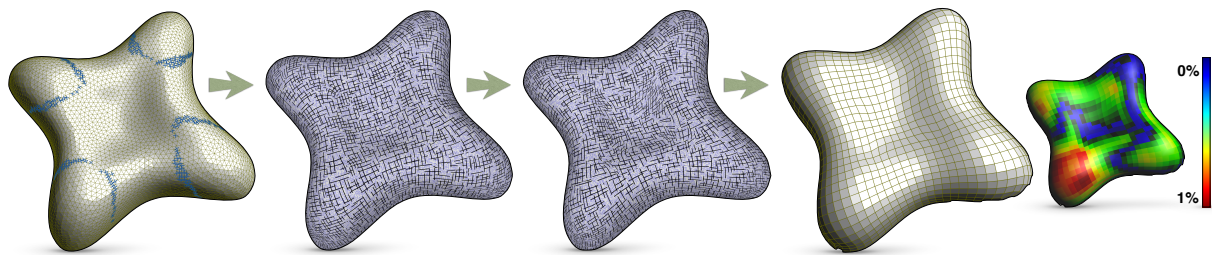


# Designing $N$ -PolyVector Fields with Complex Polynomials

Olga Diamanti<sup>1</sup> Amir Vaxman<sup>2</sup> Daniele Panozzo<sup>1</sup> Olga Sorkine-Hornung<sup>1</sup>

<sup>1</sup>ETH Zurich, Switzerland

<sup>2</sup>Vienna Institute of Technology, Austria



**Figure 1:** A smooth 4-PolyVector field is generated from a sparse set of principal direction constraints (faces in light blue). We optimize the field for conjugacy and use it to guide the generation of a planar-quad mesh. Pseudocolor represents planarity.

## Abstract

We introduce  $N$ -PolyVector fields, a generalization of  $N$ -RoSy fields for which the vectors are neither necessarily orthogonal nor rotationally symmetric. We formally define a novel representation for  $N$ -PolyVectors as the root sets of complex polynomials and analyze their topological and geometric properties. A smooth  $N$ -PolyVector field can be efficiently generated by solving a sparse linear system without integer variables. We exploit the flexibility of  $N$ -PolyVector fields to design conjugate vector fields, offering an intuitive tool to generate planar quadrilateral meshes.

## 1. Introduction

The design of tangent vector fields on discrete surfaces is a basic building block for many geometry processing applications. Such fields are often used as target gradient vectors for surface remeshing [BLP\*13, NPPZ12], parametrization [LZX\*08] and texture synthesis [LH06]. In addition, they can be used to study and detect symmetries on surfaces [BCBSG10], and for architectural geometric design [LXW\*11, PBSH13].

Many applications require the design of multiple vector fields (vector sets) coupled in a nontrivial way. Notable examples are principle curvature directions, which are defined up to a sign permutation, or vectors representing compression forces in architectural structures. Sets of more than two vectors are used for meshing of triangular, quadrilateral and hexagonal meshes [NPPZ12].

The ubiquitous *Rotationally-Symmetric* fields, commonly denoted as “ $N$ -RoSy fields”, are special vector sets compris-

ing  $N$  unit-length vectors related by a rotation of an integer multiple of  $2\pi/N$ . The 4-RoSy fields are particularly attractive since they commonly represent the ideal candidate gradients of a surface parametrization. The  $\pi/2$ -rotation invariance is employed in the generation of quad meshes on surfaces with a nontrivial topology [BLP\*13].

However,  $N$ -RoSy fields are often too restrictive for practical applications. Designers often wish to manually control the anisotropy of the field and allow deviation from uniformly-sized quads and right angles in order to increase the design space and, e.g., better adapt the discretization to the underlying shape, its semantics and articulation [TPSHSH13].

We introduce a novel representation for general unordered vector sets in which no vector is necessarily related by any symmetry or magnitude to another. We call such vector sets and their representation  $N$ -PolyVectors ( $N$ -PV). We support arbitrary angles between vectors and an arbitrary magnitude for each vector. We represent an  $N$ -PolyVector by the set of

coefficients of a complex polynomial, where the individual vectors are the roots of that polynomial. This representation generalizes  $N$ -RoSy vector sets in an intuitive way: an  $N$ -RoSy is the root set of the polynomials of the form  $z^N - u^N$ . Our representation supports efficient design of vector sets requiring neither integer period jumps, nor explicit pairings of vectors between adjacent sets on a manifold. We formally define parallel transport, smoothness and singularities of  $N$ -PolyVector fields of arbitrary degree  $N$ .

$N$ -PolyVector fields can be used to design *discrete conjugate vector fields* [LPW\*06, BS08]. Such fields are necessary in the remeshing of surfaces using *planar quad* (PQ) and planar hexagonal elements. These meshes recently gained considerable interest for their use in architectural applications, since they can be realized by planar glass, steel or wooden structures. Using orthonormal  $N$ -RoSy fields is far too restrictive for that purpose because only principal direction fields are both conjugate and orthogonal. Moreover, minimal and maximal principal directions cannot be interchanged, and therefore the singularities of a principal-direction field are limited to  $\pm k/2$  indices, instead of the  $\pm k/4$  indices possible for general 4-RoSy fields.

The contributions of this paper are the following:

1. We formally define  $N$ -PolyVector fields, describe their topology, and provide a consistent definition of parallel transport, smoothness and singularities.
2. We show how to compute smooth  $N$ -PolyVector fields with prescribed bounds on angles.
3. We show how to compute conjugate  $N$ -PolyVector fields for remeshing with planar elements.

## 2. Related work

**$N$ -RoSy fields.** 4-RoSy fields, commonly called *cross fields*, were introduced in [HZ00] for cross hatching, a technique for non-photorealistic rendering of surfaces. They were generalized to  $N$ -RoSy fields of arbitrary degree  $N$  in [PZ07] and [RVLL08].  $N$ -RoSy fields represent natural quantities on surfaces, such as the principal curvature directions, but they can alternatively be designed using a given set of constraints. Common design approaches generate smooth  $N$ -RoSy fields with a prescribed set of topological singularities [RVLL08, CDS10], or alternatively opt for smooth fields that satisfy some directional constraints [BZK09, RVAL09, KCPS13].

**Non-orthogonal vector sets.** Zdravec et al. [ZSW10] generalize cross fields, allowing them to be non-orthogonal, while focusing on the generation of conjugate vector fields with strong restrictions on the allowed singularities. Non-orthogonal conjugate vector-field design was further expanded in [LXW\*11]; however, their method still requires explicit derivation of the matchings between the vector sets on the two sides of a face. In order to avoid the hard, nonlinear and integer program involved in finding these matchings, they replace it with an equivalent (but still significantly involved)

nonlinear problem involving only real variables and periodic functions. As we show in Section 4.3, our representation makes it possible to optimize for the same constraints, incorporating the vector magnitude, providing a larger solution space and a simpler, continuous optimization. [PPTSH14] introduces *frame fields*, which are a composition of a cross field and a field of affine transformations of the tangent planes. Frame fields can be used for anisotropic quadrilateral remeshing: the field induces an altered Euclidean metric that leads to an anisotropic parametrization. However, the frame field representation they introduce cannot be extended to handle arbitrary vector sets, in contrast with the one we present in this paper (see Section 4.4).

**Applications.** As already mentioned, 4-RoSy fields can be used to create non-photorealistic renderings based on a cross-hatching pattern [HZ00, PZ07]. A 4-RoSy field can also guide the remeshing of a triangle mesh into a pure quadrilateral mesh whose edges are aligned with the field [BLP\*13]. Similarly, in [NPPZ12] a 6-RoSy field is employed to generate isotropic triangle meshes. Cross fields are used in [PBSH13] to estimate the flow of forces in self-supporting (masonry) structures. Finally, a non-orthogonal cross field, also called conjugate direction field, has been proposed in [LXW\*11] to create planar tilings for architectural applications.  $N$ -PolyVector fields provide a general framework to represent and process the fields used by all these applications.

## 3. $N$ -PolyVector fields with complex polynomials

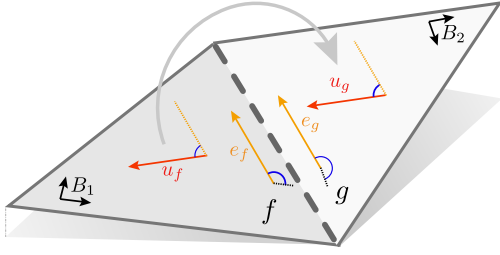
Given a triangle mesh  $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$ , we consider the plane of every triangle  $f \in \mathcal{F}$  as a discrete tangent space, and the entire collection of these planes as the *tangent bundle* of the surface. Consequently, in our framework, tangent vectors are defined on the faces of the triangles, and the tangent vector fields are thus piecewise constant. A vector field is defined as a given assignment of a single vector per face (formally, a *section* of the tangent bundle). A general  $N$ -PolyVector,  $N > 0$ , is an unordered set of  $N$  tangent vectors in a single face, and an  $N$ -PolyVector field is defined accordingly. In the following, we provide some background on a representation of  $N$ -RoSy by the complex parametrization of faces, and then show how it generalizes to represent  $N$ -PolyVector fields.

### 3.1. Piecewise-constant complex $N$ -RoSy fields

We identify every tangent space (face)  $f$  with the complex plane  $\mathbb{C}$  by choosing an arbitrary orthonormal basis (Figure 2). A vector in the tangent plane is then defined by a single complex number. However, two adjacent faces have different bases, and in order to compare adjacent vectors we require a discrete *connection* [dC76], which defines a *parallel transport* between neighboring bases.

The discrete Levi-Civita (LC) connection [CDS10] is the change of basis between two neighboring faces  $f, g \in \mathcal{F}$





**Figure 2:** Complex Levi-Civita connection. The vectors  $u_f, u_g$ , though represented by different complex numbers in different coordinate systems, are LC-parallel since they are the same vector w.r.t. a unified representation, as in Eq. (1).

across a common edge  $e \in \mathcal{E}$ , which is represented in the respective bases by the edge vectors  $e_f, e_g \in \mathbb{C}$ . Formally, the vector  $u_f$  in the tangent space of face  $f$  is LC-parallel to the corresponding vector  $u_g$  in the tangent space of face  $g$  if and only if  $u_g = u_f(e_f)^{-1}e_g$ ; for normalized edge vectors this reduces to  $u_g = u_f \bar{e}_f e_g$ , where we denote by  $\bar{x}$  the complex conjugate of  $x$ . Note that the complex conjugate should not be confused with the notion of conjugate vector fields that we introduce in Section 4.3.

This formulation is conceptually equivalent to unfolding the triangle flap  $(f, g)$  onto a plane and simply translating the vector. This intuition leads to an equivalent, symmetric and more intuitive formulation: Following the flattening of the flap, we may unify the two representation bases by assigning a new mutual basis, so that the mutual edge vector  $e$  becomes the canonical real axis in both. Then the two vectors, one in each face, are LC-parallel if and only if they have an identical representation with respect to the unified basis (see Figure 2):

$$u_g \bar{e}_g = u_f \bar{e}_f. \quad (1)$$

An  $N$ -RoSy vector field consists of the following  $N$  vectors per face:

$$\left\{ u_f \exp\left(i \frac{2\pi k}{N}\right) \mid 0 \leq k \leq N-1 \right\}, \quad (2)$$

which can be compactly expressed as the  $N$ -th order roots of a complex number  $(u_f)^N$ . Two adjacent  $N$ -RoSy vectors  $u_f, u_g$  are LC-parallel if there is a matching of the individual roots such that matching roots are LC-parallel. This reduces to the following condition:

$$(u_g \bar{e}_g)^N = (u_f \bar{e}_f)^N. \quad (3)$$

### 3.2. Complex $N$ -PolyVectors as complex polynomials

An  $N$ -RoSy field can be represented as the variety (root set) of the following complex polynomial:  $P_f(z) = z^N - (u_f)^N$ . It is then immediately clear how to generalize this representation

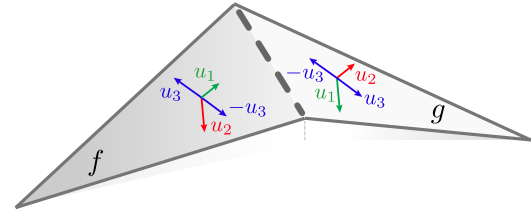
to unambiguously encode a general unordered set of face-based vectors  $\{u_0, u_1, \dots, u_{N-1}\}$ :

$$P_f(z) = (z - u_0) \cdot (z - u_1) \cdot \dots \cdot (z - u_{N-1}). \quad (4)$$

Alluding to the polynomial representation, we denote these unordered vector sets and their polynomial representation as  $N$ -PolyVectors, or  $N$ -PV in short.

If there is a subset of  $M$  symmetric vectors ( $M \leq N$ ) within an  $N$ -PolyVector, its term within the polynomial is  $z^M - u^M$ . An example is shown in Figure 3.

In essence, a polynomial represents an unordered root set by the polynomial coefficients in the monomial basis. For example, consider a 2-PolyVector consisting of two vectors  $u$  and  $v$ : the corresponding polynomial is  $P(z) = (z - u)(z - v) = z^2 - (u + v)z + uv$ . The coefficients we use to represent this 2-PolyVector are then  $-(u + v)$  and  $(uv)$  (the coefficient of  $z^N$  is always 1), and they determine the  $u, v$  explicitly up to order. Also notice that if  $u = -v$ , the 2-PolyVector is in fact a 2-RoSy, and the coefficients “degenerate” to 0 for  $z^1$  and  $-u^2$  for  $z^0$ , as expected.



**Figure 3:** An LC-parallel  $N$ -PolyVector field comprising two independent vector fields  $u_1, u_2$  and a 2-RoSy field  $(u_3)^2$ . Its polynomial is  $P(z) = (z - u_1)(z - u_2)(z^2 - u_3^2)$ . Notice that the equivalence is invariant to change of order and sign across the common edge. The resulting coefficients are  $(-u_1 - u_2, u_1 u_2 - u_3^2, (u_1 + u_2)u_3^2, -u_1 u_2 u_3^2)$ .

**LC-transport for  $N$ -PV fields.** We can generalize Eq. (1) (see Section 3.1) in a similar spirit as above. LC-parallel  $N$ -PolyVectors are defined by having individual matching LC-parallel roots across edges. The main advantage of  $N$ -PolyVector fields, compared to [LXW\*11], is that we can define transport *without* the explicit matchings, avoiding integer variables and/or challenging nonlinear energies in the subsequent optimizations.

Instead of transporting the roots directly, we transport the coefficients of the polynomials (which are also “vectors”, or complex numbers) by substituting the polynomial in Eq. (4) with Eq. (1). Suppose we have two  $N$ -PolyVectors  $\{u_{f,0}, u_{f,1}, \dots, u_{f,N-1}\}$  and  $\{u_{g,0}, u_{g,1}, \dots, u_{g,N-1}\}$  on face  $f$  and face  $g$ , respectively. Then the two  $N$ -PolyVectors are LC-parallel if and only if the following polynomials:

$$\begin{aligned} P_{f|e_f}(z) &= (z - u_{f,0} \bar{e}_f) \dots (z - u_{f,N-1} \bar{e}_f) \\ P_{g|e_g}(z) &= (z - u_{g,0} \bar{e}_g) \dots (z - u_{g,N-1} \bar{e}_g) \end{aligned} \quad (5)$$

are equivalent (respective coefficients are equal). See Figure 3 for an example. Notice that every coefficient is then transported by the corresponding power of the common edge vector. For instance, the constant coefficient  $\prod_{m=0}^{N-1} (-u_{f,m})$  is transported by  $(\bar{e}_f)^N$ . This generalizes the LC-transport in the  $N$ -RoSy case.

### 3.3. Smooth $N$ -PolyVector fields

Connections, as parallel transports, induce curvature inside closed domains. This curvature is measured by the change in vectors as they are parallel-transported around the boundary of that domain. The LC connection induces the Gaussian curvature, which, in the discrete setting, amounts to the sum of angle defects enclosed within the transport path [CDS10]. Thus, Eq. (5) cannot be satisfied globally on a surface unless it is developable (has vanishing Gaussian curvature). Consistent vector field design is equivalent to defining a *trivial connection* between faces. A  $N$ -PV field is considered perfectly smooth across an edge if its polynomial coefficients are LC-transported, which then means that every vector on one face is LC-transported to a single vector on another face. Therefore, we wish to solve for  $N$ -PV fields that define a trivial connection that is “as-LC-as-possible”. We model this by satisfying Eq. (5) in the least squares sense. Suppose that the coefficients of the  $N$ -PolyVector in face  $f$  are denoted as  $\{x_{f,m}\}$ , where coefficient  $x_m$  belongs to the monomial  $z^m$ . Thus, the energy we seek to minimize is:

$$E_{\text{smooth}} = \sum_{m=0}^N \sum_{(f,g) \in \mathcal{E}} |x_{f,m}(\bar{e}_f)^m - x_{g,m}(\bar{e}_g)^m|^2. \quad (6)$$

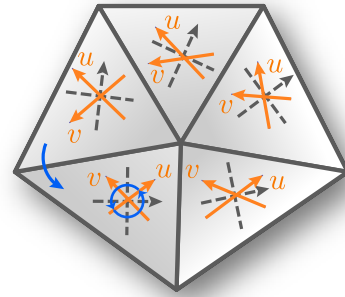
The connection vectors (the deviation from LC) are defined as:  $T_{e,m} = x_{g,m}(\bar{e}_g)^m / x_{f,m}(\bar{e}_f)^m$ . In terms of curvature, trivial connections induce zero curvature around any closed path that does not contain singularities. The argument of a connection vector is called a *connection angle*, encoding the rotation deviation from the LC-transport. Although this term is defined here to refer to the angles between the transported coefficients, we will use the same term (with appropriate clarification) to refer to the transport angles between individual vectors/roots.

### 3.4. PolyVector-field topology

Knowledge of the connection vectors  $T_{e,m}$  is not sufficient to disambiguate the relations between pairs of single (individual) vectors, since the sets are unordered. Therefore, the topology of the field can only be fully determined by defining the explicit matchings between individual vectors across edges. Given such explicit matchings, singularities are defined unambiguously as closed 1-ring routes in which the matches are not consistent [RVLL08]. We next show that there is a canonical and simple way to define singularities of an  $N$ -PolyVector field, by showing that its topology is *equiv-*

*alent* to the topology of the associated “canonical”  $N$ -RoSy field.

Consider an  $N$ -PV field  $\{u_{f,0}, u_{f,1}, \dots, u_{f,N-1}\}$ , and assume w.l.o.g. that the vectors on each face are indexed in a counterclockwise (CCW) order. Again, w.l.o.g., we consider per-vector transports that preserve CCW order, i.e., if vector  $u_{f,m}$  is matched with vector  $u_{g,n}$  in the adjacent face, then  $u_{f,m+1}$  is matched with  $u_{g,n+1}$  as well (indices are always considered modulo  $N$ ). The entire matching can then be parametrized by  $(m,n)$ . Suppose that a closed matching path around a 1-ring of vertex  $v$  returns to vector  $m+k$  in the original face. We then define the singularity index of vertex  $v$  as  $\frac{k}{N}$ . Regular vertices are defined by  $k=0$ , where no mismatch occurs, and we return to the same initial vectors. Notice that this complies with the regular definition of singularities for  $N$ -RoSy fields.



**Figure 4:** A singularity of an  $N$ -PolyVector field (orange) is also a singularity of the  $N$ -RoSy field induced by its coefficient  $x_0$  (dotted grey). In the special case of 4-PolyVector fields, the associated 4-RoSy is made of the bisectors of the PolyVectors.

Since we define the matching combinatorially, it is natural to try and quantify this topology in a manner that is invariant to the individual angle spacings between vectors in the same face. To achieve that, we look at the total sum of angles that each vector in the set  $\{u_{f,m}\}$  accumulates after a closed path around  $v$ , where  $v$  is a singularity of order  $\frac{k}{N}$ . Since  $m$  returns to vector  $m+k$  in the same face,  $m+1$  returns to vector  $m+k+1$  and so on, and since the total sum of ordered angles between consecutive vectors in the set is obviously  $2\pi$ , we get that the sum of angles accumulated in the set  $\{u_{f,m}\}$ , returning to  $u_{f,m+k}$ , is exactly  $2\pi k$ , regardless of the individual angles involved. The proof for the inverse direction (sum leads to singularity index) follows a similar claim. For an intuitive explanation of an index  $k$  of singularity in that manner, imagine a group of people sitting on chairs organized in a circle. The positions of the chairs along the circle may be irregularly spaced, corresponding to the tips of the vectors of our  $N$ -PV. When everyone walks and moves  $k$  chairs counterclockwise around the circle, the total walking distance of

the group is  $2\pi k$ , regardless of the spacing between the chairs on the circle.

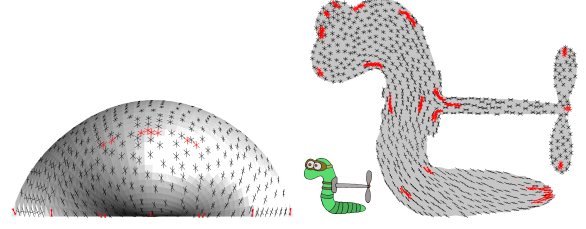
In order to quantify the topology with PolyVector coefficients, let us consider again the matching  $(m, n)$  between adjacent faces  $f, g$ . Denote the sum of the individual transport angles (i.e., the arguments of the individual transport vectors for each of the vectors in the  $N$ -PV) as  $S_{m,n}$  ( $f, g$  indices are dropped for clarity). In light of the above, if we choose a different matching  $(m, n + j)$ , we get the sum  $S_{m,n} + 2\pi j$ . We thus get that all the possible order-preserving matchings are encoded in a set of identical connection vectors, i.e., those whose argument is  $S_{m,n} + 2\pi j$ . The corresponding connection vector is in fact nothing more than the connection vector between the free coefficients  $T_{e,0} = x_{g,0}/x_{f,0}$ ,  $\arg(T_{e,0}) = S_{m,n} + 2\pi j$ . Therefore, the singularities of an  $N$ -PV field are the singularities of the  $N$ -RoSy field represented by its free coefficient  $x_0$ . By this definition, we generalize the results obtained by [LXW\*11], as they proved that the singularities of their 2-vector fields are equivalent to those of their bisector 4-RoSy field, which is described exactly by the constant coefficient of the corresponding PolyVector field.

However, note that by knowing the connection vectors, for both our  $N$ -PV and its associated  $N$ -RoSy, we merely enumerate all the possible matchings (since the actual connection vectors are identical for any  $j$ ). The topology of the field still has a single matching degree of freedom per edge, for choosing the actual  $j$ . This is equivalent to choosing a branch for the complex logarithm, in order to uniquely retrieve the argument of the connection vector  $T_{e,0}$ . A natural choice, which we employ in our work, is to choose the principal branch, i.e., choose a  $j$  such that  $\arg(T_{e,0}) \in (-\pi, \pi]$ . For  $N$ -RoSy fields, this is equivalent to choosing a matching between a (parallel-transported) vector in  $f$  to its closest vector in  $g$ . We call this choice *principal matching*.

The coefficients of an  $N$ -PV thus have a geometric meaning; as we established, the  $N$ -RoSy field of the free coefficient is topologically equivalent to the  $N$ -PV field, and we can therefore treat it as the best approximating  $N$ -RoSy field to the given  $N$ -PV field. Furthermore, other (order  $> 0$ ) coefficients vanish in the limit where  $N$ -PV becomes a RoSy field. Thus, they measure the deviation from symmetry as well. We analyze the coefficients of 4-PV fields (frame fields) in Section 4, and leave the general analysis of the higher-order coefficients of  $N$ -PV fields for future work.

### 3.5. Face-based and vertex-based tangent spaces

Our entire framework is face-based, which is a common approach to tangent vector-field design methods. As we apply our framework to parametrization, for the purpose of remeshing, this framework is compatible with the definition of vertex-based scalar functions that interpolate linearly across triangles, since our vector sets can serve as candidate gradient fields. Moreover, our representation generalizes [KCPS13],



**Figure 5:** Two examples of designing a 6-PolyVector field (black) from a sparse set of directional constraints (red). The field degenerates into a 4-PolyVector only in the vicinity of the degenerate constraints in the bottom part of both shapes.

which is a special case limited to the representation of  $N$ -RoSy fields. In their formulation, the vector sets are stored on vertices and interpolated on triangles. Since their discretization is not directly applicable to linear parametrization problems, we opt for a piecewise-constant discretization which is numerically more robust (it does not require Chebyshev expansions). Note that, much like all other piecewise-constant discretizations of  $N$ -RoSy fields, it is not convergent under refinement.

### 3.6. Constrained $N$ -PolyVector field design

$N$ -PolyVector fields can be interactively designed by specifying alignment constraints on a sparse set of faces and by automatically finding the globally optimal, smoothest field that interpolates them. Our design algorithm is divided into three steps:

1. The alignment constraints on the faces are converted to the PolyVector representation.
2. The coefficients of the polynomials are harmonically interpolated over the entire surface.
3. The directions per face are reconstructed by finding the roots of the per-face polynomials.

While the first step is trivial, since it only involves the computation of the coefficients of the polynomials which can be expressed in closed form, the other two steps are more involved and are described in detail in the next two paragraphs.

#### Harmonic interpolation of the polynomial coefficients.

We extend the well-known uniform face-based Laplace operator  $L : \mathbb{R}^{\mathcal{F}} \rightarrow \mathbb{R}^{\mathcal{F}}$  that operates on scalar functions on  $\mathcal{M}$  that are constant on faces. The operator can be written as a  $|\mathcal{F}| \times |\mathcal{F}|$  matrix:

$$L = \sum_{(f,g) \in \mathcal{E}} L_{(f,g)}, \quad (7)$$

$$\text{where } L_{(f,g)}[f, f] = -1, \quad L_{(f,g)}[f, g] = 1.$$

The  $L_{f,g}$  are  $|\mathcal{F}| \times |\mathcal{F}|$  matrices whose entries are zeros except entries  $[f, f]$  and  $[f, g]$  as specified above. We define a corresponding complex Laplacian matrix for our problem

by taking the parallel transport into account (see Section 3.3). Observe that Eq. (6) is separable with respect to the coefficients of the polynomial, and we can thus separately interpolate each coefficient. The Laplacian matrix  $L_m$  for the coefficient of degree  $m$  is similarly defined as:

$$L_m = \sum_{(f,g) \in e \in \mathcal{E}} L_{m,(f,g)}, \quad (8)$$

$$\text{where } L_{m,(f,g)}[f,f] = (\bar{e}_f)^m, \quad L_{m,(f,g)}[f,g] = -(\bar{e}_g)^m.$$

The polynomial coefficients are harmonically interpolated by solving a sparse linear system, which is prefactored using a sparse LU factorization:

$$L_m \hat{x} = 0, \quad \text{s.t. } \hat{x}_j = \hat{x}_j^0, \quad j \in \mathcal{C}, \quad (9)$$

where  $\hat{x}$  stacks all the degree- $m$  coefficients for all faces in a single vector. The variables that correspond to the user constraints ( $\mathcal{C}$  is the set of constrained faces) are fixed to the values  $\hat{x}^0$ , extracted from the directional constraints.

**Root factoring.** After obtaining the coefficient fields  $x_m$  for every monomial degree  $m$ , we can factor the face-based polynomials to produce the roots, i.e., the actual vector sets. The root-finding is done by finding the eigenvalues of the companion matrix [HJ85]. An important property arises from this computation: the coefficient of each monomial degree is computed independently from others, and since the interpolation is harmonic, it obeys the maximum principle. Therefore, if the interpolation constraints for a certain coefficient degree are all zero, the entire mesh will get a zero coefficient for that degree. For instance, if the provided constraints are purely  $N$ -RoSy, the entire interpolation result is  $N$ -RoSy by definition since only the constant coefficient ( $x_0$ ) will be interpolated. Moreover, degenerate configurations which are the result of coefficients  $x_m$ ,  $m \geq 1$ , with a large magnitude can only be obtained if forced by the user-provided constraints (see Figure 5).

#### 4. Frame fields

For the rest of the paper, we focus on the special case of *frame fields*, which are 4-PolyVector fields comprising two coupled (i.e., interchangeable) 2-RoSy fields  $u, v$ . They are represented by the following polynomial:

$$P_f(z) = (z^2 - u^2)(z^2 - v^2). \quad (10)$$

For brevity, we denote the resulting coefficients as  $x_1 = -(u^2 + v^2)$  and  $x_0 = u^2 v^2$ .

Frame fields are particularly important in geometry processing applications, since they can be used for anisotropic [PPTSH14] or planar [LXW\*11] quad remeshing. As we target anisotropic remeshing using frame fields, we detail two design options with our framework: computing fields with bounds on angles between  $u$  and  $v$  (Section 4.2), and computing conjugate vector fields (Section 4.3), that are used to generate PQ meshes for architectural rationalization.

The constraints for all our results (unless explicitly stated in the captions) were generated by sparsely sampling from the principal directions. The constrained faces can be picked from mesh regions that are most curved, or closer to being parabolic [BZK09].

##### 4.1. Frame field coefficients

The complex coefficients  $x_1, x_0$  have an intuitive geometric meaning that derives from the general property explored in Section 3.4. Cross fields, represented by the polynomials of the form  $P(z) = z^4 - w^4$ , are special cases of frame fields in which  $x_1 = -(u^2 + v^2) = 0$ . In this case, the coefficient  $x_0 = -w^4$  is a dual to the represented cross field  $w^4$ . To put it geometrically,  $x_0$  is the bisector cross-field to  $w^4$ . A more general claim is the following:

**Lemma 4.1** Given a set of two general 2-RoSy,  $u$  and  $v$ , the coefficient  $x_0 = u^2 v^2$  represents the cross of bisectors between  $u$  and  $v$  by its root set  $\{\sqrt[4]{|x_0|} \exp(i \frac{k\pi}{2}) \mid 0 \leq k \leq 3\}$ .

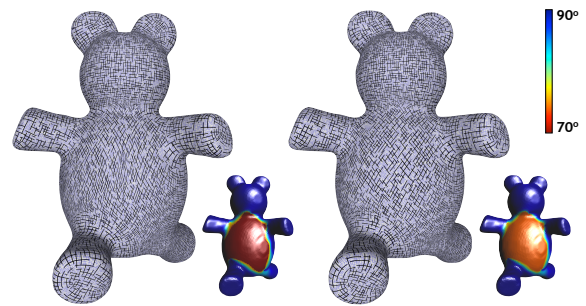
The coefficient  $x_1 = -(u^2 + v^2)$  represents the deviation of  $u, v$  from forming a perfect cross field.

*Proof.* One bisector between  $u$  and  $v$  is the result of rotating mid-way between the two, i.e.,  $b = u \sqrt{v \bar{u}}$ , and we immediately get  $b^4 = |u|^2 u^2 v^2$ . Thus  $b$  is one of the bisectors up to magnitude, and  $(x_0)^4$  represents the bisector cross field.  $\square$

The exact nature of the coefficient  $x_1$  is a bit harder to discern in general. However, notice that when both  $u$  and  $v$  have unit length, we may apply a rotation to  $u$  and  $v$  so that the bisector  $x_0$  becomes the canonical basis for the 2D plane. Then, the new vectors  $u', v'$  are complex conjugates  $v' = \bar{u}'$ , and  $x_1 = u'^2 + \bar{u}'^2 = 2 \cos(2\psi)$ , where  $\psi$  is the angle between  $u$  or  $v$  and their bisector, measuring the deviation from being a perfect cross.

##### 4.2. Angle-bounded fields

When designing frame fields for the purpose of quadrilateral remeshing, it is useful to bound the minimal angle in each



**Figure 6:** A smooth 4-PolyVector field (left) is optimized to satisfy a global angle bound of 70 degrees (right). Note the small angles in the left model are caused by manually placing constraints with an angle defect in the torso.



frame to help avoid creating degenerate quadrilaterals. We present an algorithm that efficiently generates a smooth frame field that satisfies a given minimal angle bound in each mesh face.

**Formulation.** We can express the bound on the angle as a nonlinear inequality constraint. Let  $\theta_{\min}$  be a given angle tolerance; we wish to find a field which satisfies:

$$\langle u, v \rangle \leq |u| |v| \cos \theta_{\min}. \quad (11)$$

Unfortunately, there is no trivial way to express this angle bound in terms of our coefficients  $x_0, x_1$ . However, note that our conditions are local and per-face, whereas our smoothness, measured in  $x_0, x_1$ , is global. We can therefore employ an effective local-global algorithm, as described next.

**Algorithm.** Our algorithm iteratively alternates between two steps: (1) A local step, which changes the field per face to the closest field that satisfies the angle bound constraints, followed by (2) a global step that generates a smooth field that is close to the field computed in step (1).

In the local step, we extract the  $u, v$  per face from the result of the previous global step, and rotate them until they agree with the angle bound. We find the bisector vector that partitions the smallest angle  $\theta$  between any pair of vectors  $\{\pm u, \pm v\}$ . The angle between each of those vectors and the bisector is then  $\theta/2$ . To locally enforce the minimum angle bound  $\theta_{\min}$ , if  $\theta < \theta_{\min}$ , we symmetrically rotate the two vectors away from the bisector.

In the global step, we compute the coefficients  $\hat{x}_0, \hat{x}_1$  per face from the  $u, v$  obtained in the local step, and look for a solution which is both smooth and close to the solution computed in the local step. Therefore, we minimize the following smoothness energy, with an additional term that attempts to align to the prescribed frames:

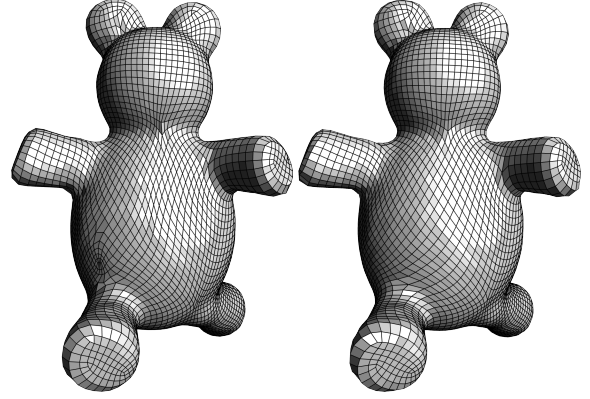
$$E_{\text{global}}^m = E_{\text{smooth}}^m + \lambda |x_m - \hat{x}_m|^2, \quad (12)$$

where  $\lambda$  is initialized at 100, and is multiplied by 1.5 at each step, to ensure convergence.

Note that any solution obtained from the local step is already guaranteed to satisfy the angle bound. However, the global step might violate it by enforcing smoothness. We consider the algorithm as having converged when the global step does not cause angle bound violations (in which case the local step does not change the field).

**Convergence.** In our experiments, the algorithm converges in 50-100 iterations, always enforcing the angle bound in all faces, with minimal impact on the field smoothness, as shown in Figure 6. Note that the field is practically unaffected on the faces that already satisfy the bound.

**Anisotropic quad remeshing.** We apply the angle bound to improve anisotropic quad remeshing in Figure 7. We compare a smooth field, interpolated from a sparse set of constraints, with the the result of its optimization to satisfy an angle



**Figure 7:** Enforcing an angle bound on a frame field results in an anisotropic quadrilateral mesh with a higher quality (right) than the mesh obtained with a field that is as smooth as possible (left). The fields used to guide the quadrangulation are shown in Figure 6.

bound of 70 degrees. The quality of the elements increases considerably, generating a better-quality mesh that is more suitable for numerical simulation or Catmull-Clark subdivision. The anisotropic quadrangulation is created using the deformation-based algorithm proposed in [PPTSH14].

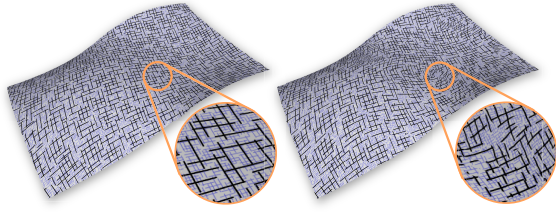
#### 4.3. Conjugate vector fields

In a nutshell, conjugate vector fields describe directions on a manifold that correspond to infinitesimally small planar quads. Therefore, creating a quadrangulation where opposite edges are aligned with conjugate directions is a suitable approximation of planar quads and a good starting point for a quad planarization algorithm. We briefly detail the definition of conjugacy and propose an algorithm to efficiently generate conjugate vector fields using 4-PolyVector fields while satisfying user-provided constraints.

Two tangent vectors  $p, q$  in the same tangent space are conjugate iff they are orthogonal w.r.t. the second fundamental form on that tangent space:  $\mathbb{I}(p, q) = 0$ . In a tangent space parametrization, this amounts to  $[p_x, p_y] S [q_x, q_y]^T = 0$ , where  $S \in \mathbb{R}^{2 \times 2}$  is the shape operator [LPW\*06]. Two vectors are both orthogonal (in the metric) and conjugate iff they are the principal directions at that point, since these are the eigenvectors of the shape operator. Note that the shape operator is only positive definite in elliptic regions (where the Gaussian curvature is positive), and that it is scalar in umbilical regions where both principal curvatures  $\kappa_1, \kappa_2$  are equal. Therefore, asymptotic directions in hyperbolic regions are self-conjugate, and all directions in a parabolic region are conjugate to the direction of zero curvature.

**Formulation.** Following [LXW\*11], we express the conjugacy condition in a face as:

$$\kappa_1 (u^T d_1) (v^T d_1) + \kappa_2 (u^T d_2) (v^T d_2) = 0 \quad (13)$$



**Figure 8:** A smooth frame field (left) is optimized into a conjugate frame field (right).

where  $\kappa_1, \kappa_2$  are the principal curvatures,  $d_1, d_2$  are the corresponding, normalized principal directions, and  $u$  and  $v$  are two vectors of the frame field in explicit column vector form. Since the principal directions are fixed, we can rewrite Eq. (13) as a real quadratic form w.r.t. the coordinates of  $u$  and  $v$ :

$$(u, v)^T H (u, v) = 0, \text{ where} \quad (14)$$

$$H = \begin{pmatrix} 0 & G_1 \\ G_2 & 0 \end{pmatrix}, \quad G_j = \kappa_j d_j d_j^T. \quad (15)$$

Similarly to the angle bound, minimizing the PolyVector smoothness energy subject to these nonlinear constraints is difficult in general, and a direct nonlinear optimization is slow and prone to getting stuck at local minima. Fortunately, the constraints exhibit the same local-global properties as the angle-bound constraints and we can use a similar algorithm.

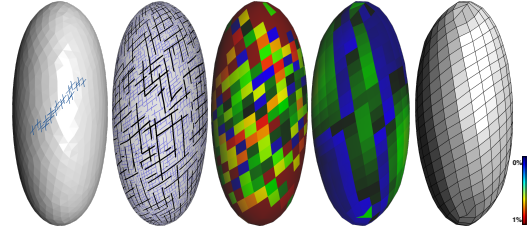
**Local step.** In the local step, we want to find  $u, v$  closest to given  $u_0, v_0$ , obtained from the global step, so that Eq. (13) is satisfied. The problem reduces to projecting  $u_0, v_0$  on the quadric parametrized by  $H$ , which is a specific *quadratically-constrained quadratic program* problem. We explain the details of this projection in Appendix A.

**Global step.** The global step is similar to that of the angle bound algorithm (Eq. (12)), but with a meaningful change: principal curvatures and directions are less reliable in near-planar regions, where the shape operator is almost an identity. Moreover, there is a greater flexibility in the generation of planar quads in such regions, and we should thus weigh smoothness over strict conjugacy there. We therefore compute the absolute Gaussian curvature of each face:  $\forall f \in \mathcal{F}, K_f = |\kappa_{f,1} \kappa_{f,2}|$ , normalize it over the entire mesh so that  $\hat{K}_f \in [0, 1]$ , and adapt the data term in the global energy accordingly. We then get the following:

$$E_{\text{global}}^m = E_{\text{smooth}}^m + \lambda K |x_m - \hat{x}_m|^2, \quad (16)$$

where  $K = \text{diag}(\hat{K}_f)$ .

**Convergence.** We stop the iterations when the sum of the absolute values of the residuals of Eq. (13) is less than  $10^{-4}$  (in our experiments, the input field is rescaled so that  $u, v$  are close to having unit length). The algorithm converged in less than 50 iterations for all our examples. We show the result of this optimization in Figure 8. The input is a smooth field, not



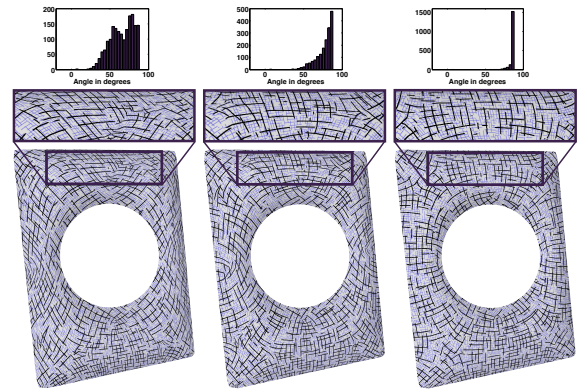
**Figure 9:** Our algorithm supports constraints which are not aligned with principal directions. Left to right: directional constraints (generated by manually drawing one of the directions and setting the other to its conjugate), optimized smooth conjugate field, planarity before and after planarization, final quad mesh.

aligned with the principal directions. The iterations produce a conjugate field that is both smooth and nearly-orthogonal in places where the principal curvatures are strong. Our algorithm also handles directional constraints which are not necessarily aligned with the principal directions (see Figure 9). We next show how to regularize orthogonality and avoid asymptotic directions.

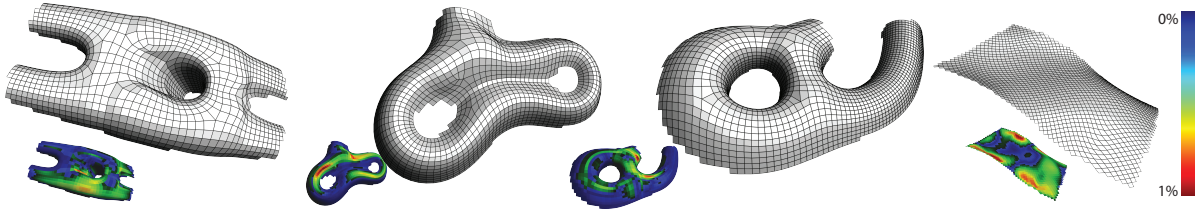
**Orthogonality control.** We can additionally control the orthogonality of the frame field by minimizing the magnitude of the coefficient  $x_1$  (recall from Section 4.1 that when  $x_1$  is zero, the field is a perfect cross field). This is achieved by adding a term to the energy of the global step for the  $x_1$  coefficient (Eq. (12)):

$$E_{\text{orth}}^1 = E_{\text{global}}^1 + w |x_1|^2. \quad (17)$$

In Figure 10, we show the effect of varying the parameter  $w$ ; as the value increases, the field becomes more similar to a cross field, and aligns to the principal curvature directions, since it is constrained to stay conjugate.



**Figure 10:** Left to right: conjugate fields regularized for orthogonality with different values for  $w$  (0, 1 and 10). The rightmost example is extreme: the regularization dominates and leads to a field that is close to the principal directions.



**Figure 11:** Planar quad remeshing of architectural models with our framework. The colors represent face planarity values.

Optimizing for orthogonality can be particularly beneficial for the design of conjugate vector fields, as it avoids asymptotic self-conjugate directions that can occur in hyperbolic regions. However, an excessive regularization of  $x_1$  simply reduces the field to a cross field, which greatly restricts the design space, leading to a result similar to the principal direction field, which can be noisy and unstable.

**Planar quadrangulation.** We demonstrate the effectiveness of our conjugacy optimization by generating planar quad remeshing of a variety of architectural surfaces (Figures 1 and 11). We start by generating a smooth frame field that is aligned with a sparse set of constraints extracted from curvature using the thresholding algorithm of [BZK09]. The field is optimized for conjugacy until convergence and used to guide a mixed-integer parametrization [BZK09]. The quad mesh is then extracted with [EBCK13] and planarized with [LPW\*06] until it reaches the desired planarity threshold of 1% (the percentual ratio of the distance between the two diagonals in the quad to the average diagonal length).

#### 4.4. Comparisons

**Frame field interpolation.** [PPTSH14] proposes an algorithm to interpolate frame fields for the purpose of anisotropic quad remeshing. Our algorithm produces very similar results when the same constraints are used, as shown in Figure 12. The two fields are very similar, with a tendency of our method to add a few extra singularities in highly curved regions. The main advantage of [PPTSH14] is that it can interpolate scale and non-orthogonality separately, but it comes at the price of not being extendable to general PolyVector fields.

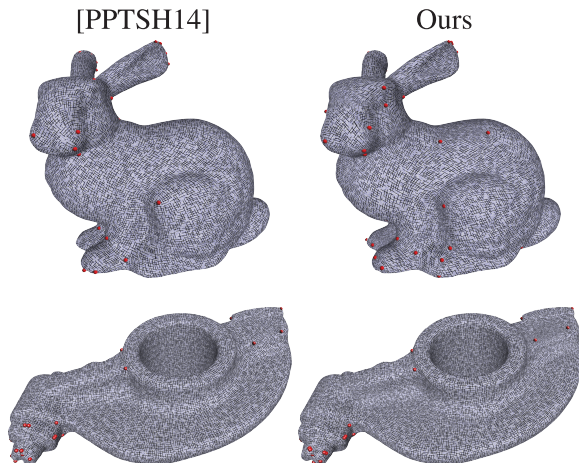
**Conjugate vector fields.** [LXW\*11] proposes an algorithm to generate conjugate vector fields. As we demonstrated above, our algorithm can also be used for the same purpose and it generates slightly smoother results, as shown in Figure 13. While the results are comparable, our algorithm is simpler to implement as it does not require the solution of a difficult nonlinear optimization problem.

#### 5. Concluding remarks

We formally introduced  $N$ -PolyVector fields, including their topology, parallel transport, smoothness and singularities. We demonstrated that they can be employed in many practical applications in geometry processing, such as anisotropic and planar remeshing.

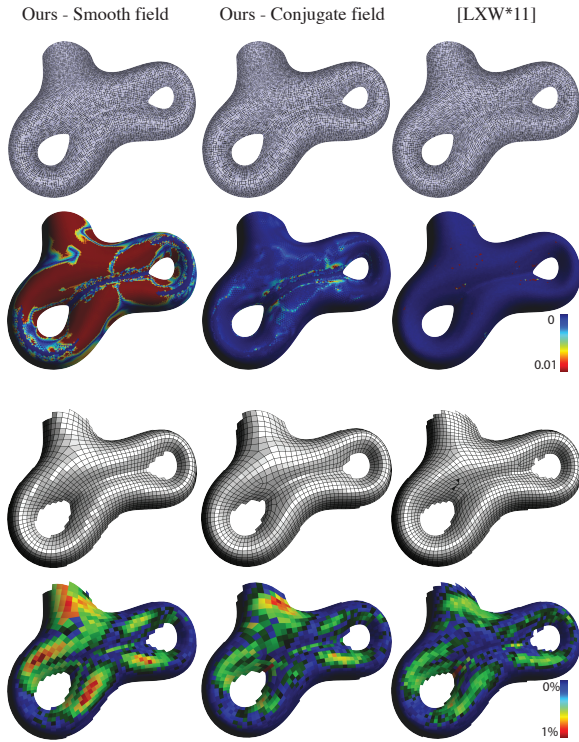
**Limitations.** Our framework has two major limitations: the first is that, similarly to [KCPS13], we do not handle singularities larger than  $\pm \frac{1}{N}$ , as our singularities are measured as arguments of numbers to the power of  $N$ , which cannot express angles greater than  $\pm \pi$  unless the angles are treated as variables. A possible direction for future work here is to enforce such singularities (as in [CDS10]) and incorporate the prescribed rotation into the same framework.

The second limitation is that in our framework, scale is coupled with rotation. Thus, when the constraints are assigned very sparsely, the energy might suppress the effect of rotations by reducing the scale (see Figure 14 for such an example). This is not the case in methods that explicitly interpolate rotations, such as [PPTSH14]. Fortunately, we do not observe the same phenomenon when using constraints that are reasonably distributed over the mesh, even when they are quite different from each other in terms of alignment.



**Figure 12:** Our design algorithm (right) generates a result similar to [PPTSH14] (left) for the special case of frame fields.

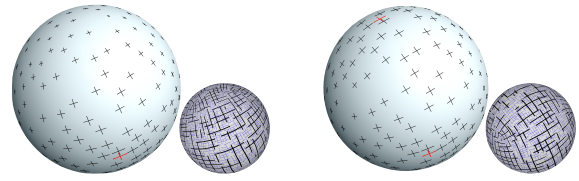




**Figure 13:** We generate a quad mesh from three different fields: a smooth N-PV field (left column), an optimized conjugate field (middle column), both generated with our algorithm, and a conjugate field optimized with [LXW\*11] (right column). The different stages of the algorithms are depicted in the rows. Top to bottom: frame field, per-face deviation from conjugacy, planarized quad mesh, face planarity values.

**Continuity and smoothness.** Our algorithm creates smooth fields when the variables are the PolyVectors coefficients, by minimizing their Dirichlet energy. However, given an explicit matching, the Dirichlet energy of the individually-matched roots (single vectors) is not necessarily minimized when the energy of the PV coefficients is. It is well-known [HM87] that the roots are a continuous function of the polynomial coefficients, but discerning the exact measure of root-based smoothness is an involved analysis, and we leave the exact derivation for future work. We have not found any case of instability in root-based smoothness in practice.

**Future work.** The most direct continuation of this work is to study *integrable* PolyVectors fields that would allow generating parameterizations whose gradients match the PolyVector fields exactly. In addition, quality parameterization should require the study of vector-set fields that guarantee globally bijective parameterizations. We further plan to apply our method in other scenarios that require independent vector-set fields, such as deformations, physical simulations (with a proper generalization to three dimensions) and field design that respects symmetries on surfaces [PLPZ12].



**Figure 14:** If the constraints are very sparse (left), we observe a reduction in scale, caused by our smoothness energy. However, this is not a practical problem in our experiments, since this phenomenon disappears with just a few more constraints (right).

## Acknowledgments

We thank Mirela Ben-Chen for her insightful comments, Helmut Pottmann and Johannes Wallner for their major support, and Evolute GmbH, Alec Jacobson, Takeo Igarashi and Yang Liu for providing the models for our experiments. This work was supported in part by the ERC Starting Grant iModel (StG-2012-306877), and by the FWF Lise-Meitner grant M1618-N25.

## References

- [BCBSG10] BEN-CHEN M., BUTSCHER A., SOLOMON J., GUIBAS L.: On discrete Killing vector fields and patterns on surfaces. *Comput. Graph. Forum* 29, 5 (2010), 1701–1711. 1
- [BLP\*13] BOMMES D., LÉVY B., PIETRONI N., PUPPO E., SILVA C., TARINI M., ZORIN D.: Quad-mesh generation and processing: A survey. *Comput. Graph. Forum* 32 (2013), 51–76. 1, 2
- [BS08] BOBENKO A. I., SURIS Y. B.: *Discrete differential geometry. consistency as integrability*, vol. 98. American Math. soc., 2008. 2
- [BZK09] BOMMES D., ZIMMER H., KOBELT L.: Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), 77:1–77:10. 2, 6, 9
- [CDS10] CRANE K., DESBRUN M., SCHRÖDER P.: Trivial connections on discrete surfaces. *Comput. Graph. Forum* 29, 5 (2010), 1525–1533. 2, 4, 9
- [dC76] DO CARMO M. P.: *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976. 2
- [EBCK13] EBKE H.-C., BOMMES D., CAMPEN M., KOBELT L.: Qex: Robust quad mesh extraction. *ACM Trans. Graph.* 32, 6 (2013), 168:1–168:10. 9
- [HJ85] HORN R. A., JOHNSON C. R.: *Matrix analysis*. Cambridge University Press, 1985. 6
- [HM87] HARRIS G., MARTIN C.: Shorter notes: The roots of a polynomial vary continuously as a function of the coefficients. *Proceedings of the American Mathematical Society* 100, 2 (1987), pp. 390–392. 10
- [HZ00] HERTZMANN A., ZORIN D.: Illustrating smooth surfaces. In *Proc. ACM SIGGRAPH* (2000), pp. 517–526. 2
- [KCPS13] KNÖPPEL F., CRANE K., PINKALL U., SCHRÖDER P.: Globally optimal direction fields. *ACM Trans. Graph.* 32, 4 (2013), 59:1–59:10. 2, 5, 9
- [LH06] LEFEBVRE S., HOPPE H.: Appearance-space texture synthesis. *ACM Trans. Graph.* 25, 3 (2006), 541–548. 1



- [LPW\*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph.* 25, 3 (2006), 681–689. [2](#), [7](#), [9](#)
- [LXW\*11] LIU Y., XU W., WANG J., ZHU L., GUO B., CHEN F., WANG G.: General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.* 30, 6 (2011). [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [9](#), [10](#)
- [LZX\*08] LIU L., ZHANG L., XU Y., GOTSMAN C., GORTLER S. J.: A local/global approach to mesh parameterization. In *Proceedings of SGP* (2008), pp. 1495–1504. [1](#)
- [NPPZ12] NIESER M., PALACIOS J., POLTHIER K., ZHANG E.: Hexagonal global parameterization of arbitrary surfaces. *IEEE Trans. Vis. Comput. Graph.* 18, 6 (2012), 865–878. [1](#), [2](#)
- [PBSH13] PANOZZO D., BLOCK P., SORKINE-HORNUNG O.: Designing unreinforced masonry models. *ACM Trans. Graph.* 32, 4 (2013), 91:1–91:12. [1](#), [2](#)
- [PLPZ12] PANOZZO D., LIPMAN Y., PUPPO E., ZORIN D.: Fields on symmetric surfaces. *ACM Trans. Graph.* 31, 4 (2012), 111:1–111:12. [10](#)
- [PPTSH14] PANOZZO D., PUPPO E., TARINI M., SORKINE-HORNUNG O.: Frame fields: Anisotropic and non-orthogonal cross fields. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 33, 4 (2014). [2](#), [6](#), [7](#), [9](#)
- [PZ07] PALACIOS J., ZHANG E.: Rotational symmetry field design on surfaces. *ACM Trans. Graph.* 26, 3 (2007). [2](#)
- [RVAL09] RAY N., VALLET B., ALONSO L., LEVY B.: Geometry-aware direction field processing. *ACM Trans. Graph.* 29 (2009), 1:1–1:11. [2](#)
- [RVLL08] RAY N., VALLET B., LI W. C., LÉVY B.: N-symmetry direction field design. *ACM Trans. Graph.* 27, 2 (2008), 10:1–10:13. [2](#), [4](#)
- [TPSHSH13] TAKAYAMA K., PANOZZO D., SORKINE-HORNUNG A., SORKINE-HORNUNG O.: Sketch-based generation and editing of quad meshes. *ACM Trans. Graph.* 32, 4 (2013), 97:1–97:8. [1](#)
- [ZSW10] ZADRAVEC M., SCHIFTNER A., WALLNER J.: Designing quad-dominant meshes with planar faces. *Comput. Graph. Forum* 29, 5 (2010), 1671–1679. [2](#)

consequently:

$$y = U(I + \lambda \Sigma)^{-1} U^T y_0 \quad (21)$$

We next plug Eq. (21) into our condition  $y^T H y = 0$  to obtain the following:

$$\begin{aligned} y_0^T U(I + \lambda \Sigma)^{-1} U^T H U(I + \lambda \Sigma)^{-1} U^T y_0 &= 0 \Leftrightarrow \\ y_0^T U(I + \lambda \Sigma)^{-1} \Sigma(I + \lambda \Sigma)^{-1} U^T y_0 &= 0 \Leftrightarrow \\ z^T D z &= 0, \end{aligned} \quad (22)$$

where  $z = U^T y_0$ . Since  $(I + \lambda \Sigma)^{-1}$  is diagonal, we have a simple expression for  $D$ :

$$D = \text{diag} \left( \frac{\sigma_m}{(1 + \lambda \sigma_m)^2} \right), \quad (23)$$

where  $\sigma_m$  are the diagonal elements of  $\Sigma$ . As a result, we need to solve:

$$\sum_m z_m^2 \frac{\sigma_m}{(1 + \lambda \sigma_m)^2} = 0. \quad (24)$$

This is a rational equation in  $\lambda$ . We solve it by computing the polynomial numerator and extracting its roots.

## Appendix A: Projection on a quadric

As mentioned in Section 4.3, finding the nearest conjugate vectors for prescribed vectors reduces to the problem of projecting a point  $y_0$  onto a quadric  $y^T H y = 0$ , where the quadric is only assumed to be symmetric, but not necessarily positive or negative definite (since it stems from the shape operator). We solve the following problem:

$$\begin{aligned} \min \|y - y_0\|^2 \\ \text{s.t. } y^T H y = 0. \end{aligned} \quad (18)$$

We solve this nonlinear problem by Lagrange multipliers. The Lagrangian of the system is:

$$\Lambda(y, \lambda) = y^T y + 2y_0^T y + \lambda y^T H y. \quad (19)$$

The stationary points of the Lagrangian are:

$$2y + 2y_0 + 2\lambda H y = 0 \Leftrightarrow y = (I + \lambda H)^{-1} y_0 \quad (20)$$

Since  $H$  is symmetric, its singular value decomposition is  $H = U \Sigma U^T$ . We then have that  $I + \lambda H = U(I + \lambda \Sigma)U^T$ , and