

# Hexagonal Global Parameterization of Arbitrary Surfaces

Matthias Nieser, Jonathan Palacios, Konrad Polthier, and Eugene Zhang



**Abstract**—We introduce *hexagonal global parameterizations*, a new type of surface parameterizations in which parameter lines respect six-fold rotational symmetries (6-RoSy). Such parameterizations enable the tiling of surfaces with nearly regular hexagonal or triangular patterns, and can be used for triangular remeshing.

To construct a hexagonal parameterization on a surface, we provide an automatic algorithm to generate a 6-RoSy field that respects directional and singularity features of the surface. This field is then used to direct a hexagonal global parameterization. The framework, called HEXCOVER, extends the QUADCOVER algorithm and formulates necessary conditions for hexagonal parameterization.

We demonstrate the usefulness of our geometry-aware global parameterization with applications such as surface tiling with nearly regular textures and geometry patterns, as well as triangular and hexagonal remeshing.

**Index Terms**—Surface parameterization, rotational symmetry, hexagonal tiling, triangular remeshing, pattern synthesis on surfaces, texture synthesis, geometry synthesis, regular patterns.

## 1 INTRODUCTION

IN this article we introduce *hexagonal global parameterizations*, a new type of global parameterizations that map a surface into the two-dimensional plane while respecting the symmetries of a regular hexagonal grid. A hexagonal global parameterization is equivalent to a texture map for tileable texture images with additional six-fold rotational symmetries, e.g. a regular triangular or hexagonal pattern. Such a parameterization is ideal for tiling surfaces with hexagonal and triangular patterns in texture and geometry synthesis, as well as for triangular remeshing.

**Tiling.** Regular hexagonal patterns are one of the three regular patterns that can seamlessly tile a plane. They provide an optimal approximation to circle packings [1] which have been linked to the wide appearances of hexagonal

patterns in nature, such as honeycombs, insect eyes, fish eggs, and snow and water crystals, as well as in man-made objects such as floor tiling, carpet patterns, and architectural decorations (Figure 1).

Tiling a surface with regular texture and geometry patterns is an important yet challenging problem in pattern synthesis [2], [1]. Methods based on some local parameterization of the surface often lead to visible breakup of the patterns along seams, i.e., where the surface is cut open during parameterization. Global parameterizations can alleviate this problem when the translational and rotational discontinuity in the parameterization is compatible with the tiling pattern in the input texture and geometry. For example, a quadrangular global parameterization is designed to be compatible with square patterns (Figure 2 (a)). On the other hand, it is incompatible with hexagonal patterns (Figure 2 (b)). In contrast, a hexagonal global parameterization is compatible with hexagonal or triangular patterns (Figure 2 (c)). The hexagonal global parameterizations we introduce in this article are designed to facilitate the tiling of surfaces with hexagonal texture and geometry patterns.

**Remeshing.** Another motivation of our work is triangular remeshing, which refers to generating a triangular mesh from an input triangular mesh to improve its quality. (Note that triangular and hexagonal meshes are dual to each other, and triangular remeshing can also be used to perform hexagonal remeshing.) In triangular remeshing, it is often desirable to have all the triangles in the mesh being nearly equilateral and of uniform sizes, and the edges following the intrinsic curvature and feature directions in the surface. In addition, special treatment is needed for irregular vertices (whose valence is not equal to six) since they impact the overall appearance and quality of the mesh.

A hexagonal parameterization transforms these challenges to that of control over the singularities in the parameterization as well as the spacing and direction of parameter lines. Smooth parameter lines and a reduced number of singularities leads to highly regular meshes. Such meshes are desirable for subdivision surface applications [3].

**Rotational Symmetry.** Inspired by recent developments in constructing a quadrangular global parameterization [4], [5], [6], we construct a global parameterization given a 6-way rotational symmetry field, or 6-RoSy field, an abbreviation introduced in [7]. An  $N$ -RoSy refers to a

M. Nieser is with the Institute of Mathematics, Freie Universität Berlin, Arnimallee 6, D-14195 Berlin, Germany. Email: matthias.nieser@fu-berlin.de.

J. Palacios is with the School of Electrical Engineering and Computer Science, Oregon State University, 1148 Kelley Engineering Center, Corvallis, OR 97331. Email: palacijo@eecs.oregonstate.edu.

K. Polthier is MATHEON-Professor with the Institute of Mathematics, Freie Universität Berlin, Arnimallee 6, D-14195 Berlin, Germany. Email: konrad.polthier@fu-berlin.de.

E. Zhang is an Associate Professor with the School of Electrical Engineering and Computer Science, Oregon State University, 2111 Kelley Engineering Center, Corvallis, OR 97331. Email: zhang@eecs.oregonstate.edu.

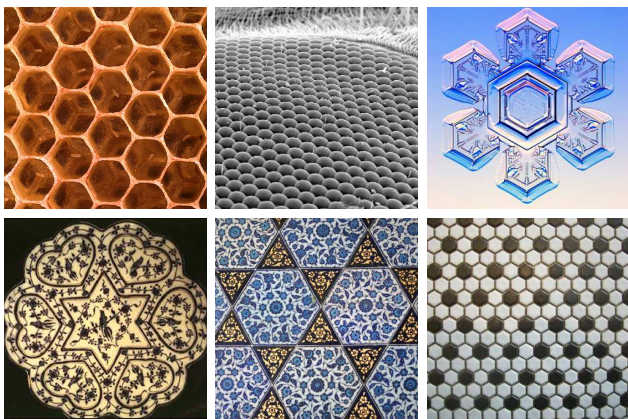


Fig. 1. Hexagonal patterns in nature (top): honeycombs, insect eyes, snowflakes. Appearance in design (bottom): star of David, Islamic pattern, floor tiling.

set of  $N$  vectors with evenly-spaced angles, and a 1-, 2-, and 4-RoSy can represent a vector, a line segment, or a cross, respectively. While a 1-, 2-, and 4-RoSy field can each be used to compute a quadrangular parameterization, a 4-RoSy field provides the most flexibility in terms of modeling branch points, and thus the types of irregular vertices in a quad mesh. Specifically, a 1- or 2-RoSy field can always be converted into a 4-RoSy field with the unfortunate constraint that a first-order singularity in the 1- or 2-RoSy field becomes a higher-order singularity in the resulting 4-RoSy field. Consequently, when performing quadrangular remeshing with a 1- or 2-RoSy field, it is in general impossible to obtain a valence three or five vertex.

Similarly, while 1-, 2-, 3-, and 6-RoSy fields can all be used for triangular remeshing, only 6-RoSy fields can be used to model fundamental irregular vertices that have a valence of either five or seven.

**Parameterization.** Automatic generation of a hexagonal parameterization from an input surface faces a number of challenges. First, unlike quadrangular parameterizations whose parameter lines are parallel to either the major or the minor principal curvature directions, in hexagonal parameterizations only one of the two directions can be used at each point on the surface. One must decide which direction to choose, and how to propagate such choices from a relatively small set of points to the whole surface to maintain the smoothness of the resulting parameterization. Second, existing techniques to explicitly control the singularities in a parameterization are user-driven, and it is not an easy task to provide automatic control over the number and location of such singularities. Third, the continuity conditions developed for quadrangular parameterizations are not appropriate for hexagonal parameterizations.

**Pipeline.** To address these challenges, we present a two-step pipeline to generate a geometry-aware hexagonal global parameterization. First, we automatically select the most appropriate principal direction with which we align our 6-RoSy field. The singularities in the field are related

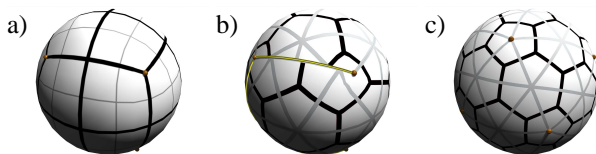


Fig. 2. A quadrangular parameterization ensures that the discontinuity along the cut is invisible (a). The same parameterization is incompatible with a hexagonal pattern (b), which leads to seams (yellow). In this case a hexagonal parameterization is needed (c).

to regions of high Gaussian curvature regions. Moreover, we introduce an automatic singularity clustering algorithm that allows nearby singularities to be either canceled or merged into a higher-order singularity, thus reducing the total number of singularities in the field.

In the second step of the pipeline, we generate a global parameterization which is aligned to the 6-RoSy field as well as possible. The QUADCOVER algorithm [5] is adapted for handling the symmetries of a hexagonal parameterization. We minimize a quadratic energy which measures the  $L^2$  distance of the parameter lines to the field. During minimization, some variables are constrained to an integer grid. We point out that in the hexagonal parameterization this grid is the set of *Eisenstein integers*, which is different from the *Gauss integers* used in the quadrangular case. This leads to a parameterization method that we refer to as HEXCOVER. The resulting parameters can then be used to generate triangular meshes free of T-junctions as well to tile a surface seamless with a hexagonal pattern.

**Contributions.** In summary, our contributions in this article are as follows:

- 1) We introduce hexagonal global parameterization and demonstrate its uses with applications such as triangular remeshing and pattern synthesis on surfaces. For remeshing we point out the need for a geometry-aware 6-RoSy field when generating a hexagonal global parameterization.
- 2) We present the first technique to construct a hexagonal global parameterization given an input surface with a guidance 6-RoSy field. We formulate the energy term as well as the continuity condition for hexagonal global parameterizations.
- 3) We propose an automated pipeline for generating geometry-aware 6-RoSy fields. As part of the pipeline, we point out how to align the field with principal curvature directions as well as develop a way of automatically clustering singularities.

The remainder of this article is organized as follows. We first cover work in relevant research areas in Section 2. Next, we describe our pipeline for generating a geometry-aware 6-RoSy field given an input surface in Section 3, and our parameterization technique in Section 4. In Section 5, we demonstrate the usefulness of our techniques with

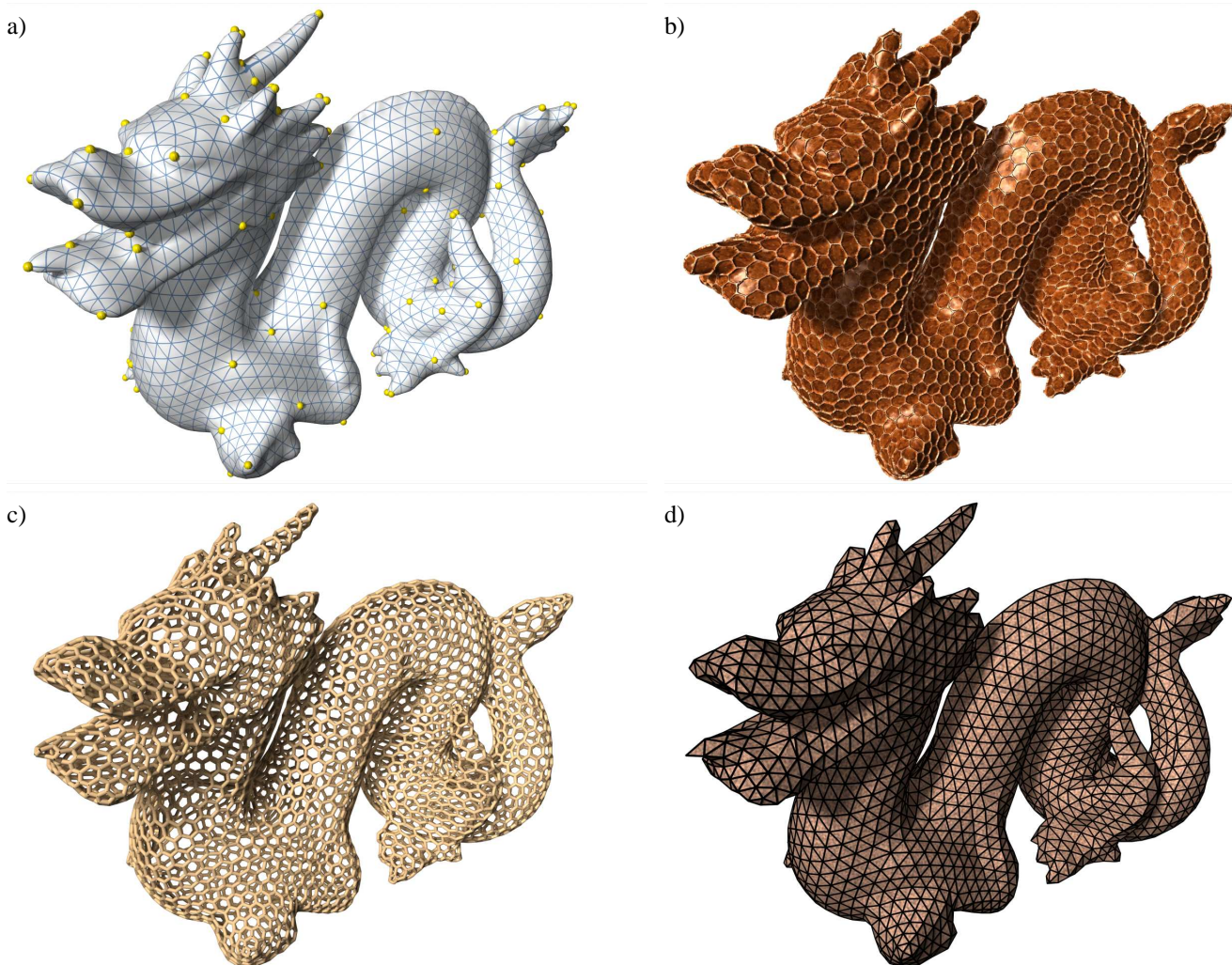


Fig. 3. Hexagonal global parameterization (a), used for regular texture (b) and geometry pattern synthesis (c) with hexagonal patterns and for geometry-aware triangular remeshing (d).

applications in triangular remeshing and surface tiling with regular texture and geometry patterns. Lastly, we close in Section 6 with future work.

## 2 RELATED WORK

**Surface Parameterization.** Surface parameterization is a well-explored research area. We will not attempt a complete review of the literature but instead refer the reader to surveys by Floater and Hormann [8] and Hormann et al. [9].

Early global parameterization methods focus on conformal parameterizations [10], [11], [12], which are aimed at angle preservation at the cost of length distortion. To reduce length distortion, Kharevych et al. [13] use cone singularities for conformal parameterization, which relax the constraint of a flat domain at few isolated points. Singularities have proven essential for high quality parameterizations and have been used in other parameterization schemes as well.

Dong et al. [14] perform quadrangulation based on harmonics functions. Later, Dong et al. [15] use a similar

idea for parameterization but create the quadrilateral meta layout automatically from the Morse-Smale complex of Eigenfunctions of the mesh Laplacian.

Tong et al. [16] use singularities at the vertices of a hand-picked quadrilateral meta layout on the surface. The patches of the meta layout are then parameterized by solving for a global harmonic one-form. Ray et al. [4] parameterize surfaces of arbitrary genus with periodic potential functions guided by two orthogonal input vector fields, or a 4-RoSy field. This leads to a continuous parameterization except in the vicinity of singularities on the surface. These singular regions are detected and reparameterized afterwards.

The QUADCOVER algorithm [5] builds upon this idea by using the input 4-RoSy field to generate a global parameterization, based on a quadratic energy formulation. Also the notion of covering spaces is used to describe a 4-RoSy field as vector field and to provide a clear theoretical setting. Our algorithm to generate a parameterization from a 6-RoSy field is an adaptation of the QuadCover method.



Bommes et al. [6] propose a method similar to the aforementioned techniques based on the same energy formulation as in [5], but provides several advancements. Beside a robust generation of 4-Rosy fields, they propose to use a mixed-integer-solver for improving the rounding of integer variables. They also add constraints that force parameter lines to capture sharp edges.

**Field Processing.** Much work has been done on the subject of vector (1-RoSy) and tensor (2-RoSy) field analysis. Note that a line field is equivalent to a symmetric tensor field [17]. To review all of this work is beyond the scope of this article; here we refer to only the most relevant work. Helman and Hesselink [18] propose a method of vector field visualization based on topological analysis and provide methods of extracting vector field singularities and separatrices. Topological analysis techniques for symmetric second-order tensor fields are later introduced in [19]. In the context of vector field design, numerous systems have been developed for the purpose of vector field, most of which have been for specific graphics applications such as texture synthesis [20], [21], [22], fluid simulation [23], and vector field visualization [24]. Fisher et al. [25] propose a vector field design system based on discrete one-forms. Note that the above systems do not employ any methods of topological analysis, and do not extract singularities and separatrices. Systems providing topological analysis include [26], [27] and [28]. The last has also been extended to design tensor fields [17], [29]. In contrast, relatively little work has been done on  $N$ -RoSy fields when  $N > 2$ . Hertzmann and Zorin [30] utilize *cross* or 4-RoSy fields in their work on non-photorealistic pen-and-ink sketching, and provide a method for smoothing such fields. Ray et al. [31] extend the surface vector field representation proposed in [27] into a design system for  $N$ -RoSy fields of arbitrary  $N$ . Palacios and Zhang [7] propose an  $N$ -RoSy design system that allows initialization using design elements as well as topological editing of existing fields. They also provide analysis techniques for the purpose of locating both singularities and separatrices, and a visualization technique in [32]. Lai et al. [33] propose a design method based on a Riemannian metric, that gives the user control over the number and locations of singularities. Their system also allows for mixed  $N$ -RoSy fields, with different values of  $N$  in different regions of the mesh. However, this method is based on user design while we focus on automatic and geometry-aware generation. Bommes et al. [6] offer a method of producing a smooth 4-RoSy field from sparse constraints, formulated as a mixed-integer problem. Zhang et al. introduce a quadrangulation method based on the notion of waves. Their method can also be used to generate 4-RoSy fields [34]. Krane et al. [35] handle cone singularities by using the notion of *trivial connection* in the surface. These singularities include those seen in 6-RoSy fields.

Ray et al. [36] propose a framework to generate an  $N$ -RoSy field that follows the natural directions in the surface and has a reduced number of singularities which tends to fall into natural locations. In this article, we make use of this

framework but automatically generate the input constraints, which relieves the user from labor-intensive manual design. Furthermore, we introduce to our knowledge the first automatic singularity clustering algorithm to reduce the number of singularities in the field.

### 3 GEOMETRY-AWARE 6-ROSY FIELD GENERATION

In this section, we describe our pipeline for generating a geometry-aware 6-RoSy field  $F$  given an input surface  $S$ . This field will then be used to guide the parameterization stage of our algorithm (Section 4).

We first review some relevant properties of 6-RoSy fields [7], [31]. An  $N$ -RoSy field  $F$  has a set of  $N$  directions at each point  $\mathbf{p}$  in the domain of the field:  $F(\mathbf{p}) = \{R_N^i v(\mathbf{p})\}$ ,  $i \in \{0, \dots, N-1\}$ . where the vector  $v(\mathbf{p}) = \rho(\mathbf{p})(\cos \theta(\mathbf{p}), \sin \theta(\mathbf{p}))^T$  is one of the  $N$  directions, and  $R_N^i$  is the linear operator that rotates a given vector by  $\frac{2i\pi}{N}$  in the corresponding tangent plane. A *singularity* is a point  $\mathbf{p}_0$  such that  $\rho(\mathbf{p}_0) = 0$  and  $\theta(\mathbf{p}_0)$  is undefined;  $\mathbf{p}_0$  is *isolated* if the value of  $\rho \neq 0$  for all points in a sufficiently small neighborhood of  $\mathbf{p}_0$ , except at  $\mathbf{p}_0$ . An isolated  $N$ -RoSy singularity can be measured by its *index*, which is defined in terms of the Gauss map [7] and has an index of  $\frac{I}{N}$ , where  $I \in \mathbb{Z}$ . A singularity  $\mathbf{p}_0$  is of *first-order* if  $I = \pm 1$ . When  $|I| > 1$ ,  $\mathbf{p}_0$  is referred to as a *higher-order* singularity. A higher-order singularity with an index of  $\frac{I}{N}$  can be realized by merging  $I$  first-order singularities.

**Requirements and Pipeline.** There are a number of goals that we wish to achieve with our automatic field generation.

First, we wish to control the number, location, and type of the singularities in the field. When performing quadrangular and triangular remeshing using global parameterizations, the singularities in the guiding 4- or 6-RoSy field correspond to the irregular vertices in the mesh. Such singularities can also lead to the breakup of texture and geometry patterns during pattern synthesis on surfaces. Consequently, the ability to control the number, location, and type of singularities in the field can improve quality of the remeshes and surface tilings.

Second, the field needs to be smooth, or distortion can occur in the resulting parameterizations that have undesirable effects for triangular remeshing and surface tiling.

Third, we need the parameter lines in the parameterization to be aligned with the feature lines in the surfaces, such as ridge and valley lines (see Figure 4). In addition, it has been documented that having texture directions aligned with the feature lines in the mesh can improve the visual perception of the texture synthesis [37].

Note that these requirements may conflict with each other. For example, excessive reduction of singularities can lead to high distortion in the field, and an overly-smoothed field may deviate from feature lines. To deal with this we adopt

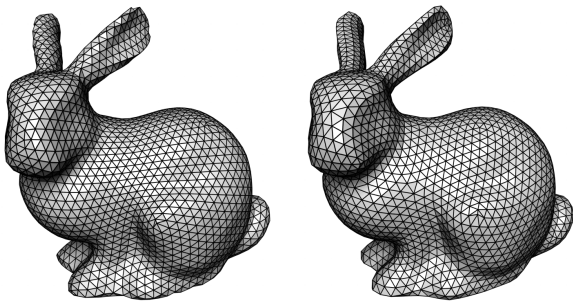


Fig. 4. For remeshing, edges should follow principal curvature directions (right). Edges ignoring surface features (right) cause "twisting" artifacts (on the ears).

the framework of Ray et al. [36]. In their framework, a set of user-specified constraints and a modified Gaussian curvature  $\bar{K}$  defined at the vertices are used to generate a sparse linear system whose solution (after several iterations) is the RoSy field that matches the constraints and  $\bar{K}$  in the least square sense. Each constraint represents a desired  $N$ -RoSy value, i.e.,  $N$  directions, at a given point. In our case we wish to have our field aligned with principal curvature directions. The user-specified  $\bar{K}$  is a vertex-based function defined on the mesh, whose value at a vertex represents the desired discrete Gauss curvature at this vertex to be reflected by resulting field curvature. The integral of  $\bar{K}$  over  $S$  must be equal to  $2\pi\chi(S)$  where  $\chi(S)$  is the *Euler characteristic* of the surface  $S$ . It allows the user to specify the location and type of singularities in the field. For example, a vertex with a  $\bar{K}$  value of  $\frac{2k\pi}{N}$  should have a singularity of index  $\frac{k}{N}$  in the resulting field. We would like to note that other field generations systems that allow directional constraints the specification of singularities of index greater than  $\frac{1}{N}$  can also be used (such as the one described in [35]). We use the method of Ray et al. because it gives additional control over the initial number singularities if desired.

Given a surface with complex geometry and topology, it can be labor intensive to provide all necessary constraints through a lengthy trial-and-error process. Consequently, we automatically generate the directional constraints as well as  $\bar{K}$ , which is at the core of our algorithm for field generation. Our algorithm consists of two stages. First, we identify a set of directional constraints based on the curvature and solve for an initial 6-RoSy field using these constraints only. Second, we extract all the singularities in the initial field and perform iterative singularity pair clustering until the distance between any singularity pair is above a given threshold. The remaining singularities will be used to generate new values for the vertex function  $\bar{K}$ , which will be used to generate the final RoSy field with reduced singularities. We describe each of these stages in more detail next.

**Automatic Constraint Identification.** To automatically identify directional constraints, we need to answer the questions of where to place constraints and what direction is assigned to each constraint.

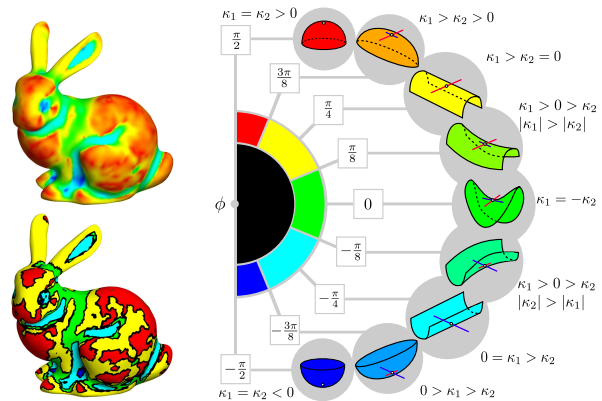


Fig. 5. Surface classification scheme to determine directional constraints.  $\phi \in [-\pi/2, \pi/2]$  is color mapped to the [BLUE, RED] arc in HSV color space: Left top: continuous mapping. Bottom: binned classification. The legend (right) shows surfaces patches which are locally similar to points with given values.

Recall that we wish to align the parameter lines with feature lines such as ridges and valleys, i.e., the principal direction in which the least bending occurs. Note that the directions in the 6-RoSy field is the gradient of the parametrization (Section 4). Consequently, we will choose the principal direction that has the most bending, i.e., maximum absolute principal curvature, as one of the directions in the 6-RoSy. We estimate the curvature tensor of the mesh using the method of Meyer et al. [38].

Principal curvature directions are most meaningful in cylindrical and hyperbolic regions due to the strong anisotropy there. However, while purely hyperbolic regions possess strong anisotropy, the absolute principal curvatures are nearly indistinguishable, thus making both principal curvature directions candidates. Moreover, the two bisectors between the major and minor principal curvature directions can also provide viable choices for the edge directions in hyperbolic regions. Due to the excessive choices of directions in hyperbolic regions and insufficient choice of directions in planar and spherical regions, we only generate directional constraints in cylindrical regions. Note that using the asymptotic directions could result in neighboring triangles being constrained with directions that differ by rotations of  $\frac{\pi}{2}$ ; while this causes no problems in 4-RoSy field generation, such constraints conflict in the case of 6-RoSy field generation.

We make use of a representation of the curvature tensor that readily exposes where on this spectrum of classification any point on a given surface falls. Using the trace-and-deviator decomposition similar to those employed in [39], the curvature tensor  $T$  at a point  $\mathbf{p} \in S$  can be rewritten as:

$$\begin{aligned} T &= \left( \frac{\kappa_1 - \kappa_2}{2} \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} + \frac{\kappa_1 + \kappa_2}{2} \text{Id} \right) \\ &= \frac{\rho}{\sqrt{2}} \left( \cos \phi \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix} + \sin \phi \cdot \text{Id} \right) \quad (1) \end{aligned}$$

where  $\kappa_1$  and  $\kappa_2$  are the principal curvatures at  $\mathbf{p}$ ,  $\rho = \sqrt{\kappa_1^2 + \kappa_2^2}$ ,  $\phi \in [-\pi/2, \pi/2] = \arctan(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2})$ ,  $\theta \in [0, \pi)$  is the angular component of the maximum principal direction measured in the local frame at  $\mathbf{p}$ , and  $\text{Id}$  denotes the identity map. Note that the first component in the sum is traceless and symmetric, while the second is a multiple of the identity matrix.  $T(\mathbf{p})$  can now be classified using  $(\rho(\mathbf{p}), \phi(\mathbf{p}))$ , which spans a half plane. There are six special configurations on this half plane, the first satisfying  $\rho(\mathbf{p}) = 0$ , i.e., the local geometry near  $\mathbf{p}$  is planar. For the remaining five configurations we have  $\rho(\mathbf{p}) > 0$ . Respectively, they correspond to  $\phi(\mathbf{p}) = \frac{\pi}{2}$  (spherical),  $\phi(\mathbf{p}) = \frac{\pi}{4}$  (cylindrical),  $\phi(\mathbf{p}) = 0$  (purely hyperbolic),  $\phi(\mathbf{p}) = -\frac{\pi}{4}$  (inverted cylindrical), and  $\phi(\mathbf{p}) = -\frac{\pi}{2}$  (inverted spherical). With this representation, we can classify any point  $\rho(\mathbf{p})$  as being planar if  $\phi(\mathbf{p})$  is smaller than a given threshold  $\delta$ , elliptical if  $\phi(\mathbf{p}) \geq \delta$  and  $|\phi(\mathbf{p})| > \frac{3\pi}{8}$ , hyperbolic if  $\phi(\mathbf{p}) \geq \delta$  and  $|\phi(\mathbf{p})| < \frac{\pi}{8}$ , and cylindrical otherwise, i.e.,  $\phi(\mathbf{p}) \geq \delta$  and  $\frac{\pi}{8} \leq |\phi(\mathbf{p})| \leq \frac{3\pi}{8}$ . We wish to point out the tensor-based decomposition is equivalent to the concept of *shape index* [40].

Given the classification, we propagate the directions in the cylindrical regions into non-cylindrical regions (planar, spherical, hyperbolic) using energy minimization, an approach taken in [6]. To accomplish this, we pick the points where  $\rho$  (the tensor magnitude) is above certain a threshold  $t_\rho$ , and label these points as having “strong” curvature (in all of our examples, we have chosen  $t_\rho$  so that 35 percent of the area of  $S$  is so-labeled). From this set of points, we use only the directions of the cylindrical points as constraints; that is, the points for which  $\phi \in [-3\pi/8, -\pi/8] \cup [\pi/8, 3\pi/8]$  (Figure 6). Finally, we select the maximum direction  $\theta$  as the constraint direction at points where  $\phi > 0$  and the minimum direction  $\theta + \pi/2$  where  $\phi < 0$ . Recall that the directions in the output field specify the gradients in our resulting parameterizations, and we wish one of the isolines of the parameters to be orthogonal to the direction in which the surface is bending the most. Clearly, the above directions satisfy this requirement (See the shapes on the right side of the right image in Figure 5). Finally, the constraints are used to set up a linear system [36] whose solution gives rise to our initial RoSy field.

For our solver, we use the geometry-aware  $N$ -RoSy field generation technique proposed by [36], as it allows us to control the level of geometric detail that is reflected by singularities, and also plays a role in the implementation of our singularity clustering technique. This system, based on discrete exterior calculus (DEC) [41], filters (locally averages) the Gauss Curvature  $K$  of  $S$  to produce  $\bar{K}$  and then computes a target field curvature  $C^t$  using the difference between  $K$  and  $\bar{K}$ .  $C^t$  is then used to modify the angles by which directions rotate when parallel transported along mesh edges. This compensates for the actual curvature of  $S$ , and directional fields computed on  $S$  under these conditions behave as though  $S$  has a Gauss curvature of  $\bar{K}$ . Since  $\bar{K}$  is smoother than  $K$ , such fields have reduced



Fig. 6. Selection of constraints. Left: Color mapping of  $\rho$ . Middle: Highest 35% of values; colors are based on  $\phi$  as in Figure 5. We use maximum curvature directions where  $\phi > 0$  (yellow) and minimum directions where  $\phi < 0$  (cyan) as being orthogonal to the direction in which the surface is bending the most (see close-up, right). Notice that chosen directions in nearby yellow and cyan regions agree as they would not if we had selected only one of the curvature directions everywhere.

topological noise, which makes them more suitable for our parameterization algorithm.

**Automatic Singularity Clustering.** Our initial field was obtained from directional constraints only. Consequently, it typically consists of only first-order singularities. Given a surface with rather complex geometry and topology, the number of singularities can be rather large. Furthermore, while the location of the singularities tend to be appropriate (in high curvature regions), many of them form dense clusters. Having singularities in closer proximity can lead to difficulties in the resulting parameterization. This is because the singularities will be constrained to be on a lattice in the parameter space as typically required by most global parameterization methods [5], [6]. Consequently, the smallest distance between any singularity pair will be mapped to a unit in the parameter space. If the smallest distance is too small, the two involved singularities may be mapped to the same point on the lattice, leading to a locally infinite stretching in the parameterization. Figure 15 illustrates this.

To address this, many field generation techniques constrain the number of singularities to be as few as possible [31], but this represents another extreme, where the field directions can become highly distorted in some regions. Furthermore, many of the aforementioned approaches require much user interaction [7], [31], [36], which can be time-consuming for models with complex geometry and topology.

Our goal is to automatically reduce the number of singularities in the field while retaining the locations of the remaining singularities inside high curvature regions. To achieve this we employ the following process.

First, we extract the singularities in the initial RoSy field (using the method described in [36]) which we use to build a graph embedded in the surface. The nodes of this graph are the singularities in the field, and the edges representing



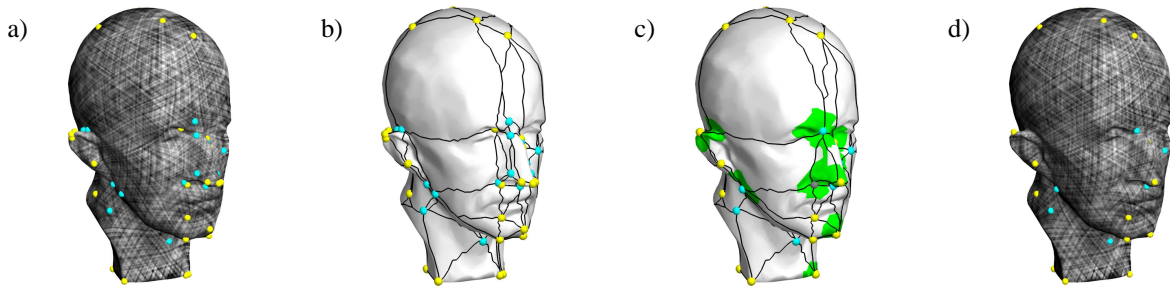


Fig. 7. Clustering pipeline: (a) Initial field. (b) Singularity graph  $G$ . (c) Reduced graph obtained by performing edge-collapses. The region  $R$  is shown in green. (d) Reduced field generated by resolving in  $R$  with singular constraints at the nodes of  $G$  and directional constraints at the boundary of  $R$ .

proximity information between singularity pairs. We refer to this graph as the *singularity graph*  $G$ . To construct  $G$ , we compute a Voronoi diagram with the singularities as sites. The dual graph gives rise to the singularity graph [42].

Second, we iteratively perform edge collapses on this graph, which is equivalent to performing singularity pair clustering (merging or cancellation), until the minimal surface distance between any singularity pair is above a given threshold. Every time a singularity pair is clustered, we compute the sum of the singularity indexes and place a singular constraint with the sum as its desired index. Note that we do this even if the sum is zero, i.e., singularity pair cancellation. The singularity constraint is placed on the path between the two original singularities, closer to the one with the Gaussian curvature of highest magnitude. This is an attempt to keep singularities near the features that caused them to originally appear during initialization and is accomplished by interpolating along the geodesic from  $\mathbf{p}_0$  to  $\mathbf{p}_1$  using the value  $|K(\mathbf{p}_1)|/(|K(\mathbf{p}_0)| + |K(\mathbf{p}_1)|)$ , where  $K(\mathbf{p})$  is the Gaussian curvature at  $\mathbf{p} \in S$ . We continue to collapse edges in the order of increasing edge-length on  $G$  until no edge of length less than  $d_{sing}$  remains. At the end of this process, we will have generated a set of singularity constraints, i.e., the remaining vertices in the graph, which is then used to update the field in the vicinity of these singularities. In the case of fields generated for remeshing,  $d_{sing}$  can be selected based on the edge-length of the output mesh. We choose  $d_{sing}$  to be  $0.1B$  where  $B$  is the size of the bounding box for the model. For a visual summary of the algorithm, see Figure 7.

Third, we modify  $\bar{K}$  based on the singularity constraints. Recall that the  $\bar{K}$  is simply a smoothed version of the discrete Gauss curvature during the generation of the initial field. The singularity constraints, produced in the previous step, consists of a set of vertices in the mesh and a desired singularity index  $t(\mathbf{p})$  for each such singularity constraint  $\mathbf{p}$ . We modify  $\bar{K}$  such that it is zero everywhere on the surface except at singularity constraints where the value of  $\bar{K}$  is  $\frac{2\pi}{N}t(\mathbf{p})$ . Notice that such assignment satisfies the constraint that the integral of  $\bar{K}$  over  $S$  is equal to  $2\pi\chi(S)$ . We now modify the 6-RoSy field by solving the same system used to generate the initial field, with one difference: we do not

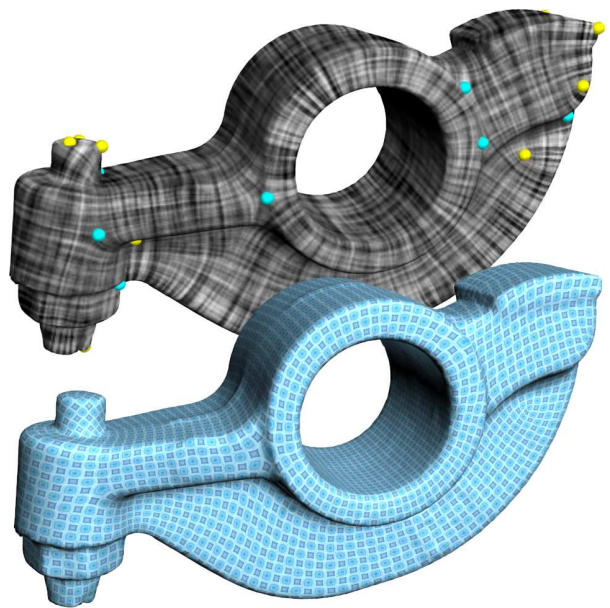


Fig. 8. Geometry-aware 4-RoS field and corresponding texture tiling.

resolve for the updated field everywhere on the surface. Instead, we generate a region  $R = \{\mathbf{p} | d(\mathbf{p}, V_{collapse}) < d_{sing}\}$ , where  $V_{collapse}$  is the set of vertices that were members of collapsed edges in  $G$ , and resolve for the field only in  $R$ . That is, the field values are fixed in the complement of  $R$  and the values on the boundary of  $R$  will serve as the boundary conditions when resolving; the original directional constraints are ignored in this step. In this way, we largely preserve the results of the field generated from the directional constraints, but force the merging and cancellation of singularities in the regions where large clusters had appeared before. The field values for vertices inside  $R$  are then updated. We have found this to be fast and efficient in controlling the location, type, and number of singularities.

We wish to point out that our automatic field generation method can be applied to  $N$ -RoSy field generation for any  $N$  that is even, in particular 4-RoS fields. Figure 8 shows an example generated using our method. The only change in the whole field generation pipeline occurs during automatic

identification of directional constraints. Instead of choosing  $\theta$  or  $\theta + \frac{\pi}{2}$  as one of the six directions for constraints, we choose both for the case of 4-RoSy fields.

#### 4 HEXCOVER PARAMETERIZATION

In this section we describe the second stage of our pipeline, which constructs a hexagonal global parameterization given an input triangular mesh surface along with a 6-RoSy field defined on it. We will first introduce the notion of hexagonal parameterizations before describing our HEXCOVER parameterization technique which is an extension of the QUADCOVER method for quad remeshing.

**Hexagonal Parameterization and Energy.** Given a triangular mesh surface  $S$  with  $|T|$  triangles, a *global parameterization*  $\varphi : S \rightarrow \mathbb{R}^2$  respecting an  $N$ -RoSy symmetry is a collection of linear maps  $\{\varphi_i | 1 \leq i \leq |T|\}$  where each  $\varphi_i : t_i \rightarrow \mathbb{R}^2, p \mapsto (u_i, v_i)^T$  maps triangle  $t_i \in S$  onto  $\mathbb{R}^2$  with the following property. For any pair of adjacent triangles  $t_i$  and  $t_j$  we have:

$$\varphi_j(\mathbf{p}) = R_N^{r_{ij}} \varphi_i(\mathbf{p}) + w_{ij}, \quad \forall \mathbf{p} \in t_i \cap t_j, \quad (2)$$

where  $r_{ij} \in \{0, 1, \dots, N-1\}$  and  $w_{ij} \in \mathbb{R}^2$  are the rotational and translational discontinuities, respectively. Recall that  $R_N^k$  is the linear operator that rotates a vector by  $\frac{2k\pi}{N}$  in its tangent plane (Section 3). The maps  $\varphi_i$  are restricted to be linear on each triangle. They are defined by their values at vertices, while  $r_{ij}$  and  $w_{ij}$  are defined on edges.

In quadrangular case where  $N = 4$ , the parameter lines can be visualized by treating  $\varphi^{-1}$  as the map that textures the surface with a 2D regular unit grid. To ensure the continuity in parameter lines, the translational discontinuities  $w_{ij}$  are required to be on the set of *Gauss integers*  $G_4 := \{(a, b)^T | a, b \in \mathbb{Z}\}$ .

A hexagonal parameterization  $N = 6$  can be considered similarly, except that in this case the texture image needs to respect hexagonal rotational symmetries. A canonical choice is a hexagonal or triangular pattern as shown in Figure 9 (left). The texture image has an aspect ratio of  $1 : \sqrt{3}$  and tiles the plane seamlessly. It is furthermore invariant under rotations of  $\frac{\pi}{3}$  around the center of each hexagon. The set of these center points is known as *Eisenstein integer lattice*, shown in Figure 9 (right):

$$G_6 := \left\{ a \begin{pmatrix} 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 1/2 \\ \sqrt{3}/2 \end{pmatrix} \mid a, b \in \mathbb{Z} \right\}. \quad (3)$$

Beside the rotational invariance, the hexagonal grid also remains invariant under translations by any vector in  $G_6$ . While a hexagonal parameterization is a discontinuous map, the discontinuities are not visible if all  $w_{ij}$  are in  $G_6$  because of the repeating structure of the texture image (Figure 2).

A hexagonal parameterization can be generated according to a guidance 6-RoSy field  $F$ . Given a point  $\mathbf{p}$ , the edges of the hexagons are aligned with the 6 vectors of  $F$  in  $\mathbf{p}$ . We generate a hexagonal parameterization which optimizes this

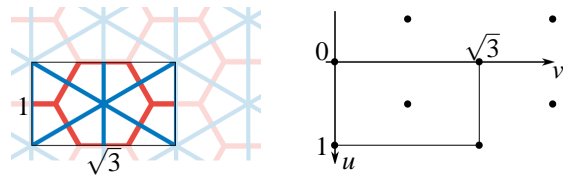


Fig. 9. Left: Texture with hexagonal rotational symmetries. Right: Eisenstein integer lattice  $G_6$ .

alignment in  $L^2$ -sense. Specially, we minimize the quadratic energy:

$$E(u, v) := \int_S (\|\nabla u - F_u\|^2 + \|\nabla v - F_v\|^2) dA, \quad (4)$$

where  $F_u(\mathbf{p})$  is one of the six vectors of  $F$  at  $\mathbf{p} \in S$  and  $F_v(\mathbf{p}) := R_4^1 F_u(\mathbf{p})$  is perpendicular to it.

The parameterization must fulfill the integer constraints in Equation (2), whereas the values  $r_{ij}$  encode which of the 6-RoSy vectors in adjacent triangles  $t_i$  and  $t_j$  are paired, i.e.  $F_u$  in  $t_i$  is paired with  $R_6^{r_{ij}} F_u$  in  $t_j$ . The  $r_{ij}$  are held fixed during energy minimization, whereas  $u, v$  and  $w_{ij}$  are optimized.

Notice that the energy is independent of the choice of  $F_u$  (there are six choices per triangle) due to the rotational symmetries of  $\varphi$  from Equation (2). A different choice of  $F_u$  in one triangle will result in the same change in the  $r_{ij}$ 's along all adjacent edges. The resulting minimizer of the energy (Equation (4)) is then locally rotated by a multiple of  $\frac{\pi}{3}$  in this triangle, resulting in the same pattern.

A key observation in QUADCOVER [5] is that the optimization can be divided into two subproblems and solved independently:

- 1) **Local step.** Minimize energy (Equation (4)) for  $u_i, v_i, w_{ij} \in \mathbb{R}$ , ignoring the integer constraint on  $w_{ij}$ . In QUADCOVER the minimizer is computed by removing the curl of  $F$ , making it locally integrable, and defining  $u_i, v_i$  as its potential. This leads to a local parameterization  $\varphi'$ .
- 2) **Global step.** Convert  $\varphi'$  into a global parameterization by incorporating the aforementioned integer constraints.

**HEXCOVER and Covering Spaces.** While possible, solving Equation (4) directly presents some challenges due to the fact that  $F_u$  and  $F_v$  are both multi-valued (there are six values per triangle). Here we make use the notion of *covering space*, which transforms the problem of computing a global parameterization on  $S$  under a guiding 6-RoSy field  $F$  to generating a global parameterization on an  $N$ -fold cover  $S'$  of  $S$  under a guiding vector field  $F'$ . The benefit of doing this is that we can use standard vector field calculus without having to deal with an  $N$ -RoSy field.

In fact, the covering is just used as theoretical foundation and is not explicitly computed in either QUADCOVER or HEXCOVER. The covering is implicitly represented by the



values  $r_{ij}$  resulting in additional constraints (Equation (2)) during optimization. Note that covering spaces are used implicitly by other approaches optimizing a piecewise-linear global parameterization [16], [6].

In the hexagonal case,  $F$  can be lifted to  $F'$  on a six-fold covering surface  $S'$  of  $S$ , which is defined as follows: every triangle  $t_i$  in  $S$  will have six corresponding triangles in  $S'$ :  $t_{i,0}, \dots, t_{i,5}$ . The vector field  $F'$  distributes the six vectors of  $F$  onto the six copies, i.e.,  $F'(t_i) = R_6^i F_0(t_i)$  where  $F_0(t_i)$  is one of the six directions of  $F$  in  $t_i$ . For adjacent triangles  $t_i, t_j$  in  $S$ , the corresponding copies are combinatorially connected, depending on the rotational discontinuity  $r_{ij}$ . The triangles  $t_{i,k}, k \in \{0, \dots, 5\}$  are thereby connected with  $t_{j,k+r_{ij} \bmod 6}$ . Note that  $S'$  is a Riemannian surface with branch points at those positions where the original 6-RoSy field has singularities. All six copies of a triangle are geometrically identical, so there is not necessarily an embedding without self-intersections. This does not present any difficulty for us, however, since the algorithm does not rely on an explicit embedding of  $S'$ .

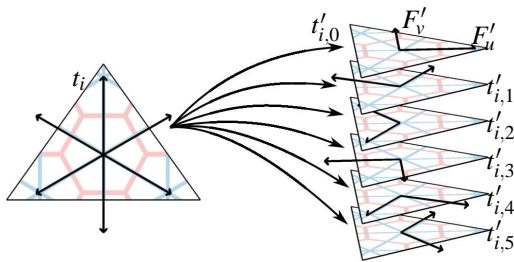


Fig. 10. Left: Triangle  $t_i$  with 6-RoSy field. Right: 6-fold covering of  $t_i$  with vector fields  $F'_u, F'_v$ .

The problem now turns into minimizing the energy in Equation (4) on the covering space  $S'$ , using  $F'_u := F'$ ,  $F'_v := R_4^1 F'$  (see Figure 10). Due to the symmetry of the covering surface and the symmetric behavior of the algorithm, the resulting texture images on different copies of each triangle are congruent and their projection onto the domain  $S$  is a global parameterization which satisfies Equation (2). Again, the use of coverings is only a theoretical view, the algorithm itself will not compute the covering, but represents it implicitly by storing the values  $r_{ij}$ . They must be considered when minimizing the energy.

**Local Step.** In the local step, Energy (4) is minimized for values of the parameterization  $u_i(p_j), v_i(p_j)$  at each vertex  $p_j$  in all incident triangles  $t_i$ , and for the translational discontinuities  $w_{ij} \in \mathbb{R}^2$ . Due to the high number of variables and additional constraints (Equation (2)), QUADCOVER proposes to solve an alternative energy providing the same result but with a much smaller system of equations and no constraints. We use a similar reduction for HEXCOVER.

Let  $\varphi = (u, v)^T$  be the minimizer of Energy (4). A key observation is derived from the discrete Hodge-Helmholtz decomposition of vector fields [43]: The field  $(F_u - \nabla u, F_v - \nabla v)$  is exactly a *co-gradient field* ( $R_4^1 \nabla u^*, R_4^1 \nabla v^*$ ) which

minimizes the energy

$$E^*(u^*, v^*) := \int_S (\|R_4^1 \nabla u^* - F_u\|^2 + \|R_4^1 \nabla v^* - F_v\|^2) dA. \quad (5)$$

Here,  $u^*$  and  $v^*$  are scalar non-conforming finite element functions, which are linear in each triangle and defined by values on edge midpoints. At boundary edges,  $u^*$  and  $v^*$  are fixed to 0. The constraints (Equation (2)) simplify to

$$\begin{pmatrix} u^*_{|t_i} \\ v^*_{|t_i} \end{pmatrix} = R_6^{r_{ij}} \begin{pmatrix} u^*_{|t_j} \\ v^*_{|t_j} \end{pmatrix} \quad (6)$$

for adjacent triangles  $t_i, t_j$ . Notice that the translational discontinuities  $w_{ij}$  do not appear in this formulation.

Equation (6) directly relates the values of  $u^*$  and  $v^*$  in both adjacent triangles of each edge, therefore only one free  $u^*$ -variable and one  $v^*$ -variable remains left per edge. We build up a system of linear equations by setting all partial derivatives of Energy (5) for the free variables to 0. The matrix of this system has dimension  $2|E| \times 2|E|$ , where  $|E|$  is the number of edges in the mesh. We solve this system and obtain  $(u^*, v^*)$  from which we compute  $(\nabla u, \nabla v)$ .

The parameterization  $(u, v)$  is computed by first cutting the mesh open to a simply connected disk and then directly integrate the gradients. We cut the surface at the shortest homotopy generators similar to [44]. The result is a graph  $G$  on edges, such that the complement  $S \setminus G$  is simply connected. We also need to connect all singularities with the cut graph, since they can be seen as infinitesimal small holes. For this purpose, the method was adapted allowing also surface boundary and singularities by [45].

The gradient fields  $(\nabla u, \nabla v)$  are integrated by setting  $(u, v) = (0, 0)$  in an arbitrary root vertex  $v_0$  in triangle  $t_0$  and directly integrating the piecewise constant vectors in  $t_0$  and adjacent triangles until the whole surface is covered. When passing an edge, the values of  $(u, v)$  must be rotated according to Equation (2). Note that the translational discontinuities are set to 0 in the inner of  $S \setminus G$ . The solution is consistent and does not depend on the traversal of the triangles, as long as the edges in the cut graph  $G$  are not involved in this propagation.

**Global Step.** While the obtained parameterization  $(u, v)$  is a minimizer of Equation (4), it may be discontinuous along the edges of  $G$ . For a global hexagonal parameterization, such discontinuity will lead to seams in the parameter lines if the  $w_{ij}$ 's are not in the set of  $G_6$  (the Eisenstein integer lattice). However, when performing local integration in the previous step we only require that  $w_{ij} \in \mathbb{R}$ . In this section we discuss how to modify the initial parameterization to enforce the integer constraints.

$G$  can be considered as union of paths  $\gamma_i$ , each of which is either a closed loop or starts and ends at a singularity. An important property of the solution of Energy (4) is that the translational discontinuity  $w_{ij}$  is constant for all edges on the same path  $\gamma_i$ . Let  $w_i$  be the constant for path  $\gamma_i$ , which can be computed from the coordinates of  $(u, v)$  at both sides

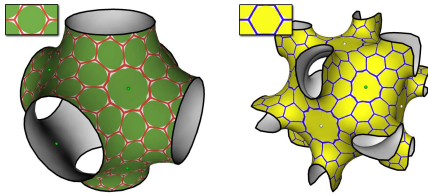


Fig. 11. Minimal surfaces. Left: Schwarz surface with 8 singularities of index  $-1/2$ . Right: Neovius surface with 8 index  $-1/2$  and 6 index  $-1$  singularities.

of an edge of  $\gamma_i$ . Note that the translational discontinuities can add up if two paths partially overlap.

To enforce the integer constraints, we modify the translational discontinuity  $w_{ij}$  for every edge in  $G$  by rounding them to the nearest integer in  $G_6$ . Then, Energy (4) is minimized, holding all discontinuities  $w_{ij}$  fixed.

The coordinates of a singularity are uniquely determined by the  $w_i$  of all of its incident paths  $\gamma_i$ . For a singular vertex with valence  $v$ . There are  $v$  constraints (Equation (2)) that relate the  $v$  coordinate vectors of the vertex in its adjacent triangles. Thus, rounding the values  $w_i$  is similar to prescribing the coordinates of singularities.

For each regular vertex of valence  $v$ , one of the  $v$  relations (Equation (2)) is redundant since the total discontinuity adds up to zero, reflecting a zero Poincaré index. Therefore, its coordinates are determined by the coordinates in one of its incident triangles, we therefore obtain one free variable for  $u$  and one for  $v$  per vertex. Energy (4) is minimized by setting all partial derivatives to 0 resulting in a sparse linear system. The matrix has dimension  $2|\bar{V}| \times 2|\bar{V}|$  with  $|\bar{V}|$  being the number of regular vertices.

Figure 11 shows the hexagonal parameterization of two minimal surfaces using our technique.

**Rounding Technique.** The presented rounding technique for the  $w_i$  is just a heuristic for the problem of finding an optimal parameterization yielding the integer conditions. In general, this problem is NP-hard, since it is equivalent to minimizing a quadratic function on a given lattice (also called the *closest vector problem*).

The rounding technique used in QUADCOVER [5] where all integer variables are rounded at once compares to that from Mixed Integer Quadrangulation (MIQ) [6], which iterates between rounding the integer variables and solving the system with the new boundary condition. In QUADCOVER, the translational discontinuities  $w_{ij}$  are used as integer variables, whereas MIQ uses the coordinates of singularities. Since the coordinates of singularities are uniquely determined by the  $w_{ij}$  (up to global translation), both approaches consider a similar space, but use a different representation basis.

In all our tests, both rounding techniques (direct and mixed integer rounding) give nearly similar results. We conjecture that the reason behind this is our use of the shortest cut

graph  $G$ . It appears that the shorter the paths  $\gamma_i$ 's are, the more local the influence of  $w_i$ 's are and the more accurate a direct rounding of all integer variables is.

Another aspect is computing time. In MIQ, the number of linear systems of equations need to be solved is equal to the number of paths  $\gamma_i$  in  $G$ . Note that the matrix in the system is different every time due to the variable reduction when rounding. The rounding technique from QUADCOVER only needs to solve two systems of equations in total. We therefore decided to use the direct rounding. However, we do not anticipate any difficulty in adapting the MIQ solver to hexagonal parameterizations.

## 5 RESULTS AND APPLICATIONS

Here, we apply hexagonal parameterization to two graphics applications: pattern synthesis, and triangular remeshing.

**Pattern Synthesis on Surfaces.** Example-based texture and geometry synthesis on surfaces has received much attention from the Graphics community in recent years. We refer to [46] for a complete survey. Here we will refer to most relevant work.

Wei and Levoy [22] are the first who point out that  $N$ -RoSy fields of  $N > 1$  are suitable for specification of special symmetries in textures. Liu et al. [47] propose techniques for the analysis, manipulation, and synthesis of *near-regular* textures (i.e. very structured textures with repeating patterns) in the plane. Kaplan and Salesin [2] address the design of Islamic star patterns in the plane. There has been some recent work in constructing circle patterns from a triangular mesh for architectural models [1].

Generating regular patterns on a surface can be greatly facilitated given an appropriate global parameterization. Given a regular hexagonal texture or geometry pattern, it is simply tiled in the parameter-space of the mesh and the texture should stitch (relatively) seamlessly everywhere (Figure 12). For example, to achieve circle packing for architectural patterns, our hexagonal parameterization allows nice hexagonal patterns to be generated from a surface, which can be used as input to such algorithms as shown in Figure 12 (right). Our method provides necessary smoothness and feature alignment, thus leading to a high-quality model, even in the case of relatively high geometric and topological complexity. Figure 3 (b, c) provides some additional examples in which regular hexagonal texture and geometry patterns are placed on the dragon.

We also comment that quadrangular global parameterizations can be used to synthesis square patterns on surfaces. Figure 8 provides one such example. To the best of our knowledge this realization has not been published. Notice that our field generation algorithm can also automatically generate geometry-aware 4-RoSy fields, which lead to coherent synthesized patterns that align with surface features.

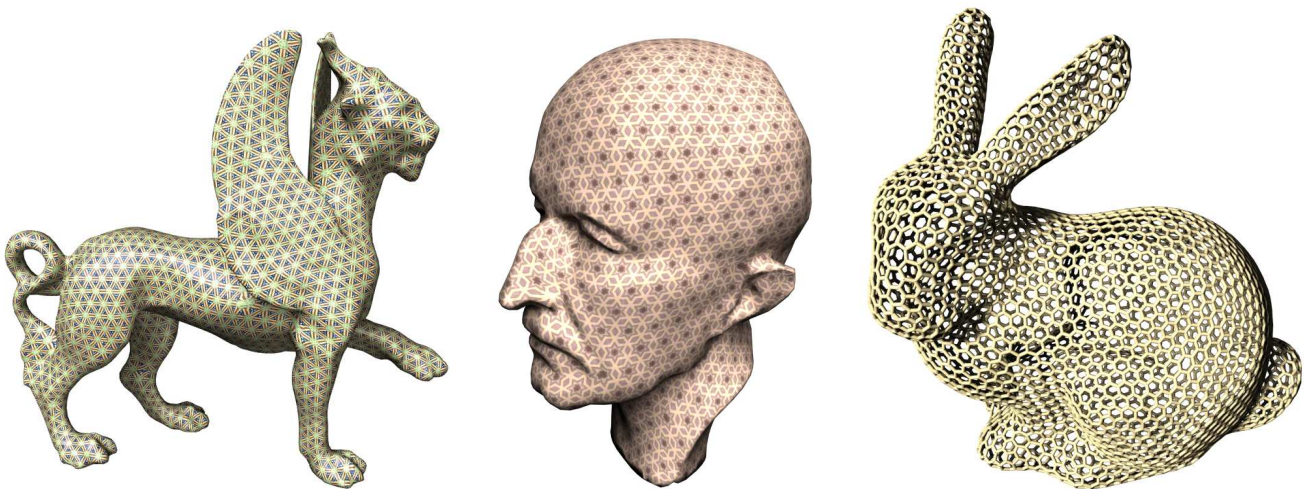


Fig. 12. Seamless tiling of hexagonal textures (left, middle) and geometry patterns (right).

**Triangular Remeshing.** There has been much work in triangular remeshing. To review all past work is beyond the scope of this article. We refer the reader to [48] for a complete survey of triangular remeshing literature, and review only the most relevant work here. Common methods of mesh triangulation are typically based on either a parameterization [49], [50], [51], [52], local optimization methods [53], [54], [55], or Delaunay triangulations and centroidal Voronoi tessellations [56], [57].

The focus of triangular remeshing is on shape preservation, good triangle aspect ratio, feature-aware triangle sizing, and control of irregular vertices (valence not equal to six). These object functions are often conflicting with one another, and the output mesh is a result of compromise among these factors. For example, many parameterization-based methods suffer from artifacts in the triangulation at the locations of the chart boundaries (though this problem can be alleviated by using a global parameterization as in [52]). Direct and local optimization methods suffer from a lack of global control over the structure of the triangulation such as the location and number of irregular vertices.

In this article, we perform triangular remeshing using a hexagonal global parameterization derived a shape-aware 6-RoSy field. There are a number of benefits to this. First, such an approach can lead to overall better aspect ratio for triangles in the remesh (equilateral). Second, the number of irregular vertices can be reduced and their locations can be controlled as these vertices correspond exactly to the set of singularities in the 6-RoSy field. Third, we have incorporated the ability to match the orientations of the RoSy field based on natural anisotropy in the surfaces. Fourth, the size of the triangles can be controlled through a scalar sizing function. The frames are just scaled by the corresponding sizing value. A smaller scaling results in bigger triangles whereas a high value generates a finer triangle mesh (Figure 13).

We can influence the number of singularities in the mesh by

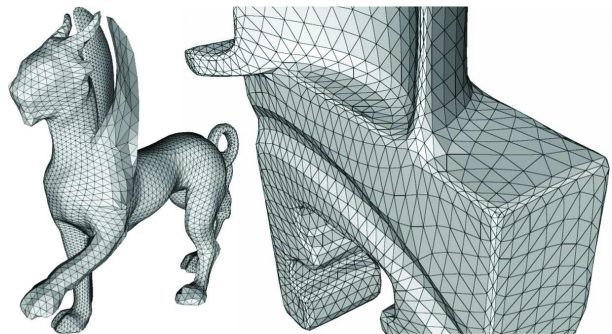


Fig. 13. Adaptive sizing of triangles. Left: Linear scaling along the  $y$ -axis. Right: Scaling by the absolute maximal principle curvature value.

| model name           | Hausdorff distance | min angle | max angle | SD angle | irregular vertices |
|----------------------|--------------------|-----------|-----------|----------|--------------------|
| Foot [50]            | 0.3373             | 2.82      | 173.88    | 11.92    | 146                |
| Foot [57]            | 0.0094             | 26.92     | 115.85    | 7.40     | 3287               |
| Foot                 | 0.0129             | 22.65     | 125.09    | 5.11     | 13                 |
| Venus [50]           | 0.1005             | 0.42      | 178.99    | 17.48    | 38                 |
| Venus [57]           | 0.0439             | 19.89     | 121.13    | 10.37    | 1449               |
| Venus                | 0.0543             | 24.87     | 114.80    | 6.84     | 36                 |
| Max Planck Fig.12    | 0.00263            | 12.44     | 145.79    | 5.00     | 44                 |
| Bunny Fig.14, left   | 0.6581             | 22.43     | 128.08    | 8.23     | 23                 |
| Bunny Fig.14, middle | 0.0198             | 18.03     | 138.54    | 7.54     | 65                 |
| Bunny Fig.14, right  | 0.0309             | 16.99     | 133.8     | 8.50     | 151                |
| Feline Fig.12        | 0.02695            | 5.75      | 167.34    | 11.09    | 121                |
| Dragon Fig.3         | 0.00762            | 4.80      | 151.88    | 9.10     | 181                |
| Blade Fig.13         | 0.84233            | 0.67      | 178.18    | 26.41    | 55                 |

TABLE 1

Quality of meshes: Hausdorff distance (% of bounding box); minimum, maximum, and standard deviation (SD) of angles, and number of irregular vertices.

singularity clustering as described in Section 3. Figure 14 shows that the distances of the singularities impact the smoothness of the parameterization, with more singularities reproducing more feature details of the surface. However, metric distortion also increases when more singularities are used as can be represented with the actual mesh resolution (see Figure 15). Choosing the number of singularities can



be considered as a tradeoff between smoothness of mesh elements and feature preservation. In Table 1, we compare the statistics for the three bunny remeshing results. Notice that the Hausdorff error and the standard deviation in angles of the triangles in the remesh is the lowest for the case when there are 65 singularities, corresponding to the parameter values that we used to generate all our models. The other two models have 23 and 151 singularities, respectively. They were the results of more and less aggressive singularity clustering. Figure 14 compares the three models visually. Notice that features such as ridges along the ears are usually less preserved when there are too few singularities.

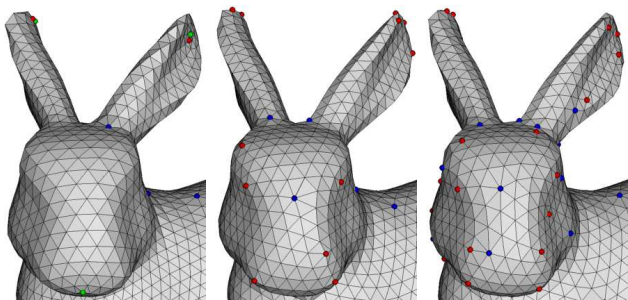


Fig. 14. Remeshing with 23, 65, and 151 singularities.

Figure 16 compares the results of the foot and Venus models using our method with that of [50] and [57]. Table 1 provides the quality statistics of all tested models and the comparison. Notice that our method has better overall aspect ratios of triangles in the mesh (larger minimum angle, smaller maximum angle, and small standard deviation of angles) than [50]. All three methods capture the underlying geometry well (comparable Hausdorff distances to the original input mesh) but our method tends to have the fewest irregular vertices among all three methods. This is a direct result of our automatic singularity clustering in the field generation (Section 3) while achieving good triangle aspect ratios is due to the nature of the hexagonal parameterization. In addition, our method tends to produce edge directions that better align with the features in the mesh, such as along Venus’ nose ridge than [50]. Additional remeshing results can be found in Figure 3.

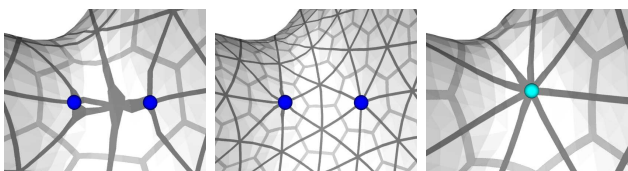


Fig. 15. Singularities which are closer than the grid size may force the parameterization to degenerate locally (left). It can be avoided by either choosing a finer grid size (middle) or by merging nearby singularities with our clustering approach (right).

**Performance.** The amount of time to automatically generate a geometry-aware 6-RoSy field is on average 40 seconds for a model of 40K triangles, measured on a PC

with a dual-core CPU of 2.8GHz CPU and 4GB RAM. The time to generate the parameterization is approximately 120 seconds per model, measured on a PC with 2.13GHz four-core CPU with 8GB RAM. The running time of both stages are impacted by the mesh size as well as the number of singularities in the RoSy field. The computation time for both the field generation and parameterization stages is dominated by solving linear systems whose size is  $O(|E|)$  where  $|E|$  is the number of edges in the mesh. We solve these systems using a biconjugate gradient solver, whose complexity is sub-quadratic.

## 6 FUTURE WORK

There are a number of possible future research directions. First, we plan to add the capability to have parameter lines passing through sharp edges in the model, as concerned in the quadrangulation case by [6]. Second, we wish to study objects that are close to  $N$ -RoSy, which we refer to as near-regular RoSy’s. In these objects the  $N$  member vectors do not have identical magnitude nor even angular spacings. Such objects can allow more flexibility in both quadrangular and triangular remeshing. Third, pentagonal symmetry appears in many natural objects such as flowers. We wish to pursue graphics applications that deal with pentagonal symmetry. While an  $N$ -gon can tile a plane only if  $N = 3, 4,$  and  $6$ , it can tile a hyperbolic surface for any  $N > 2$ . Consequently, pentagonal patterns have the potential of being used to tile hyperbolic regions in a surface or for a hyperbolic parameterization. Notice our parameterization technique can actually handle a parameterization based on an  $N$ -RoSy field for any  $N \geq 2$ . In another direction we plan to investigate appropriate mathematical representations that handle other types of wallpaper textures which may contain reflections and gliding reflections. Surface tiling with at least two different types of rotational symmetries is another potential future direction. Such patterns have applications in cyclic weaving over surfaces [58] and remeshing [33].

## ACKNOWLEDGEMENTS

The authors wish to thank Felix Kälberer and Ulrich Reitebuch for fruitful discussions on parameterization and help on remeshing. Craig Anderson helped with the video production. Many thanks to all reviewers for their helpful comments which have led to significant improvements of the article. The 3D models used in this article are courtesy of Marc Levoy and the Stanford graphics group, and the AIMshape repository. The work is partially sponsored by the DFG research center MATHEON and the US National Science Foundation (NSF) grants IIS-0546881, CCF-0830808 and IIS-0917308.

## REFERENCES

- [1] A. Schiftner, M. Höbinger, J. Wallner, and H. Pottmann, “Packing circles and spheres on surfaces,” in *Sigraph Asia*, 2009, pp. 1–8.

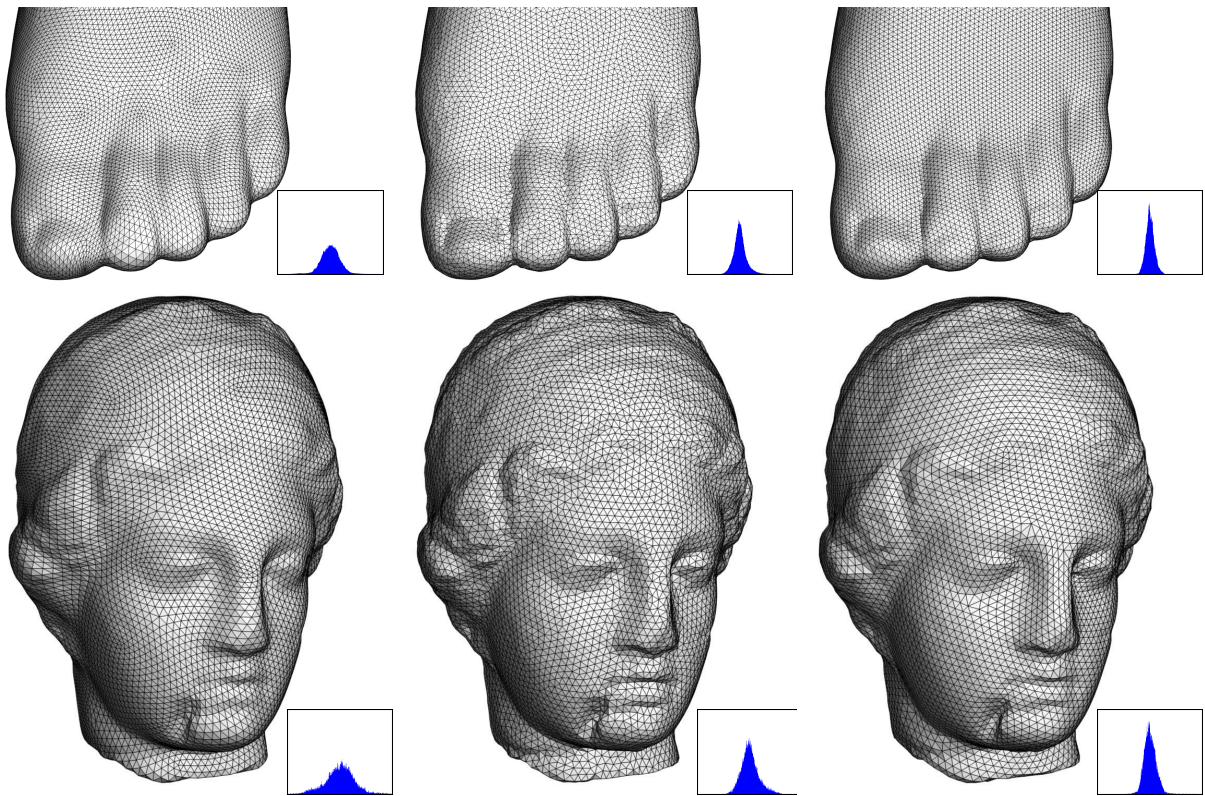


Fig. 16. Comparison of our method (right) to those from [50] (left) and [57] (middle). The histograms show occurring inner angles (on the  $X$ -axis from 0 to  $\pi/3$ . For each model, the scale on the  $Y$ -axis is the same.

- [2] C. S. Kaplan and D. H. Salesin, "Islamic star patterns in absolute geometry," *Trans. Graph.*, vol. 23, no. 2, pp. 97–119, 2004.
- [3] Z. J. Wood, P. Schröder, D. Breen, and M. Desbrun, "Semi-regular mesh extraction from volumes," in *VIS*, 2000, pp. 275–282.
- [4] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez, "Periodic global parameterization," *TOG*, vol. 25, no. 4, pp. 1460–1485, 2006.
- [5] F. Kälberer, M. Nieser, and K. Polthier, "QUADCOVER - surface parameterization using branched coverings," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 375–384, 2007.
- [6] D. Bommes, H. Zimmer, and L. Kobbelt, "Mixed-integer quadrangulation," *Trans. Graph.*, vol. 28, no. 3, pp. 1–10, 2009.
- [7] J. Palacios and E. Zhang, "Rotational symmetry field design on surfaces," in *Siggraph*, 2007, p. 55.
- [8] M. S. Floater and K. Hormann, "Surface parameterization: a tutorial and survey," in *Advances in multiresolution for geometric modelling*. Springer Verlag, 2005, pp. 157–186.
- [9] K. Hormann, K. Polthier, and A. Sheffer, "Mesh parameterization: Theory and practice," in *Siggraph Asia, Course Notes*, no. 11, 2008.
- [10] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle, "Conformal surface parameterization for texture mapping," *Trans. Vis. Comp. Graph.*, vol. 6, no. 2, pp. 181–189, 2000.
- [11] X. Gu and S.-T. Yau, "Global conformal surface parameterization," in *SGP*, 2003, pp. 127–137.
- [12] M. Jin, Y. Wang, S.-T. Yau, and X. Gu, "Optimal global conformal surface parameterization," in *IEEE Visualization*, 2004, pp. 267–274.
- [13] L. Kharevych, B. Springborn, and P. Schröder, "Discrete conformal mappings via circle patterns," *Trans. Graphics*, vol. 25, no. 2, 2006.
- [14] S. Dong, S. Kircher, and M. Garland, "Harmonic functions for quadrilateral remeshing of arbitrary manifolds," *Comput. Aided Geom. Des.*, vol. 22, pp. 392–423, 2005.
- [15] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart, "Spectral surface quadrangulation," in *Siggraph '06*, pp. 1057–1066.
- [16] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun, "Designing quadrangulations with discrete harmonic forms," *SGP*, 2006.
- [17] E. Zhang, J. Hays, and G. Turk, "Interactive tensor field design and visualization on surfaces," *Trans. Vis. Comp. Graph.*, vol. 13, no. 1, pp. 94–107, 2007.
- [18] J. L. Helman and L. Hesselink, "Visualizing vector field topology in fluid flows," *Comp. Graph. Appl.*, vol. 11, pp. 36–46, 1991.
- [19] T. Delmarcelle and L. Hesselink, "The topology of symmetric, second-order tensor fields," *Comp. Graph. Appl.*, pp. 140–147, 1994.
- [20] E. Praun, A. Finkelstein, and H. Hoppe, "Lapped textures," *Siggraph*, pp. 465–470, 2000.
- [21] G. Turk, "Texture synthesis on surfaces," *Siggraph*, 2001.
- [22] L. Y. Wei and M. Levoy, "Texture synthesis over arbitrary manifold surfaces," *Siggraph*, pp. 355–360, 2001.
- [23] J. Stam, "Flows on surfaces of arbitrary topology," *Siggraph*, 2003.
- [24] J. J. van Wijk, "Image based flow visualization for curved surfaces," *IEEE Visualization*, pp. 123–130, 2003.
- [25] M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe, "Design of tangent vector fields," in *Siggraph*, 2007, p. 56.
- [26] H. Theisel, "Designing 2d vector fields of arbitrary topology," in *Eurographics*, vol. 21, 2002, pp. 595–604.
- [27] W.-C. Li, B. Vallet, N. Ray, and B. Lévy, "Representing higher-order singularities in vector fields on piecewise linear surfaces," in *Trans. Vis. Comp. Graph.*, 2006.
- [28] E. Zhang, K. Mischaikow, and G. Turk, "Vector field design on surfaces," *Trans. Graph.*, vol. 25, no. 4, pp. 1294–1326, 2006.



- [29] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang, "Interactive procedural street modeling," *TOG*, vol. 27, no. 3, pp. 1–10, 2008.
- [30] A. Hertzmann and D. Zorin, "Illustrating smooth surfaces," *Siggraph*, pp. 517–526, 2000.
- [31] N. Ray, B. Vallet, W. C. Li, and B. Lévy, "N-symmetry direction field design," *Trans. Graph.*, vol. 27, no. 2, pp. 10:1–13, 2008.
- [32] J. Palacios and E. Zhang, "Interactive visualization of rotational symmetry fields on surfaces," in *Trans. Vis. Comp. Graph., to appear*, 2011.
- [33] Y.-K. Lai, M. Jin, X. Xie, Y. He, J. Palacios, E. Zhang, S.-M. Hu, and X. Gu, "Metric-driven rosy field design and remeshing," *Trans. Vis. Comp. Graph.*, vol. 16, no. 1, pp. 95–108, 2010.
- [34] M. Zhang, J. Huang, X. Liu, and H. Bao, "A wave-based anisotropic quadrangulation method," *Trans. Graph.*, vol. 29, pp. 118:1–8, 2010.
- [35] K. Crane, M. Desbrun, and P. Schröder, "Trivial connections on discrete surfaces," *Eurographics '10*, vol. 29, no. 5, pp. 1525–1533.
- [36] N. Ray, B. Vallet, L. Alonso, and B. Lvy, "Geometry aware direction field design," *Trans. Graph.*, 2009.
- [37] K. Xu, D. Cohen-Or, T. Ju, L. Liu, H. Zhang, S. Zhou, and Y. Xiong, "Feature-aligned shape texturing," *Trans. Graph.*, vol. 28, no. 5, pp. 1–7, 2009.
- [38] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential geometry operators for triangulated 2-manifolds," *VisMath*, 2002.
- [39] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramée, "Asymmetric tensor analysis for flow visualization," *Trans. Vis. Comp. Graph.*, vol. 15, pp. 106–122, 2009.
- [40] J. J. Koenderink and A. J. van Doorn, "Surface shape and curvature scales," *Image Vision Comput.*, vol. 10, pp. 557–565, October 1992.
- [41] A. N. Hirani, "Discrete exterior calculus," Ph.D. dissertation, 2003.
- [42] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," in *Siggraph*, 1995, pp. 173–182.
- [43] K. Polthier and E. Preuss, "Identifying vector field singularities using a discrete Hodge decomposition," in *Visualization and Mathematics III*. Springer, 2003, pp. 113–134.
- [44] J. Erickson and K. Whittlesey, "Greedy optimal homotopy and homology generators," in *Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2005, pp. 1038–1046.
- [45] F. Kälberer, M. Nieser, and K. Polthier, "Stripe parameterization of tubular surfaces," in *Topological Data Analysis and Visualization: Theory, Algorithms and Applications*. Springer Verlag, 2009.
- [46] L.-Y. Wei, S. Lefebvre, V. Kwatra, , and G. Turk, "State of the art in example-based texture synthesis," *Eurographics*, 2009.
- [47] Y. Liu, W.-C. Lin, and J. Hays, "Near-regular texture analysis and manipulation," in *Siggraph*, 2004, pp. 368–376.
- [48] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene, "Recent advances in remeshing of surfaces," *Research Report, AIM@Shape*, 2005.
- [49] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," vol. 21, no. 3, 2002, pp. 355–361.
- [50] P. Alliez, M. Meyer, and M. Desbrun, "Interactive geometry remeshing," in *Siggraph*, 2002, pp. 347–354.
- [51] P. Alliez, E. C. de Verdiere, O. Devillers, and M. Isenburg, "Isotropic surface remeshing," in *SMI*. IEEE Computer Society, 2003, p. 49.
- [52] A. Khodakovsky, N. Litke, and P. Schröder, "Globally smooth parameterizations with low distortion," in *Siggraph '03*.
- [53] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *Siggraph*, 1993, pp. 19–26.
- [54] P. Lindstrom and G. Turk, "Image-driven simplification," *Trans. Graph.*, vol. 19, no. 3, pp. 204–241, 2000.
- [55] V. Surazhsky and C. Gotsman, "Explicit surface remeshing," in *SGP*, 2003.
- [56] G. Turk, "Re-tiling polygonal surfaces," *Siggraph*, 1992.
- [57] D.-M. Yan, B. Lévy, Y. Liu, F. Sun, and W. Wang, "Isotropic remeshing with fast and exact computation of restricted voronoi diagram," in *SGP*, 2009, pp. 1445–1454.
- [58] E. Akleman, J. Chen, Q. Xing, and J. L. Gross, "Cyclic plain-weaving on polygonal mesh surfaces with graph rotation systems," *Trans. Graph.*, vol. 28, no. 3, pp. 1–8, 2009.



**Matthias Nieser** is a PhD student at the Department of Mathematics at Freie Universität Berlin, studying under Konrad Polthier. He is funded by the DFG research center MATH-EON. His current research focuses on discrete differential geometry, in particular the parameterization and structuring of surfaces and volumes.



**Jonathan Palacios** is currently a PhD student in the Department of Electrical Engineering and Computer Science at Oregon State University, studying under Dr. Eugene Zhang. His primary research areas are computer graphics, geometric modeling, symmetry, and higher-order tensor field visualization and analysis. He is an NSF IGERT fellow, and a member of the ACM.



**Konrad Polthier** is professor of mathematics at Freie Universität Berlin and DFG research center MATHEON, and chair of the Berlin Mathematical School. He received his PhD from University of Bonn in 1994, and headed research groups at Technische Universität Berlin and Zuse-Institute Berlin. His current research focuses on discrete differential geometry and geometry processing. He co-edited several books on mathematical visualization, and co-produced mathematical video films. His recent video MESH ([www.mesh-film.de](http://www.mesh-film.de), joint with Beau Janzen) has received international awards including "Best Animation" at the New York International Independent Film Festival. He served as paper or event co-chair on international conferences including Symposium on Geometry Processing 2006 and 2009.



**Eugene Zhang** received the PhD degree in computer science in 2004 from Georgia Institute of Technology. He is currently an associate professor at Oregon State University, where he is a member of the School of Electrical Engineering and Computer Science. His research interests include computer graphics, scientific visualization, and geometric modeling. He received a National Science Foundation (NSF) CAREER award in 2006. He is a member of the IEEE and a senior member of ACM.