PH-CPF: Planar Hexagonal Meshing using Coordinate Power Fields

KACPER PLUTA, Technion - Israel Institute of Technology, Israel
MICHAL EDELSTEIN, Technion - Israel Institute of Technology, Israel
AMIR VAXMAN, Utrecht University, The Netherlands
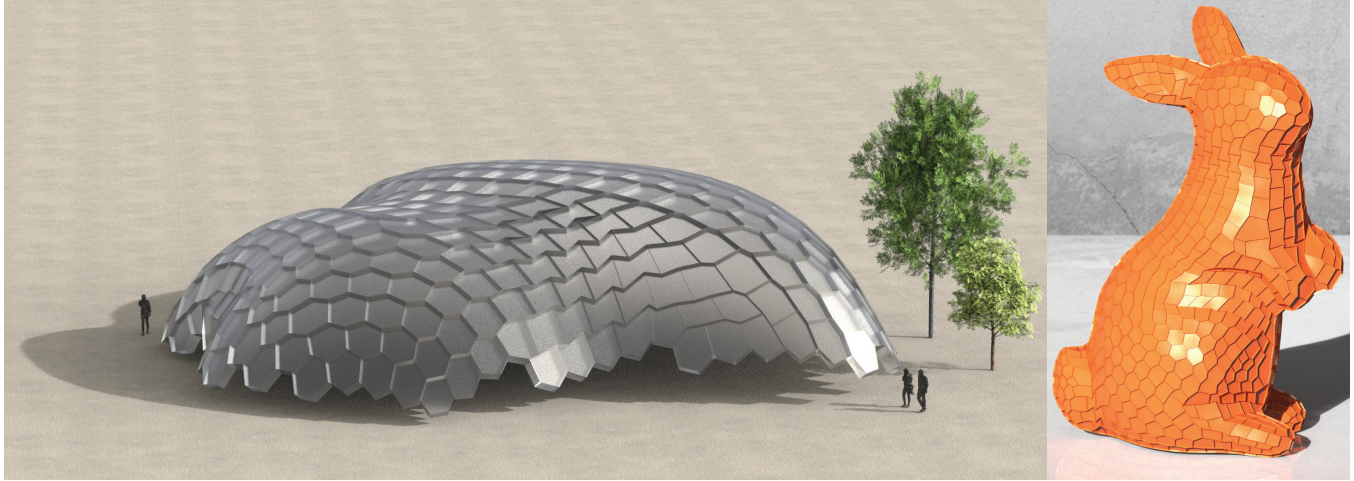MIRELA BEN-CHEN, Technion - Israel Institute of Technology, Israel

Fig. 1. Planar hexagonal meshes of computed with our approach.

We present a new approach for computing planar hexagonal meshes that approximate a given surface, represented as a triangle mesh. Our method is based on two novel technical contributions. First, we introduce *Coordinate Power Fields*, which are a pair of tangent vector fields on the surface that fulfill a certain *continuity* constraint. We prove that the fulfillment of this constraint guarantees the existence of a seamless parameterization with quantized rotational jumps, which we then use to regularly remesh the surface. We additionally propose an optimization framework for finding Coordinate Power Fields, which also fulfill additional constraints, such as alignment, sizing and bijectivity. Second, we build upon this framework to address a challenging meshing problem: planar hexagonal meshing. To this end, we suggest a combination of conjugacy, scaling and alignment constraints, which together lead to planarizable hexagons. We demonstrate our approach on a variety of surfaces, automatically generating planar hexagonal meshes on complicated meshes, which were not achievable with existing methods.

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

Additional Key Words and Phrases: geoemtry processing, planar hexagonal meshing, parameterization, tangent vector fields

Authors' addresses: Kacper Pluta, Michal Edelstein, Mirela Ben-Chen, The Henry and Marilyn Taub Faculty of Computer Science, Technion - Israel Institute of Technology, Haifa, Israel, 3200003; Amir Vaxman, Princetonplein 5, De Uithof, 3584 CC Utrecht, The Netherlands.

1 INTRODUCTION

Trianglular meshes are the most prevalent surface representation in geometry processing. However, for many applications, such as architectural design, animation and numerical simulation, a more structured representation is required, such as a quadrangular or hexagonal mesh. Specifically in architectural design, meshes with planar faces are amenable to realization using wood or glass panels, and are thus highly sought after [Li and Knippers 2015; Sonntag et al. 2019]. While planar quadrangular meshes have been intensively researched in recent years [Liu et al. 2006; Pottmann et al. 2015], planar hexagonal meshes remain an open challenge. Due to considerable technical difficulties, such as a strict restriction on allowable element orientations, and the necessary existence of non-convex faces, existing approaches are either limited to specific types of surfaces, or are not fully automatic, or both [Pottmann et al. 2015].

From the perspective of the geometry processing framework available for meshing, much progress has been made in recent years [Bommes et al. 2013a, 2009; Campen et al. 2015]. Still, the "standard" approach relies on computing vector fields on the surface, which are the *candidate gradients* of the parameterization functions [Bommes et al. 2013b]. However, with the exception of a few recent works, there is no guarantee that these candidates are in fact

integrable, i.e. that there exist parameterization functions whose gradients are these vector fields. Furthermore, while using the candidate gradients as the primary variables is convenient in some applications, in many cases, such as the planar hexagonal meshing application, and especially when the gradients are not expected to be orthogonal, this is not the most natural choice.

We propose a novel optimization framework for parameterization-based meshing, whose primary variables are a pair of tangent vector fields, which are candidate *Coordinate Power Fields* (CPF). CPFs are a generalization of *coordinate vector fields*, which are the pushforward of the 2*D* coordinate grid under a given local patch parameterization. Unlike coordinate vector fields, coordinate power fields allow for rotational jumps in the parameterization, and thus (up to global holonomy constraints) lead to a seamless global paramterization. Specifically, at the heart of our framework lies a new *continuity* constraint, which two tangent vector fields should fulfill in order to be CPFs, and which provably guarantees the local existence of a seamless parameterization with quantized rotational jumps.

We leverage our framework to compute planar hexagonal meshes. Specifically, we define additional constraints on the CPFs, using a novel *Dupin metric* approach, such that a pushed forward hexagonal grid would be planarizable. We demonstrate that this holds if the CPFs are *conjugate*, and are additionally aligned and sized according to the *Dupin Indicatrix* [Do Carmo 2016]. By combining these constraints with our optimization framework we automatically generate planar hexagonal meshes that approximate a wide variety of shapes, including complex shapes which, to the best of our knowledge, are not achievable by existing automatic methods. Furthermore, on meshes where existing approaches are applicable, we achieve a comparable or better planarity error as existing approaches, while using a considerably smaller number of elements. Finally, we demonstrate that our framework is additionally applicable to planar quad meshing.

1.1 Related Work

Global seamless quantized parameterization. Global parameterization, meshing and directional field design are vast topics in geometry processing, with ample amount of existing work. For a general overview of these topics we refer the reader to the excellent surveys on parameterization [Hormann et al. 2007; Sheffer et al. 2006], quad meshing [Bommes et al. 2013b] and directional field design [Vaxman et al. 2016].

Our meshing is parameterization-guided, namely, we compute an *integer grid-map* [Bommes et al. 2013a; Campen et al. 2015] and push forward a regular grid through this map to the surface to generate the new mesh elements. Within this framework, our main contributions is the computation of the gradients of a *seamless* global parameterization with quantized rotational continuity across the cuts. The remaining parts of the algorithm, namely integrating the parameterization, quantizing the translations, generating the mesh [Vaxman et al. 2019], and planarization [Tang et al. 2014] are based on existing work.

One challenge in this framework is achieving smooth gradient vector fields while allowing for discontinuities which lead to quantized singularities (or *cone points*). The most prevalent approach

to designing such fields is working with rotationally symmetric vector fields (RoSy-fields), which are computed by solving a mixed-integer [Bommes et al. 2009] or integer-valued [Farchi and Ben-Chen 2018] optimization problem, or with a reformulation to a complex-valued problem [Azencot et al. 2017; Knöppel et al. 2013; Viertel and Osting 2019] (which led to the notion of *power fields*). More versatility is achieved with complex polynomials (PolyVectors) [Diamanti et al. 2014], as they relax the requirement of rotational symmetry. Additionally, generalization beyond cross fields has been suggested by using a non-Euclidean metric, either extrinsically [Panozzo et al. 2014] or intrinsically [Jiang et al. 2015a].

These rotational fields are then integrated into an *integer grid-map* [Bommes et al. 2013a], where care is taken to guarantee that some translational components are also quantizable [Lyon et al. 2019]. However, with the exception of a few methods, most approaches that compute direction fields do not guarantee their *integrability*. Thus, the resulting parameterization gradients often deviate from the computed direction fields. Notable exceptions are PGP [Ray et al. 2006], that attempt to reduce the curl from the computed field before integrating, and the work of Myles et al. [Myles et al. 2014], that explicitly optimizes for a parameterization that agrees with an input cross field, yet it may introduce additional cone points. Furthermore, both approaches use a cross-field, which cannot, by definition, define the scale of a non-uniform parameterization, or the orientation of a non-conformal one (since cross-fields are unit-length and orthogonal). Harmonic cone-based seamless parameterization methods (e.g., HGP [Bright et al. 2017], and the orbifold methods [Aigerman and Lipman 2016]) guarantee local injectivity for given cone values, yet do not consider the field, hence they also cannot generate non-uniform parameterizations. The simplex assembly method [Fu and Liu 2016] uses affine transformations as the main variables and guarantees injectivity, yet it does not allow for quantized rotational discontinuities, nor does it consider the alignment. The only method that allows for non-orthogonal fields, considers alignment, and also optimizes integrability is the work by Diamanti et al. [2015] on integrable PolyVector fields. However, PolyVector fields allow for *any* set of consistently oriented vector fields, and thus form a much bigger space than the space of coordinate vector fields of parameterizations.

Our method, on the other hand, uses the vectors tangent to the meshing elements as the primary variables, thus considerably simplifying the formulation of constraints. In addition, we allow for non-orthogonal gradients, by posing the rotationally-symmetric constraint in the *parametric domain*. Thus, our variables are exactly as general as required to allow for any seamless parameterization with quantized rotations, without spurious degrees of freedom. The methods closest to our approach, are the recent algorithms for computing Chebyshev nets [Liu et al. 2020; Sageman-Furnas et al. 2019]. The second also used the tangent vector fields as primary variables and introduced an integrability constraint that is formulated using the *Lie Bracket operator*. Notably, this operator is *discretized*, and not *discrete*, hence it will not be exactly zero even if a seamless parameterization exists. We, on the other hand, present a novel *discrete* integrability constraint, which is provably exactly zero if and only if a seamless parameterization exists locally (without considering global homology obstructions). Moreover, our formulation

is a generalization of the classic integrability of curl-free piecewise constant vector fields [Polthier and Preuß 2003], which is obtained for the special case $N = 1$. The second method used both the tangent vector fields and the candidate gradients as variables, defining the smoothness on the first representation with PolyVectors, and the integrability on the second, and used alternating descent for the optimization. Our approach simplifies this formulation by removing the need for PolyVectors, while keeping the discrete integrability guarantee. Finally, both approaches were tailored for Chebyshev quad meshes, and a generalization to planar hexagonal meshes (or any other general formulation) was not presented.

We further elaborate on the relations between our formulation and the most similar approaches in section 3.4.

Planar hexagonal meshing. Planar hexagonal (PH) and planar quad (PQ) meshes are of paramount importance in architecture [Pottmann 2007; Pottmann et al. 2015]. PH meshes in particular are highly sought after for grid-shell constructions with wood panels [Li and Knippers 2015; Sonntag et al. 2019; Wagner et al. 2020]. This is due to their valence-3 vertices which lead to (conical) meshes with a natural face offset that can be realized with panels of constant thickness. They are further known to have beneficial mechanical properties [La Magna et al. 2012; Li 2017]. A few attempts have been made in the architecture literature to generate such meshes given an input smooth surface [Contestabile and Iuorio 2019; Henriksson and Hult 2015; Li 2017; Mesnil et al. 2017], yet the resulting algorithms often require manual work and are mostly limited to simple surfaces. It is worth noting that finding the PH mesh, or the flat segments that will form the panels, is only the first step in the realization of the surface. Further structural analysis is required to validate mechanical stability (see e.g. the work done on the Landesgartenschau exhibition hall [Li and Knippers 2015]), and thus a fully automatic approach that can generate a variety of meshes based on a small number of parameters (such as element size) is critical.

In the graphics and geometry processing literature there has also been a limited amount of work on this topic, unlike PQ meshes which were extensively researched [Liu et al. 2006, 2011; Zadravec et al. 2010]. PH meshes are in general more constrained than PQ meshes, as they, for example, require non-convex faces to create a watertight cover of a surface with negative Gaussian curvature [Pottmann et al. 2015]. While purely convex hexagonal faces have been used for architectural structures [De Leon 2012; Romero and Ramos 2013], the resulting facade is not watertight. One approach for hexagonal meshing is using the duality with triangular meshes, either by triangulating the Gauss map [Almegaard et al. 2007], or by tangent plane intersections [Troche 2008, 2009; Zimmer et al. 2012]. These approaches often do not lead to regular planar hexagonal shapes.

Better hexagonal shapes can be achieved if the primal triangular mesh is aligned with the principal directions, and sized according to the *Dupin indicatrix* [Li et al. 2014; Wang et al. 2008]. To achieve such an "ideal triangulation" the authors first construct an anisotropic quadrangular mesh, which is then decomposed into a triangular mesh, whose dual is the sought after planar hexagonal mesh. These multiple transitions require special care for singularities regions. Furthermore, to obtain aesthetic hexagon shapes in transition regions between positive and negative Gaussian curvature, these transition

regions must be specified manually. A related approach [Wang and Liu 2010] also starts from a quad mesh obtained from a conjugate curve network, and then shifts the quads to obtain a "brick-wall" pattern which is later optimized for planarization, which will again require special care for singularities. Our method, on the other hand, directly computes a triangulation that is aligned and sized according to the Dupin indicatrix, whose barycetric dual is a hexagonal mesh which is planarizable. Our approach does not require any special treatment of singularities, or manual adjustments in curvature transition regions, and generates planar hexagonal meshes on surfaces more complex than previously possible.

Other approaches aim at generating different variations of hexagonal meshes. For example, Nieser et al. [2011] generate curvature-aligned triangular meshes (without considering the planarity of the dual meshes), and we follow a similar approach for choosing alignment directions. Other variants include conformal planar hexagonal meshes [Müller 2011], which are only applicable to a restricted set of surfaces, honeycomb structures [Jiang et al. 2014] and caravel meshes [Tellier et al. 2020a,b], and polyhedral patterns [Jiang et al. 2015b]. The latter work focuses on the post-processing of a polygonal mesh (PH meshes as a special case), to generate an aesthetic planar polygonal pattern. We note that our approach generates the initial polygonal mesh which this approach receives as input. We use a similar method for post-processing, yet with some modifications as we do not have the regular stripe structure used there.

1.2 Contributions:

Our main contributions are:

- Coordinate Power Fields (CPFs) - a novel discrete formulation of integrability for seamless global parameterizations with quantized rotations.
- A corresponding general optimization framework that is applicable to different meshing applications.
- A new parameterization-based method for generating planar hexagonal meshes, using CPFs and a Dupin-indicatrix based metric, that exceeds the state of the art in both the quality of the resulting meshes, and the complexity of the surfaces that can be approximated.

2 BACKGROUND

Some of this material, e.g. on patch parameterization [Sageman-Furnas et al. 2019] and on the Dupin indicatrix [Wang and Liu 2009], has appeared previously. We repeat it here for completeness and for a quick reference using our notations, and limit ourselves to the concepts required for our exposition. For a full treatment of patch parameterization see the book by do Carmo [2016, Chap. 2].

2.1 Meshing by pushforward through a patch parameterization

2.1.1 Patch parameterization. Given an oriented embedded surface $\mathcal{M}$, let $r : \mathcal{U} \subset \mathbb{R}^2 \rightarrow \Omega \subset \mathcal{M}$ be a regular parameterization of a planar domain to a patch on the surface (see Figure 2) [1]. Let $\hat{u}, \hat{v}$ be the Cartesian unit orthogonal axes in the parameterization domain,

---

[1]Note that we use the classical plane-to-surface definition, rather than the common computer graphics one which is surface-to-plane.
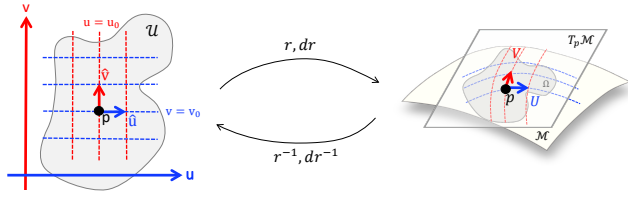
Fig. 2. Notations, patch parameterization and coordinate vector fields.

Fig. 3. Notations, triangular meshing.

and u, v be the coordinate functions on $\mathcal{U}$. Thus, r is defined on points $(u, v) \in \mathcal{U}$ and

$$p = r(u, v) \in \mathcal{M}, \quad r^{-1}(p) = (u, v) \in \mathcal{U}.$$

We define two scalar functions $u, v : \Omega \to \mathbb{R}$ as the pullback of the functions u, v : $\mathcal{U} \to \mathbb{R}$ through r^{-1}:

$$u(p) = u(r^{-1}(p)), \quad v(p) = v(r^{-1}(p)).$$

The gradients $\nabla u, \nabla v : \Omega \to T\mathcal{M}$ are tangent vector fields on Ω, whereas the gradients $\nabla u, \nabla v : \mathcal{U} \to \mathbb{R}^2$ are planar vector fields. Specifically, $\nabla u = \hat{u}, \nabla v = \hat{v}$.

The partial derivatives $U = \frac{\partial r}{\partial u}, V = \frac{\partial r}{\partial v}$ are the corresponding *coordinate vector fields* [Lee 2013, pp. 176] on Ω. Equivalently, we write $U = dr(\hat{u}), V = dr(\hat{v})$, where dr is the *differential* of r. The vector fields U, V are tangent to the isolines $v = v_0, u = u_0$, respectively.

2.1.2 Meshing by grid-pushforward. A common approach for regular meshing [Bommes et al. 2013a], is pushing forward a regular grid through a parameterization. We first consider a single patch and quadrangular remeshing. Then, the locations of the new mesh vertices are given by $r(t\mathbb{Z}^2 \cap \mathcal{U})$, where $t \in \mathbb{R}$ is a constant scalar coefficient that controls the size of the new mesh elements. Let $p = (u_0, v_0) \in t\mathbb{Z}^2 \cap \mathcal{U}$, and consider the quad $\{r(p), r(p + t\hat{u}), r(p + t\hat{v} + t\hat{u}), r(p + t\hat{v})\}$. In the limit of increasing refinement, we have, by the definition of the differential, $\lim_{t \to 0} \frac{r(p + t\hat{u}) - r(p)}{h} = dr_p(\hat{u}) = U(r(p))$, and similarly for V. Hence, the vectors $U(p), V(p), p = r(p)$ are closely related to the edges of the generated quad [2].

Consequentially, many constraints on the mesh elements, whether planarity, sizing, orientation or angles, are easily specified using constraints on the coordinate vector fields U, V. Furthermore, the pullback map dr^{-1}, as well as the gradient vector fields $\nabla u, \nabla v$ all have simple expressions in terms of U, V. Specifically, let B_p be a local orthonormal basis of $T_p\mathcal{M}$ at $p \in \Omega$, and $(U_p\ V_p)$ be the 2×2 matrix whose columns are the coefficients of $U(p), V(p)$ in the basis B_p. Then, it is straightforward to show (see supplemental material) that:

$$dr^{-1}(X) = (U_p\ V_p)^{-1}X, \quad \forall X \in T_p\mathcal{M}, \tag{1}$$

and

$$(\nabla u)_p = (1\ 0)(U_p\ V_p)^{-1}, \quad (\nabla v)_p = (0\ 1)(U_p\ V_p)^{-1}, \tag{2}$$
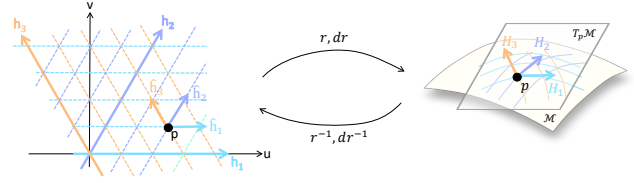
where all the coefficients of vectors in $T_p\mathcal{M}$ are with respect to the local basis B_p. A straightforward calculation then gives:

$$(\nabla u)_p = -\frac{1}{s_p}(JV_p)^T, \quad (\nabla v)_p = \frac{1}{s_p}(JU_p)^T, \tag{3}$$

where $s_p = \langle U_p, -JV_p \rangle$, and $J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. Finally, by combining the previous three equations, we have an explicit expression of dr^{-1}:

$$dr_p^{-1} = \frac{1}{s_p}(-V_p\ U_p)^T J^T. \tag{4}$$

We note that this discussion appears in [Sageman-Furnas et al. 2019], albeit using differential one-forms instead of gradient vector fields.

2.1.3 Triangular meshing. Our PH-meshing algorithm is based on tangent duality with a triangular mesh, which we compute using our parameterization framework. Here, the grid $\mathbb{Z}^2$ is replaced by the triangular grid $\mathbb{A}^2$ (see Figure 3), where $\hat{h}_1, \hat{h}_2, \hat{h}_3$, are the triangular unit regular coordinate axes in the parameterization domain $\mathcal{U}$.

There is a linear relationship between the Cartesian and triangular grids, thus we can alternate between the two representations using matrix multiplication. Specifically, we have $(\hat{h}_1\ \hat{h}_2\ \hat{h}_3)_{2\times3} = (\hat{u}\ \hat{v})_{2\times2} A$, where $A = \begin{pmatrix} 1 & 1/2 & -1/2 \\ 0 & \sqrt{3}/2 & \sqrt{3}/2 \end{pmatrix}_{2\times3}$. We additionally define, $h_1, h_2, h_3 : \mathcal{U} \to \mathbb{R}$ as the triangular coordinate functions, and we have that $(h_1\ h_2\ h_3) = (u\ v) A$. Finally, H_i are given by the pushforward of $\hat{h}_i$ through r, so $H_i = dr(\hat{h}_i)$. Since U, V are the coordinate vector fields of r, and since dr is linear, we have that $(H_1\ H_2\ H_3)_{2\times3} = (U\ V)_{2\times2} A$, where H_i, U, V and A are represented in an arbitrary orthonormal local basis at each $p \in \Omega$.

For a point $p \in t\mathbb{A}^2$, the pushforward of one of its neighboring triangles has vertices at $\{r(p), r(p + t\hat{h}_1), r(p + t\hat{h}_2)\}$, and similarly for the other 5 neighboring triangles. As in the quadrangular case, in the refinement limit we have $\lim_{t \to 0} \frac{r(p + t\hat{h}_1) - r(p)}{t} = dr(\hat{h}_1) = H_1$, and similarly for H_2 (and the other 5 triangles, by symmetry). Hence, the edges of the pushed forward triangles are closely related to the vector fields H_1, H_2, H_3. The dual hexagonal edges are also closely related to H_i and U, V, as we discuss in the next sections.

2.2 Discrete surfaces

We represent the input surfaces using triangle meshes $M = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where $m = |\mathcal{F}|, n = |\mathcal{V}|$. Vector fields are piecewise constant per face, either in the trivial basis in $\mathbb{R}^3$ as $V \in \mathbb{R}^{3m}$, or using a local basis per face as $V \in \mathbb{R}^{2m}$. The orthonormal (arbitrary) local basis per face, which we use to represent vector fields, the shape operator and the pullback maps, is given by a matrix $B \in \mathbb{R}^{3m \times 2m}$.

---

[2]Note that we abuse notations slightly and treat $r(p) \in \mathcal{M}$ as its corresponding embedded point in $\mathbb{R}^3$ here.

3 COORDINATE POWER FIELDS

3.1 Rotational Continuity across patches

Consider two parameterizations $r_i : \mathcal{U}_i \to \Omega_i \subset \mathcal{M}$, $i \in [1, 2]$. In general, if r_i are part of a valid atlas of $\mathcal{M}$, then for any $p \in \Omega_1 \cap \Omega_2$ and any $X \in T_p \mathcal{M}$, the pullback maps agree. Namely: $r_1^{-1}(p) = r_2^{-1}(p) \in \mathbb{R}^2$ and $dr_1^{-1}(X) = dr_2^{-1}(X) \in \mathbb{R}^2$.

In the context of meshing, however, we are only interested in the *pushforward of a grid* through these parameterizations. Hence, the pullback maps are part of a valid *grid atlas*, if they agree up to a *grid automorphism* (see e.g. [Nieser et al. 2011]). These consist of quantized rotations of integer multiples of $2\pi/N$ (where $N = 4, 6$ for quads and triangles respectively), and quantized translations. In the vector field optimization step we consider only quantized rotations, as these are achievable without introducing integer variables.

Definition 3.1. Two parameterizations r_1, r_2 are part of a valid *rotational grid atlas of degree N* iff their corresponding pullback maps dr_1^{-1}, dr_2^{-1} agree up to a rotation by an integer multiple of $2\pi/N$. Specifically, for any $p \in \mathcal{M}$, $X \in T_p \mathcal{M}$ there exists an integer $\mathsf{k} \in [0, .., N - 1]$, such that:

$$dr_1^{-1}(X) = R^{\frac{2\pi}{N}\mathsf{k}} dr_2^{-1}(X), \tag{5}$$

where R^θ is the counter-clockwise rotation by the angle θ.

By identifying points $(x, y) \in \mathbb{R}^2$ with points $x + iy \in \mathbb{C}$, we have (following Knöppel et al. [2013]):

$$\left(dr_1^{-1}(X)\right)^N = \left(dr_2^{-1}(X)\right)^N. \tag{6}$$

Using Equations (1) and (2), we have that:

$$\left((D_X u_1)_p, (D_X v_1)_p)\right)^N = \left((D_X u_2)_p, (D_X v_2)_p)\right)^N, \tag{7}$$

$$u_i(p) = \mathsf{u}(r_i^{-1}(p)), \quad v_i(p) = \mathsf{v}(r_i^{-1}(p)), \quad i = [1, 2], \tag{8}$$

where $(D_X f)_p = \langle (\nabla f)_p, X \rangle_p$, and $\langle , \rangle_p$ is given by the metric at p.

3.2 Smooth CPFs

The partial derivatives of a regular parameterization $r : \mathcal{U} \to \Omega$ are its coordinate vector fields. Analogously, given a collection of parameterizations, we define *Coordinate Power Fields*.

Definition 3.2. Let $r_i : \mathcal{U}_i \to \Omega_i \subset \mathcal{M}$, $i \in [1, .., m]$ be regular parameterizations such that $\bigcup_{i=1}^m \Omega_i = \mathcal{M}$, and let U_i, V_i be their partial derivatives. The set of pairs of vector fields $\{(U_i, V_i) \mid i \in [1, .., m]\}$ is a set of *Coordinate Power Fields of degree N* iff every pair of parameterizations is part of a rotational grid atlas of degree N.

3.3 Discrete Rotational Continuity and CPFs

Given an oriented manifold triangle mesh M embedded in $\mathbb{R}^3$, let $U, V \in \mathbb{R}^{2m}$ be a pair of piecewise constant (PC) vector fields represented in a local basis.

Definition 3.3. Two PC vector fields U, V are *Linearly Independent and Consistently Oriented* (LICO) iff $\langle U_i, -JV_i \rangle > 0$, $\forall t_i \in \mathcal{F}$.

Following previous work [Bommes et al. 2013a; Kälberer et al. 2007], we set each triangle to be the image of a different parameterization from some triangle domain in $\mathbb{R}^2$. Given two LICO vector fields, the parameterization to the face $t_i \in \mathcal{F}$ is induced by the vectors U_i, V_i. Specifically, let $(p_i^1, p_i^2, p_i^3) \in \mathbb{R}^{3 \times 3}$ be the embedding of the triangle $t_i \in \mathcal{F}$, and consider the triangle

$$\mathsf{t}_i = \left((0; 0), (\ U_i\ V_i\ )^{-1}(p_i^2 - p_i^1), (\ U_i\ V_i\ )^{-1}(p_i^3 - p_i^1)\right) \in \mathbb{R}^{2 \times 3}, \tag{9}$$

where $p_i^2 - p_i^1$ and $p_i^3 - p_i^1$ are expressed in the local basis of t_i. We define $r_i : \mathsf{t}_i \to t_i$ as the unique linear map between the two triangles. Clearly, $dr_i = (\ U_i\ V_i\ )$, and $dr_i^{-1} = (\ U_i\ V_i\ )^{-1}$. In the discrete case, the pairwise intersections of the images of the parameterizations are the *mesh edges*, leading to the following definition.

Definition 3.4. A pair of LICO vector fields $U, V \in \mathbb{R}^{2m}$ are *discrete coordinate power fields of degree N* iff

$$\left(dr_i^{-1}(e_{ij})\right)^N = \left(dr_j^{-1}(e_{ij})\right)^N, \quad \forall e_{ij} \in \mathcal{E}^I, \tag{10}$$

where $\mathcal{E}^I$ are the interior edges of M, and $e_{ij} = t_i \cap t_j$, $t_i, t_j \in \mathcal{F}$, with a fixed orientation for both triangles.

Discrete CPFs guarantee the existence of a corresponding seamless, rotationally quantized, global parameterization, as follows.

THEOREM 3.5. *Let U, V be discrete CPFs of degree N. Then there exist functions $u^l, v^l \in \mathbb{R}^m$, $l = [1, 2, 3]$, with u_i, v_i piecewise linear per face $t_i \in \mathcal{F}$ (yet discontinuous between faces), such that:*

(1) $\nabla(u_i) = (\ 1\ 0\ )(\ U_i\ V_i\ )^{-1}$, $\nabla(v_i) = (\ 0\ 1\ )(\ U_i\ V_i\ )^{-1}$, $\forall t_i \in \mathcal{F}$.
(2) *The triangle* $\mathsf{t}_i = (\mathsf{p}_i^1, \mathsf{p}_i^2, \mathsf{p}_i^3) \in \mathbb{R}^{2 \times 3}$ *with coordinates* $\mathsf{p}_i^l = (u_i^l, v_i^l)$ *is positively oriented.*
(3) *Let* $e_{ij} \in \mathcal{E}^I$, *with* $e_{ij} = t_i \cap t_j$, *and set* $\alpha_i, \beta_i \in [1, .., 3]$ *the indices of the vertices of* e_{ij} *in* t_i, *and similarly for* t_j. *Thus,* $e_{ij} = p_i^{\alpha_i} - p_i^{\beta_i} = p_j^{\alpha_j} - p_j^{\beta_j} \in \mathbb{R}^3$. *Then there exists* $\mathsf{k}_{ij} \in \mathbb{Z}$ *such that* $R^{2\pi\mathsf{k}_{ij}/N}(\mathsf{p}_i^{\alpha_i} - \mathsf{p}_i^{\beta_i}) = \mathsf{p}_j^{\alpha_j} - \mathsf{p}_j^{\beta_j}$.

The proof is a straightforward application of the CPF condition (10), and is provided in the supplemental material.

Note 1. For the special case $N = 1$ and a simply connected mesh, Theorem 3.5 leads to a global parameterization $u, v \in \mathbb{R}^{|\mathcal{V}|}$. Its existence is guaranteed since Equation (10) implies that the classic discrete *edge-based* curl operator [Polthier and Preuß 2003, Def. 3] is 0 for the vector fields given by the rows of dr_i^{-1}.

Note 2. Our definition is related to Integer Grid Maps [Bommes et al. 2013a], but not exactly the same. Specifically, we do not enforce integer transitions between charts, nor integer singularity locations. We further discuss this in Section B.

3.4 Relation to other approaches

The existence conditions of an Integer Grid Map, i.e. a parameterization s.t. the pushforward of an integer grid leads to a valid mesh, have been given previously [Bommes et al. 2013a; Kälberer et al. 2007]. The full problem leads to a difficult mixed integer optimization problem, which, to the best of our knowledge, is not solved directly in any approach. Methods vary in the *decomposition* of the problem, and in the method that is used to solve each part.

QuadCover, MIQ, Globally Optimal (GO), and IGM. The most common decomposition is to first compute a *cross-field* either by smoothing the principal curvature directions (QuadCover [Kälberer et al. 2007]), by solving a mixed integer problem (MIQ [Bommes et al.

2009]), or by solving a convex complex-valued problem (GO [Knöppel et al. 2013]). In the second step, a parameterization is computed which aligns with the computed cross field and adheres to the integer transition constraints, either by a mixed integer formulation (MIQ [Bommes et al. 2009]), or using a branch-and-cut algorithm (IGM [Bommes et al. 2013a]).

This decomposition has several disadvantages. First, the sizing of the elements is only addressed in the second step, and thus cannot influence the topology (the singularity locations) of the resulting mesh. Furthermore, integrability is only enforced in the second step (by explicitly computing the parameterization functions), and thus the resulting parameterization can be arbitrarily far from the input cross-field. Finally, using a cross field implies that the resulting gradient directions are orthogonal, which is not always preferable.

Our approach does a *different decomposition*. The only part which is not addressed in our optimization are the integer transitions between charts. All the other components: integrability, sizing and alignment are all addressed in the vector field optimization stage. This allows, for example, the singularities to move to modify the sizing of the elements and not only the directions.

FrameFields, Metric Customization, Simplex Assembly and PolyVectors. A different first step is to allow for non-orthogonal non-unit length fields, and then proceed with the second parameterization step as described above. This is done by either computing a new 3D geometry that is prescribed by a given metric (FrameFields [Panozzo et al. 2014]), by enforcing a different intrinsic metric using a different connection on the input mesh (Metric Customization [Jiang et al. 2015a]), or by generalizing the complex-valued formulation to a complex-polynomial one (PolyVectors [Diamanti et al. 2014]). None of these approaches guarantees integrability, and thus when computing the parameterization in the second step, the gradients of the parameterization may differ greatly from the computed fields. The FrameFields and Metric Customization approaches optimize for a 2x2 matrix per face, as do we. However, they consider only the SPD component (without the planar rotation), whereas we compute the full linear map (without the translation). Some notion of integrability of the computed metric exists for both approaches, the first since they construct a new mesh embedding, and the second since the metric of two neighboring triangles should agree on the length of the edge. Note, however, that this constraint is weaker than ours, as we enforce a seamless quantized rotation. The Simplex Assembly [Fu and Liu 2016] method computes the full affine map per triangle (including the translation part), yet they do not take into account field alignment, or allow for quantized rotational jumps.

Integrable PolyVectors, Commuting PolyVectors and Discrete Integrable Frame Fields. The only three approaches that allow for non-orthogonal fields and optimize integrability are Integrable PolyVectors (IPV) [Diamanti et al. 2015], Commuting PolyVectors (CPV) [Sageman-Furnas et al. 2019] and Discrete Integrable Frame Fields (DIFF) [Liu et al. 2020]. Integrability is defined for IPVs through the *PolyCurl*, which is 0 if there exists a matching between neighboring faces which leads to a zero edge-based curl for all the vectors of the PolyVector. However, IPVs describe a larger space than CPFs, and it is not clear how to restrict them to represent only vector fields that are the coordinate vector fields for some parameterization. In

CPVs, integrability is defined through the *discretization* of the Lie bracket, which in general does not guarantee discrete integrability. Finally, in DIFFs integrability is defined on the pulled-back edges yet with a known matching, unlike our approach where we allow for rotational jumps in the pulled back edges.

4 CPFS OPTIMIZATION FRAMEWORK

Given an input mesh M, we setup an optimization framework for computing CPFs U, V. The CPFs are then used to compute a parameterization and a regular mesh, as explained in Section 4.6.

4.1 Variables

Our main variables are $U, V \in \mathbb{R}^{2m}$, the candidate CPFs, represented using a local basis B.

4.2 Hard Constraints

We have two hard constraints. The LICO constraint:

$$\langle U_i, -JV_i \rangle > 0, \quad \forall t_i \in \mathcal{F}, \tag{C1}$$

and the CPF constraint:

$$\left(dr_i^{-1}(e_{ij}) \right)^N = \left(dr_j^{-1}(e_{ij}) \right)^N, \quad \forall e_{ij} \in \mathcal{E}^I. \tag{C2}$$

4.3 Soft Constraints

Conditions on alignment, sizing and orthogonality are required in general for parameterization.

4.3.1 Alignment. Let $d_i \neq 0$ be a direction in the face $t_i \in \mathcal{F}$, and n_i be an even divisor of N. Our alignment constraint is:

$$\text{Im} \left(\left(dr_i^{-1}(d_i) \right)^{n_i/2} \right) = 0. \tag{C3}$$

LEMMA 4.1. *If U, V are LICO and the constraint (C3) is fulfilled, then one of the pushed forward grid directions in the face t_i will be parallel to d_i.*

The proof is provided in Appendix A.

4.3.2 Sizing. The element sizing is determined by the scale of the CPFs U, V. We add a constraint on the size of the elements under a user-defined pointwise metric g. The metric is the identity for uniform scale, a diagonal matrix for isotropic scale, or any symmetric positive definite matrix for anisotropic scale. We have:

$$\|U_i\|_{g_i}^2 = 1, \quad \|V_i\|_{g_i}^2 = 1, \quad \forall t_i \in \mathcal{F}, \tag{C4}$$

where $g_i \in \mathbb{R}^{2 \times 2}$ is a positive definite matrix defined in the local basis of $t_i \in \mathcal{F}$, and $\|x\|_g^2 = x^T g x$.

4.3.3 Orthogonality. The elements' angles are determined by the pointwise inner product between U, V. We add an orthogonality constraint under a symmetric inner product s chosen by the user. The inner product can be the identity for orthogonality, or the shape operator for conjugacy, for example. We have:

$$\langle U_i, V_i \rangle_{s_i} = 0, \tag{C5}$$

where $s_i \in \mathbb{R}^{2 \times 2}$ is a symmetric matrix, defined using the local basis of the face $t_i \in \mathcal{F}$, and $\langle x, y \rangle_g = x^T g y$.
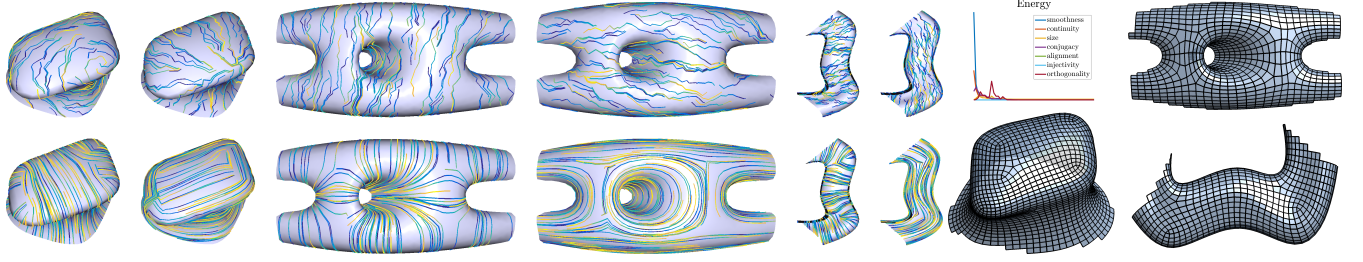
Fig. 4. (left) Examples of convergence from a random initialization, with the initial (top) and final (bottom) U, Vs shown. (right) the resulting quad meshes, and the energy graph for the train station model.

4.4 Relaxed Problem and Optimization Approach

The constraints we require lead to a non-linear non-convex optimization problem. As a design choice, we opted for posing the problem as an unconstrained non-linear least squares problem, and solving it with the Levenberg-Marquardt [Gavin 2020] algorithm.

For the LICO constraint (C1), we follow previous work on preserving local injectivity [2013], and formulate it using a spline barrier function. This approach requires a valid initial guess, and therefore is not applicable to the CPF constraint (C2). Hence, we opt for a penalty approach [Leyffer and Mahajan 2010, Sec 4.2]. Specifically, we reformulate the constraint as a quadratic objective, and use an increasing sequence of penalty parameters. For simplicity, we introduce auxiliary variables $z \in \mathbb{C}^{|\mathcal{E}^I|}$, which represent the "consensus" pulled back edge through the local transformations at its neighboring triangles. The soft constraints (C3), (C4), (C5) are reformulated as (potentially weighted) quadratic objectives.

Our relaxed optimization problem is given by:

$$\underset{U,V \in \mathbb{R}^{2m}, z \in \mathbb{C}^{|\mathcal{E}^I|}}{\text{minimize}} \quad E^s(U,V,z) + \beta E^{\Psi_c}(U,V,z) + E^{\Psi}(U,V) \tag{11}$$

where E^s is a smoothness objective, E^{Ψ_c} is the penalty objective for the CPF constraint (C2), and β is the penalty parameter. Finally, E^{Ψ} is the objective for the rest of the constraints:

$$E^{\Psi}(U,V) = E^{\Psi_{lico}}(U,V) + E^{\Psi_l}(U,V) + E^{\Psi_s}(U) + E^{\Psi_s}(V) + E^{\Psi_o}(U,V), \tag{12}$$

where $E^{\Psi_{lico}}, E^{\Psi_l}, E^{\Psi_s}, E^{\Psi_o}$ are penalty objectives for the LICO (C1), alignment (C3), sizing (C4) and orthogonality (C5) constraints.

The objectives are defined either per interior edge (smoothness and CPF penalty), or per face (all the others). For each such objective we specify the formulation for a single element. The total energy is computed by averaging all the elements' objectives values. All the gradients are provided in the supplemental material.

4.4.1 LICO penalty.
We use the barrier function designed for Locally Injective Mappings (LIM) [Schüller et al. 2013], where the barrier term is applied to s_i:

$$E_i^{\Psi_{lico}}(U,V) = \phi_i^2\left(s_i(U,V) - \epsilon^{\Psi_{lico}}\right), \quad s_i(U,V) = \langle U_i, -JV_i \rangle, \tag{13}$$

where ϕ_i is [Schüller et al. 2013, Eq.(5),(6)]:

$$\phi_i(x) = \begin{cases} \infty, & x \le 0 \\ \frac{1}{g_i(x)} - 1, & 0 < x < \bar{s}_i, \\ 0, & x \ge \bar{s}_i \end{cases} \quad g_i(x) = \frac{1}{\bar{s}_i^3}x^3 - \frac{3}{\bar{s}_i^2}x^2 + \frac{3}{\bar{s}_i}x. \tag{14}$$

Here $\bar{s}_i$ is the value of s_i in the previous iteration times a constant $s = 0.1$, and we take $\epsilon^{\Psi_{lico}} = 10^{-3}$. We take the barrier with respect to the previous iteration instead of a fixed constant to be more resilient to different scalings.

4.4.2 CPF penalty.
The z variables conciliate the pullback of t_i and the pullback of t_j:

$$E_{ij}^{\Psi_c}(U,V,z) = \left|\left(dr_i^{-1}(e_{ij})\right)^N - z_{ij}\right|^2 + \left|\left(dr_j^{-1}(e_{ij})\right)^N - z_{ij}\right|^2, \tag{15}$$

where we measure absolute difference between complex numbers.

4.4.3 Soft constraints.
Let $\Psi(U,V) = 0$ be one of the constraints (C3), (C4) or (C5). Then the corresponding objective E^{Ψ} is given by $|\Psi(U,V)|^2$.

4.4.4 Smoothness.
The regularizer that is often used in N-RoSy field generation is the vector Laplacian, modified to accommodate the rotational symmetry [Knöppel et al. 2013]. This formulation favors vector fields which are similar in neighboring faces when raising them to the N-th power in the appropriate basis. The integrability constraint requires that the N-th power of the pulled back edge agrees across neighboring triangles, yet the orthogonal direction is not constrained. Hence, for the regularizer we use:

$$E_{ij}^S(U,V,z) = \left|\left(dr_i^{-1}(Je_{ij})\right)^N - z_{ij}\right|^2 + \left|\left(dr_j^{-1}(Je_{ij})\right)^N - z_{ij}\right|^2. \tag{16}$$

Note that, similarly to the CPF constraint, for $N = 1$ Equation (16) is equivalent to requiring that the discrete *edge-based* divergence [Polthier and Preuß 2003, Def. 4] is 0 for the vector fields given by the rows of dr_i^{-1}.

4.5 Implementation Details

Initialization. The only constraint on our initialization is that the initial U, V are LICO. The initial z is always computed as the average of the N-th power of the pulled-back edges, namely: $z_{ij} = \left(dr_i^{-1}(e_{ij})\right)^N + dr_j^{-1}(e_{ij}))^N\right)/2$.

The best initialization is highly dependent on the application. For simple surfaces, we can obtain a good solution from random (valid) initialization. For example, taking U to be a random unit length vector field, $V_i = JU_i$, and rescaling both according to g_i if
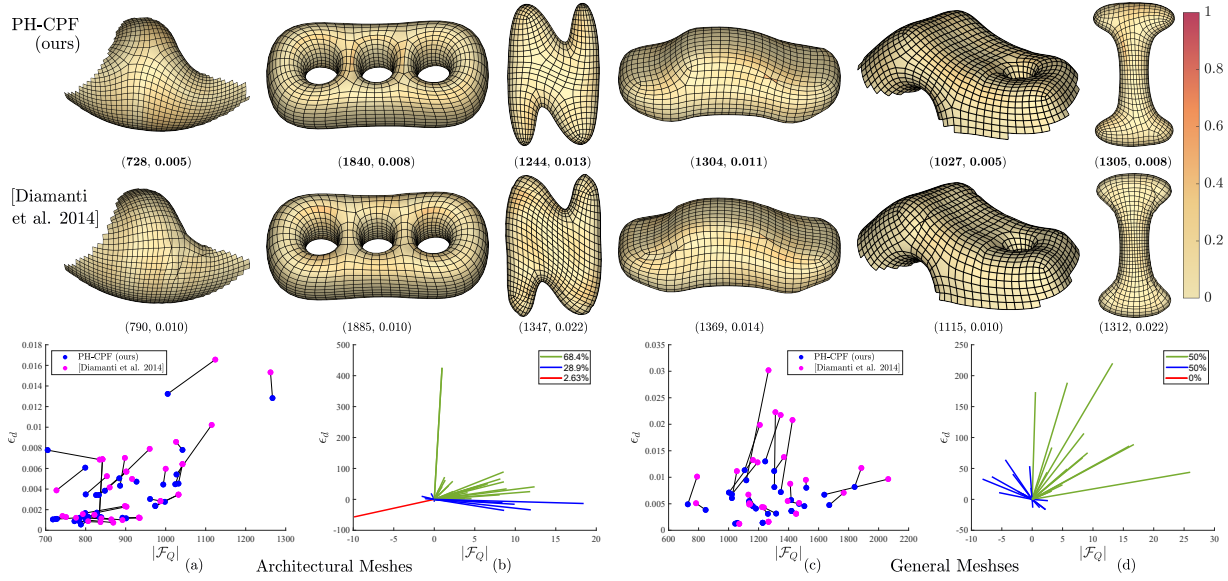
Fig. 5. Computing planar quad meshes using our method and PolyVectors [Diamanti et al. 2014]. (top) Some examples, with the number of faces $|\mathcal{F}_Q|$ and the Hausdorff distance ϵ_d to the input mesh in parentheses. (bottom) (a,c) Scatter graph of $|\mathcal{F}_Q|$ vs ϵ_d, (b,d) tradeoff graph, see the text for details. Green lines indicate our method achieves a lower distance with fewer faces.

a sizing constraint given. Figure 4 (left) shows examples of optimized CPFs U, V (bottom) computed from a random initialization (top), with curvature based sizing (Section 5.6) and the conjugacy and orthogonality constraints. Figure 4 (right) shows the resulting quad meshes and the convergence graph of the energy for the train station model (the other graphs are similar). Note that for quad meshes, conjugacy and orthogonality combined imply alignment to curvature directions, although we did not enforce this explicitly.

Optimization. We use the Levenberg-Marquardt algorithm [Gavin 2020] with Jacobian scaling for optimization, as implemented in Matlab, with $\beta_0 = 1$. If required, we repeat the optimization, using the optimized variables as the initial solution, and increasing β by a multiplicative factor of 2, until mesh integration with integer seams succeeds, or a maximum of 10 iterations is reached. In most of the results we show, the first iteration sufficed.

Balancing the energy terms. Different energy terms in Equations (11), (12) have different units, which are mesh-dependent. Without balancing, different weighting parameters are required for different meshes. Specifically, E^s, E^{Ψ_c}, E^{Ψ_I} are defined on the N-th power of a pulled back vector, whereas the rest of the objectives are defined on the input mesh. Hence, we rescale these energies, taking $1/\lambda^{2N}E$, where λ is the average expected 2D edge length, given by $\lambda = \frac{1}{3m}\sum_{i=1}^{m}\sum_{e_{ij}\in t_i}\|e_{ij}^T g_i e_{ij}\|$.

4.6 Meshing with CPFs

Given the computed U, V vector fields, the seamless parameterization can be theoretically constructed using the definitions of the parameterization functions u, v that are given in the proof. However, even if the CPF constraint is accurate to machine precision,

such an implementation would require computing integer translations per edge, which is unnecessarily expensive. Thus, we follow the standard procedure for extracting a mesh [Vaxman et al. 2019]. Specifically, integrating the parameterization with the prescribed quantized rotations on the edges using a Poisson solve, computing a full integer grid map by quantizing the translations [Vaxman 2021], and extracting the mesh by overlaying a grid on the integer grid map. We provide the details in Section B.

4.7 Example Application: Planar quad meshing

Our main interest in this paper is planar hexagonal meshing, which is explained in detail in the next section. As we introduce a general scheme for parameterization-based meshing, we provide also an example for planar quad meshing, and compare with existing work. We set $N = 4$, set the sizing to be curvature dependent (Section 5.6), set alignment with $N = 4$ to the prominent curvature directions (Section 5.5.2), and set the conjugacy and orthogonality constraints. For initialization we use the smoothed curvature directions.

For the comparison, we use the PolyVectors method [Diamanti et al. 2014] using the publicly available implementation from the Directional library [Vaxman et al. 2019]. We note that an implementation of Integrable PolyVectors [Diamanti et al. 2015] that includes conjugacy optimization of the vector fields (which is necessary for planar quad meshing) is not publicly available. For the competing code we we used the same shape operator as ours and applied the same planarization post processing as for our approach, setting the bound on the same maximal planarity of 1%. We compare the Hausdorff distance to the source mesh ϵ_d and the number of faces $|\mathcal{F}_Q|$ produced by both methods for a set of architectural meshes and a set of general meshes. Figure 5 (top) shows a few qualitative examples
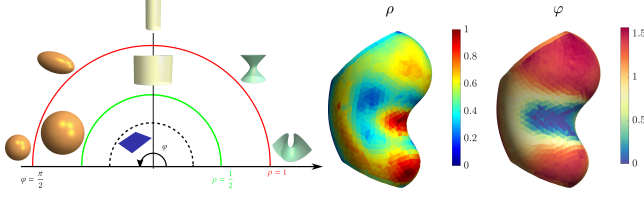
Fig. 6. Curvature regions and their polar representation with φ and ρ.

from the sets, as well as $(|\mathcal{F}_Q|, \epsilon_d)$. Figure 5 (bottom, (a,c)) shows a scatter plot of ϵ_d vs $|\mathcal{F}_Q|$ for both methods, where connected points represent the same mesh. The graphs (b,d), show the difference between our results, represented as the origin, to Diamanti's results, in percentages. Green lines, in the first quadrant, represent meshes for which our method produces a lower ϵ_d, with less elements. Blue lines, in the second and third quadrant, represent the trade-off between the methods, one achieved a better distance, while the other, less elements. The fourth quadrant, with a single red line, shows a case where Diamanti's method produced better results in both criteria. Note that most meshes fall in the first quadrant, indicating that for the same planarity values, our method achieved a lower Hausdorff distance with a smaller number of elements.

5 PLANAR HEX MESHING

Our main goal is generating planar hexagonal meshes which approximate a given surface, and we use the CPF framework for the optimization. In the following we describe the constraints that we apply, which are unique for hexagonal meshes.

5.1 Background: Curvature and the Dupin Indicatrix

Notations and polar curvature representation. The maximal and minimal curvatures at $p \in \mathcal{M}$ are denoted by $\kappa_1(p)$, $\kappa_2(p) \in \mathbb{R}$, respectively, with corresponding maximal and minimal curvature directions denoted by $d_1(p), d_2(p) \in T_p\mathcal{M}$. The Gaussian and mean curvatures are denoted by $K(p), H(p)$, respectively, and the shape operator by S_p. The normal curvature in the direction $v \in T_p\mathcal{M}$ is denoted by $\kappa_n(v)$. Two directions $x, y \in T_p\mathcal{M}$ are conjugate if they are orthogonal with respect to the shape operator, namely $\langle x, y \rangle_{S_p} = x^T S_p y = 0$. We drop p when the meaning is clear.

Curvature on triangle meshes is notoriously difficult to compute robustly, requiring special treatment in the context of curvature aligned meshing [Campen et al. 2016]. Thus, similarly to Niesser et al. [2011], we use a polar representation of the curvatures:

$$\varphi = \left| \text{atan}\left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right) \right|, \quad \rho = \sqrt{\kappa_1^2 + \kappa_2^2},$$

where φ is a smooth representation of the curvature type of the point (elliptic $K > 0$, parabolic $K = 0$ or hyperbolic $K < 0$), and ρ represents distance from planarity (we normalize it between 0 and 1). Thus, φ is well defined only for $\rho > 0$ (see Figure 6).

The Dupin Indicatrix. An important notion for computing PH meshes, as has been shown by Wang et al. [2009, Thm. 1], is the concept of the *Dupin Indicatrix* [Eisenhart 1960, Chap. IV].

The Dupin indicatrix at an elliptic point $p \in \mathcal{M}$ is a curve on the tangent plane $T_p\mathcal{M}$, given by the points $p + |\kappa_n(v)|^{-1/2}\hat{v}$ for all unit length vectors $\hat{v} \in T_p\mathcal{M}$. In local curvature coordinates, i.e. taking p as the origin and d_1, d_2 as the local x, y axes, the Dupin indicatrix is the ellipse $|\kappa_1|x^2 + |\kappa_2|y^2 = 1$. At a hyperbolic point it is given by the conjugate hyperbolas $\kappa_1 x^2 + \kappa_2 y^2 = \pm 1$, and at parabolic points by two parallel lines, either $|\kappa_1|x^2 = 1$ or $|\kappa_2|y^2 = 1$ (see Fig. 7 (top)).

The relation between the Dupin indicatrix and planar meshing is clearly visualized when considering the local geometry at p (see Figure 8, inspired by [Lisle 2003, Fig.2]). Consider a normal plane in the direction $\hat{v} \in T_p\mathcal{M}$ at p, and its intersection curve with the surface. Further, take a plane parallel to $T_p\mathcal{M}$ and at distance t from it, such that it intersects the surface, and let $2d$ be the distance between the two intersection points of the surface and these two planes. If r is the radius of curvature at p, then for small enough t we have $d^2 = r^2 - (r-t)^2 = 2t|\kappa_n(\hat{v})|^{-1} + O(t^2)$. Hence, up to first order in t, the intersection curve between the cutting plane and the surface is a uniform scale of the dupin indicatrix (see Figure 7 (bottom)). We use this property when we design our sizing constraints for PH meshing, as we discuss next.

5.2 Dual Meshing

Givan a parameterization $r : \mathcal{U} \subset \mathbb{R}^2 \to \Omega \subset \mathcal{M}$, and a point on the triangular grid $p \in t\mathbb{A}^2 \cap \mathcal{U}$, let p_i, $i \in [1, .., 6]$ be its one-ring grid neighbors, and $\bar{p}_i = (p + p_i + p_{i+1})/3$ be the barycentric dual vertices (see Figure 8 (right)). Next, consider the points $r(\bar{p}_i) \in \mathcal{M}$ and the corresponding hexagon $\mathcal{H}_p$ generated by linearly connecting the pushed forward points to yield a closed polyline. This hexagon represents one element in the hexagonal remeshing of the input surface. Our general goals are:

- *Planarity.* The vertices of $\mathcal{H}_p$ are co-planar, and $\mathcal{H}_p$ is a simple polygon.
- *Approximation.* The Hausdorff distance between the surface and $\mathcal{H}_p$ is minimal among all hexagons with the same area approximating the surface at $p = r(p)$.
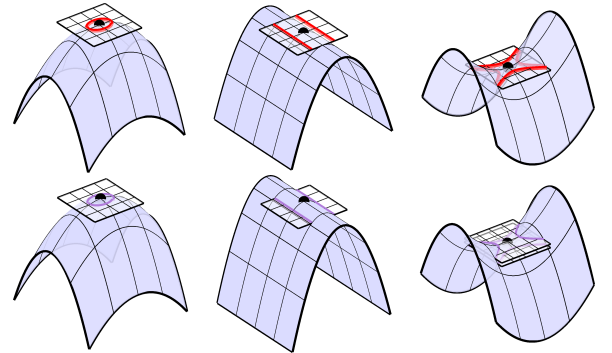- *Fairness.* $\mathcal{H}_p$ has horizontal and vertical reflection symmetry.

Fig. 7. (top) The Dupin indicatrix at $p \in \mathcal{M}$. (bottom) The intersection of a plane parallel to the tangent plane and at a small distance from it with the surface is similar to the Dupin Indicatrix. When p is (left) elliptic, (center) parabolic and (right) hyperbolic.
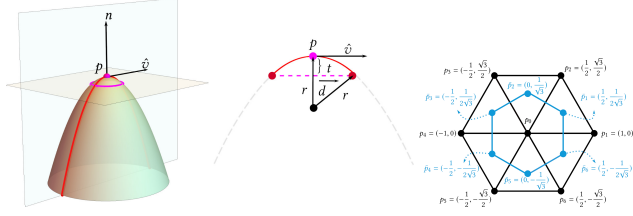
Fig. 8. (left) The local geometry near a point $p \in \mathcal{M}$, with a normal plane in the direction $\hat{v} \in T_p \mathcal{M}$, and a cutting plane parallel to $T_p \mathcal{M}$ at distance t from it. (right) Notations for the pushforward of one ring of triangles, and the corresponding dual hexagon.

5.3 Obstruction to convexity.

Parameterization-based dual meshing generates *convex* hexagonal faces. However, the faces of a PH mesh approximating a given surface are inscribed in a quadric which is a uniform scale of the Dupin indicatrix [Wang and Liu 2009, Thm. 1] (see also Fig. 8 (left)). Hence, for elliptic, parabolic and hyperbolic regions the expected face shapes are *convex* hex, *brick* hex, and *bow-tie* hex, (see Fig 9).

Thus, regardless of the parameterization chosen, we *cannot* expect the mesh faces to be planar, unless the surface has only positive Gaussian curvature. There exist methods for *planarizing* a hexagonal mesh using a non linear optimization [Jiang et al. 2015b; Tang et al. 2014]. These approaches generate a valid PH mesh, which fulfills our stated goals, *given a good initialization*. We denote a mesh that can serve as the initial solution for a planarization approach as *planarizable*. We compute planarizable meshes using the CPF framework, by designing constraints on the CPFs U, V. In the following we describe our pipeline, and the required CPF constraints.

5.4 Algorithm

Our algorithm is outlined in the following (see Figure 10).

5.5 Planarity by Conjugacy and Alignment

5.5.1 Conjugacy and developable strips.
Following Wang et al. [2009] we consider strips of hexagons which are developable (see Figure 11 (left)).
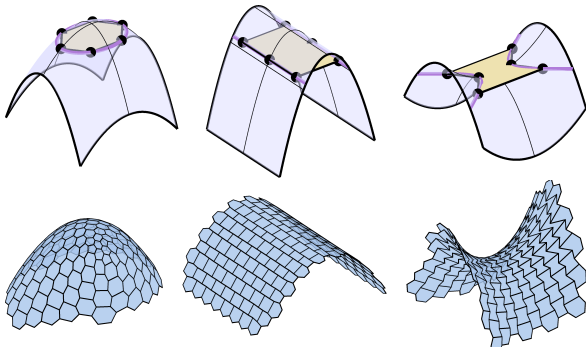

Fig. 9. (top) Expected shapes of planar hexes in different curvature regions: convex, brick and bow-tie. (bottom) Quadric surfaces exhibiting these face shapes: elliptic paraboloid, parabolic cylinder, and hyperbolic paraboloid.

ALGORITHM 1: An outline of our algorithm

Input: M, δ, η, t
Output: M_H

Compute guiding field from curvature, and use as alignment constraints (Sec. 5.5.2 to 5.5.5) ;
Compute scaling constraints from δ, η (Sec. 5.6.2) ;
Add orthogonality (Sec. 5.7) and conjugacy (Sec. 5.5.1) constraints;
Solve optimization problem for U, V (Sec. 4.4) ;
Convert U, V to H_1, H_2, H_3 (Sec. 2.1.3) ;
Integrate H_i into parameterization functions h_i (Sec. B.1) ;
Extract primal triangular mesh M_R from h_i (Sec. B.2) ;
Extract barycentric dual mesh M_D from M_R (Sec. B.2) ;
Planarize dual mesh M_D to the final PH mesh M_H (Sec. C)

The relation between conjugate direction fields and planar meshing is well established [Liu et al. 2006, 2011; Wang and Liu 2009]. Specifically, consider a hexagon $\mathcal{H}_p = \{\bar{p}_i \mid i \in [1,..,6]\}$ at the tangent plane at $p \in \mathcal{M}$. Using the argument that the discrete ruling directions are conjugate to the central curve of a developable strip, Wang et al. [2009] have shown that for a convex hexagon, every pair of consecutive edge vectors $e_i = \bar{p}_i - p$ fulfills $\langle e_i + e_{i+1}, e_i - e_{i+1}\rangle_{S_p} = 0$.

Plugging in the expressions for the edge vectors as a function of U_p, V_p it is straightforward to show (see the supplemental material) that in elliptic regions, these conditions are equivalent to:

$$\langle U_p, V_p\rangle_{S_p} = 0, \tag{17}$$

$$\langle U_p, U_p\rangle_{S_p} = \langle V_p, V_p\rangle_{S_p}. \tag{18}$$

Therefore, in the CPF framework, we use the orthogonality constraint $\langle U_i, V_i\rangle_{s_i} = 0$, taking s_i to be the shape operator at face $t_i \in \mathcal{F}$. Note that Equation (18) implies $\langle U_p, U_p\rangle_{S_p} = \kappa_n(U_p)\|U_p\|^2 = \kappa_n(V_p)\|V_p\|^2 = \langle V_p, V_p\rangle_{S_p}$. In elliptic regions, where S_p is positive definite and the normal curvatures have the same sign, this is equivalent to $|\kappa_n(U_p)|\|U_p\|^2 = |\kappa_n(V_p)|\|V_p\|^2$, and hence U_p, V_p trace a uniform scale of the Dupin indicatrix. Thus, this constraint will be covered by the scaling constraint as we explain in Section 5.6.

5.5.2 Alignment.
A strip of convex hexes is planarizable if there are degrees of freedom allowing the hex vertices to move to the required hex shapes (see Figure 11 (right)). Hence, in parabolic regions the U direction *must* be aligned with the ruling direction, i.e. the direction of *minimal absolute curvature*, which we denote by $\bar{d}$. In hyperbolic regions, the U direction should be aligned with either d_1 or d_2, and in elliptic regions alignment is flexible, yet hexes would be easier to planarize and better shaped if U is aligned either with d_1 or d_2.

The minimal absolute curvature direction d is defined by:

$$\bar{d} = \begin{cases} d_1 & |\kappa_1| < |\kappa_2| \\ d_2 & \text{otherwise} \end{cases}, \tag{19}$$

and it is not necessarily smooth on the surface. For example, in parabolic regions of positive mean curvature, we have $\kappa_1 = 0, \kappa_2 > 0$, and thus $\bar{d} = d_1$. On the other hand, in parabolic regions of negative mean curvature, we have $\kappa_1 < 0, \kappa_2 = 0$ and thus $\bar{d} = d_2$. Furthermore, when the discrete curvature is noisy $\bar{d}$ is not stable in umbilical regions where $\kappa_1 = \kappa_2$, nor in regions where $\kappa_1 = -\kappa_2$.
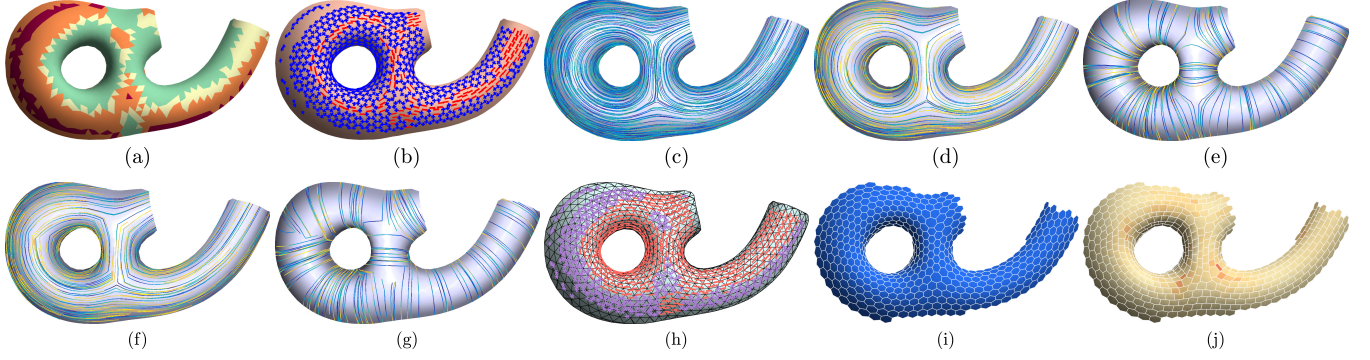
Fig. 10. Our algorithm: (a) curvature regions, (b) alignment constraints for guiding field (Equation (21)), (c) guiding field, (d,e) initial U, V, (f,g) optimized U, V, (h) extracted primal triangular mesh M_R, overlayed with the CPF alignment constraints (Equation (23)),(i) dual hexagonal mesh M_D, (j) final planar mesh M_H.
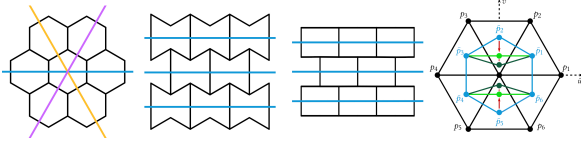

Fig. 11. (left) Developable strips of hexes in different curvature regions. (right) A dual convex hex (blue) can be modified to the brick or bow-tie shapes (green) if it is aligned correctly.

To consolidate the different alignment constraints in different regions, we define *curvature regions* in terms of ρ, φ and then compute a smooth 2-RoSy *guiding field* which will provide alignment constraints for the CPFs computation.

5.5.3 Curvature regions. Following Nieser et al. [2011], we use ρ, φ to robustly define curvature regions that contribute to the alignment constraints as previously discussed. Let $\mathcal{F}_e, \mathcal{F}_p, \mathcal{F}_h \subset \mathcal{F}$ denote the faces sets of elliptic, parabolic and hyperbolic curvature regions, respectively. They are defined as follows:

$$\begin{aligned}
\mathcal{F}_e &= \{\, t_i \in \mathcal{F}_{npl} \setminus \mathcal{F}_u \quad | \quad \varphi - \pi/4 \geq \varphi_p \,\}, \\
\mathcal{F}_p &= \{\, t_i \in \mathcal{F}_{npl} \quad | \quad |\varphi - \pi/4| < \varphi_p \,\} \\
\mathcal{F}_h &= \{\, t_i \in \mathcal{F}_{npl} \quad | \quad \varphi - \pi/4 \leq -\varphi_p \,\},
\end{aligned} \tag{20}$$

where $\mathcal{F}_{npl} = \{t_i \in \mathcal{F} \mid \rho > \rho_m\}$ are the non planar regions, $\mathcal{F}_u = \{t_i \in \mathcal{F} \mid \varphi \geq \pi/2 - \varphi_e\}$ are the umbilical regions, and we take $\rho_m = 0.01, \varphi_p = 0.05, \varphi_e = 0.2$. Figure 12 shows a few examples of color coded curvature regions.
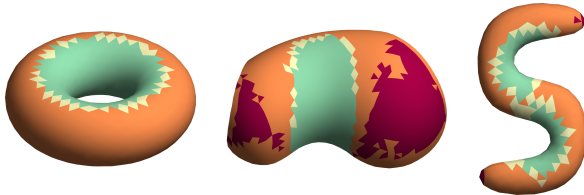
5.5.4 Guiding field. We define an optimization problem for a 2-RoSy *guiding field* y, that is aligned with either d_1 or d_2 according to the curvature regions. Let $\mathcal{F}_2 = \mathcal{F}_p, \mathcal{F}_4 = \mathcal{F}_e \cup \mathcal{F}_h$. Then, for all faces in $\mathcal{F}_2$, the guiding field should be aligned with $\pm\bar{d}$, where as for faces in $\mathcal{F}_4$ it should be aligned either with $\pm\bar{d}$ or $\pm\mathrm{i}\bar{d}$. Hence, we obtain the following optimization problem.

$$\begin{aligned}
\underset{y \in \mathbb{R}^{2m}}{\text{minimize}} \quad & y^T \Delta_2 y \\
\text{subject to} \quad & y_i - \bar{d}_i^2 = 0, \quad t_i \in \mathcal{F}_2, \\
& y_i^2 - \bar{d}_i^4 = 0, \quad t_i \in \mathcal{F}_4,
\end{aligned} \tag{21}$$

where $\Delta_2 \in \mathbb{R}^{2m \times 2m}$ denotes the 2-RoSy Laplacian [Knöppel et al. 2013], expressed in the local face basis.

This problem has a convex quadratic objective and linear and quadratic constraints. We reformulate it as an unconstrained non-linear least squares problem, as follows.

$$\underset{y \in \mathbb{R}^{2m}}{\text{minimize}} \quad \frac{1}{\lambda} y^T \Delta_2 y + \beta(E^{\Psi_2} + E^{\Psi_4}), \tag{22}$$

where E^{Ψ_2}, E^{Ψ_4} are the average $L2$ errors of the first and second constraints, respectively, λ is the first non-zero eigenvalue of Δ_2, and we take the penalty weight $\beta = 10^5$.

The problem (21) without the second constraint is convex, and we take its solution as the initial solution for the problem (22), for which we use the Levenberg-Marquardt algorithm. Figure 13 (left) shows an example of 2- and 4-RoSy constraints and the resulting guiding field y (middle).

Fig. 12. Elliptic (orange, $\mathcal{F}_e$), parabolic (yellow, $\mathcal{F}_p$) and hyperbolic (green, $\mathcal{F}_h$) curvature regions. Red regions are umbilical.
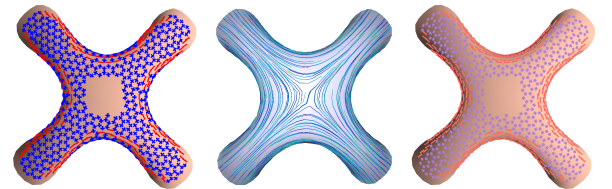

Fig. 13. (left) Constraints from curvature regions, (middle) resulting 2-RoSy guiding field y, (right) alignment constraints for CPF optimization.
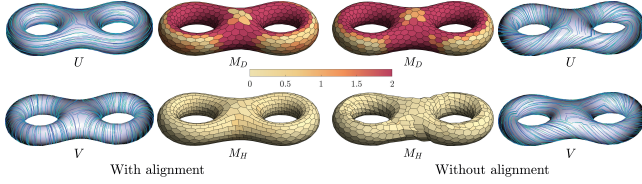
Fig. 14. Our results with (left) and without (right) alignment constraints. We show the final U, V fields, and the dual and final meshes. Note that without alignment constraints the U field is not aligned with the minimal absolute curvature direction, leading to a non-planarizable mesh.

5.5.5 CPF Alignment constraints.

Finally, we use the guiding field to setup the alignment constraints for the CPF framework. As discussed, in elliptic regions there are three developable strips, thus the guiding field can be aligned with any of the pushed forward grid directions. On the other hand, in parabolic and hyperbolic regions there is a single strip, which should be aligned with the pushed forward $\pm\hat{u}$ to generate a planarizable dual hex, as in Figure 11 (right). Hence, we set the constraints:

$$d_i = \sqrt{y_i}, \quad n_i = 6, \quad t_i \in \mathcal{F}_e,$$
$$d_i = \sqrt{y_i}, \quad n_i = 2, \quad t_i \in \mathcal{F}_p \cup \mathcal{F}_h. \tag{23}$$

The CPF alignment constraints are visualized in Figure 13 (right).

These constraints are essential for achieving a planarizable mesh. In Figure 14 we compare the results of our algorithm with and without alignment constraints, shown in the left and right part of the Figure, respectively. We show the resulting U, V fields, the dual mesh M_D and the final PH mesh M_H. In both experiments we used conjugacy and scaling constraints. Note that without alignment constraints, the resulting U field is not aligned with the minimal absolute curvature direction, and the planarization leads to deformed hexagons. In this figure, and all the figures that show color coded planarity error, all the faces that have a planarity error larger than the maximal value on the color scale are showed with the color of the maximal value. This is done to better visualize the low planarity regions on meshes which have large maximal planarity error.
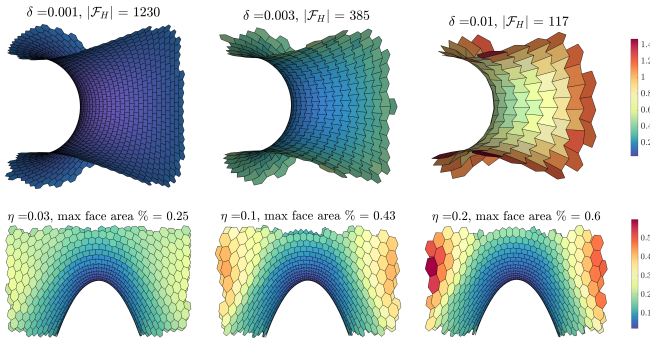

Fig. 15. Effect of scaling parameters δ, η. (top) Decreasing δ while keeping η fixed allows a closer approximation of the surface at non-degenerate curvature regions. (bottom) Increasing η while keeping δ fixed allows for larger elements in flat regions. We show the color coding of the face area's percentage of the total surface area.
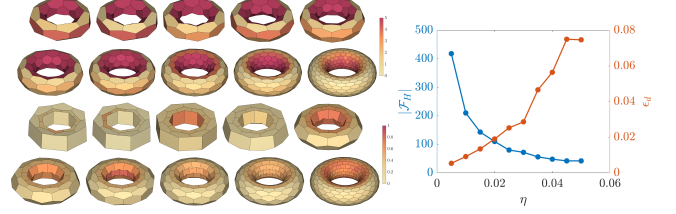
Fig. 16. Varying η, δ together while keeping their percentage ratio fixed allows us to smoothly vary the mesh resolution. (left) The two top rows show the dual mesh before planarization, and the two bottom rows the final PH mesh, all color coded by the planarity value (with maximal values 5 and 1 resp.). (right) The trade-off between the number of elements $|\mathcal{F}_H|$ and the Hausdorff distance ϵ_d as a function of η. As expected, lower values of η lead to a smaller Hausdorff distance at the expense of a larger number of faces.

5.6 Approximation by Scaling

5.6.1 Regularized Dupin Indicatrix.

Our conjugacy and alignment constraints set the *orientation* of the pushed forward hexagon. For the resulting hexagons to be circumscribed in a uniform scale of the Dupin indicatrix, as required, we additionally constrain the *scaling* of a pushed forward vector $w \in T_p\mathcal{M}$. In general, in an elliptic region, w should fulfill $\|w\|^2|\kappa_n(w)| = \eta, \forall w$ s.t. $\kappa_n(w) \neq 0$, where $\eta > 0$ controls the area of the element.

There are two issues to note. First, the vector size $\|w\|$ becomes unbounded as $\kappa_n \to 0$. Second, our hexagons are regular hexagons which are pushed forward through a parameterization, and thus are *always* convex. Thus, w should always trace an ellipse, even in hyperbolic regions, and it should have the aspect ratio of the Dupin indicatrix. We address both issues with the following definition.

Definition 5.1. Let $p \in \mathcal{M}$, with curvature directions d_1, d_2, and consider the local curvature coordinates in $T_p\mathcal{M}$ where p is the origin, and d_1, d_2 are the x, y axes, respectively. Given $\epsilon > 0$, the ϵ-*regularized Dupin indicatrix* $D_\epsilon(p)$ is the ellipse $x^2\tilde{\kappa}_{1,\epsilon} + y^2\tilde{\kappa}_{2,\epsilon} = 1$, where $\tilde{\kappa}_{1,\epsilon} = \sqrt{\kappa_1^2 + \epsilon^2}$, $\tilde{\kappa}_{2,\epsilon} = \sqrt{\kappa_2^2 + \epsilon^2}$. The ϵ-*regularized normal curvature* in direction $w \in T_p\mathcal{M}$ is given by $\tilde{\kappa}_{n,\epsilon}(w) = \tilde{\kappa}_{1,\epsilon}\cos^2\theta + \tilde{\kappa}_{2,\epsilon}\sin^2\theta$, where θ is the angle of w w.r.t d_1.

The function $x \to \sqrt{x^2 + \epsilon^2}$ is a well known smoothing of the function $|x|$ that is known to be convex, differentiable, positive, and within an ϵ-dependent bound from $|x|$ [Argaez et al. 2011]. Geometrically, we replace $|x|$ with the hypotenuse of a right angle triangle with leg lengths $|x|$ and $\epsilon > 0$, and thus we are guaranteed that the result is positive, and has the desired approximation effect. Furthermore, the regularized Dupin indicatrix is always well defined, and provides a smooth, non-degenerate transition between elliptic and hyperbolic regions. We further have the following.

LEMMA 5.2. *Let* $p \in \mathcal{M}$, *with curvatures* κ_1, κ_2, *and let* $D_\epsilon(p)$ *be its regularized Dupin indicatrix. Then, the area* $\alpha_\epsilon(p)$ *of* $D_\epsilon(p)$ *fulfills:*

$$\lim_{\substack{|\kappa_1| \gg 0 \\ |\kappa_2| \gg 0}} \alpha_\epsilon(p) = \frac{\pi}{\sqrt{|\kappa_1|}\sqrt{|\kappa_2|}}, \qquad \lim_{\substack{\kappa_1 \to 0 \\ \kappa_2 \to 0}} \alpha_\epsilon(p) = \frac{\pi}{\epsilon}$$

$$\lim_{\substack{\kappa_1 \to 0 \\ |\kappa_2| \gg 0}} \alpha_\epsilon(p) = \frac{\pi}{\sqrt{\epsilon}\sqrt{|\kappa_2|}}, \qquad \lim_{\substack{\kappa_2 \to 0 \\ |\kappa_1| \gg 0}} \alpha_\epsilon(p) = \frac{\pi}{\sqrt{\epsilon}\sqrt{|\kappa_1|}}. \tag{24}$$
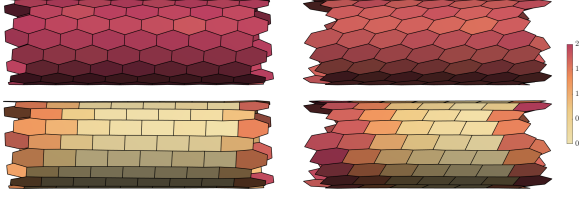
Fig. 17. The dual mesh (left) with and (right) without orthogonality constraints, (top) before and (bottom) after planarization. Without orthogonality hexes are skewed, but remain planar.

The proof follows directly from the fact $\lim_{x \to 0} \sqrt{x^2 + \epsilon^2} = \epsilon$.

Hence, by choosing ϵ, γ, and requiring $\|w\|^2 |\tilde{\kappa}_{n,\epsilon}(w)| = \gamma^2, \forall w = dr(\mathbf{w}) \in T_p\mathcal{M}$, we constrain the pushforward unit circle to be a *uniform γ-scale of the ϵ-regularized Dupin indicatrix*. This allows us to control the areas of the pushed forward hexagons, and provide a smooth transition between elliptic, parabolic and hyperbolic regions.

5.6.2 CPF scaling constraints. We set up the scaling constraints:

$$\|U_i\|^2_{g_i(\epsilon,\gamma)} = 1, \quad \|V_i\|^2_{g_i(\epsilon,\gamma)} = 1,$$
$$g_i(\epsilon,\gamma) = \frac{1}{\gamma^2}\begin{pmatrix} d_1 & d_2 \end{pmatrix}\begin{pmatrix} \tilde{\kappa}_{1,\epsilon} & 0 \\ 0 & \tilde{\kappa}_{2,\epsilon} \end{pmatrix}\begin{pmatrix} d_1 & d_2 \end{pmatrix}^T = \frac{1}{\gamma^2}\tilde{S}_\epsilon. \quad (25)$$

We discuss the effect of the combined constraints in different regions in Section 5.8.

5.6.3 Choosing ϵ and γ. The choice of the scaling parameters ϵ, γ affects the aspect ratio of the hexagonal elements in parabolic regions, as well as the maximal element size in planar regions. The area of the γ-scaled ϵ-regularized Dupin indicatrix is $\gamma^2 \alpha_\epsilon(p)$, which is equal to $\gamma^2 \pi/\epsilon$ in planar regions (Equation (24)).

Let δ be the desired elements' distance to the surface, and η the maximal element size. Thus, we have $\pi\gamma^2/\epsilon = \eta$, where $\gamma^2 = 2\delta + O(\delta^2)$ (see Figure 8 (left)). Hence, we take $\epsilon_{\delta,\eta} = \frac{2\pi\delta}{\eta}, \gamma_{\delta,\eta} = \sqrt{2\delta}$.

Figure 15 shows the effects of choosing different δ, η. Varying δ while keeping η fixed (top) allows a closer approximation of the surface at non-degenerate locations. Varying η while keeping δ fixed (bottom) allows for larger elements in flat regions. The values of δ, η are taken as a percentage of the bounding box diagonal, and
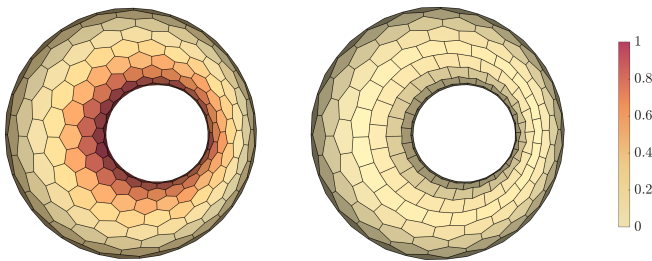

Fig. 18. The dual mesh (left) before and (right) after planarization. Hexagonal faces in elliptic regions are nearly planar even before planarization, since the hexagons are inscribed in a scaled Dupin indicatrix.
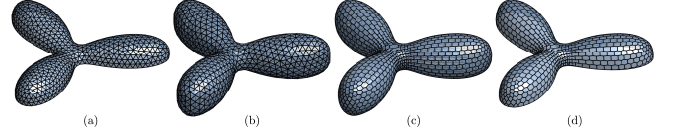

Fig. 19. Preserving the input's (a) single planar boundary curve by doubling the mesh (b), PHexing it (c), and intersecting the result with the boundary plane (d).

total surface area, respectively. The figure shows the color coding of the face areas percentage of the total area of the mesh.

Figure 16 shows how we can control the mesh resolution by scaling both η and δ while keeping their percentage ratio fixed. The values of η vary linearly from 5% (top left) to .5% (bottom right) of the total surface area, and δ varies from 10% to 1% of the bounding box diagonal. The color scale is uniform for all images, and varies from 0 to 2. The two top rows show the dual mesh before planarization, and the two bottom rows show the final PH mesh. Note that here the initial planarity error is quite high, yet we still obtain a coarse planar hex mesh after planarization. In general, for simple surfaces with low curvature variation, we take δ to be 0.5% of the bounding-box diagonal of the input surface, and η to be 0.25% of the total surface area. For more challenging general meshes, we take the same δ, but reduce η to be 0.1%, to generate smaller elements.

5.7 Fairness by Orthogonality

Our last objective is *fairness*, formalized as horizontal and vertical symmetry of the hexagonal elements. We achieve this by requiring the orthogonality of U_i, V_i, namely by adding the CPF constraints:

$$\langle U_i, V_i \rangle = 0, \quad \forall t_i \in \mathcal{F}. \quad (26)$$

In parabolic and hyperbolic regions the planarity optimization modifies the shapes of the pushed-forward hexes away from convexity. If the symmetry axes of the pushed forward hexagon are not aligned with the curvature directions, this modification can lead to skewed hexes, for example parallelograms instead of rectangular bricks in parabolic regions. By enforcing orthogonality as well as conjugacy, we effectively require that U_i, V_i are aligned with the curvature directions, and thus that the planarization preserves the horizontal and vertical reflectional symmetry of the hexes. Note that, unlike for PQ meshes, these combined constraints do not replace alignment, since for PH meshing the U direction must be aligned with the minimal absolute curvature direction, whereas for PQ meshes U, V can be aligned with either minimal or maximal directions.
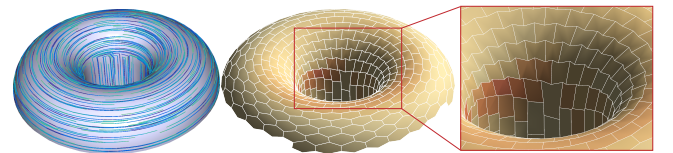

Fig. 20. Failure case: neighboring parabolic regions with conflicting maximal absolute curvature directions lead to a non-smooth guiding field (left), which leads to multiple singularities and badly shaped hexagons (right).
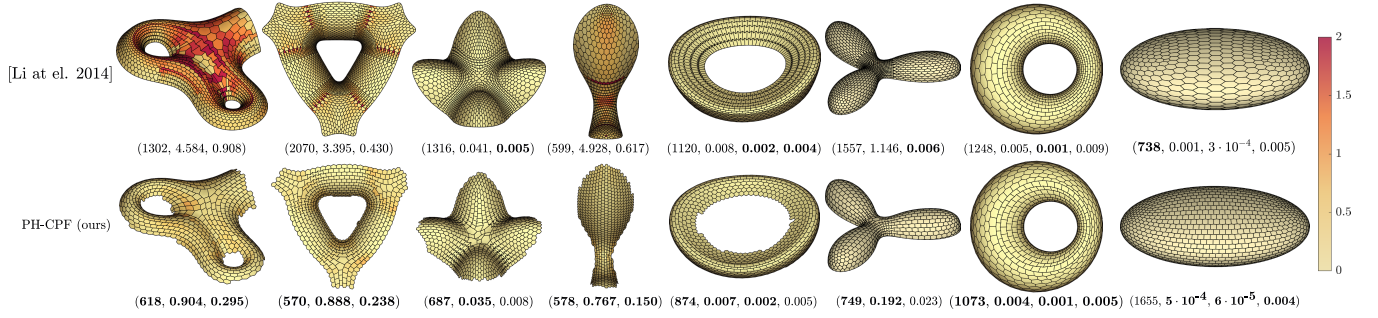
Fig. 21. Comparison of our results with Li et al. [2014], color coded with the face planarity values. Below each mesh, we show ($|\mathcal{F}_H|, \epsilon_p, \bar{\epsilon}_p, \epsilon_d$) where ϵ_d is displayed only if Li's input mesh was provided. Note that we obtain better or similar planarity with a smaller number of faces.

Figure 17 demonstrates the effect of adding orthogonality constraints. On the left we show the result with orthogonality, and on the right without it. We show the dual mesh before planarization M_D (top) and the final PH mesh M_H (bottom). Note that without orthogonality the hexes of M_H are still planar, but lose horizontal and vertical symmetry with respect to the ruling direction.

5.8 Constraint analysis

5.8.1 Elliptic regions. When $\kappa_1 \kappa_2 \gg 0$ we have $\tilde{S}_\epsilon \approx S$, and thus the Dupin Equation (18) holds approximately. Furthermore, in these regions, for any $w \in \mathbb{R}^2$, $|w| = 1$ we have that $\|dr(w)\|^2_{g_i(\epsilon,\gamma)} = \frac{1}{\gamma^2} w^T (U_i V_i)^T \tilde{S}_\epsilon (U_i V_i) w \approx w^T \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} w = 1$. Hence, in elliptic regions the scaling and conjugacy constraints imply that pushing forward the unit circle yields (approximately) a uniform γ-scale of the ϵ-regularized Dupin indicatrix. Indeed, in elliptic regions (when γ is small enough) the initial planarity error is considerably lower compared to the rest of the surface (see Figure 18).

5.8.2 Parabolic regions. When either $\kappa_1 \rightarrow 0$ or $\kappa_2 \rightarrow 0$, we have $\langle \bar{d}, w \rangle_S \rightarrow 0$ for any $w \in T_p \mathcal{M}$. Thus, if Eq. (23) holds and U is aligned with $\bar{d}$, then the conjugacy Eq. (17) holds for any tangent vector w. The dimensions of the resulting element are restricted

in the ruling direction by the scaling parameters ϵ, γ, and in the orthogonal direction by the non-zero curvature, as given by Eq. (24).

5.8.3 Hyperbolic regions. If the alignment and conjugacy equations (23), (17) hold, then U, V are aligned with the curvature directions. In this case, it is easy to check that $\langle U_i, V_i \rangle_{\tilde{S}_\epsilon} = 0$, and if additionally the scaling equation (25) holds we have again that $\|dr(w)\|^2_{g_i(\epsilon,\gamma)} = 1$ for any unit vector $w \in \mathbb{R}^2$, and thus pushing forward the unit circle yields a uniform γ-scale of the ϵ-regularized Dupin indicatrix.

5.8.4 Planar regions. When both $\kappa_1 \rightarrow 0, \kappa_2 \rightarrow 0$, we have that $g_i(\epsilon, \gamma) \rightarrow \frac{\epsilon}{\gamma^2} Id$, thus the sizing is uniform, and determined by the scaling parameters. If we additionally require orthogonality, this amounts to requiring conformality in planar regions.

5.9 Initialization

We initialize the variables U, V as follows $\forall t_i \in \mathcal{F}$:

$$\tilde{U}_i^0 = \sqrt{y_i}, \qquad \tilde{V}_i^0 = -J\tilde{U}_i^0,$$
$$U_i^0 = \tilde{U}_i^0 / \|\tilde{U}_i^0\|_{g_i(\epsilon,\gamma)}, \quad V_i^0 = \tilde{V}_i^0 / \|\tilde{V}_i^0\|_{g_i(\epsilon,\gamma)}, \tag{27}$$

and z as $\forall e_{ij} \in \mathcal{E}^I$:

$$z_{ij}^0 = \left(\left((dr_i^0)^{-1}(e_{ij}) \right)^N + \left((dr_j^0)^{-1}(e_{ij}) \right)^N \right) / 2. \tag{28}$$

Thus, the initial solution fulfills the conjugacy, scaling, orthogonality and alignment constraints, and is of course LICO as required. The optimization improves the continuity objective while maintaining smoothness and the other constraints as best as possible.
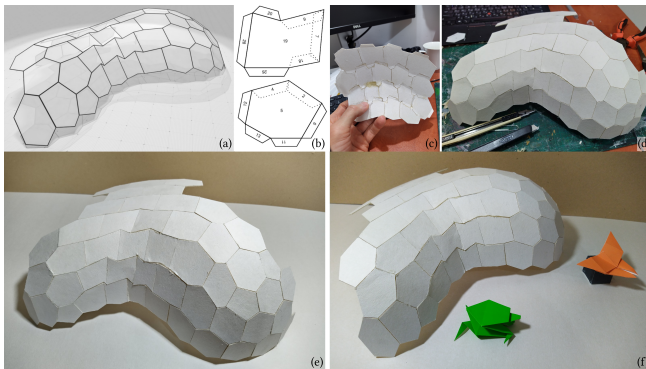

Fig. 22. Fabrication of the Laga model from construction paper. (a) PH mesh, (b) 2D face templates for cutting, (c-d) intermediate and (e-f) final constructions.
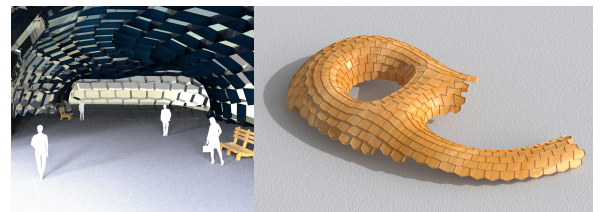

Fig. 23. Face offset meshes computed from our PH meshes. Note that these are computable only for conical meshes.
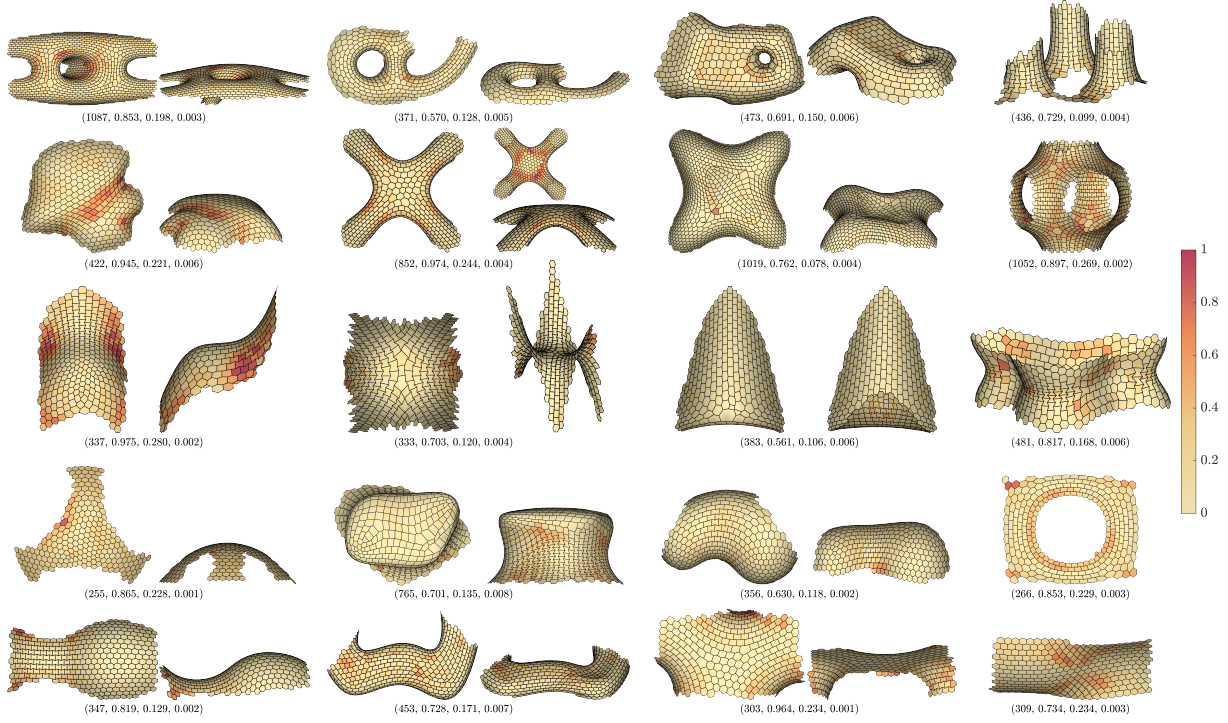
Fig. 24. Gallery of results for architectural models. We show the PH mesh, color coded with planarity values, and below $(|\mathcal{F}_H|, \epsilon_p, \bar{\epsilon}_p, \epsilon_d)$.

6 IMPLEMENTATION DETAILS

We provide the details on mesh generation and planarization in the Appendices.

6.1 Constraints computation

6.1.1 Shape operator. For the shape operator we take the tangential gradient of the normals at the vertices, symmetrized and projected to the tangent plane [Azencot et al. 2015]. If normals at the vertices are available (e.g. if the surface was sampled from a parametric surface) we use those, otherwise we take the area-weighted average of the normals of the neighboring faces. For the large majority of meshes, the normals are computed.

6.1.2 Boundary faces. Boundary faces are defined as faces with at least one vertex on the boundary. On these faces, the computation of vertex normals (and the shape operator) is not reliable, and therefore we do not constrain these faces for alignment, nor conjugacy.

6.1.3 Guiding field, special cases. If there are no constraints, the guiding field is given by the lowest eigenvector of the 2-RoSy Laplacian. If there are only 2-RoSy constraints, the optimization problem (21) is convex quadratic and is solved directly using constrained linear least squares. If there are only 4-RoSy constraints, then the curvature is uniform, either hyperbolic or elliptic, and we take the minimal curvature direction d_1 in constrained regions as the 2-RoSy constraints for the guiding field optimization.

6.1.4 Alignment weighting. As is common in meshing, we add diagonal weights to the alignment constraint, to reduce the weight in nearly planar regions. Specifically, for $N = 4$ we weigh the alignment constraint on the face $t_i \in \mathcal{F}$ with ρ_i^2, and for $N = 6$ with $\left(\rho_i^3 - 3\rho_i^2 + 3\rho_i\right)^2$, a polynomial which resembles a Sigmoid function and has vanishing first and second derivatives at 1.

6.2 Boundaries

Unlike quadrangular meshes, hexagonal elements cannot naturally conform to the surface boundaries, as in convex regions the hexagons will form "jagged" boundaries. Thus, a boundary-conforming mesh will have many non-hexagonal faces. Furthermore, if the boundaries are parallel to the maximal absolute curvature direction, arbitrarily shaped low quality faces may be obtained when the curvature-aligned hexagons intersect the non-aligned boundary. Thus, many recent works on hexagonal remeshing [Jiang et al. 2015b; Müller 2011; Wang and Liu 2009; Wang et al. 2008] have chosen to generate meshes whose boundaries do not conform to the input shape. Following the large body of existing work, we have chosen a similar path. Note, however, that methods that preserve boundaries, such as Li et al. [2014], are advantageous in many scenarios, especially in the architectural context.

When there is a single planar boundary curve, one possible solution is *doubling* the input surface by reflecting it with respect to the boundary plane. This leads to a closed surface, which is then be intersected with the boundary plane after hex-meshing. See Figure 19 for such an example. Finding a robust, general solution for handling the boundary is an interesting avenue for future work.
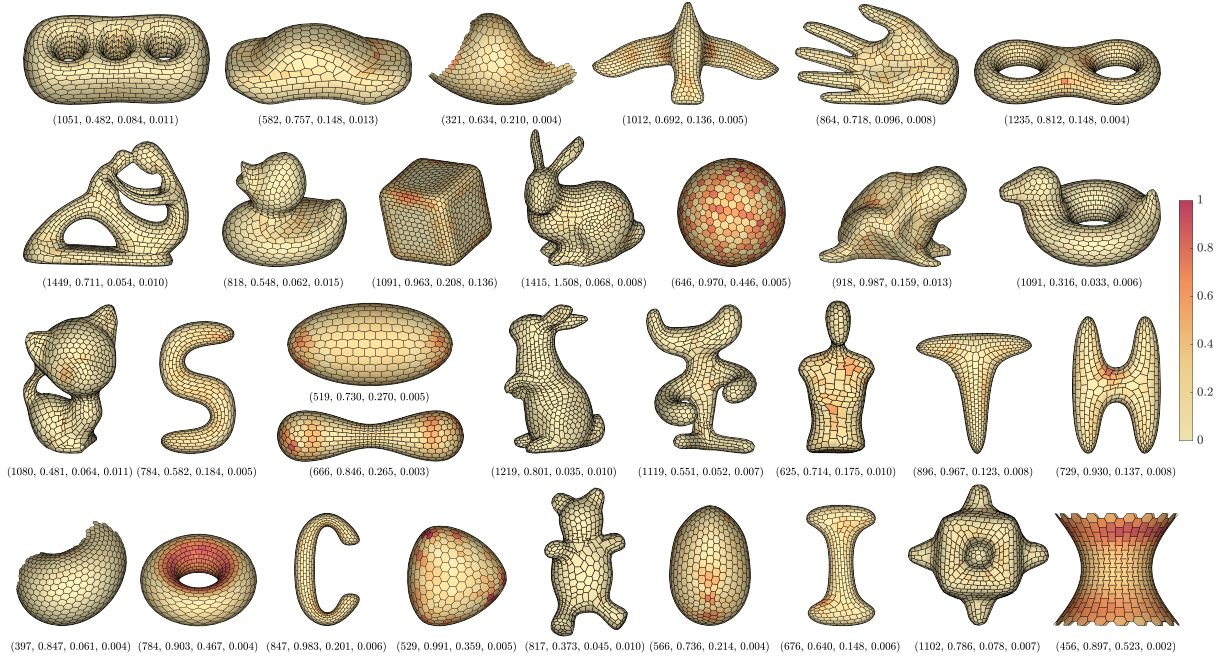
(1051, 0.482, 0.084, 0.011) (582, 0.757, 0.148, 0.013) (321, 0.634, 0.210, 0.004) (1012, 0.692, 0.136, 0.005) (864, 0.718, 0.096, 0.008) (1235, 0.812, 0.148, 0.004)

(1449, 0.711, 0.054, 0.010) (818, 0.548, 0.062, 0.015) (1091, 0.963, 0.208, 0.136) (1415, 1.508, 0.068, 0.008) (646, 0.970, 0.446, 0.005) (918, 0.987, 0.159, 0.013) (1091, 0.316, 0.033, 0.006)

(519, 0.730, 0.270, 0.005)

(1080, 0.481, 0.064, 0.011) (784, 0.582, 0.184, 0.005) (666, 0.846, 0.265, 0.003) (1219, 0.801, 0.035, 0.010) (1119, 0.551, 0.052, 0.007) (625, 0.714, 0.175, 0.010) (896, 0.967, 0.123, 0.008) (729, 0.930, 0.137, 0.008)

(397, 0.847, 0.061, 0.004) (784, 0.903, 0.467, 0.004) (847, 0.983, 0.201, 0.006) (529, 0.991, 0.359, 0.005) (817, 0.373, 0.045, 0.010) (566, 0.736, 0.214, 0.004) (676, 0.640, 0.148, 0.006) (1102, 0.786, 0.078, 0.007) (456, 0.897, 0.523, 0.002)

Fig. 25. Gallery of results for general models. We show the PH mesh, color coded with planarity values, and below ($|\mathcal{F}_H|, \epsilon_p, \bar{\epsilon}_p, \epsilon_d$).

6.3 Limitations

Our method relies on the existence of a smooth 2-RoSy guiding field, which is aligned to the maximal absolute curvature direction in parabolic regions. In some cases, when there are adjacent parabolic regions with a maximal absolute curvature direction which changes abruptly (see Fig. 20 (a)), there exists no such smooth field. In these cases, multiple singularities are introduced along the discontinuity line, leading to badly shaped hexagons (Fig. 20 (b)). This is a limitation that can potentially be addressed by allowing both quads and hexagons, a direction we leave for future work.

We solve a non-linear non-convex optimization problem, and the solution that we obtain depends on the initialization. While our algorithm is quite stable, as we have demonstrated using convergence from a random initialization, we cannot guarantee that the global optimum would be reached. Furthermore, we rely heavily on curvature information, for all our constraints except orthogonality. While we have used a standard off-the-shelf algorithm for computing curvatures, using more elaborate curvature computation schemes can potentially improve the accuracy of our results.

7 EXPERIMENTAL RESULTS

7.1 Evaluation metrics for PH meshes

Let $M_H = (\mathcal{V}_H, \mathcal{F}_H, \mathcal{E}_H)$ be the output hexagonal mesh.

Planarity. The planarity error ε_p is computed per face $f_i = \{p_1, .., p_d\} \in \mathcal{F}_H$ according to standard practice [Vaxman et al. 2017]:

$$\varepsilon_p(f_i) = \left(\frac{1}{d} \sum_{j=1}^{d} \varepsilon_p^2(f_{q_j})\right)^{1/2}, \qquad \varepsilon_p(f_q) = \frac{|\langle \hat{n}, p_{21}\rangle|}{\left(\|p_{31}\| + \|p_{42}\|\right)/2},$$
$$\hat{n} = \frac{p_{31} \times p_{42}}{\|p_{31}\| \|p_{42}\|}, \qquad\qquad p_{ij} = p_i - p_j. \tag{29}$$

where $q_j = \{p_j, p_{j+1}, p_{j+2}, p_{j+3}\}$, and the indexing is modulo $d = |f_i|$. We denote by ϵ_p the maximal planarity error over all the faces, and by $\bar{\epsilon}_p$ the average planarity error. Planarity errors are given in percentages, e.g. when we refer to planarity error 1 in the results, the meaning is 1%.

Hausdorff distance. We use the one sided Hausdorff distance ϵ_d from M_H tp M using the VCG package [Cignoni et al. 2020] with face sampling, and measured w.r.t the bounding box diagonal.

7.2 Comparison with Li et al. [2014]

We compare our PH meshes with the results of the method by Li et al. [2014] which were supplied by the authors. As input to our method, we used either Li's input mesh if it was provided, or a remeshed version of their output. Figure 21 shows the qualitative results, color coded with the face planarity values, and the quantitative results ($|\mathcal{F}_H|, \epsilon_p, \bar{\epsilon}_p, \epsilon_d$). Note that we obtain better or similar planarity values with a smaller number of faces. Furthermore, the transition between element shapes in our approach is smoother. We choose the target planarity value for our planarization as the minimum between the maximum planarity error for the Li model and 1%. An important advantage of Li et al. [2014] is that their result preserves the input boundary, which is important in some applications. For some special cases we can generate a boundary conforming mesh (third result from the right, see also Section 6.2).

7.3 Fabrication

To demonstrate the planarity of our result, we generated a very coarse PH mesh of the Laga model, having only 49 faces, and realized it using construction paper. The fabrication has been done by hand

using printed 2D face templates that were transferred over a sheet of construction paper and cut using a precision crafting knife. We chose this approach—over options such as laser cutting and plywood or plexiglass—mainly because construction paper is easy to work with and allows a straightforward design of connectors between faces. Nevertheless, this approach has low accuracy, and is likely not suitable for crafting meshes with a large number of faces. The results are shown in Figure 22.

7.4 Face offset meshes

Fig. 1 and Fig. 23 show face offset meshes, which are in general only computable for conical meshes. In Figure 1 (right) and Fig. 23 we show only the PH mesh, and the extruded edges to the face offset mesh for better visualization.

7.5 Robustness

We show results for a variety of architectural and general meshes (Fig. 24 and 25). For each mesh we show the output PH mesh M_H color coded with planarity errors, and specify how many faces it has $|\mathcal{F}_H|$, the maximal planarity error ϵ_p, the average planarity error $\bar{\epsilon}_p$ and the Hausdorff distance to the source mesh ϵ_d. To the best of our knowledge, PH meshes of such complicated general models have not been demonstrated before. The most complex PH-Meshes appear in Li et al. [2014], and have limited curvature variation. Models such as the kitten, bunny, teddy bear and fertility, with large curvature variation, have not been PHexed previously.

8 CONCLUSIONS

We proposed a novel framework for parameterization-driven meshing, at the heart of which lies a novel integrability constraint, which is provably zero when a quantized rotationally-seamless parameterization exists. We have leveraged this framework for designing planar hexagonal meshes, succeeding to mesh a variety of complex models, which have not been demonstrated to be PH-meshable previously. We have additionally showed that our method is general, and is easily applicable to planar quad meshing as well.

We believe that both our general framework, as well as the PH meshing application would be highly useful to upstream geometry processing applications and to practitioners. Due to the flexibility of our framework, it is interesting to investigate how additional constraints, e.g. physical constraints, can be added. Furthermore, using our approach for quick validation during form-finding is a fruitful direction for future work. Finally, generalizing the approach to work directly on a free-form parameterized surface instead of a surface mesh is both theoretically interesting and practically useful.

REFERENCES

Noam Aigerman and Yaron Lipman. 2016. Hyperbolic Orbifold Tutte Embeddings. *ACM Trans. Graph.* 35, 6, Article 217 (Nov. 2016), 14 pages.

Henrik Almegaard, Anne Bagger, Jens Gravesen, Bert Jüttler, and Zbynek Šír. 2007. Surfaces with Piecewise Linear Support Functions over Spherical Triangulations. In *Mathematics of Surfaces XII*, Ralph Martin, Malcolm Sabin, and Joab Winkler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 42–63.

Miguel Argaez, Carlos Ramirez, and Reinaldo Sanchez. 2011. An l1-algorithm for underdetermined systems and applications. In *2011 Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, 1–6.

Omri Azencot, Etienne Corman, Mirela Ben-Chen, and Maks Ovsjanikov. 2017. Consistent functional cross field design for mesh quadrangulation. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.

Omri Azencot, Orestis Vantzos, Max Wardetzky, Martin Rumpf, and Mirela Ben-Chen. 2015. Functional thin films on surfaces. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 137–146.

David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013a. Integer-grid maps for reliable quad meshing. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013b. Quad-Mesh Generation and Processing: A Survey. *Comput. Graph. Forum* 32, 6 (2013), 51–76.

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

Alon Bright, Edward Chien, and Ofir Weber. 2017. Harmonic Global Parametrization with Rational Holonomy. *ACM Trans. Graph.* 36, 4, Article 89 (2017), 15 pages.

Marcel Campen, David Bommes, and Leif Kobbelt. 2015. Quantized global parametrization. *Acm Transactions On Graphics (TOG)* 34, 6 (2015), 1–12.

Marcel Campen, Moritz Ibing, Hans-Christian Ebke, Denis Zorin, and Leif Kobbelt. 2016. Scale-invariant directional alignment of surface parametrizations. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 1–10.

Paolo Cignoni et al. 2020. vcglib: Visualization and Computer Graphics Library. https://github.com/cnr-isti-vclab/vcglib/.

Marinella Contestabile and Ornella Iuorio. 2019. A digital design process for shell structures. *Proceedings of IASS Annual Symposia* 2019, 15 (2019), 1–8.

Alexander Peña De Leon. 2012. Rationalisation of freeform façades: A technique for uniform hexagonal panelling. In *In Beyond codes and pixels: Proceedings of the 17th International Conference on Computer-Aided Architectural Design Research in Asia*. 243–251.

Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing N-PolyVector fields with complex polynomials. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 1–11.

Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Integrable PolyVector Fields. *ACM Trans. Graph.* 34, 4, Article 38 (2015), 12 pages.

Manfredo P Do Carmo. 2016. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications.

Luther Pfahler Eisenhart. 1960. *A treatise on the differential geometry of curves and surfaces*. Dover, New York.

Nahum Farchi and Mirela Ben-Chen. 2018. Integer-only cross field computation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Xiao-Ming Fu and Yang Liu. 2016. Computing Inversion-Free Mappings by Simplex Assembly. *ACM Transactions on Graphics (SIGGRAPH Asia)* 35, 6 (2016).

Henri Gavin. 2020. The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University* (2020), 1–19. http://people.duke.edu/~hpgavin/ce281/lm.pdf

Viktoria Henriksson and Maria Hult. 2015. *Rationalizing freeform architecture-Surface discretization and multi-objective optimization*. Master's thesis. https://hdl.handle.net/20.500.12380/231658

Kai Hormann, Bruno Lévy, and Alla Sheffer. 2007. Mesh parameterization: Theory and practice. *SIGGRAPH 2007 Course Notes* (2007).

Caigui Jiang, Chengcheng Tang, Amir Vaxman, Peter Wonka, and Helmut Pottmann. 2015b. Polyhedral Patterns. *ACM Trans. Graph.* 34, 6, Article 172 (2015), 12 pages.

Caigui Jiang, Jun Wang, Johannes Wallner, and Helmut Pottmann. 2014. Freeform honeycomb structures. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 185–194.

Tengfei Jiang, Xianzhong Fang, Jin Huang, Hujun Bao, Yiying Tong, and Mathieu Desbrun. 2015a. Frame Field Generation through Metric Customization. *ACM Trans. Graph.* 34, 4, Article 40 (2015), 11 pages.

Felix Kälberer, Matthias Nieser, and Konrad Polthier. 2007. Quadcover-surface parameterization using branched coverings. In *Computer graphics forum*, Vol. 26. Wiley Online Library, 375–384.

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

Riccardo La Magna, Frédéric Waimer, and Jan Knippers. 2012. Nature-inspired generation scheme for shell structures. *Proceedings of the International Symposium of the IASS-APCS Symposium, Seoul, South Korea, 2012* (2012).

John M. Lee. 2013. *Introduction to smooth manifolds* (2nd ed. ed.). Springer, New York.

Sven Leyffer and Ashutosh Mahajan. 2010. Nonlinear constrained optimization: methods and software. *Argonee National Laboratory, Argonne, Illinois* 60439 (2010).

Jian-Min Li. 2017. *Timber shell structures: form-finding and structural analysis of actively bent grid shells and segmental plate shells.* Stuttgart: Institute for Structural Design and Structural Design, University of Stuttgart.

Jian-Min Li and Jan Knippers. 2015. Segmental Timber Plate Shell for the Landesgartenschau Exhibition Hall in Schwäbisch Gmünd—the Application of Finger Joints in Plate Structures. *International Journal of Space Structures* 30, 2 (2015), 123–139.

Yufei Li, Yang Liu, and Wenping Wang. 2014. Planar hexagonal meshing for architecture. *IEEE transactions on visualization and computer graphics* 21, 1 (2014), 95–106.

Richard J Lisle. 2003. Dupin's indicatrix: a tool for quantifying periclinal folds on maps. *Geological magazine* 140, 6 (2003), 721–726.

Hao-Yu Liu, Zhong-Yuan Liu, Zheng-Yu Zhao, Ligang Liu, and Xiao-Ming Fu. 2020. Practical Fabrication of Discrete Chebyshev Nets. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 13–26.

Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. 2006. Geometric Modeling with Conical Meshes and Developable Surfaces. In *ACM SIGGRAPH 2006 Papers*. 681–689.

Yang Liu, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. 2011. General Planar Quadrilateral Mesh Design Using Conjugate Direction Field. *ACM Trans. Graph.* 30, 6 (2011), 1–10.

Max Lyon, Marcel Campen, David Bommes, and Leif Kobbelt. 2019. Parametrization quantization with free boundaries for trimmed quad meshing. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Romain Mesnil, Cyril Douthe, Olivier Baverel, and Bruno Léger. 2017. Generalised cyclidic nets for shape modelling in architecture. *International Journal of Architectural Computing* 15, 2 (2017), 148–168.

Christian Müller. 2011. Conformal hexagonal meshes. *Geometriae dedicata* 154, 1 (2011), 27–46.

Ashish Myles, Nico Pietroni, and Denis Zorin. 2014. Robust Field-Aligned Global Parametrization. *ACM Trans. Graph.* 33, 4, Article 135 (2014), 14 pages.

Matthias Nieser, Jonathan Palacios, Konrad Polthier, and Eugene Zhang. 2011. Hexagonal global parameterization of arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics* 18, 6 (2011), 865–878.

Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. 2014. Frame Fields: Anisotropic and Non-Orthogonal Cross Fields. *ACM Trans. Graph.* 33, 4, Article 134 (July 2014), 11 pages.

Konrad Polthier and Eike Preuß. 2003. Identifying Vector Field Singularities Using a Discrete Hodge Decomposition. In *Visualization and Mathematics III*, Hans-Christian Hege and Konrad Polthier (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 113–134.

Helmut Pottmann. 2007. *Architectural geometry*. Vol. 10. Bentley Institute Press.

Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner. 2015. Architectural geometry. *Computers & Graphics* 47 (2015), 145 – 164.

Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. 2006. Periodic Global Parameterization. *ACM Trans. Graph.* 25, 4 (2006), 1460–1485.

Fernando Romero and Armando Ramos. 2013. Bridging a Culture: The Design of Museo Soumaya. *Architectural Design* 83, 2 (2013), 66–69.

Andrew O Sageman-Furnas, Albert Chern, Mirela Ben-Chen, and Amir Vaxman. 2019. Chebyshev nets from commuting PolyVector fields. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–16.

Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. 2013. Locally injective mappings. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 125–135.

Alla Sheffer, Emil Praun, and Kenneth Rose. 2006. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision* 2, 2 (2006), 105–171.

Daniel Sonntag, Lotte Aldinger, Simon Bechert, Martin Alvarez, Abel Groenewolt, Oliver David Krieg, Hans Jakob Wagner, Jan Knippers, and Achim Menges. 2019. Lightweight segmented timber shell for the Bundesgartenschau 2019 in Heilbronn. *Proceedings of IASS Annual Symposia* 26 (2019), 1–8.

Chengcheng Tang, Xiang Sun, Alexandra Gomes, Johannes Wallner, and Helmut Pottmann. 2014. Form-Finding with Polyhedral Meshes Made Simple. *ACM Trans. Graph.* 33, 4, Article 70 (2014), 9 pages.

Xavier Tellier, Cyril Douthe, Laurent Hauswirth, and Olivier Baverel. 2020a. Caravel meshes: A new geometrical strategy to rationalize curved envelopes. *Structures* 28 (2020), 1210 – 1228.

Xavier Tellier, Sonia Zerhouni, Guillaume Jami, Alexandre Le Pavec, Thibault Lenart, Mathieu Lerouge, Nicolas Leduc, Cyril Douthe, Laurent Hauswirth, and Olivier Baverel. 2020b. The Caravel heX-Mesh pavilion, illustration of a new strategy for gridshell rationalization. *SN Applied Sciences* 2 (2020), 1–12.

Christian Troche. 2008. Planar hexagonal meshes by tangent plane intersection. *Advances in architectural geometry* 1, 57-60 (2008), 2.

C Troche. 2009. Planar hexagonal tesselation of freeform surfaces—manufacturing solutions and design potentials. In *Proc. Design Modelling Symposium Berlin.* 331–9.

Amir Vaxman. 2021. Directional Technical Reports: Seamless Integration. https://doi.org/10.17605/OSF.IO/S5ACK

Amir Vaxman et al. 2017. libhedra: geometric processing and optimization of polygonal meshes. https://github.com/avaxman/libhedra.

Amir Vaxman et al. 2019. *Directional: A library for Directional Field Synthesis, Design, and Processing.*

Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional field synthesis, design, and processing. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 545–572.

Ryan Viertel and Braxton Osting. 2019. An approach to quad meshing based on harmonic cross-valued maps and the Ginzburg–Landau theory. *SIAM Journal on Scientific Computing* 41, 1 (2019), A452–A479.

Hans Jakob Wagner, Martin Alvarez, Abel Groenewolt, and Achim Menges. 2020. Towards digital automation flexibility in large-scale timber construction: integrative robotic prefabrication and co-design of the BUGA Wood Pavilion. *Construction Robotics* (2020), 1–18.

Wenping Wang and Yang Liu. 2009. A note on planar hexagonal meshes. In *Nonlinear Computational Geometry*. Springer, 221–233.

Wenping Wang and Yang Liu. 2010. A Note on Planar Hexagonal Meshes. In *Nonlinear Computational Geometry*, Ioannis Z. Emiris, Frank Sottile, and Thorsten Theobald (Eds.). Springer New York, New York, NY, 221–233.

W Wang, Y Liu, D Yan, B Chan, R Ling, and F Sun. 2008. Hexagonal meshes with planar faces (TR-2008-13).

Ron Wein, Eric Berberich, Efi Fogel, Dan Halperin, Michael Hemmer, Oren Salzman, and Baruch Zukerman. 2020. 2D Arrangements. In *CGAL User and Reference Manual* (5.2 ed.). CGAL Editorial Board. https://doc.cgal.org/5.2/Manual/packages.html#PkgArrangementOnSurface2

Mirko Zadravec, Alexander Schiftner, and Johannes Wallner. 2010. Designing Quad-dominant Meshes with Planar Faces. *Computer Graphics Forum* 29, 5 (2010), 1671–1679.

Henrik Zimmer, Marcel Campen, Ralf Herkrath, and Leif Kobbelt. 2012. Variational tangent plane intersection for planar polygonal meshing. In *AAG*. 319–332.

A PROOF OF LEMMA 4.1

LEMMA 4.1. *If U, V are LICO and the constraint (C3) is fulfilled, then one of the pushed forward grid directions in the face t_i will be parallel to d_i.*

PROOF. Let $(dr_i^{-1}(d_i))^{n_i/2} = a + \mathrm{i}b$. Since the constraint holds, $b = 0$, and since U, V are LICO, $a \neq 0$. Thus, $(dr_i^{-1}(d_i))^{n_i} \in \mathbb{R}_{>0}$, and $\exists \alpha \in \mathbb{R}_{>0}$ s.t. $(\alpha dr_i^{-1}(d_i))^{n_i} = 1$.

Let $\omega_0, \omega_1, ..., \omega_{n_i-1}$ be the n_i-th unit roots, namely $\omega_l^{n_i} = 1, l \in [0, ..., n_i - 1]$. Hence, $\exists \alpha \in \mathbb{R}_{>0}$ s.t. $(\alpha dr_i^{-1}(d_i))^{n_i} = \omega_l^{n_i}$. Therefore, $\exists l \in [0, .., n_i - 1], \alpha \in \mathbb{R}_{>0}$ s.t. $dr_i(\omega_l) = \alpha d_i$. On the other hand, by definition, the pushed-forward grid in the face t_i is parallel to $dr_i(\tilde{\omega}_{\tilde{l}}), \tilde{l} \in [0, ..., N-1]$, where $\tilde{\omega}_{\tilde{l}}$ are the N-th unit roots. Since n_i is a divisor of N, $\{\omega_l\} \subseteq \{\tilde{\omega}_{\tilde{l}}\}$, which completes the proof. □

B INTEGRATION AND MESH GENERATION

For integration and meshing we use standard practice and existing software that we do not claim as a contribution; however we discuss here specific consequences to our configuration choices.

B.1 Integration with seams in $\mathbb{Z}$

From the CPFs U_i, V_i we first compute the gradients $(\nabla u)_i, (\nabla v)_i$ for every face $t_i \in \mathcal{F}$ using Equation (2). Let $h_{1,2,3}$ denote the pullback of the triangular grid coordinates to M. The linear relation between the Cartesian and triangular grid carries over to the gradients, and we use it to compute $F_{1,2,3}$, the candidate gradient fields of the triangular grid functions.

We generate the mesh by integrating the field $F_{1,2,3}$ into the seamless multivariate function $h_{1,2,3}$ defined on the vertices on the mesh M. We use the new seamless integration package in Directional [Vaxman 2021; Vaxman et al. 2019]. This guarantees that the solution is order-preserving and faithful to our original field as much as possible, as our field is near-integrable (with quantized rotational jumps). The algorithm first "combs" $F_{1,2,3}$ according to our prescribed matching (the integer jumps on interior edges), cutting the mesh into a topological disc with the singularities on the boundaries, and

creating an order-preserving function on the vertices of the cut mesh. The function is seamless across these cuts, which means there exist permutations and integer translational jumps, where the permutations correspond to the matching of the field and the singularities indices. We further prescribe that $h_1 - h_2 + h_3 = 0$ to get a consistent triangle mesh, which matches the symmetry of the field F.

We choose a configuration where we round the seam translational jumps to integers, rather than the singularity coordinates, to relax the constraints on our otherwise very-constrained field design; that means that a singularity coordinate can fall on any point in $\frac{1}{2}\mathbb{Z}^2$ [Nieser et al. 2011]. The practical implication is that the resulting primal mesh M_R can be only triangle-dominant rather than fully triangular. To mitigate that we triangulate any resulting quadrilateral faces by adding the Delaunay diagonal.

B.2 Mesh extraction

To get M_R from the parameterization functions $h_{1,2,3}$, we trace their isolines on the original mesh and enumerate their intersections and faces; this is effectively and exactly done using the CGAL arrangement package [Wein et al. 2020] also as a part of the seamless integration package.

Hex-dominant dualization. We dualize the mesh M_R into the barycentric hex-dominant mesh M_D by inserting the barycenteric dual vertex in each interior face. Special measures are taken for the dualization of boundary faces, to ensure that the output mesh is consistent. We first remove one strip of boundary faces from the primal mesh and then another strip of faces from the dual mesh. In doing so, we avoid boundary faces with very small area, which may lead to instabilities during planarization.

C PLANARIZATION

The dual meshes we obtain are only approximately planar, with the planarity error concentrated in non-elliptic regions. However, the meshes are *planarizable* using a standard approach inspired by Jiang et al. [2015b], and planarization is necessary to generate brick and bowtie concave hexagonal shapes. We optimize for face planarity (29) under regularization with several objectives.

Surface distance. To prevent the vertices from drifting from the original surface M, we minimize the distance $\|\hat{v}_i - v_i\|$ between the vertices v_i and their closest projections $\hat{v}_i$ on M.

Symmetry. To preserve the symmetry of the even-degree faces, we consider the vectors connecting the barycenter c_f with every $v_i \in f$. We then require each pair of opposite vectors to be as equal as possible. In detail, we consider the vectors $p_i = c_f - v_i$ and $q_i = c_f - v_{(i+d/2)}, i \in [1, .., d/2]$, where d is the face degree. Finally, f is symmetric when $p_i = -q_i, \forall i \in [1, .., d/2]$.

Tangential drift. To prevent normal drift, we favor vertex movement in the tangent plane, computed at vertex locations in the previous iteration. We minimize $\langle n_{\tilde{v}_i}, (v_i - \tilde{v}_i) \rangle^2$, where $\tilde{v}_i$ is the previous value of v_i, and $n_{\tilde{v}_i}$ is the average face normal around $\tilde{v}_i$.

Edge and diagonal length. We prevent the degenration of edges and diagonals using the geometric barrier in Equation (14) applied to the edges' and diagonals' length.

Quad fairness. For quad meshes we add an extra fairness term, the squared-norm of the difference between a vertex and its averaged neighborhood $\|v_i - (w_1 + \cdots + w_n)/n\|^2$, where $w_1, \ldots, w_n$ are the n nearest neighbors of v_i. For more details see Tang et al. [2014].

We provide the planarization formulation and optimization in the supplemental material. Meshes before and after planarization are shown in Figures 10, 14, 17 and 18.