Classical and computational differential geometry offer methods to design and fabricate complex geometric structures from simple elements. This is of great significance, especially in architecture, where individual, large-scale designs demand bespoke solutions that cannot be mass-produced but need to stay within a given budget. A large amount of contributions to architectural geometry thus deals with the simplification of construction elements, such as panels, joints or beams.

The present paper is motivated by the recent emergence of various types of architectural structures that are fabricated by bending originally flat straight pieces of material, typically wood or metal. Fig. 2 shows three built examples of such gridshells. They differ in how the individual elements are placed with respect to an underlying reference surface and in the pattern that is formed by them. If a bent element lies tangential (Fig. 2 a) to the reference surface  $S$ , it follows a shortest path (geodesic) on  $S$ . If it is placed orthogonal (Fig. 2 b) to  $S$ , it is aligned with an asymptotic curve of  $S$ , i.e., a curve of vanishing normal curvature. Very recently, even the case of a

constant angle (Fig. 2 c) (different from 0 and  $\pi/2$ ) of the bent elements against  $S$  has been considered. They follow so-called pseudo-geodesics on  $S$  and thus one speaks of pseudo-geodesic gridshells [Mesnil and Baverel 2023].



Fig. 2. Various types of gridshells fabricated by bending straight lamellas. (a) The Polydôme by Julius Natterer using stacked geodesic planks, and a diagonal geodesic cover [Natterer et al. 2000]; (b) Timber gridshell by Eike Schling et al. combining asymptotic and geodesic lamellas in a trihex grid (<https://eikeschling.com>). (c) Pseudo-geodesic gridshell by [Mesnil and Baverel 2023], using inclined strips.

The pattern of elements on the final structure may just be a single family of elements. Commonly, gridshells are constructed from two families of beams and a third set of elements that triangulate and thus brace the structure. This leads to three families of curved elements arranged in a so-called 3-web or Kagome pattern. In the present paper we are interested in all these potential patterns, thus generalizing the geodesic patterns of [Pottmann et al. 2010].

For some of the gridshells discussed above, there are efficient computational design methods. This includes asymptotic gridshells and geodesic gridshells. However, turning to pseudo-geodesic gridshells or hybrid asymptotic-geodesic shells, this is no longer true. There are computational models based on optimization, which however lack especially in the difficult initialization step. In other words, it is within some limits possible to modify a given design while keeping certain constraints, but it is hard to access the design space. The present paper aims to fill this gap.

While being motivated by specific gridshells and their underlying patterns, we present a more complete study of the problem area. We turn practical constraints on fabrication (such as torsion-free nodes) and function (e.g. shading) into geometric ones and show how to use them for computational design.

### 1.1 Contributions and Overview

Bent strips of material that are straight and flat prior to bending are already determined by their central curve or one of the boundary curves, as their binormals are strip surface normals (Sec. 2.1). This reduces a major part of our work to a computational design problem of families of surface curves with constraints on their binormals. For its solution, we propose a novel level set formulation in the spirit of discrete differential geometry, by discretizing the geometry rather than equations. This methodology part of Section 3 is surveyed in Fig. 3. The second major contribution of our paper are individual applications, illustrated in Fig. 4. We conclude in Sec. 7 with statistics on parameters and performance, along with a few directions for future research.

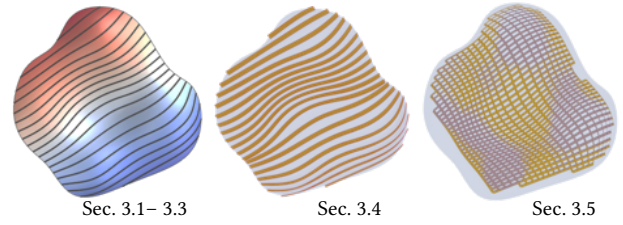


Fig. 3. Overview of methodology. Given a surface that guides the shape of a strip structure, we compute base curve families of strips with a level set algorithm (Sec. 3.1,3.2 and 3.3), attach strips to each family (Sec. 3.4) and optimize meshes which represent base curve families of strips (Sec. 3.5).



Fig. 4. Overview of applications. Gridshells formed by two or three families of rectifying strips are presented in Sec. 4 along with physical models. The difficult topic of gridshells with torsion-free nodes, where transversal strips meet at straight line segments, is addressed in Sec. 5. In Sec. 6 we design shading systems via a single family of rectifying strips.

### 1.2 Related Work

*Differential Geometry.* Geodesics and asymptotic curves on surfaces are a well-studied topic in classical differential geometry [do Carmo 1976], and the same is true for their discrete counterparts [Bobenko and Suris 2008]. Wunderlich [1950] introduced and studied pseudo-geodesic curves on surfaces, especially on cones and cylinders. The quadrilateral nets of asymptotic curves on negatively curved surfaces (A-nets) are very well understood, including their discrete versions [Bobenko and Suris 2008]. A-nets with a constant intersection angle  $\gamma$  of asymptotic curves represent surfaces with a constant ratio of principal curvatures [Wang and Pottmann 2022], which include minimal surfaces for  $\gamma = \pi/2$ .

Quad nets of geodesics (G-nets) provide more freedom than A-nets. Remarkable are surfaces which possess conjugate G-nets, also known as Voss nets [Sauer 1970; Voss 1888]. Discrete Voss nets are flexible quad meshes with planar faces, i.e., they become mechanisms when one considers faces rigid and places hinges at the edges [Bobenko and Suris 2008; Sauer 1970; Wunderlich 1951]. Orthogonal G-nets exist exactly on developable surfaces. Their discrete models have been developed by Rabinovich et al. [2018] for geometric design with developable surfaces. Orthogonal nets where only one family of curves are geodesics exist on all surfaces. Discrete versions for design and fabrication are found in [Wang et al. 2019].

*Geometry of webs.* In the present paper, we are mostly interested in *triangular webs (3-webs)*, which consist of three families of curves on a surface, such that through each surface point there is a curve from each family. Such a 3-web can be represented by a parameterization  $x(u, v)$  where the three families of web curves possess constant  $u$ , constant  $v$  and constant  $u+v$ , respectively. For the geometry of webs in general, we refer to the monograph by Blaschke and Bol [Blaschke and Bol 1938]. In web geometry, these curve families

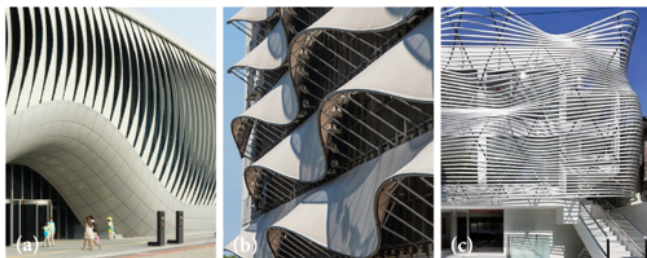


Fig. 5. Twisted louvers are used in architectural facades to control lighting and ventilation. (a) Soma Architects worked together with the University of Stuttgart to develop a kinetic louver facade for the One Ocean Pavilion at the EXPO 2012 (<https://soma-architecture.com>). (b) The Wave Car-Park by Scott Brownrigg (<https://scottbrownrigg.com>). (c) Aluminium wrapping by Amano Design in Omotesando, Japan (<https://amanod.com/jingumae>).

are mostly considered as dense, but also discrete webs have been studied. There, one picks just iso-lines of integer values of  $u, v, u + v$ .

Our focus is on webs of asymptotic, geodesic and pseudo-geodesic curves. The best-studied ones are the 3-webs of geodesics. In the plane, these are webs of straight lines, which are formed by the tangents of an algebraic curve of class 3 [Graf and Sauer 1924]. R. Sauer [1926] studied geodesic 3-webs on surfaces on hand of a discrete model. The characterizing partial differential equation for geodesic webs is found in [Mayrhofer 1931], but its explicit solution appears to be difficult. Geodesic webs exist on all surfaces of constant Gaussian curvature, since those can be mapped to the plane so that all geodesics get mapped to straight lines [Graf and Sauer 1924; Volk 1929].

*Geometry Processing and Computational Design.* There is a wealth of contributions on the efficient computation of geodesics (see [Crane et al. 2017] and references therein). Computational design of webs of geodesics and/or planar curves has been addressed in [Deng et al. 2011; Pottmann et al. 2010]. Geodesic webs also appear as weaving patterns generalizing the traditional craft of basket weaving to more complex freeform shapes [Ayres et al. 2020, 2018; Vekhter et al. 2019]. The use of initially straight ribbons limits the space of potential shapes and thus this weaving technique has been extended to curved ribbons [Baek et al. 2021; Ren et al. 2021]. [Pillwein et al. 2020; Pillwein and Musialski 2021] address the computational design of elastic geodesic gridshells, that are formed by an originally flat arrangement of straight planks, including solutions of the inverse problem of approximating a given shape with such a structure. Their shells are special X-shells [Panetta et al. 2019], for which a solution to the inverse problem is not known. [Liu et al. 2023] present structures that deploy from a stack of initially straight or circular elastic strips. They exhibit close relations to Chebyshev nets, which previously occurred in computational fabrication [Garg et al. 2014; Masson 2017; Sageman-Furnas et al. 2019].

*Architectural Geometry and Gridshells.* Gridshells have been of great interest in the architectural and engineering community for over a century. We point to early work by Vladimir Suchoy (see [Schling and Barthel 2021]) and Frei Otto [Happold and Liddell 1975; Henricke 1974]. Natterer stacked flat planks along geodesic paths to create ribbed gridshells [Natterer et al. 2000]. Geodesic gridshells

recently gained popularity [Martín-Pastor and González-Quintal 2021; Montagne et al. 2020], also as deployable structures [Soriano 2017]. Asymptotic gridshells are still a novelty in architecture. The tall orientation of planks creates a high stiffness, while offering an elastic erection process from flat to curved grid. Asymptotic gridshells have been built from steel, with constant node angle [Schling et al. 2018], and as web bisecting the principal curvature lines to allow for a simplified facade from developable or planar quad panels [Schling and Wan 2022]. Asymptotic grids can be actuated to become highly controlled transformable structures [Schikore et al. 2021]. Recently, hybrid asymptotic and geodesic 3-webs have been described [Schling et al. 2022], and implemented to build a robust tri-hex gridshell from timber planks (see Fig. 2 b). Mesnil and Baverel [2023] introduced pseudo-geodesic curves on rotational surfaces as a possible construction strategy for gridshells and recently built a large-scale prototype using timber planks and repetitive steel joints (Fig. 2 c).

Apart from practical applications as load-bearing gridshells, slender louvers are used in architecture for shading and ventilation, see Fig. 5. The controlled incline of louvers offers the possibility to redirect light and control views.

## 2 PRELIMINARIES

### 2.1 Bending straight strips: rectifying developables

Bending a rectangular strip  $D_0$  of inextensible flat material yields a developable surface strip  $D$  (Fig. 6). The straight central line  $c_0$  of the strip  $D_0$  is bent to a geodesic curve  $c \subset D$ . A geodesic on a surface is characterized by vanishing geodesic curvature. Equivalently, its principal normals are surface normals.

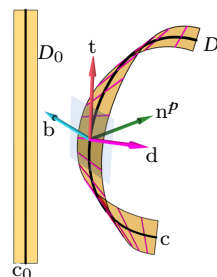


Fig. 6. Rectifying developable strip  $D$  along a space curve  $c$  and its straight unfolding  $D_0$ . The ruling  $d$  through a point of  $c$  is given by the Darboux vector  $d = \tau t + \kappa b$ , where  $\kappa$  and  $\tau$  denote curvature and torsion, respectively.

This shows how to construct the strip when  $c$  is given: We consider the orthonormal Frenet frame  $(t, n^P, b)$  along  $c$ , formed by the unit tangent vector  $t$ , principal normal vector  $n^P$  and binormal vector  $b$ . Vectors  $t$  and  $n^P$  span the osculating plane, and vectors  $t$  and  $b$  span the *rectifying plane*. For a geodesic on a surface, the rectifying planes are tangent to the surface. Therefore, the developable surface carrying the strip  $D$  has the rectifying planes of  $c$  as tangent planes. Since a developable surface contains only a one-parameter family of tangent planes, the surface  $D$  lies in the envelope  $R$  of rectifying planes, which is called the *rectifying developable of  $c$* . As a developable surface,  $R$  contains a family of straight lines, called rulings (see Fig. 6).  $R$  is also the rectifying developable of the strip boundaries. In the following, we call any developable strip with a straight development a *rectifying strip*. Sec. 5.1 presents a closer local analysis of rectifying strips to better understand tolerances at nodes of a gridshell.

A *discrete curve* is a polyline  $C$  with vertices  $c_i$ . Its edges can be seen as discrete tangents. The connecting planes  $\sigma_i$  of 3 consecutive vertices  $c_{i-1}c_i c_{i+1}$  are discrete osculating planes at  $c_i$ . The circle  $k_i$  through  $c_{i-1}c_i c_{i+1}$  is a discrete osculating circle. If edges are of constant length (discrete arc length parameterization), the tangent  $t_i \in \sigma_i$  of the circle  $k_i$  at  $c_i$  is an outer bisector of the polyline, and the other bisector, containing the center of  $k_i$ , is a discrete principal normal. This yields a discrete Frenet frame  $(t_i, n_i^p, b_i)$ . We use the definition via bisectors also for a general discrete parameterization. With unit edge vectors  $e_i = (c_{i+1} - c_i) / \|c_{i+1} - c_i\|$ , we have

$$t_i = \frac{e_i + e_{i-1}}{\|e_i + e_{i-1}\|}, \quad n_i^p = \frac{e_i - e_{i-1}}{\|e_i - e_{i-1}\|}, \quad b_i = t_i \times n_i^p. \quad (1)$$

The discretization of rectifying strips is addressed in Section 3.4.

## 2.2 Attaching strips at constant angle: pseudo-geodesics

A rectifying strip can be attached tangentially to a surface  $S$  exactly along a geodesic of  $S$ . We also require strips  $D$  that are inclined under a constant angle  $\alpha \neq 0$  against  $S$ . Equivalently, we look at curves  $c \subset S$  whose rectifying planes form a constant angle  $\alpha \neq 0$  with the tangent planes of  $S$ . In other words, the binormals of  $c$  form the constant angle  $\theta = \frac{\pi}{2} - \alpha$  with the normals of  $S$ . According to W. Wunderlich[1950], such curves are called *pseudo-geodesics* or shortly *P-curves* in the following. The case  $\alpha = \pi/2$  characterizes asymptotic curves.

We complete the tangent vector  $t$  of the surface curve  $c$  and surface unit normal vector  $n$  via the side vector  $u = n \times t$  to the Darboux frame. The curvature vector  $\kappa n^p$  of  $c$  is decomposed into a tangential and normal component via  $\kappa n^p = \kappa_g u + \kappa_n n$ , with  $\kappa_g, \kappa_n$  as geodesic and normal curvature, respectively and  $n^p$  as principal normal vector. For a P-curve  $c$ , the angle between  $n^p$  and  $n$  equals  $\alpha$ , i.e.  $n^p = \sin \alpha u + \cos \alpha n$ . Hence, a P-curve is characterized by a constant ratio of geodesic and normal curvature,  $\tan \alpha = \kappa_g / \kappa_n$ . Not only  $\kappa_n$ , but also  $\alpha$  and  $\kappa_g$  have to be considered with a sign. The signs determine whether  $c$  turns to the left or right side of  $t$ .

Later we also consider surface curves along which the angle  $\alpha$  is not constant, but varies according to the desired application.

For the main part of this paper, the surface  $S$  is given as a triangle mesh, and the curve is a polyline on it, namely a level set of a function  $F$  defined on the mesh. In the final steps of the individual applications, the curve may also be a parameter line in a constrained mesh (quad mesh for grid-like patterns and triangle mesh for 3-webs). In all cases, the pseudo-geodesic property is expressed by a constant angle  $\theta$  between discrete binormal  $b$  and surface normal  $n$ ,

$$b \cdot n - \cos \theta = 0. \quad (2)$$

[Jiang et al. 2019] proposed a tracing algorithm to find P-curves on triangle meshes, but their shape control is cumbersome, depending highly on the choices of  $\theta$  and tangent direction at the starting point. Moreover, it is hard to avoid intersections of neighboring P-curves or self-intersections. Therefore, we turn to a level-set formulation.

## 3 METHOD

In this section, we introduce our level-set based formulation and the optimization framework that forms the core of our computational

design tools. The major part is devoted to the level-set algorithm. In Sec. 3.4 we describe the computation of rectifying strips and Sec. 3.5 deals with constrained mesh optimization for strip structures and user-guided shape modification. We found it easier to describe initially just the design of patterns formed by P-curves. In Sections 5 and 6 we show how to easily adapt our algorithms to similar constraints that occur in practical applications.

### 3.1 Curve families on surfaces via level sets

Given a surface  $S$  with disk topology and a function  $F$  defined on it, its level sets  $F = \text{const.}$  form a family of curves that cover the surface without intersections of different curves. If  $\nabla F \neq 0$  everywhere, there are no self-intersections of individual curves and through each surface point there is exactly one level curve, orthogonal to  $\nabla F$ .

It is well known (see e.g. [Pottmann et al. 2010]) that geodesic and normal curvature of level sets can be computed via

$$\kappa_n = \frac{\text{II}(J\nabla F)}{\|\nabla F\|^2}, \quad \kappa_g = \text{div} \left( \frac{\nabla F}{\|\nabla F\|} \right),$$

where  $\text{II}$  is the second fundamental form of  $S$  and  $J$  is rotation by 90 degrees. Pseudo-geodesic level sets can then be expressed via the constant ratio  $\kappa_g / \kappa_n$ . Since we have to discretize later anyway and we have to include other constraints on binormals of level sets, we do not discretize equations, but take a *geometric discrete approach*: We express the binormal vectors of the discrete curves explicitly and control the behavior of the curves by applying constraints to the binormal vectors.

*Controlling binormal vectors of the level curves.* Let the surface  $S$  be represented by a triangle mesh  $(\mathcal{V}, \mathcal{E}, \mathcal{F})$  that is topologically equivalent to a disk. We want to compute a scalar function  $F$  on  $S$ , with non-vanishing gradient  $\nabla F$ , whose level sets have binormal vectors that are constrained by the application. For pseudo-geodesic level sets, this is the constant angle constraint Eq. (2).

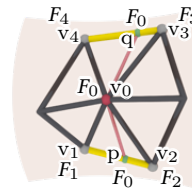


Fig. 7.  $v_0$  is an interior vertex of the triangle mesh  $S$ . The level curve (red) passing through  $v_0$  has value  $F_0$ , and has intersection points  $p, q$  with two edges (yellow) of the vertex star around  $v_0$ .

The function  $F$  is defined by its values on the mesh vertices and linearly interpolated in the triangles. It is sufficient to express the constraint on binormals locally at each vertex  $v_i$  as well as the level polyline passing through  $v_i$ . This is done as follows. Consider the level polyline  $c$  passing through a vertex  $v_0$  of  $S$  (Fig. 7) and let  $F_0$  be the function value at  $v_0$ . If  $v_0$  is not on the boundary of  $S$  there is a unique level curve  $c$  passing through  $v_0$ , that intersects the boundary of the vertex star in two further points. These points  $p$  and  $q$  define two edges  $e_p = v_0 - p$  and  $e_q = v_0 - q$ . Assuming that  $e_p$  and  $e_q$  are linearly independent, they span the discrete osculating plane and the discrete binormal vector  $b$  of the curve  $c$  can be computed as  $b = \pm e_p \times e_q / \|e_p \times e_q\|$ .

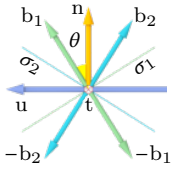


Fig. 8. For a given angle  $\theta$ , Eq. (2) holds with appropriately oriented binormals for two osculating planes  $\sigma_1$  and  $\sigma_2$ . Equation (3) helps to select the desired osculating plane without having to care about the orientation of the binormal vector.

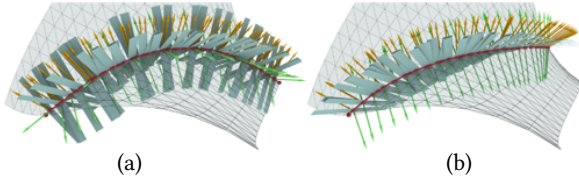


Fig. 9. The discrete curve (red) in (a) satisfies Eq. (2), but the osculating planes (blue rectangles) are not continuous, meaning non-continuity of the discrete Frenet frames. The curve in (b) has continuous Frenet frames since it follows Eq. (3). The normal vectors of the surface are marked in yellow.

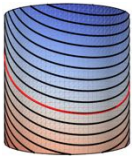


Fig. 10. P-curves to  $\theta = 60^\circ$  as level sets on a triangulated rotational cylinder (optimization error  $E_{\text{angle}} = 9 \times 10^{-6}$ ; see Eq. 12). The parametric P-curve (red) computed according to [Wunderlich 1951] matches the level set (black) well. The Hausdorff distance of the two curves is 0.47% of the curve length.

*Handling the angle constraint between surface normals and binormals of level sets.* Here we have to correctly select one of the two possible osculating planes  $\sigma_1, \sigma_2$  that contain a given curve tangent and form an angle  $\theta$  with the tangent plane; see Fig. 8. In order to simplify the subsequent optimization, we want a criterion that is agnostic to the change  $b \rightarrow -b$ . We use the side vector  $u$  and normal vector  $n$ . For one choice of osculating plane, the scalar products  $b \cdot n$  and  $b \cdot u$  have the same sign ( $\sigma_1$  in Fig. 8) and for the other choice they have a different sign ( $\sigma_2$  in Fig. 8), leading to the angle constraints

$$\begin{aligned} (b \cdot n)^2 &= \cos^2 \theta, \\ (b \cdot n)(b \cdot u) &= \sin \theta \cos \theta. \end{aligned} \quad (3)$$

Clearly this depends on the sign of  $\theta$  and the side vector  $u$ . In Section 3.2.2, we show how to achieve a consistent side vector field. Fig. 9 illustrates the necessity of caring about the sign of  $\theta$ .

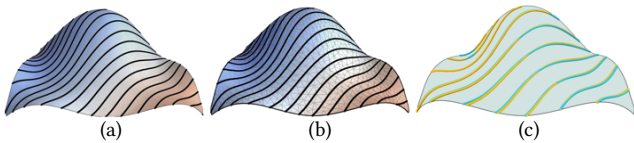


Fig. 11. Stability under re-meshing. The triangle mesh in (a) has 8960 vertices, and the level sets are P-curves of  $\theta = 60^\circ$ . The mesh is simplified to 657 vertices using fTetWild [Hu et al. 2020], but the shape is kept. We initialize  $F$  of (b) by linear interpolating the function values of (a), then optimize for P-curves (Section 3.2.2). We extract 10 level sets from each of (a) and (b) (Section 3.4) and compare them (c). The red and blue curves are for (a) and (b), respectively. The Hausdorff distance of the curves of (a) and (b) is 0.76% of the diagonal length of the bounding box of the mesh.

*Validation of the discretization.* We validate our discretization on a cylinder of revolution, whose pseudo-geodesics have known explicit parametric representations [Wunderlich 1951]; see Fig. 10. In Fig. 11, we also compare level sets on two models with the same shape, but different triangulations. The differences between the corresponding curves are minor, indicating consistency of our method under different triangulations.

### 3.2 Optimization framework

We apply Eq. (3) as our constraint on level sets to represent P-curves. For that, we propose the following 3 steps to construct a function whose level curves are P-curves of an input angle  $\alpha$  (see Fig. 12). The first step is to create a few guiding pseudo-geodesic curves on the reference triangle mesh using a tracing algorithm (Fig. 12 a). The second step initializes the scalar function  $F$  and optimizes it so that all level sets become pseudo-geodesics (Fig. 12 b-d). In the third step, we extract the level curves from the scalar function and optimize the corresponding rectifying strips in a post-processing step (Fig. 12 e).

*3.2.1 Tracing guide curves.* We first create a few guide curves on the reference triangle mesh with a robust version of the tracing algorithm in [Jiang et al. 2019].

Starting from a segment  $p_{i-1}p_i$  on the triangle mesh (see Fig. 13), we look for the next point  $p_{i+1}$  on an appropriate edge  $v_s v_e$ , such that the discrete osculating plane  $\sigma_i$  spanned by  $p_{i-1}, p_i, p_{i+1}$  forms the desired angle  $\alpha$  with the normal vector  $n_i$  at  $p_i$ . Equivalently, the binormal vector  $b_i$  has to form the angle  $\theta = \pi/2 - \alpha$  with  $n_i$ . All planes through  $p_i$  that form a constant angle  $\alpha$  with  $n_i$  are tangent to a rotational cone  $\Gamma_i$  with axis vector  $n_i$  and thus there are two candidate planes for  $\sigma_i$  which also contain  $p_{i-1}$ . We compute the intersections of these two planes and the edges through the two vertices  $v_0$  or  $v_2$  in the following way. Parametrizing an edge  $v_s v_e$  via  $p_{i+1} = (1-t)v_s + tv_e, t \in [0, 1]$ , the angle constraint reads

$$[(g_i \times g_{i+1}) \cdot n_i]^2 - \|g_i \times g_{i+1}\|^2 \cdot \cos^2 \theta = 0, \quad (4)$$

where  $g_i = p_i - p_{i-1}$ ,  $g_{i+1} = p_{i+1} - p_i$ . It is a quadratic equation in  $t$ ; only one of the solutions yields the correct sign of the angle against a local Darboux frame (as explained in Sec. 3.1). We select as next point  $p_{i+1}$  the nearest intersection with an edge in the chosen neighborhood so that  $\|p_{i+1} - p_i\| > d_0$ . The threshold  $d_0$  avoids too close vertices and unstable discrete osculating planes. The traced curves are used to guide level sets to desired directions, and quickly show if P-curves of angle  $\theta$  behave as desired for design.

*3.2.2 Optimization of level sets towards P-curves.* To compute a function  $F$  on the given mesh whose level sets are P-curves, we first compute an initial guess using the traced polylines. Then, we optimize the level sets so that Eq. (3) is satisfied.

*Initialization.* Starting from a set  $C_t$  of  $n$  ordered intersection-free traced curves  $c^i, i = 1, \dots, n$ , (Fig. 12a), we set the function value of each curve  $c^i \in C_t$  to a constant value  $F^i$ , e.g.  $F^i = i$ .

Having these function values  $F^i$  at the vertices  $p_j$  of the curves  $c^i$ , we initialize the scalar function  $F$  on a parameter domain. To map the mesh plus guide curves into the plane, we use a conformal mapping algorithm implemented in [Jacobson and Panozzo

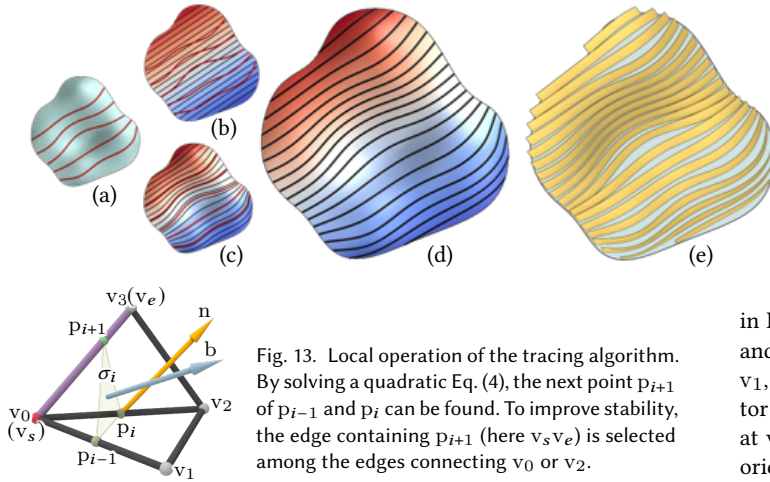


Fig. 12. Overview of our algorithm for generating a family of pseudo-geodesic curves: (a) Starting with tracing guide curves, our method first initializes the scalar function on a parameter domain of the surface (b) and optimizes the scalar function so that its level sets interpolate the guide curves (c). Then, we optimize the scalar function so that the level sets are pseudo-geodesic curves (d). The quality of the extracted curves and rectifying strips is improved through post-processing procedures (e). The detailed evaluation for (d) is shown in Table 2 and Fig. 40. The evaluation for (e) is shown in Table 4.

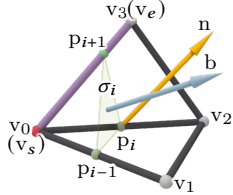


Fig. 13. Local operation of the tracing algorithm. By solving a quadratic Eq. (4), the next point  $p_{i+1}$  of  $p_{i-1}$  and  $p_i$  can be found. To improve stability, the edge containing  $p_{i+1}$  (here  $v_s v_e$ ) is selected among the edges connecting  $v_0$  or  $v_2$ .

2017]. A point  $p \in S$  is mapped onto  $m(p) \in \mathbb{R}^2$ . To obtain non-intersecting level sets, we first initialize  $F$  as a linear function in 2D, as illustrated in Fig. 12 (b):

$$F(p) = F^1 + (F^n - F^1) \frac{(m(p) - m(c_0^1)) \cdot (m(c_0^n) - m(c_0^1))}{\|m(c_0^n) - m(c_0^1)\|^2},$$

where  $c_0^1$  and  $c_0^n$  are the starting points of  $c^1$  and  $c^n$ , respectively.

Then we have to optimize the values of  $F$  on all vertices  $v_i$  of the mesh  $S$  so that the level sets are smooth and interpolate the guide curves. Using the notation in Fig. 13 for the local situation near a vertex  $p$  of a polyline  $c^i \in C_t$ , we define an objective function

$$E_{\text{trace}} = \sum_{c^i \in C_t} \sum_{p \in c^i} \left( \frac{F^i - F_0}{F_1 - F_0} - \frac{\|p - v_0\|}{\|v_1 - v_0\|} \right)^2. \quad (5)$$

Here  $v_0 v_1$  is the edge of  $S$  that contains  $p$ , and  $F_0$  and  $F_1$  are the function values at  $v_0$  and  $v_1$ , respectively.  $E_{\text{trace}}$  relates the value  $F^i$  at  $p$  by linear interpolation to the values  $F_0, F_1$  at the end points of the edge  $v_0 v_1$ .

To obtain smooth level sets, we adopt the squared Hessian energy  $E_{\text{fair}}$  proposed by [Stein et al. 2018] and implemented in [Jacobson and Panozzo 2017]:

$$E_{\text{fair}} = \sum_{v \in \mathcal{V}} \|H(v)\|^2 \mathcal{A}(v). \quad (6)$$

$\mathcal{A}(v)$  is the area of the Voronoi cell of vertex  $v$ , and  $\|H(v)\|^2$  is the squared Frobenious norm of the Hessian [Stein et al. 2018]. This energy yields smoother level sets of  $F$  on the mesh boundaries than the Laplacian energy. Our final objective function for initialization is

$$E_{\text{init}} = \lambda_0 E_{\text{trace}} + \lambda_1 E_{\text{fair}}, \quad (7)$$

where  $\lambda_0$  and  $\lambda_1$  are properly chosen weights. Typically we choose  $\lambda_0 = 3, \lambda_1 = 1$ . A result of this step is shown in Fig. 12 (c).

*Optimization for pseudo-geodesics.* We optimize the scalar function  $F$  such that the level set through each vertex is locally a P-curve of angle  $\theta$ . Firstly, for each inner vertex  $v_0$  of  $S$ , we search among the boundary edges of its vertex star to find the edges that have intersections with the level curve passing through  $v_0$ , as shown

in Fig. 7. If  $v_0$  is not a singularity of  $F$ , there are two edges  $v_1 v_2$  and  $v_3 v_4$  that intersect the level set at  $p$  and  $q$ , respectively. Let  $v_1, v_2, v_3, v_4$  be counterclockwise arranged w.r.t. the normal vector  $n$  at  $v_0$ ,  $F_1 > F_2$  and  $F_3 > F_4$ , the direction of the tangent vector at  $v_0$  of polyline  $\{p, v_0, q\}$  can be prescribed as from  $p$  to  $q$ . This orientation is crucial to our method, since we need to construct a continuous Darboux frame along the curves to obtain continuous osculating planes. In view of Eq. (3), the angle constraint is written as

$$E_{\text{angle}} = \sum_{v \in \mathcal{V}} ((b \cdot n)^2 - \cos^2 \theta)^2 \mathcal{A}(v) + \sum_{v \in \mathcal{V}} ((b \cdot n)(b \cdot u) - \sin \theta \cos \theta)^2 \mathcal{A}(v), \quad (8)$$

where  $b$  is the unit binormal vector satisfying

$$b \cdot (v_0 - p) = 0, \quad b \cdot (v_0 - q) = 0, \quad b^2 = 1, \quad (9)$$

and  $u$  is the unit side vector of the Darboux frame, constrained by

$$u \cdot (p - q) = 0, \quad u \cdot n = 0, \quad u^2 = 1. \quad (10)$$

The vectors  $b$  and  $u$  are used as auxiliary variables along with the values of  $F$  on mesh vertices. We note that the constraints in Eq. (10) cannot prevent  $u$  from inverting its orientation. In our implementation, in each iteration, we check if  $\det(q - p, u, n) > 0$ ; otherwise, we re-initialize  $u$  as the unit vector of  $n \times (q - p)$ .

Since we want to avoid self-intersecting level curves and singularities in the curve family, we add a term  $E_{\text{grad}}$  that prevents a vanishing gradient of  $F$ ,

$$E_{\text{grad}} = \sum_{f \in \mathcal{F}} (\|\nabla F(f)\| - r)^2 \mathcal{A}(f), \quad (11)$$

where  $\nabla F(f)$  is the gradient of  $F$  on triangle  $f$  and  $r$  is a positive constant value. As mentioned in [Pottmann et al. 2010], this term also acts as a regularizer to control the spacing of level sets:  $\|\nabla F\| - r = 0$  means that the distance between two level sets  $F = c$  and  $F = c + 1$  is roughly  $1/r$ .

Pseudo-geodesic level curves are obtained by minimizing

$$E_{\text{pg}} = \lambda_{\text{fair}} E_{\text{fair}} + \lambda_{\text{grad}} E_{\text{grad}} + \lambda_{\text{angle}} E_{\text{angle}}. \quad (12)$$

The variables are the values of  $F$  on the vertices of the mesh  $S$ , binormal vectors  $b$ , and side vectors  $u$ . We employ a Levenberg-Marquardt method as in [Madsen et al. 2004] to minimize  $E_{\text{pg}}$ . The parameters  $\lambda$  for the selected models are presented in Table 2.

To make sure that  $E_{\text{angle}}$  better reflects the numerical errors in the angles regardless of the Voronoi cell areas of the vertices, we take  $\mathcal{A}(v_i) = \frac{n \mathcal{A}(v_i)}{\sum_{k=0}^n \mathcal{A}(v_k)}$  such that the average of  $\mathcal{A}(v)$  is 1.

Signed angles are not necessary for geodesic and asymptotic level sets. These are optimized using energy terms  $E_{\text{geo}}$  and  $E_{\text{asy}}$  that slightly differ from  $E_{\text{angle}}$  in Eq. (8) to avoid auxiliary variables,

$$\begin{aligned} E_{\text{geo}} &= \sum_{v \in \mathcal{V}} \left( \frac{\det(\mathbf{n}, \mathbf{v}_0 - \mathbf{p}, \mathbf{v}_0 - \mathbf{q})}{\|\mathbf{v}_0 - \mathbf{p}\| \|\mathbf{v}_0 - \mathbf{q}\|} \right)^2 \mathcal{A}(v), \\ E_{\text{asy}} &= \sum_{v \in \mathcal{V}} \left( \left( \frac{\mathbf{n} \cdot (\mathbf{v}_0 - \mathbf{p})}{\|\mathbf{v}_0 - \mathbf{p}\|} \right)^2 + \left( \frac{\mathbf{n} \cdot (\mathbf{v}_0 - \mathbf{q})}{\|\mathbf{v}_0 - \mathbf{q}\|} \right)^2 \right) \mathcal{A}(v). \end{aligned} \quad (13)$$

$E_{\text{geo}} = 0$  means that the osculating plane spanned by  $\mathbf{e}_p = \mathbf{v}_0 - \mathbf{p}$  and  $\mathbf{e}_q = \mathbf{v}_0 - \mathbf{q}$  contains the normal  $\mathbf{n}$  at each vertex star, while  $E_{\text{asy}} = 0$  says that it is orthogonal to  $\mathbf{n}$  and thus tangent to  $S$ .

*Interactive steering of pseudo-geodesic curves.* The tracing algorithm produces P-curves which strongly depend on the starting point, starting direction, and the angle  $\theta$ . It requires the user to trace the curves one by one while avoiding intersections between them. To lift the burden of this tedious process, we also provide an interactive design method that allows the user to sketch on the surface and then solve for the best-fitting P-curves automatically. The minor adaptation required to extend this beyond P-curves, is also useful for the applications in Sections 5 and 6.

As shown in Fig. 14, non-intersecting curves represented as sequences of points are sketched by the user. Before optimization, we filter the curves such that on each triangle of  $S$  there is at most one sketched curve point. Then a smooth scalar function whose level sets follow the input curves is initialized via the method in Sec. 3.2.2, but the tracing energy  $E_{\text{trace}}$  is replaced by

$$\sum_{c \in \mathcal{C}_t} \sum_{p \in c} (F_0 u + F_1 v + F_2 t - c_c)^2.$$

Here  $F_0, F_1, F_2$  are the function values at the vertices of the triangle  $T$  in which the sketched point  $p$  lies,  $(u, v, t)$  are the barycentric coordinates of  $p$  w.r.t.  $T$ , and  $c_c$  is a value which is constant along each sketched curve. In each iteration, we take  $c_c$  as the average of the function values of the sketched points along the curve  $c$  in the previous iteration.

After obtaining a smooth scalar field whose level sets approximate the sketched curves, we compute the discrete binormal vectors on the vertices of the triangles where the sketched point lies and take the average  $\theta$  of the angles between the binormal vectors and the surface normal vectors as the target pseudo-geodesic angle. Then we solve for pseudo-geodesic level sets with angle  $\theta$  according to Sec. 3.2.2.

In our implementation, the interactive design of P-curves is fully automatic with prescribed parameters (see Table 2). To enhance the applicability of the parameters for triangle meshes with different scales or densities, we uniformly scale the meshes prior to optimization, so that their average edge length is 1. Even then we cannot guarantee that the same parameters work for all models. The reasons are not only the unavoidable limitations of geometry, but also the avoidance of singularities (Section 7). Fig. 15 shows a failure example using optimization with default parameters. The problem can be resolved by changing parameters during optimization.

*P-curve families with changing angles.* For applications to gridshells, a small angle  $\theta$  between rectifying strips and the normals

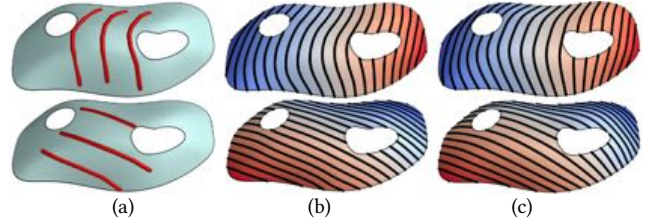


Fig. 14. Examples of interactive design of P-curves. The red curves drawn on the surfaces (a) are used as guide curves to initialize the scalar function  $F$  (b). We then average the angles  $\theta$  of the level sets around the sketched curves to determine the target angles and further optimize for P-curves (c). The level sets in the top and bottom figures in (c) are P-curves of  $\theta = 60.3^\circ$  and  $\theta = 75.36^\circ$ , respectively (for details on optimization weights and detailed evaluation, see Table 2 and Fig. 41).

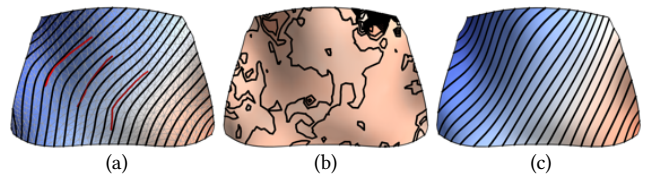


Fig. 15. Reaction to failures. This triangle mesh is the same as in Fig. 11 b. (a) The scalar field is initialized with hand-drawn strokes. (b) Optimization towards P-curves with  $\theta = 66.5^\circ$  using the default parameters (as in Fig. 14) fails. (c) We start from  $\lambda_{\text{fair}} = \lambda_{\text{grad}} = \lambda_{\text{angle}} = 10^{-4}$ , then gradually increase  $\lambda_{\text{angle}}$  to 3 and decrease  $\lambda_{\text{grad}}$  to 0. The lower  $\lambda_{\text{angle}}$  at the beginning of optimization helps to avoid singularities. The error  $E_{\text{angle}}$  is successfully reduced to  $6.69 \times 10^{-7}$  for  $\theta = 66.5^\circ$ .

of the reference surface  $S$  can be an advantage. On a negatively curved surface  $S$ , this is easy and one can even achieve  $\theta = 0$  by placing the strips along asymptotic curves. However, for positively curved  $S$ , P-curves to a constant angle  $\theta$  may be too curved and not form a family of non-intersecting curves, resulting in an optimization failure. Thus, we propose the following method which generates well-arranged P-curves with small angles  $\theta$  and allows for a variation of the angle  $\theta$  between different P-curves.

We aim at P-curves to sufficiently small angles  $\theta$  which arise as level sets of  $F$ . The computation of the angle assignment function  $\theta(F)$  proceeds as follows. Given a smooth function  $F$  on  $S$ , we compute the angle  $\theta$  between surface normal  $\mathbf{n}$  and level set binormal  $\mathbf{b}$  at each vertex. Then, the data  $(\theta, F)$  is fitted by a polynomial curve  $P_c$ , and the target angles on the vertices for the next iteration are obtained from  $P_c$ . Supervised by the fairness terms  $E_{\text{fair}}$  and  $E_{\text{grad}}$ , and iteratively updating  $P_c$ , a family of nearly parallel P-curves is generated. In our implementation, we use a polynomial of degree 3 to fit the data  $(\theta, F)$ . To yield a smaller  $\theta$ , we decrease the target angle  $\theta$  by 5% each time we update the polynomial  $P_c$ . Fig. 16 shows an example where we obtain a family of P-curves whose angles  $\theta$  vary from  $36.2^\circ$  to  $67.1^\circ$ .

### 3.3 Co-optimizing reference surface and level sets

In view of the targeted applications, we use a level-set formulation to obtain families of pseudo-geodesics where exactly one curve passes through each surface point. For a given surface and angle  $\theta$ ,

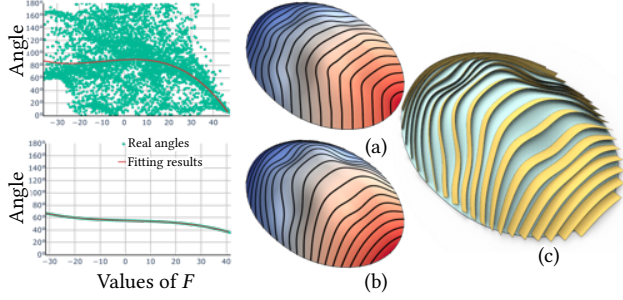


Fig. 16. Starting with an initial scalar field  $F$  (a), we compute the angles  $\theta$  between  $\mathbf{b}$  and  $\mathbf{n}$  on the vertices, and fit  $(\theta, F)$  (green dots) by a polynomial curve (red, top left). The angles  $\theta$  are far from the fitted curve and the level sets are far from P-curves, since for each value of  $F$ , the angles are not constant. The final optimized P-curves and the corresponding angles for each level set are shown in (b) and bottom left. The extracted rectifying developable strips are shown in (c).

such families may not exist. As an example, take a spherical patch and a small angle  $\theta$ . P-curves then are circles of a small radius. They will have an envelope and the area beyond the envelope will not be covered at all. Thus, we implemented algorithms that can change the surface and the angle  $\theta$ .

Given a scalar function  $F$  on a triangle mesh  $S$ , we can optimize  $S$  such that the level sets become the desired P-curves. Fig. 17 shows examples, where we changed the angles of P-curves by optimizing  $S$  while keeping the values of  $F$  at the mesh vertices unchanged.

The energy  $E_{pg}$  is written as

$$E_{pg}^M = \lambda_{fair}^S E_{fair}^S + \lambda_{close}^S E_{close}^S + \lambda_{angle}^S E_{angle}^S + \lambda_{edge}^S E_{edge}^S.$$

Here  $E_{angle}^S$  has the same expression as  $E_{angle}$ , but now values of  $F$  at mesh vertices  $\mathbf{v}$  are fixed, while vertices  $\mathbf{v} \in \mathcal{V}$ , binormal vectors  $\mathbf{b}$  and side vectors  $\mathbf{u}$  are variables.  $E_{fair}^S$  is the fairness energy, which contains 3 terms:  $E_{fair}^S = E_H^S + E_{uni}^S + E_c^S$ . The first term  $E_H^S$  is the squared Hessian energy of the  $x, y$  and  $z$  coordinates of the mesh vertices [Stein et al. 2018], providing smooth surfaces during optimization.  $E_{uni}^S$  is the uniform Laplacian energy [Taubin 1995] which prevents vanishing triangle areas. The third term  $E_c^S$  is to provide smooth level sets on the surface. Taking the notations in Fig. 7,

$$E_c^S = \sum_{\mathbf{v}_0 \in \mathcal{V}} \left\| \frac{\mathbf{p} - \mathbf{v}_0}{\|\mathbf{p} - \mathbf{v}_0\|} + \frac{\mathbf{q} - \mathbf{v}_0}{\|\mathbf{q} - \mathbf{v}_0\|} \right\|^2.$$

$E_{close}^S$  is the approximation energy:

$$E_{close}^S = \sum_{\mathbf{v} \in \mathcal{V}} ((\mathbf{v} - \mathbf{v}^*) \cdot \mathbf{n}_i^*)^2 + \|\mathbf{v} - \mathbf{v}_{ori}\|^2, \quad (14)$$

where  $\mathbf{v}^*$  is the closest point of  $\mathbf{v}$  on  $S$ ,  $\mathbf{n}_i^*$  is the normal vector of the initial surface on  $\mathbf{v}^*$ , and  $\mathbf{v}_{ori}$  is the original position of  $\mathbf{v}$ . This term allows the vertices to move within the tangent planes of the reference mesh, preventing large-scale deviations from the initial surface.  $E_{edge}^S$  restricts the changes in edge lengths,

$$E_{edge}^S = \sum_{e \in \mathcal{E}} (l_e - l_e^*)^2, \quad (15)$$

where  $l_e^*$  is the edge length of the original corresponding edge of  $e$ .  $E_{edge}^S$  plays a role as a regularizer to prevent mesh shrinkage.

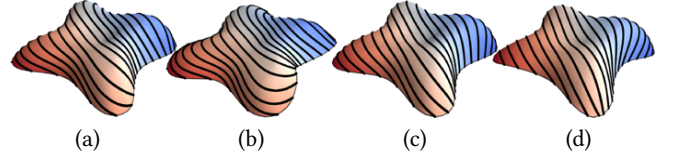


Fig. 17. The level sets in (a) are P-curves of  $\theta = 60^\circ$ . Using our triangle mesh optimization method described in Section. 3.3, we can optimize the underlying triangle mesh as shown in (b) so that  $\theta = 45^\circ$ . Similarly, (c) shows P-curves of  $\theta = 75^\circ$ . They are optimized as geodesics in (d). Observe the shape differences in the reference surface.

### 3.4 Computing rectifying strips

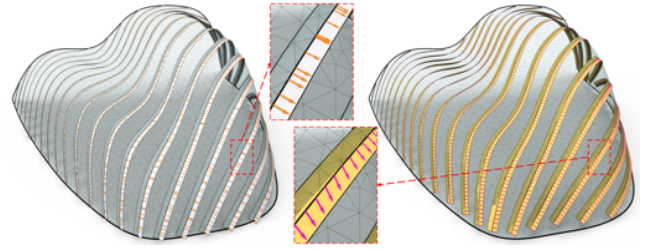


Fig. 18. Polylines and binormal vectors (left) extracted from the level sets and the optimized strips with discrete ruling vectors (right), that can be developed into straight strips. Fig. 41 and Table 4 show the evaluation for scalar field optimization and quality of the developable strips, respectively.

In this step we initially extract the level sets and improve the accuracy of angles between binormals and surface normals. The major part is devoted to the computation of the rectifying strips.

Having the scalar function  $F$  at mesh vertices, we linearly interpolate in the triangles and extract those level sets on which we build the strips. The  $n$  level sets correspond to the  $n$  scalar values sampled uniformly from  $[\min(F), \max(F)]$ . Energy  $E_{grad}$  guides towards level sets uniformly distributed on the surface. However, the level polylines are initially uneven and binormals and the angle  $\theta$  are not highly accurate, see Fig. 18 (left). We re-sample the polylines and improve the angle  $\theta$  between binormals and surface normals by optimization. We still use Eq. 8 - 10, but on vertices  $\mathbf{c}_i$  of the polylines instead of vertex stars of the triangle mesh. Next, we explain the subsequent computation of discrete rectifying strips; see also Fig. 18 (right).

*First version of rectifying strips.* The discrete Frenet frame in (1) is attached to the vertices of a discrete curve  $C$ . The discrete rectifying planes  $R_i : (\mathbf{x} - \mathbf{c}_i) \cdot \mathbf{n}_i^p = 0$  define a discrete developable surface  $R$  with intersection lines  $L_i := R_{i-1} \cap R_i$  as discrete rulings. Note that the edges of the polyline  $C$  do not lie on  $R$ . Fig. 19 illustrates the geodesic property of  $C$  w.r.t.  $R$  by the planar unfolding of  $R$ , where the vertices of  $C$  are nearly aligned along a straight line.



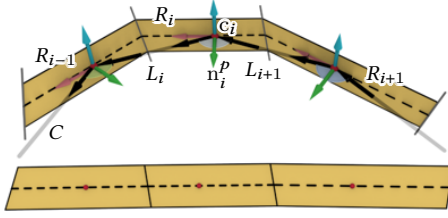


Fig. 19. The outer bisecting planes at vertices of a discrete curve (polyline)  $C$  define a discrete rectifying developable; unfolding it together with the vertices of  $C$  results in nearly collinear points.

**REMARK 1.** At each vertex  $c_i$ , the reflection of the edge line  $c_{i-1}c_i$  at the plane  $R_i$  yields the next edge line  $c_i c_{i+1}$ . Hence, the entire polyline  $C$  is obtained by subsequent reflection at the discrete surface  $R$ . Reflection polylines are one version of discrete geodesics on a surface (see e.g. [Pottmann and Wallner 2001], p. 494). A reflection polyline at a smooth surface  $S$  gets finer as the edge lines get more tangential to  $S$ , and in the limit one obtains a geodesic.

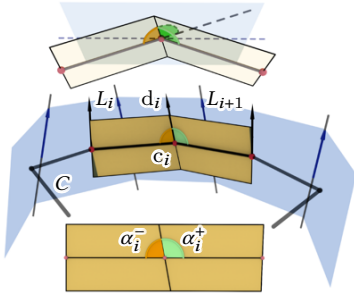


Fig. 20. Another version of a discrete rectifying developable strip, formed by a sequence of planar quads, with a precisely straight unfolding. This requires the discrete rulings to lie in the outer bisecting planes (blue) of the central polyline  $C$ .

*Preferred version of rectifying strips.* We now turn to another discrete rectifying developable, which we did not see in prior work. It is a strip of planar quads that contains the given polyline  $C$  as a medial line and unfolds to a straight rectangular strip in the plane (see Fig. 20). The straight unfolding requires us to find discrete ruling lines  $d_i$  through  $c_i$ , so that the sum of angles  $\alpha_i^-$  and  $\alpha_i^+$  between  $d_i$  and the two edges through  $c_i$  equals  $\pi$  (see Fig. 20). The extension of one edge beyond  $c_i$  forms the angle  $\pi - \alpha_i^- = \alpha_i^+$  with  $d_i$ , showing that  $d_i$  must lie in the outer bisecting plane  $R_i$  at  $c_i$ , or  $d_i$  must be orthogonal to the discrete principal normals  $n_i^p$ ; With unit edge vectors  $e_i$  this leads to the constraints

$$e_i - \frac{c_{i+1} - c_i}{\|c_{i+1} - c_i\|} = 0, \quad n_i^p - \frac{e_i - e_{i-1}}{\|e_i - e_{i-1}\|} = 0, \quad d_i \cdot n_i^p = 0. \quad (16)$$

Moreover rulings  $d_{i-1}$  and  $d_i$  must lie in a plane  $R_{i-1,i}$  through  $c_{i-1}, c_i$ . Reflecting the plane  $R_{i-1,i}$  at  $R_i$  yields the next plane  $R_{i,i+1}$ . Thus, one may choose one of these planes and get the others by successive reflection, resulting in a discrete rectifying developable. This degree of freedom, only present in the discrete model, is removed by optimization, aiming at proximity to the previously discussed model  $R$  (Fig. 19). We compute initial guesses for  $d_i \in R_i$  by averaging the unit direction vectors of the two discrete rulings  $L_i$  and  $L_{i+1}$  in  $R_i$  (see Fig. 20). This initial choice does not guarantee coplanar lines  $d_{i-1}$  and  $d_i$ . Thus, we finally optimize for planarity

while keeping  $d_i \in R_i$ . To express that rulings  $d_i, d_{i+1}$  are coplanar with the edge vector  $e_i$ , one introduces a unit normal vector  $n_{i,i+1}^p$  (discrete principal normal attached to the edge) as auxiliary variable and expresses orthogonality to the three vectors,

$$n_{i,i+1}^p \cdot d_i = 0, \quad n_{i,i+1}^p \cdot d_{i+1} = 0, \quad n_{i,i+1}^p \cdot e_i = 0, \quad \|n_{i,i+1}^p\|^2 = 1. \quad (17)$$

Constraints (16) and (17) are quadratic in the variables  $c_i, e_i, d_i, n_i^p, n_{i,i+1}^p$  if the values of the denominators are taken from the previous iteration. The constraint system is formulated as a nonlinear least squares problem and solved by a Levenberg-Marquardt method, adding the standard fairness energy  $\sum \|c_{i-1} - 2c_i + c_{i+1}\|^2$  as a regularizer with a small weight that is set to 0 in later iterations. For details on this type of optimization, see [Tang et al. 2014]. The planarity of quads, straightness of the unfolded strips and approximation errors for selected models are presented in Table 4.

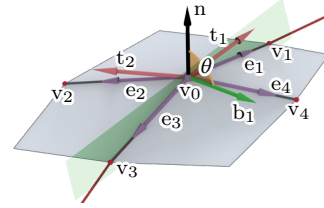


Fig. 21. Notation for optimization of a quadrilateral strip structure. To make the polyline through  $v_3 v_0 v_1$  a P-curve, the angle  $\theta$  between binormal  $b$  and surface normal  $n$  has to attain the assigned value.

### 3.5 Optimization of base meshes for strip structures

If we have more than one family of strips in a structure, it is an advantage if the strips are attached to the two or three families of major mesh polylines of a quad mesh or a triangle mesh (in case of a 3-web) with regular combinatorics. Initial versions of such a base mesh are extracted from level sets, but we need to strictly enforce above constraints per mesh polyline that defines a strip. Using the notation of Fig. 21, we discuss the case of a quad mesh and consider a vertex star with center  $v_0$  and neighbors  $v_1, \dots, v_4$ . As in Eq. (1), we add tangents  $t_1, t_2$  for mesh polylines from edge vectors via

$$t_1 = (e_1 - e_3) / \|e_1 - e_3\|, \quad t_2 = (e_2 - e_4) / \|e_2 - e_4\|.$$

The unit normal  $n$  at  $v_0$  satisfies constraints,

$$n \cdot t_1 = 0, \quad n \cdot t_2 = 0, \quad \|n\|^2 = 1.$$

To express a discrete P-curve with angle  $\theta$ , e.g. in tangent direction  $t_1$ , we use a binormal  $b_1$  and the constraints

$$e_1 \cdot b_1 = 0, \quad e_3 \cdot b_1 = 0, \quad \|b_1\|^2 = 1,$$

as well as the angle constraints Eq. (8). Again a Levenberg-Marquardt algorithm is used. The parameters and statistics for mesh optimization are presented in Table 3. Similar to [Schling et al. 2022], this mesh optimization can serve for interactive editing of structures while keeping the constraints. Examples for designing rectifying strip structures are provided in Section 4, and Section 5 extends the presented mesh optimization to torsion-free structures.

## 4 APPLICATION TO GRIDSHELLS

The construction of gridsells by bending straight flat lamellas as in [Schling et al. 2022] and [Mesnil and Baverel 2023] has been one of the motivations for the present research. These highly constrained structures are not easily designed. While optimization algorithms for design modification are known in certain cases [Schling et al. 2022], it is especially difficult to access the design space through non-trivial shapes or to approximate a given surface by such a structure. It is not surprising that the pseudo-geodesic gridsells in [Mesnil and Baverel 2023] are just rotational shapes; a main contribution in that work is the development of construction elements such as appropriate nodes. Since these are now available, one needs effective digital design tools. We show how to properly apply our optimization framework to the design of various types of gridsells.

To simplify writing, we use the following abbreviations: A=asymptotic, G=geodesic, P=pseudo-geodesic (to  $\alpha \neq 0, \pi/2$ ). Gridsells may consist of two or three families of elements. In the former case, discussed in Subsection 4.1, we have quad nets that get labeled as AG, PG, PP and so on. In the latter case (Subsection 4.2), the three families are arranged in a 3-web and labeled as AAG, AGG, PPG or similar. A Kagome pattern as in Fig. 1 arises from a 3-web by shifting the evaluation of the level-set function for one family. Table 1 provides an overview of the net/web types and their constraints discussed in our paper.

Type	$u$ constraints	$v$ constraints	$d$ constraints	surf condition	example Figs
PG	$E_{\text{angle}}$	$E_{\text{geo}}$	-	-	22
PP	$E_{\text{angle}}$	$E_{\text{angle}}$	-	-	23, 24, 25, 26
PPG	$E_{\text{angle}}$	$E_{\text{angle}}$	$E_{\text{geo}}$	-	1, 27
AAG	$E_{\text{asy}}$	$E_{\text{asy}}$	$E_{\text{geo}}$	negatively curved	28, 29
AGG	$E_{\text{geo}}$	$E_{\text{geo}}$	$E_{\text{asy}}$	negatively curved	30
TFN	Eq. 21	Eq. 21	-	-	31
PN	Eq. 22	Eq. 22	-	-	33
ACCS	Eq. 23	Eq. 23	-	parametrization (23)	32

Table 1. Different types of rectifying strip structures discussed in our paper. The  $u$ ,  $v$ , and  $d$  constraints are the constraints applied to  $u$ ,  $v$  parameter lines, and the diagonals of the structures; *surf condition* refers to conditions that the reference surfaces should satisfy. TFN denotes gridsells with torsion-free nodes (Sec. 5.2.3), PN is for gridsells with parallel node axes (Sec. 5.2.2), and ACCS is for gridsells with a family of asymptotic curves of constant slope (Sec. 5.2.4).

### 4.1 Gridsells from nets of pseudo-geodesics

The methods of Section 3 are well suited for the design of PP-nets. One needs to compute two families of P-curves which intersect at angles that do not become too small. Hence, the design of the two families in two separate steps is not ideal. Thus, we simultaneously compute both families from two level-set functions  $F, G$ . In order to constrain the angle  $\gamma$  between the level sets of  $F$  and  $G$ , we have to constrain the scalar product of normalized gradients  $g_F = \nabla F / \|\nabla F\|$  and  $g_G = \nabla G / \|\nabla G\|$  on each triangle,

$$g_F \cdot g_G = \cos \gamma, \quad g_F \cdot g_G^\perp = \sin \gamma,$$

where  $\perp$  is rotation by 90 degrees. Moreover, using our methods, we can turn asymptotic or geodesic curves into P-curves. Results are shown in Figures 22, 23, 24, 25 and 26.

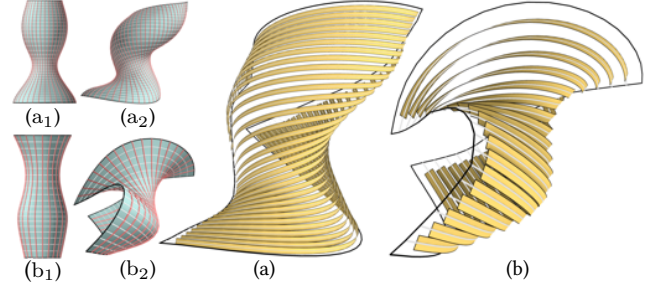


Fig. 22. Two rotational meshes (a<sub>1</sub>) and (b<sub>1</sub>) with the orthogonal PG-net of circles and profiles are isometrically deformed to meshes (a<sub>2</sub>) and (b<sub>2</sub>) using the method of [Wang et al. 2019]. Orthogonality of the net and geodesics are preserved during an isometry. While the P-curve property of circles (to different inclination angles  $\theta$ ) gets lost during isometric deformation, it is restored by optimization, yielding orthogonal PG-nets (a) and (b); for clarity only the P-strips are shown.

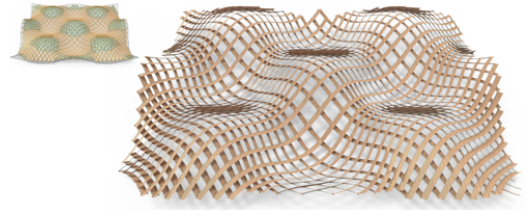


Fig. 23. A pseudo-geodesic net with angles  $\theta_1 = \theta_2 = 60^\circ$ , designed from the triangle mesh in the upper left.

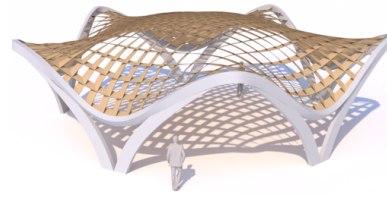


Fig. 24. PP-gridshell on a shape with varying signs of Gaussian curvature. Here, an AA-net would not be possible and GG-nets result in lower stiffness [Schikore 2023].

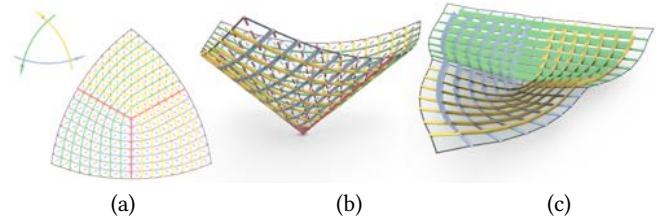


Fig. 25. A cone (a,b) is represented as a discrete orthogonal geodesic net [Rabinovich et al. 2018] with a combinatorial singularity of valence 3 at the vertex. Three straight diagonals (red) separate the net into three patches of regular combinatorics. Optimizing the geodesics to become P-curves of angle  $\theta = 50^\circ$  results in a structure (c) whose prototypical model is shown in Fig. 26.

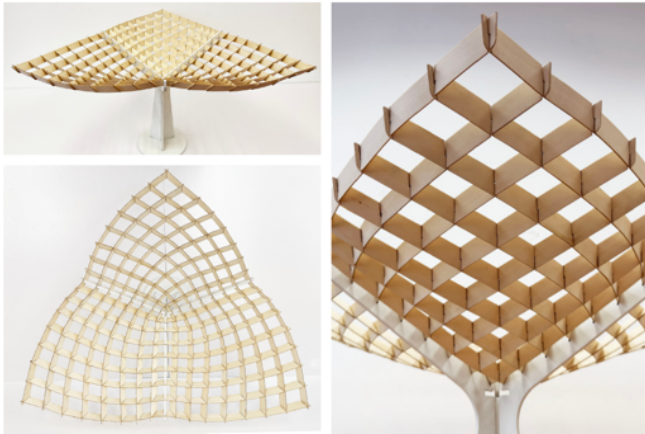


Fig. 26. Prototypical model built from timber veneer strips at an inclination angle of constant  $50^\circ$ , creating a triple symmetric cantilevering roof structure. The constant incline allows for repetition in the joints and created a beneficial orientation of beams to carry gravity loads.

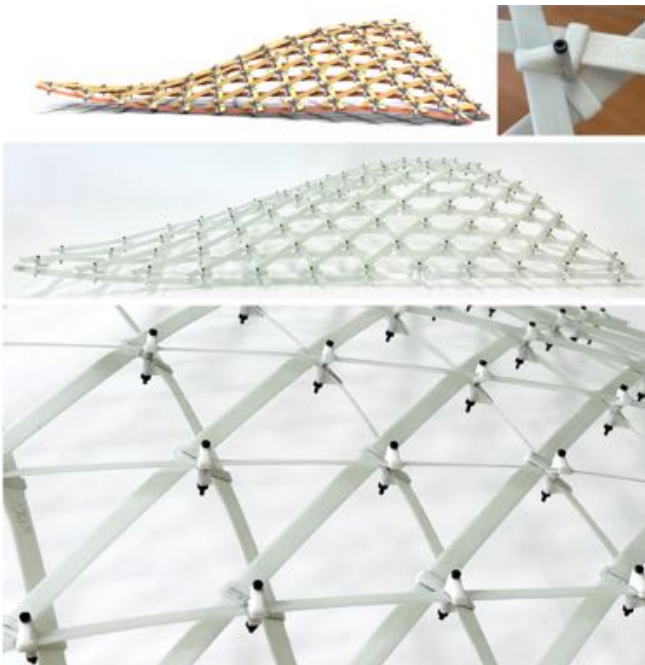


Fig. 27. Digital and physical model of a PPG-3-web. The lamellas are connected laterally through 3D-printed joints and steel bolts. The pseudo-geodesic lamellas on top (orange) and bottom (red) have  $\theta = 60^\circ$  inclination. The geodesic lamellas (yellow) are fitted in between.

## 4.2 Gridshells from pseudo-geodesic webs

It is well-known (see e.g. [Blaschke and Bol 1938]) that a 3-web on a surface can be represented by the level sets of 3 functions  $F, G, H$  which satisfy

$$F + G + H = 0. \quad (18)$$

This simple constraint makes the level-set formulation very attractive for the design of 3-webs on surfaces. However, the webs we are dealing with, do not have enough degrees of freedom to represent any shape. Hence, it is important that our framework can also optimize for the underlying surface; this is most easily done through post-processing of a web initialized via the level-set approach.

**4.2.1 AAG-webs.** The approximation of a negatively curved surface  $S$  by an AAG-web, illustrated in Fig. 28, is based on the following thoughts. Since the web consists of two families of asymptotic curves, initial representations of these two families by level-set functions  $F$  and  $H$  are easily found. However, the level-set functions are not unique; any reparameterizations  $\bar{F} = \phi(F)$  and  $\bar{H} = \eta(H)$  represent the AA-net as well, but increase the degrees of freedom to find a third function  $G = -F - H$  whose level sets are close to geodesics. Geometrically, this means discretely sampling the two families of asymptotic curves to obtain an AA-net in which one family of diagonal curves are geodesics, which slide nicely in between the other two families of lamellas and create an efficient triangulation of the web. This will in general require a shape change of  $S$ , which we want to keep minimal.

Hence, we initialize  $G = -F - H$  and then simultaneously optimize all three functions  $F, G, H$  so that  $(F+G+H)^2$  is minimized, and A- and G-properties are fulfilled as well as possible; see Fig. 28. This optimization brings level sets of  $G$  closer to geodesics, but results in some deviation of the level sets of  $F$  and  $H$  from asymptotic curves. Hence, in the final step we use the mesh optimization of Sec. 3.4 to obtain a high-quality AAG-web (Figs. 28 and 29).

**4.2.2 AGG-webs.** An AGG-web has more degrees of freedom than an AAG web, since only the asymptotic curves are already determined by the input surface and even there one can select one of the two families. Here and for AAG-webs we assume that the given surface is free of flat points so that there is no singularity in the net of asymptotic curves.

We initialize the design of an AGG-web by selecting a family of asymptotic curves and first optimize to an AG-net with a weak angle constraint between the A- and G-family. Later a third level-set function is added simultaneously we optimize to the AGG property, allowing a change of the reference surface as well; see Fig. 30.

**4.2.3 Webs involving pseudo-geodesics.** In an analogous way we can compute webs which contain P-curves. For construction reasons, there should be at least one G-family in the web; see Figs. 1 and 27.

## 5 TORSION-FREE NODES IN RECTIFYING STRIP STRUCTURES

So far, we did not address the fabrication of structures from rectifying strips. If one has two or more families of strips involved, a key issue is how to connect them at the nodes where they meet. We now turn to this problem and present special structures which simplify the construction of nodes. These are torsion-free support structures that are well-known in architectural geometry [Pottmann et al. 2015]. However, the restriction to *rectifying strips* is essential and makes the problem more interesting and challenging.

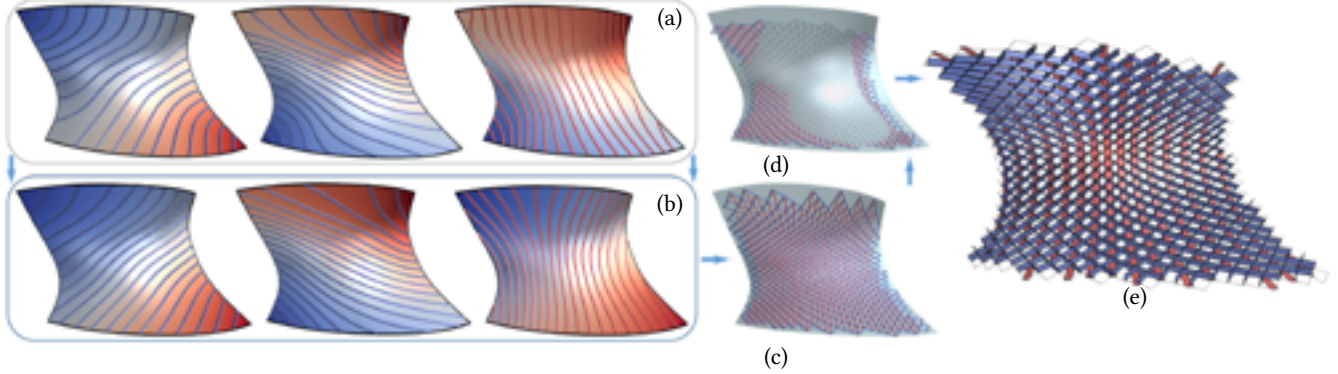


Fig. 28. Approximation of a negatively curved surface by an AAG-web. We first optimize the scalar functions  $F$  and  $H$  for the two families of asymptotic curves separately, shown as first and second image in (a), and initialize the function  $G = -F - H$ , third image in (a).  $G$ 's level sets are not geodesics yet. Then, the 3 scalar functions together are optimized towards a lower overall deviation from the AAG-property (b). Then, a web is extracted from the level sets (c); it is not yet an AAG-web with sufficient precision. Hence, mesh optimization is applied as a post-processing procedure, resulting in an accurate AAG-web (d) that exhibits some deviation from the reference surface. (e) shows the rectifying strip structure defined by the web. The evaluation of the AAG web in (e) is shown in Table 3.

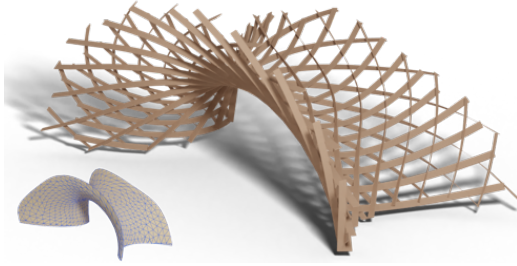


Fig. 29. AAG-web approximating a reference surface (bottom left).

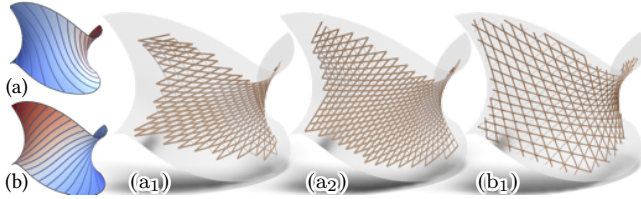


Fig. 30. AGG-webs approximating a given surface  $S$ . Choosing one family of asymptotic curves (a), the angles between A and one family of G level sets are close to  $90^\circ$  (a<sub>1</sub>) and  $120^\circ$  (a<sub>2</sub>), respectively. Choosing the other asymptotic directions (b) gives us different patterns, e.g. the one in (b<sub>1</sub>).

### 5.1 Nodes and the local geometry of rectifying strips

The rulings of a rectifying strip with central curve  $c$  are given by the Darboux vectors  $d = \tau t + \kappa b$ . They are equal to the binormals  $b$  only at points with vanishing torsion  $\tau$ . However, in all practical realizations of asymptotic gridshells, one assumes a small deviation from binormals and builds nodes with binormals as axes (see inset). In view of tolerances at nodes, this is a realistic assumption, drawing also on some non-isometric deformation behavior of the material being used for the strips (timber, metal, certain plastics). In other words, one models



the strip by the binormal surface rather than the rectifying developable. Proposition 1, whose proof is given in the Appendix, quantifies the deviation of the binormal surface from developability.

**PROPOSITION 1.** *The ruled surface  $B(u, v) = c(u) + vb(u)$  formed by the binormals of a space curve  $c$  has Gaussian curvature*

$$K(u, v) = -\left(\frac{\tau}{1 + \tau^2 v^2}\right)^2. \quad (19)$$

In particular, on each binormal ( $u = \text{const.}$ ), the point with maximal  $|K|$  belongs to  $v = 0$ , i.e., it is the curve point  $c(u)$ . Thus, the maximum deviation from developability of the binormal surface is along the curve  $c$  itself. Just for a planar curve  $c$  ( $\tau \equiv 0$ ), the binormal surface agrees with the rectifying developable and is a cylinder with rulings orthogonal to  $c$ 's plane.

It is also good to understand how much the rectifying developable  $D$  deviates from the binormal. This requires the computation of the normal curvature of  $D$  in direction  $b$ , which follows easily from Euler's formula, as shown in the Appendix.

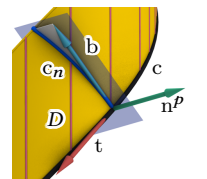
**PROPOSITION 2.** *The rectifying developable  $D$  has vanishing normal curvature  $\kappa_1 = 0$  in direction of the Darboux vector  $d$ . At  $c$ , its second normal curvature  $\kappa_2$  and normal curvature  $\kappa_n(b)$  in direction of the binormal  $b$  are*

$$\kappa_2 = \kappa(1 + k^2), \quad \kappa_n(b) = \kappa k^2, \quad (20)$$

where  $k := \tau/\kappa$  denotes the conical curvature.

The intersection curve  $c_n$  between  $D$  and the normal plane of  $c$  has curvature  $\kappa k^2$  at the curve point  $c(u)$ . The 2nd order Taylor expansion of  $c_n$  in the normal plane reads  $vb + \frac{1}{2}\kappa k^2 v^2 n^p$ , showing that a point on the binormal, at distance  $v$  from  $c$ , has an approximate distance  $|\kappa|k^2 v^2/2$  to the rectifying developable  $D$ .

By Euler's formula, the normal curvature  $\kappa_n(a)$  in any direction  $a$  of the rectifying plane, that forms the angle  $\psi$  with  $d$  equals  $\kappa_n(a) = \kappa_2 \sin^2 \psi$ . Therefore, Eq. (20) allows us to compute the



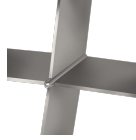
deviation of the line  $c + va$  from  $D$  near  $c$ . This is relevant if one wants to place the node axis in direction  $a$ .

The singular point  $r(u) = c(u) + v_r d(u)$  on a ruling of  $D$  is found with  $v_r = -\frac{1}{\kappa k'}$  (see e.g. [Strubecker 1964], p. 216), where  $k'$  is the derivative of  $k$  w.r.t. arc length. For our application, the singular points have to be outside the strip. Even being too close to the strip boundary is an undesirable situation for fabrication and aesthetics.

*Summary and implications on fabrication.* The binormals are close to rectifying strips for small torsion  $\tau$  only. If binormals are used as node axes on a strip along an asymptotic curve  $c$  of a reference surface  $S$ , the value  $\kappa k^2 = \tau^2/\kappa$  should be small as well. By the way, the torsion  $\tau$  of an asymptotic curve is related to the Gaussian curvature  $K$  of  $S$  via  $\tau^2 = -K$ . Large local changes  $k'$  of the conical curvature  $k$  imply that singularities come close to the strip. Conical curvature  $k$  is the curvature of the spherical curve  $t(u)$  traced out by the unit tangent vectors. Hence,  $k$  can easily be estimated from a discrete model. These practical considerations on the strips can be incorporated into optimization.

## 5.2 Torsion-free rectifying strip structures

A node where two developable strips intersect along a straight line segment is called *torsion-free* (see inset). Of course, that line segment must lie on a ruling of both strips. Torsion-free nodes provide advantages for fabrication. Thus we now focus on such nets of rectifying strips which possess only torsion-free nodes and call them *torsion-free rectifying strip structures* (see Figs. 32 and 31). We first address a way to compute them by optimization and then turn to specific cases which possess advantages in design and fabrication. Those also constitute good initial models for design modification through optimization.



*5.2.1 Torsion-free structures through optimization.* The mesh optimization algorithm of Sec. 3.5 is already equipped with everything needed for the computation of torsion-free structures. One only needs to make sure that the ruling vectors of rectifying strips agree at the nodes. To initialize the computation, one can start from a net of rectifying strips and intersect discrete rectifying planes at nodes. The resulting lines are then initial guesses for the ruling vectors at nodes. We express the common ruling vectors  $d_i$  on the vertices  $v_i$ ,  $i = 0, \dots, 4$  of each mesh vertex star (see Fig. 21), such that

$$\begin{aligned} d_0 \cdot (e_i + e_{i+2}) &= 0, i = 1, 2, \\ n_{0,i} \cdot d_i &= 0, n_{0,i} \cdot d_0 = 0, n_{0,i} \cdot e_i = 0, i = 1, \dots, 4, \\ \|d_0\|^2 &= 1, \|d_i\|^2 = 1, \|n_{0,i}\|^2 = 1, i = 1, \dots, 4, \end{aligned} \quad (21)$$

where  $n_{0,i}$  is a discrete principal normal attached to the edge  $e_i$ . The first row of the equations expresses  $d_0$  as the common ruling, and the second row ensures developability. The model in Fig. 31, left, has been computed in that way. This approach will work well and cause only small shape changes if the ruling vectors of the two strips at the nodes are nowhere too different from each other. Another way of modeling uses initial structures which already possess torsion-free nodes, as the ones discussed next.

### 5.2.2 Torsion-free rectifying strip structures with parallel node axes.

The mathematical problem underlying the construction of torsion-free rectifying strip structures is to find surface parameterizations  $x(u, v)$  along which there is a 2-parameter family of straight lines  $L(u, v) = x(u, v) + wa(u, v)$  with the property that the generated ruled surfaces along parameter lines  $u = \text{const.}$  and  $v = \text{const.}$  are the rectifying developables of the corresponding parameter lines on  $S$ . Hence,  $a(u, v)$  has to agree with the Darboux vector of both parameter lines of  $x(u, v)$ . This appears to be a difficult problem, and we are not aware of any study of it in the geometry literature. In the following, we discuss the much more accessible special case of parallel node axes ( $a = d = \text{const.}$ ), which already shows that the search for torsion-free rectifying strip structures attached to a surface  $S$  is not a shape restriction on  $S$ , at least locally.

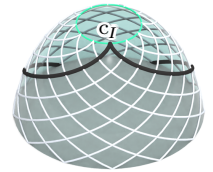
In the following we imagine the parallel node axes to be vertical and choose the constant node axis vector (Darboux vector) as  $d = (0, 0, 1)^T$ . Then, each strip  $D$  must be cylindrical with vertical rulings and the curve  $c$  is a geodesic on  $D$ . The rulings of a cylinder  $D$  appear as parallel lines in the planar unfolding and the geodesic as a straight line segment, leading to a constant intersection angle between geodesic and rulings. Thus,  $c$  intersects the vertical rulings of  $D$  under constant angle  $\gamma$  and is thus a so-called *curve of constant slope* (see e.g. [Pottmann and Wallner 2001]). The angle  $\gamma$  between curve tangent  $t$  and Darboux vector  $d$  satisfies  $\cot \gamma = \tau/\kappa = k$ . Thus, curves of constant slope are characterized by constant conical curvature  $k$ . This is in agreement with above-mentioned properties: The curve  $t(s)$  is a circle and thus has constant curvature  $k$  and the singular points on  $D$  are at infinity due to  $k' = 0$ . The slope is defined via the inclination angle  $\beta$  against horizontal planes as

$$\tan \beta = \tan(\pi/2 - \gamma) = k.$$

Slope 0 belongs to curves in horizontal planes. Principal normals of a curve of constant slope are orthogonal to the vertical cylinder  $D$  and thus horizontal; binormals thus have the constant inclination angle  $\beta$  against the vertical direction.

*5.2.3 Computational design.* Even after fixing the node axis direction  $d$ , there are still infinitely many choices for the curve net on  $S$ . However, a curve of constant slope  $k \neq 0$  cannot reach those areas of  $S$  whose tangent planes have a slope smaller than the chosen one.

The boundary of reachable points is a so-called isophote  $c_I$  on  $S$ , whose tangent planes possess the fixed slope  $k$ . Curves of the same constant slope  $k$  already form a net of curves, which ends with cusps on  $c_I$ , since in areas with steeper tangent planes there are two directions of given slope  $k$  (see the inset for an illustration).



There is no need to choose the same slope. Here our level-set method is very useful, as it can easily be adapted to curves of constant slope. One just has to replace the angle condition for P-curves by one which expresses a constant angle between unit tangents  $t$  of level sets (gradients in triangles rotated by 90 degrees) and  $d$ :

$$t \cdot d = \cos \gamma. \quad (22)$$

Now tracing gets simpler and user-guided patterns via strokes is a good design tool as well; see Fig. 33 for a result. Likewise, mesh optimization is easily adapted to this case; one just has to express the ruling vectors as described in Eq. 21.

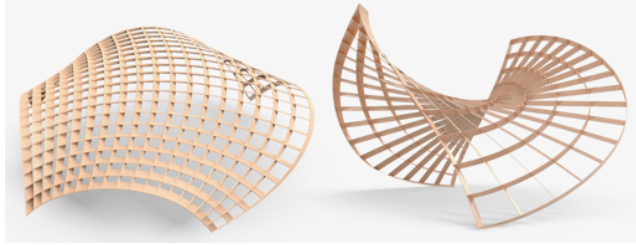


Fig. 31. Torsion-free rectifying strip structures, generated by optimization from a PP-net with torsion in the nodes (left), and by editing part of the model in Fig. 32 so that node axes are no longer parallel (right).

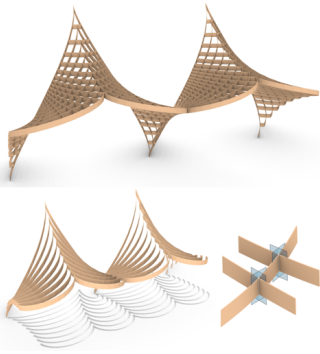


Fig. 32. Torsion-free rectifying strip structure with vertical node axes. Strips of one family (bottom left) are orthogonal to the reference surface; the other strips follow curves in horizontal planes. Tangent planes (blue) at node axes are orthogonal (bottom right); nodes along the same non-horizontal strip are congruent.

**5.2.4 Structures with parallel node axes and one family of strips orthogonal to the reference surface.** Cylindrical strips may form small angles with the reference surface  $S$ , which could be undesirable. Since both families of rectifying strips in a torsion-free structure cannot be orthogonal to  $S$  unless  $S$  is a plane, the question arises whether at least one family of them can be attached orthogonal to  $S$ . The answer is affirmative. If cylindrical strips are orthogonal to  $S$  they must follow asymptotic curves on  $S$ . Hence, we are looking for surfaces where one family of asymptotic curves are of constant slope with respect to the same reference direction, but slopes vary across asymptotic curves. All these surfaces have been determined in [Brauner 1968], resulting in an explicit parameterization that involves a solution  $f(u, v)$  of the heat equation,

$$\begin{aligned} x &= e^v(f_u \sin u + f_v \cos u), \quad y = e^v(f_v \sin u - f_u \cos u), \\ z &= \mu(f + f_v), \quad \text{with } f_{uu} - f_v = 0; \mu = \text{const.} \end{aligned} \quad (23)$$

Here, the  $u$ -lines  $v = \text{const.}$  are the asymptotic curves of constant slope. The  $v$ -lines  $u = \text{const.}$  are conjugate to the horizontal curves  $z = \text{const.}$ ; along each of them the horizontal tangents are parallel. We summarize our findings:

**PROPOSITION 3.** *Torsion-free rectifying strip structures with parallel node axes are formed by the rectifying cylindrical strips of a net*

*of curves of constant slope on the reference surface  $S$ . If, in addition, the strips of one family are orthogonal to  $S$ , they follow one family of asymptotic curves in a class of surfaces  $S$ , which in an adapted coordinate system can be parameterized as in Eq. (23).*

The case with  $f(u, v) = e^{-v}(\cos u + u \sin u + 2v \cos u)$  is the basis of the structure in Fig. 32. This periodic surface, studied in detail in [Brauner 1968], is also a translation surface. The generating curves are those in horizontal planes and the  $v$ -lines (in vertical planes). This could be interesting for a watertight skin using flat panels.

We note the following constructive advantage of all surfaces in [Brauner 1968]: Since the principal normals of the asymptotic curves of constant slope are horizontal tangents of  $S$ , the asymptotic curves are also curves of steepest descent, forming an orthogonal curve net with the horizontal curves. Using the latter for the second family of strips (as in Fig. 32) results in *repetitive torsion-free nodes*. At each node axis, the tangent planes of the two strips are orthogonal and one can build nodes, aligned with strip boundaries, that are congruent along each non-vertical strip.

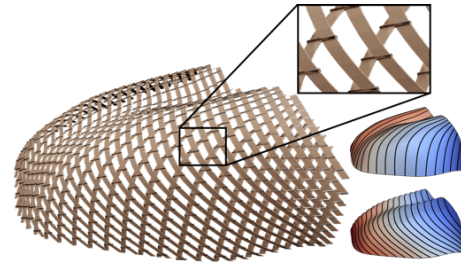
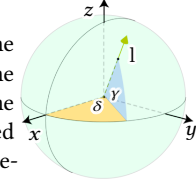


Fig. 33. The level-set method is used to design two families of curves of constant slope (right) as the basis of a torsion-free rectifying support structure with parallel node axes (left). The intersection angle at nodes is  $65^\circ$ ; the detail shows the node axes and the arrangement of strips in two layers.

## 6 SHADING SYSTEMS

A family of appropriately arranged strips, mounted on a facade or transparent roof, can function as a shading system. The topic of shading systems for freeform skins has previously been addressed by [Wang et al. 2013]. There, a torsion-free support structure composed of individually shaped flat faces is used, which is harder to fabricate than the simple system we propose. The construction of our system amounts just to fixing the originally flat straight strips of bendable material appropriately on the underlying structure. We do not address here potential ways of fixation, but clearly it is helpful to have control over the rulings of the strips or their deviation from straight lines as discussed in Sec. 5. Our focus is on the challenging computational problem of designing rectifying developable strips along a freeform surface  $S$ , so that they provide shading at chosen times of the day; here  $S$  can be a freeform architectural skin or an auxiliary design surface in front of a flat facade so that the shading system becomes a design element. We also consider the case where in some areas the system does not block the direct sunlight, while in other areas it does.

In order to function as a shading system, the rectifying developable strips have to block the light for selected light directions. We assume these to be given by unit vectors  $l_i$ , oriented from the earth towards the sun. Vectors  $l_i$  describe a set of points on the unit sphere  $S^2$ . During a single day, the points  $l_i$  lie on a circle  $\subset S^2$ , whose rotational axis is parallel to the earth's axis. Its radius depends on geographic latitude and date. Independent of latitude, it is a great circle at the equinox.



The user can select the light directions for which shading should be provided via a region  $R_l \subset S^2$ . To simplify its representation, we use a Cartesian  $(x, y, z)$  system whose  $z$ -axis is parallel to the earth axis and whose  $xz$ -plane is parallel to the meridian plane on which the building is located, as shown in the inset. In this system, we use angles  $(\delta, \gamma)$  with  $l = (\cos \delta \cos \gamma, \sin \delta \cos \gamma, \sin \gamma)$ . Within one year,  $\gamma \in [-23.5^\circ, 23.5^\circ]$ , and  $\delta \in [-90^\circ, 90^\circ]$  during a day at the equinox. In our implementation,  $R_l$  corresponds to a choice of intervals for  $\delta$  and  $\gamma$  and thus the constraint  $l = (l_x, l_y, l_z) \subset R_l$  amounts to

$$\begin{aligned} \gamma \in [\gamma_{\min}, \gamma_{\max}] &\iff l_z \in [z_{\min}, z_{\max}], \\ \delta \in [\delta_{\min}, \delta_{\max}] &\iff l_y/l_x \in [g_{\min}, g_{\max}], \end{aligned} \quad (24)$$

where  $z_{\min} = \sin \gamma_{\min}$ ,  $z_{\max} = \sin \gamma_{\max}$ ,  $g_{\min} = \tan \delta_{\min}$ ,  $g_{\max} = \tan \delta_{\max}$ . Inequality constraints Eq. (24) are incorporated into our optimization framework as equality constraints through dummy variables  $\psi$ . An inequality  $x \geq a$  is written as  $x - a - \psi^2 = 0$ .

We study two different effects w.r.t. the incoming light, namely blocking the light for shading or letting it through in certain parts. In both cases, we minimize an objective function of the form

$$E_{\text{tot}} = \lambda_1 E_{\text{fair}} + \lambda_2 E_{\text{grad}} + \lambda_3 E_L, \quad (25)$$

where  $E_L$  controls the position of rectifying planes of level sets w.r.t. the light. This depends on the purpose and is now discussed in each of the two cases, together with an appropriate initialization.

*Shading.* A rectifying developable strip along a curve  $c$  on the given surface  $S$  blocks light in direction  $l$  most efficiently, if  $c$ 's rectifying planes are orthogonal to  $l$ . This is expressed with the energy  $E_L = E_{\text{block}}$ ,

$$E_{\text{block}} = \sum_{v \in \mathcal{V}} ((l \cdot t)^2 + (l \cdot b)^2) \mathcal{A}(v), \quad (26)$$

where  $t$  and  $b$  are unit tangent and unit binormal vectors, respectively. If  $l$  was a constant light direction, minimizing  $E_{\text{block}}$  to 0 would result in a straight line  $c$  orthogonal to  $l$ , which will in general not exist on  $S$ . However, we apply the constraint with variable  $l = l_i$  for each vertex  $v_i \in S$  so that  $l_i \in R_l$ .

Our pipeline for the design of shading systems is seen in Fig. 34. To initialize optimization, we take an even simpler approximation by representing the region  $R_l$  as an arc of a great circle. For buildings not too far away from the equator, this great circle can be a circular arc in our local  $xy$ -plane. For simplicity of description, we confine ourselves here to this case. A strip  $D$  orthogonal to this choice of light must have tangent planes parallel to the  $z$ -axis and thus lie on a general cylinder with  $z$ -parallel rulings. It follows from Sec. 5.2.2 that the strip curve  $c$  is of constant slope. Hence, initialization is done with a family of curves of constant slope. The simplest

choice would be given by planar curves, which in the adapted coordinate system are the level sets of the function  $z$  restricted to  $S$ . While those curves are easiest to get, they may not be appropriate, since we need their normals to be in the selected light directions to be blocked. It is therefore important to have the entire family of curves of constant slope as candidates for initialization, as shown in Fig. 35.

Assuming  $R_l$  to be an arc of a great circle, we compute in each triangle of the mesh the range of vectors that are orthogonal to one light ray from  $R_l$ . Then, we compute their slopes and select the most common slope value for initialization (see Figs. 34 and 35).

*Not blocking light.* One may want to let the light pass the system in certain areas or at specific times of the day. This lighting effect requires that the light direction is coplanar with the rectifying planes of strip curves  $c$ , i.e.,  $l$  is orthogonal to their principal normal vectors. Hence, we use  $E_L = E_{\text{light}}$ ,

$$E_{\text{light}} = \sum_{v \in \mathcal{V}} (l \cdot n^p)^2 \mathcal{A}(v), \quad (27)$$

where  $n^p$  is the principal normal vector that satisfies  $n^p \cdot t = 0$ ,  $n^p \cdot b = 0$ ,  $\|n^p\|^2 = 1$ .

As an initialization, we choose a central light direction  $l_c \in R_l$ , corresponding to  $\delta = (\delta_{\min} + \delta_{\max})/2$ ,  $\gamma = (\gamma_{\min} + \gamma_{\max})/2$ . Initial curves  $c$  shall have rectifying planes parallel to  $l_c$ , which means that they are geodesics on cylinders whose rulings are parallel to  $l_c$ . As before, they are curves of constant slope, but now the angle  $\gamma$  is measured against  $l_c$ . The present case is a bit easier than for shading, since the constant angle  $\gamma$  is the only constraint to be imposed.

Fig. 36 shows examples for different lighting effects. The model in Fig. 37 has been split into various parts with shading systems of different function, and also includes a PP-net.

We can apply different shading/lighting constraints on the same model without splitting it into separate meshes, as shown in Fig. 38. We first selected the east side as the part where sunlight should pass through. Following that is a transition area where only smoothness is applied so that there is a fair transition into the remaining part that is optimized for shading.

## 7 DISCUSSION

Our algorithms have been implemented in C++ and tested on an Intel Xeon E5-2687W 3.0 GHz processor. Tables 2, 3 and 4 as well as Figs. 40 and 41 provide details on the used parameters and performance for selected models of various levels of complexity. Our implementation is available on our project web page.

*Limitations.* Some limitations are of a fundamental geometric nature. Heavily constrained structures such as AAG- or AGG-webs do not exist on general negatively curved surfaces. Thus our algorithms may or may not find a result, even when allowing small deviations from the reference surface. Here, the avoidance of combinatorial singularities plays a role as well. Driven by the aim to avoid self-intersections or optimization failures of level sets, we also ruled out combinatorial singularities (Fig. 42). They are however allowed for the mesh optimization in Sec. 3.4. For practical applications, the composition of combinatorially regular parts towards a final design may be sufficient, as illustrated in Fig. 26.

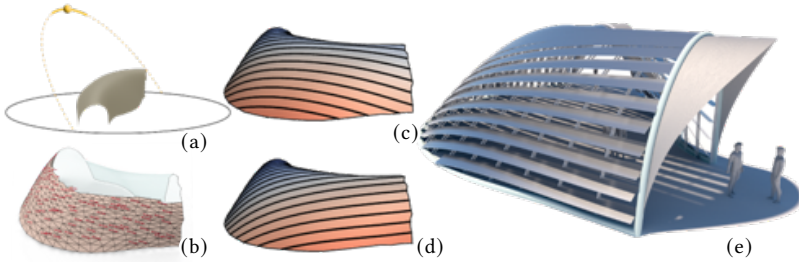


Fig. 34. Given a reference surface and the considered sunlight directions (great circular arc) (a), we select a region for placing a shading system and compute the best slope against the earth axis (red arrows in (b) show their direction). This serves for initializing the scalar function  $F$  (c). After optimizing the level-sets (d) with consideration of the more specific sunlight region, the rectifying strips of the shading system (e) are obtained as in Sec. 3.4. The time and location of the scene is August 1st, 15:00, Los Angeles.

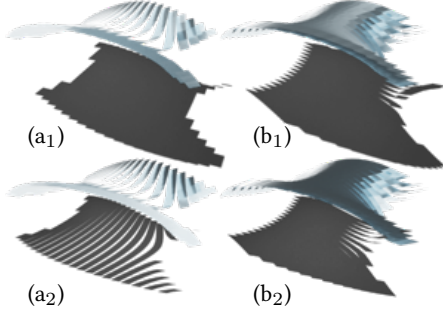


Fig. 35. Built on the Tropic of Cancer, the two families of strip curves (a) and (b) have a constant slope w.r.t. the earth axis. The curves (a) are planar ( $\gamma = 90^\circ$ ), the curves (b) have  $\gamma = 30^\circ$ . On the summer solstice, both (a<sub>1</sub>) and (b<sub>1</sub>) provide proper shading, but on the winter solstice, (b<sub>2</sub>) is more effective than (a<sub>2</sub>). Thus the slope helps to provide shading for a larger variety of sun positions.

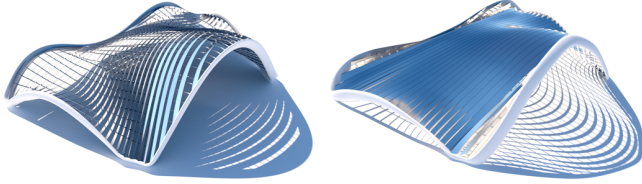


Fig. 36. A comparison of blocking light (left) and letting it through (right). Designed on the same reference surface, different target functions result in structures that generate different shadow patterns on the ground. This scene is in Makkah, 12:00 on December 1st.

Fig.	$ V $	$bbd$	$\lambda_{fair}$	$\lambda_{grad}$	$\lambda_{angle}$	$T/iter$	$E_{angle}$
1	3021	62.24	1e-3	1e-4	10	0.54	7.92e-3 6.89e-3 3.36e-3
12	3021	62.24	1e-3	1e-4	30	0.047	2.8e-6
14(bottom)	620	35.82	1e-3	1e-4	10	0.021	5.45e-7
23	1394	176.62	1e-4	1e-4	30	0.088	7.65e-3 0.12
28	323	551.27	1e-3	0	30	0.019	5.04e-3 6.10e-3 4.62e-6
33	3330	74.32	1e-2	0	1e3	0.017	5.32e-5
36(a)	4427	102.55	0.1	0.3	1e-3	0.46	0.32
36(b)	4427	102.55	0.01	0.1	3e-4	0.42	2.15

Table 2. Level set optimization statistics to selected figures.  $|V|$  is the number of vertices,  $bbd$  is the diagonal length of the axis-aligned bounding box of the surface. The multiple  $E_{angle}$  values of one model are for: Fig. 1:  $\theta_1 = 45^\circ$ ,  $\theta_2 = 30^\circ$ , G; Fig. 23:  $\theta_1 = 30^\circ$ ,  $\theta_2 = 60^\circ$ ; Fig. 28: A, A, G. Note that the shading energy (see Fig. 36) needs not to have a very small value, since in general there is no exact solution.

A restriction of the current implementation is of a topological nature. The reference surface must be a topological disk, possibly

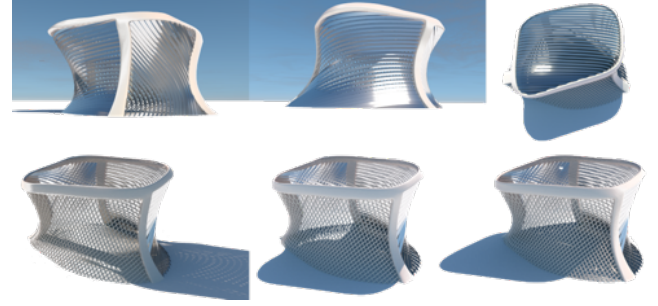


Fig. 37. Strip patterns with different functions on the same model. The roof (top right) and the south patch (top middle) are designed to block light, and the east patch (top left) is designed to let the light pass through the building. The north side of the building (bottom row) is a PP-net with P-curves of  $60^\circ$ , so that we can see through. We set the location to Vienna and the time of the bottom row to 8:00, 12:00, and 14:00, respectively, on August 1st. The shadow on the ground shows that the desired effects are achieved.

Fig.	$ V $	$bbd$	$\lambda_{fair}$	$\lambda_{edge}$	$\lambda_{close}$	$\lambda_{angle}$	$T/iter$	$E_{angle}$
1	933	61.92	1e-4	-	1e-4	10	0.14	3.92e-8 2.04e-8 1.41e-9
17(b)	1073	37.07	0.02	2e-3	3e-4	30	0.24	9.74e-5
17(d)	1073	37.07	1e-3	1e-4	3e-4	10	0.20	3.86e-5
23	584	175.65	1e-3	-	1e-4	1	0.063	2.47e-6 2.44e-6
28	616	521.59	1e-3	-	1e-4	50	0.053	4.41e-5 1.62e-5 2.59e-7
30(a <sub>1</sub> )	394	36.98	1e-4	-	1e-4	30	0.053	3.09e-6 1.82e-9 3.68e-9

Table 3. Quad/triangle mesh optimization statistics to selected figures. The multiple  $E_{angle}$  values of one model are for: Fig. 1:  $\theta_1 = 45^\circ$ ,  $\theta_2 = 60^\circ$ , G; Fig. 23:  $\theta_1 = 30^\circ$ ,  $\theta_2 = 60^\circ$ ; Fig. 28: A, A, G; Fig. 30 (a<sub>1</sub>): G, G, A.

Fig.	$ V $	$ Strip $	$\max(\text{diff}_\theta)$	$\max(e_{\text{straight}})$	$E_{\text{planar}}$	$\max(E_{\text{planar}})$	$Appro$ (%)
1	2005	111	2.72°	8.00e-4	3.48e-3	5.10e-5	1.43
12	1665	20	1.376°	5.04e-4	1.22e-4	4.9e-7	0.07
18	1819	20	1.59°	3.38e-3	0.063	3.78e-4	0.14
23	780	56	1.12°	2.75e-4	1.12e-4	2.78e-6	0.50
34	2853	30	2.07°	8.30e-5	5.84e-5	3.73e-7	1.19
38	6375	80	5.51°	0.013	0.38	8.01e-4	0.21

Table 4. Rectifying strip quality evaluation to selected figures.  $|Strip|$  is the number of strips,  $\max(\text{diff}_\theta)$  is the maximal deviation of  $\theta$  from the input binormal strips.  $\max(e_{\text{straight}})$  is the ratio between the maximal deviation of an unfolded strip from a straight line and the strip length.  $E_{\text{planar}}$  is the total energy on quad planarity, expressed as  $\sum_i \det(d_i, d_{i+1}, e_i)^2$ , where the unit vectors  $d_i$ ,  $d_{i+1}$  and  $e_i$  are defined as in Eq. (17). To avoid optimization failures for strips defined by curves with large torsion and small curvature (see Sec. 5.1), our optimization for developable strips (Sec. 3.4) can slightly change the base curve of a strip to obtain better rulings, leading to a possible loss of accuracy in  $\theta$ . This deviation is usually minor as shown by  $\max(\text{diff}_\theta)$  values.  $Appro$  is the ratio (in percentage) of  $h$  to  $bbd$ , with  $h$  as Hausdorff distance between input surface and base curves of strips and  $bbd$  as bounding box diagonal.



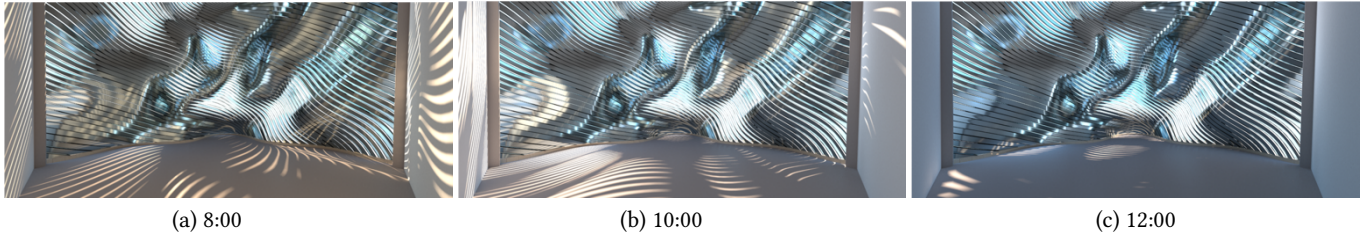


Fig. 38. This shading system has been designed so that it allows light to pass through during the morning, but blocks it at noon and in the afternoon.

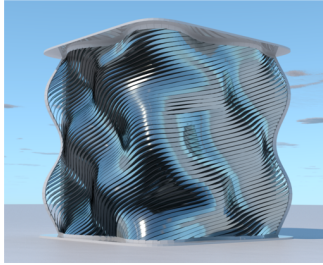


Fig. 39. Outside view onto the shading system of Fig. 38. Less extremely shaped systems may not only serve for shading a flat facade, but also as a design element. The date and location of this scene are August 15th in London.

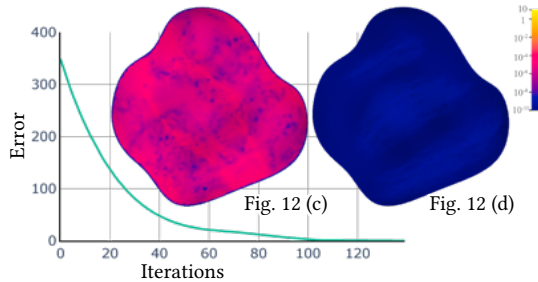


Fig. 40. The plot shows the optimization error  $E_{\text{angle}}$  vs. the number of iterations of Fig. 12. The color of the surfaces shows the distribution of the errors of Fig. 12 (c) and (d). After 140 iterations,  $E_{\text{angle}}$  has dropped from 348.87 to  $2.8 \times 10^{-6}$ . The total runtime is 6.48s for optimizing the P-curves.

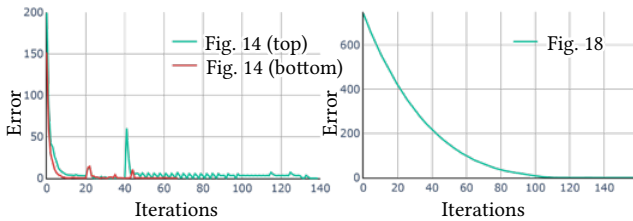


Fig. 41. Left: the plot of error  $E_{\text{angle}}$  vs. the number of iterations of the two interactive design examples in Fig. 14. The number of iterations, the last errors, and the runtime for the two examples are 140,  $1.82 \times 10^{-6}$ , 2.84s and 69,  $5.45 \times 10^{-7}$ , 1.49s, respectively. Right: The optimization of Fig. 18 takes 8.11s for 160 iterations;  $E_{\text{angle}} = 1.37 \times 10^{-5}$ .

with holes (as in Fig. 14). We did not go beyond that for a good reason: On a closed surface of any topology, there may be no patterns of the required type that extend smoothly over the entire model. That would require, e.g. closed P-curves. The existence of closed geodesics has been a topic in pure geometry, but the formation of patterns over the entire surface is certainly not possible on general

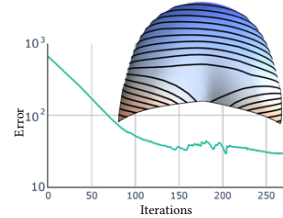


Fig. 42. A failure case where a singularity appears when optimizing for  $\theta = 60^\circ$ . The plot shows the error  $E_{\text{angle}}$  (in log) vs. the number of iterations. The error drops to around 30 after 270 iterations in 6.18s.

surfaces. We are not aware of any contribution on the existence and variety of closed P-curves on closed surfaces.

**Conclusion and Future Research.** Motivated by the recent emergence of a variety of gridshells and the aim to build complex shapes from simple elements, we presented tools for the computational design of structures that can be fabricated by bending flat straight lamellas. Various application scenarios, design options and fabrication constraints have been incorporated. We illustrated our methods at hand of computational design results and physical models.

One direction for future work are systems that block the direct sunlight, but provide indirect lighting through reflection. Another direction is to combine our tools for P-curve pattern design with the results in [Jiang et al. 2019] to generate origami models by folding flat sheets along appropriately computed curve patterns. An open mathematical problem is the explicit derivation of torsion-free rectifying strip structures with non-parallel node axes.

## ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their helpful comments, and the authors of [Laccone et al. 2023] for the nice testing surfaces. This work has been supported by KAUST baseline funding (grant BAS/1/1679-01-01) and Hong Kong, Research Grants Council: ECS 27604721.

## REFERENCES

- Phil Ayres, Soraya Bornaz, Adam Orlinski, Moritz Heimrath, and Alison Grace Martin. 2020. Architectural Scale Kagome Weaving: Design methods and fabrication concepts. In *FABRICATE: Design & Making*. UCL Press, 178–185.
- Phil Ayres, Alison Grace Martin, and Mateusz Zwierzycki. 2018. Beyond the basket case: A principled approach to the modelling of kagome weave patterns for the fabrication of interlaced lattice structures using straight strips. In *Adv. in Architectural Geometry*. 72–93.
- Changyeob Baek, Alison G Martin, Samuel Poincloux, Tian Chen, and Pedro M Reis. 2021. Smooth triaxial weaving with naturally curved ribbons. *Physical Review Letters* 127, 10 (2021), 104301.
- Wilhelm Blaschke and Gerrit Bol. 1938. *Geometrie der Gewebe*. Springer.
- Alexander I. Bobenko and Yuri B. Suris. 2008. *Discrete differential geometry: Integrable structure*. Graduate Studies in Mathematics, Vol. 98. American Mathematical Society, Providence, RI.

- Heinrich Brauner. 1968. Die Flächen mit Böschungslinien als Fallinien. *Monatsh. Math.* 72 (1968), 385–411.
- Keenan Crane, Clarisse Weischedel, and Max Wardetzky. 2017. The Heat Method for Distance Computation. *Commun. ACM* 60, 11 (2017), 90–99.
- Bailin Deng, Helmut Pottmann, and Johannes Wallner. 2011. Functional webs for freeform architecture. *Computer Graphics Forum* 30 (2011), 1369–1378.
- Manfredo do Carmo. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.
- Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire Mesh Design. *ACM Trans. Graph.* 33, 4, Article 66 (jul 2014), 12 pages.
- Heinrich Graf and Robert Sauer. 1924. Über dreifache Geradensysteme. *Sitz. Bayer. Akad. Math.-nat. Abt.* (1924), 119–156.
- Edmund Happold and Ian W. Liddell. 1975. Timber lattice roof for the Mannheim Bundesgartenschau. *Structural Engineering* 53, 3 (1975), 99–135.
- Jürgen Hemmick. 1974. IL10 Gitterschalen. Sonderforschungsbereich weitgespannte Flächentragwerke. Stuttgart, Institut für leichte Flächentragwerke.
- Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 39, 4, Article 117 (July 2020), 18 pages.
- Alec Jacobson and Daniele Panozzo. 2017. Libigl: Prototyping Geometry Processing Research in C++. In *SIGGRAPH Asia 2017 Courses*. ACM, 11:1–172.
- Caigui Jiang, Klara Mundilova, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2019. Curve-pleated structures. *ACM Trans. Graph.* 38, 6 (2019), 169:1–13.
- Francesco Laccone, Nico Pietroni, Paolo Cignoni, and Luigi Malomo. 2023. Bending-Optimized Double-Layer Grid Shells. Available at SSRN 4313774 (2023).
- Daoming Liu, Davide Pellis, Yu-Chou Chiang, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2023. Deployable Strip Structures. *ACM Trans. Graph.* 42, 4, Article 103 (jul 2023), 16 pages.
- Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. 2004. Methods for Non-Linear Least Squares Problems (2nd ed.).
- Andrés Martín-Pastor and Francisco González-Quintal. 2021. Surface discretization with rectifying strips on geodesics. *Nexus Network Journal* 23, 3 (2021), 565–582.
- Yannick Masson. 2017. *Existence and construction of Chebyshev nets and application to gridshells*. Ph. D. Dissertation. Paris Est.
- Karl Mayrhofer. 1931. Über Sechseckgewebe aus Geodätischen. *Monatshefte für Mathematik und Physik* 38 (1931), 401–404.
- Romain Mesnil and Olivier Baverel. 2023. Pseudo-geodesic gridshells. *Engineering Structures* 279 (2023), 115558.
- Nicolas Montagne, Cyril Douthe, Xavier Tellier, Corentin Fivet, and Olivier Baverel. 2020. Voss surfaces: A design space for geodesic gridshells. In *Proceedings of IASS Annual Symposia*, Vol. 2020. 1–11.
- Julius Natterer, Norbert Burger, Alan Müller, and Joh Natterer. 2000. Holzrippendächer in Brettstapelbauweise-Raumerlebnis durch filigrane Tragwerke. *Bautechnik* 77, 11 (2000), 783–792.
- Julian Panetta, Mina Konaković, Florin Ivoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Trans. Graph.* 38, 4 (2019), 83:1–15.
- Stefan Pillwein, Kurt Leimer, Michael Birsak, and Przemyslaw Musialski. 2020. On Elastic Geodesic Grids and Their Planar to Spatial Deployment. *ACM Trans. Graph.* 39, 4 (2020), 125:1–12.
- Stefan Pillwein and Przemyslaw Musialski. 2021. Generalized Deployable Elastic Geodesic Grids. *ACM Trans. Graph.* 40, 6 (2021), 271:1–15.
- Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner. 2015. Architectural Geometry. *Computers and Graphics* 47 (2015), 145–164.
- Helmut Pottmann, Qixing Huang, Bailin Deng, Alexander Schiftner, Martin Kilian, Leonidas Guibas, and Johannes Wallner. 2010. Geodesic Patterns. *ACM Trans. Graph.* 29, 4 (2010), 43:1–10.
- Helmut Pottmann and Johannes Wallner. 2001. *Computational Line Geometry*. Springer.
- Michael Rabinovich, Tim Hoffmann, and Olga Sorkine-Hornung. 2018. Discrete Geodesic Nets for Modeling Developable Surfaces. *ACM Trans. Graph.* 37, 2 (2018), 16:1–17.
- Yingying Ren, Julian Panetta, Tian Chen, Florin Ivoranu, Samuel Poincloux, Christopher Brandt, Alison Martin, and Mark Pauly. 2021. 3D weaving with curved ribbons. *ACM Trans. Graph.* 40, 4 (2021), 127:1–15.
- Andrew O. Sageman-Furnas, Albert Chern, Mirela Ben-Chen, and Amir Vaxman. 2019. Chebyshev Nets from Commuting PolyVector Fields. *ACM Trans. Graph.* 38, 6, Article 172 (nov 2019), 16 pages.
- Robert Sauer. 1926. Flächen mit drei ausgezeichneten Systemen geodätischer Linien, die sich zu einem Dreiecksnetz verknüpfen lassen. *Sitz. Bayer. Akad. Math.-nat. Abt.* (1926), 353–397.
- Robert Sauer. 1970. *Differenzgeometrie*. Springer.
- Jonas Schikore. 2023. *Compliant Grids. Theory, Design and Realization*. Ph. D. Dissertation. Technical University of Munich, Munich.
- Jonas Schikore, Eike Schling, Thomas Oberbichler, and Anna M Bauer. 2021. Kinetics and design of semi-compliant grid mechanisms. In *Adv. in Architectural Geometry*, 108–129.
- Eike Schling and Rainer Barthel. 2021. Šuchov’s bent networks: the impact of network curvature on Šuchov’s gridshell designs. In *Structures*, Vol. 29. Elsevier, 1496–1506.
- Eike Schling, Martin Kilian, Hui Wang, Denis Schikore, and Helmut Pottmann. 2018. Design and construction of curved support structures with repetitive parameters. In *Adv. in Architectural Geometry*, Lars Hesselgren et al. (Ed.). Klein Publ. Ltd, 140–165.
- Eike Schling and Zongshuai Wan. 2022. A geometry-based design approach and structural behaviour for an asymptotic curtain wall system. *Journal of Building Engineering* (2022), 104432.
- Eike Schling, Hui Wang, Sebastian Hoyer, and Helmut Pottmann. 2022. Designing asymptotic geodesic hybrid gridshells. *Computer-Aided Design* 152 (2022), 103378.
- Enrique Soriano. 2017. Low-tech geodesic gridshell: Almond pavilion. *ArchiDOCT* 4, 2 (2017), 29–38.
- Oded Stein, Eitan Grinspun, Max Wardetzky, and Alec Jacobson. 2018. Natural Boundary Conditions for Smoothing in Geometry Processing. *ACM Trans. Graph.* 37, 2 (2018), 23:1–13.
- Karl Strubecker. 1964. *Differentialgeometrie I*. De Gruyter.
- Chengcheng Tang, Xiang Sun, Alexandra Gomes, Johannes Wallner, and Helmut Pottmann. 2014. Form-finding with Polyhedral Meshes Made Simple. *ACM Trans. Graph.* 33, 4 (2014), 70:1–9.
- Gabriel Taubin. 1995. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 351–358.
- Josh Vekhter, Jiacheng Zhuo, Luisa F Gil Fandino, Qixing Huang, and Etienne Vouga. 2019. Weaving geodesic foliations. *ACM Trans. Graph.* 38, 4 (2019), 34:1–22.
- Otto Volk. 1929. Über Flächen aus geodätischen Dreiecksnetzen. *Sitzungsber. Heidelb. Akad. Wiss. Math.-Natur. Kl.* 1929, 1 (1929), 3–32.
- Aurel Voss. 1888. Über diejenigen Flächen, auf denen zwei Scharen geodätischer Linien ein conjugirtes System bilden. *Sitzungsber. Bayer. Akad. Wiss., math.-naturw. Klasse* (1888), 95–102.
- Hui Wang, Davide Pellis, Florian Rist, Helmut Pottmann, and Christian Müller. 2019. Discrete Geodesic Parallel Coordinates. *ACM Trans. Graph.* 38, 6 (2019), 173:1–13.
- Hui Wang and Helmut Pottmann. 2022. Characteristic parameterizations of surfaces with a constant ratio of principal curvatures. *Comp.-Aided Geometric Design* 93 (2022), 102074.
- Jun Wang, Caigui Jiang, Philippe Bompas, Johannes Wallner, and Helmut Pottmann. 2013. Discrete line congruences for shading and lighting. *Computer Graphics Forum* 32, 5 (2013), 53–62.
- Walter Wunderlich. 1950. Pseudogeodätische Linien auf Kegelflächen. Pseudogeodätische Linien auf Zylinderflächen. *Sitz. II, Ost. Ak. Wiss.* 158 (1950), 61–73, 75–105.
- Walter Wunderlich. 1951. Zur Differenzgeometrie der Flächen konstanter negativer Krümmung. *Österreich. Akad. Wiss. Math.-Nat. Kl. S.-B. Ila.* 160 (1951), 39–77.

## A APPENDIX

*Proof of Proposition 1.* One uses results from the differential geometry of ruled surfaces; see e.g. [Pottmann and Wallner 2001]. We assume  $c(u)$  to be an arc length parameterization ( $\|c'\|^2 = 1$ ). The parameter  $v_s$  for the striction point  $c + v_s b$  on a ruling is found as  $v_s = -(c' \cdot b') / \|c'\|^2$ , which equals 0 due to orthogonality of  $c' = t$  and  $b' = -\tau n_p$ . This shows that the parameter  $v$  is the distance to the striction point. The rest follows by Larmar’s formula for the Gaussian curvature,  $K = -\delta^2 / (\delta^2 + v^2)^2$ , where  $\delta$  denotes the distribution parameter, which is computed as

$$\delta = \frac{\det(c', b, b')}{\|b'\|^2} = \frac{\det(t, b, -\tau n_p)}{\tau^2} = \frac{1}{\tau}.$$

*Proof of Proposition 2.* We apply Euler’s formula for the normal curvature  $\kappa_n$  of a surface curve whose tangent forms the angle  $\phi$  with the first principal direction,  $\kappa_n = \kappa_1 \cos^2 \phi + \kappa_2 \sin^2 \phi$ . Each point of  $D$  is parabolic with principal curvature  $\kappa_1 = 0$  in the direction of the rulings  $d$ . The curve  $c \subset D$  has osculating planes which are orthogonal to the tangent planes of  $D$ , and thus the curvature  $\kappa$  of  $c$  is a normal curvature w.r.t.  $D$  in direction  $t$ . The angle  $\phi$  between  $t$  and  $d$  satisfies  $\sin^2 \phi = \kappa^2 / (\kappa^2 + \tau^2)$ , which yields by Euler’s formula,  $\kappa = \kappa_2 \sin^2 \phi$ , the second principal curvature  $\kappa_2 = \kappa(1 + k^2)$ , and analogously  $\kappa_n(b) = \kappa k^2$ .