

# Geometry-Aware Direction Field Processing

NICOLAS RAY, BRUNO VALLET, LAURENT ALONSO, and BRUNO LEVY  
INRIA

Many algorithms in texture synthesis, nonphotorealistic rendering (hatching), or remeshing require to define the orientation of some features (texture, hatches, or edges) at each point of a surface. In early works, tangent vector (or tensor) fields were used to define the orientation of these features. Extrapolating and smoothing such fields is usually performed by minimizing an energy composed of a smoothness term and of a data fitting term. More recently, dedicated structures ( $N$ -RoSy and  $N$ -symmetry direction fields) were introduced in order to unify the manipulation of these fields, and provide control over the field's topology (singularities). On the one hand, controlling the topology makes it possible to have few singularities, even in the presence of high frequencies (fine details) in the surface geometry. On the other hand, the user has to explicitly specify all singularities, which can be a tedious task. It would be better to let them emerge naturally from the direction extrapolation and smoothing.

This article introduces an intermediate representation that still allows the intuitive design operations such as smoothing and directional constraints, but restates the objective function in a way that avoids the singularities yielded by smaller geometric details. The resulting design tool is intuitive, simple, and allows to create fields with simple topology, even in the presence of high geometric frequencies. The generated field can be used to steer global parameterization methods (e.g., QuadCover).

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; G.1.6 [Numerical Analysis]: Optimization; J.6 [Computer Applications]: Computer Aided Engineering

General Terms: Algorithms

Additional Key Words and Phrases: Vector field design, topology, filtering

## ACM Reference Format:

Ray, N., Vallet, B., Alonso, L., and Levy, B. 2009. Geometry-Aware direction field processing. *ACM Trans. Graph.* 29, 1, Article 1 (December 2009), 11 pages. DOI = 10.1145/1640443.1640444 <http://doi.acm.org/10.1145/1640443.1640444>

**Introduction.** Many algorithms in computer graphics require to decorate a surface with various visual features. Most of the time, the algorithm has to answer the question “How do I orient these features on the surface?”. For texture synthesis [Lefebvre and Hoppe 2006; Turk 2001], a smooth tangent vector field can provide the orientation to give to the sample image onto the surface. However, feature orientation on a surface cannot always be defined by a tangent vector field. For example, in nonphotorealistic rendering [Praun et al. 2001], the orientation of strokes is defined up to a rotation of  $\pi/2$ , that is, it is given by two orthogonal directions (which is called a “cross field” in Hertzmann and Zorin [2000]). The same type of fields is also needed to orient edges in quad-dominant remeshing [Alliez et al. 2003], or to determine the orientation of iso- $u$  and iso- $v$  in global parameterization [Ray et al. 2006; Kalberer et al. 2007]. Li et al. [2006b] have unified the representation of all types of orientations by introducing the notion of  $N$ -symmetry direction field. The orientation is then given by a set of  $N$  unit vectors in the tangent plane of the surface such that turning them by  $2\pi/N$  generates the same set of vectors.

The first processing algorithms for direction fields [Praun et al. 2001; Ray et al. 2006] are based on the minimization of the field curvature estimated by the angle deviation between the direction sampled on adjacent triangles (or vertices). The benefits of this approach are that it is easy to handle user-defined directional constraints and to smooth an existing field (usually initialized with the main directions of the curvature tensor). Another important feature is that the singularities of the field (pole, saddles, bisectors, trisectors, etc.) are automatically generated in a way that minimizes the energy. On the one hand, this is interesting because the field topology captures the shape of the object by placing singularities such as poles at the tips of fingers in Figure 3. On the other hand, geometric details will yield many singularities (see Figure 1, left). Such a complex topology is difficult to manage in the application. For example, in quad-remeshing, each singularity will generate an extraordinary vertex (or facet).

Undesirable singularities can be avoided by algorithms that control the direction field topology [Ray et al. 2008; Palacios and Zhang 2007], but at the expense of losing the ability of automatically

Authors' address: N. Ray (contact author), B. Vallet, L. Alonso, B. Levy, INRIA, 615 rue du jardin botanique, CS 20101, 54603 Villers les Nancy Cedex, France; email: ray@loria.fr.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).  
© 2009 ACM 0730-0301/2009/12-ART1 \$10.00  
DOI 10.1145/1640443.1640444 <http://doi.acm.org/10.1145/1640443.1640444>

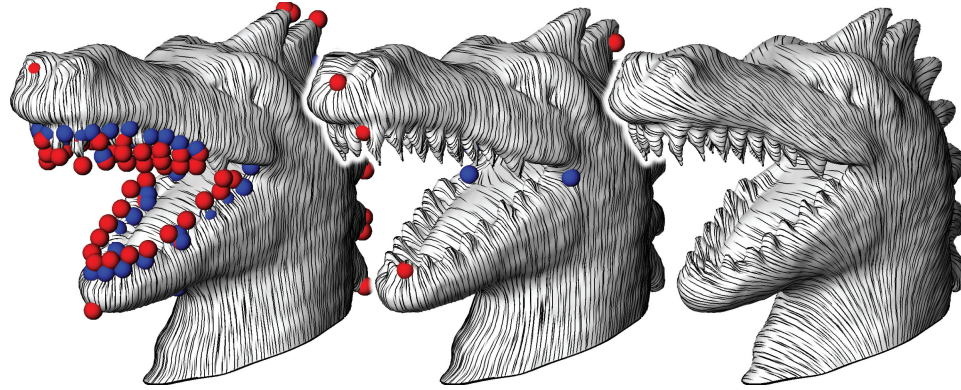


Fig. 1. Our new direction field processing method lets the user choose the size of geometric features for which the algorithm is allowed to create singularities. Classical algorithms can either automatically place (too many) singularities (left), or avoid creating any singularity (right) at the expense of distorting the field a lot (like in the neck of the monster). In this article, we present a trade-off between these two extreme configurations that places singularities only for significant features (middle).

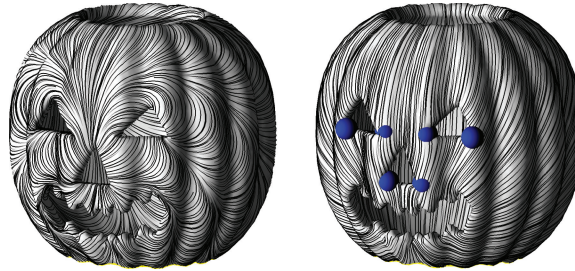


Fig. 2. Generating a direction field without any singularity may lead to high curvature of the field (left). Introducing singularities (saddles represented by blue spheres) allows to minimize the field curvature (right).

capturing the object shape. As a consequence, the user must manually set all singularities, which can be a painful task for complex objects.

As illustrated in Figure 2, there is a trade-off between the field curvature and the number of singularities. As a consequence, methods that are solely based on curvature minimization may generate a large number of singularities whereas in methods that provide topology control, the user must carefully place singularities in order to avoid large field curvature. In practice, current solutions to define the field topology either suffer a lack of user control, or require too much user interaction. This article proposes an intermediate solution that is able to automatically generate the field topology, but places only singularities that capture “meaningful” geometric features. The user can control what “meaningful” means by setting a minimal feature size for which the algorithm is allowed to create a singularity, and can edit the field topology by moving singularities. For example, in Figure 1 (middle) the teeth of the dinosaur are considered to be too small to be “meaningful” features, whereas singularities are still generated to capture the global shape of the head.

**Problem Statement.** Our objective is to design a direction field processing algorithm that is easy to use and gives suitable results for applications. The desired features for such a tool are listed next.

- (1) *Direction smoothness.* The direction field should be as smooth as possible. This ensures visual quality, but also makes singularities emerge naturally. Since only orientation, and not size, matters in these applications, it is more meaningful to

smooth *directions* (normalized vectors) than *vectors*. Notice that the field curvature (variation of orientation) captures both the straightness of streamlines (in the direction of the streamlines) and the parallelism of streamlines (in the orthogonal direction).

- (2) *Rotational symmetry.* The orientation of features is generally defined by  $N$ -symmetry directions (sets of  $N$  directions that form angles multiple of  $2\pi/N$ ). In particular 1,2 and 4-symmetry direction fields correspond to direction ( $\rightarrow$ ), line ( $-$ ), and cross ( $+$ ) fields. Line fields are used in surface hatching and cross fields in quad remeshing and global parameterization. Using  $N$ -symmetry leads to a more accurate control of the field topology.
- (3) *Geometry control.* We want either to extrapolate user constraints or to smooth an existing field such as the principal curvature directions field. In both cases this will result in hard or smooth constraints on the geometry of the field.
- (4) *Topology control.* It is very important for the direction field to have a minimum number of singularities that will capture the global shape of the object. Explicit control can require too much user interaction, whereas no topology control at all may lead to too many singularities. In practice, a good solution would automatically place singularities, but only for meaningful geometric features. Our goal is to define a new algorithm that offers a good trade-off between these two extreme configurations. In other words our algorithm will automatically place singularities only for “meaningful” geometric features.

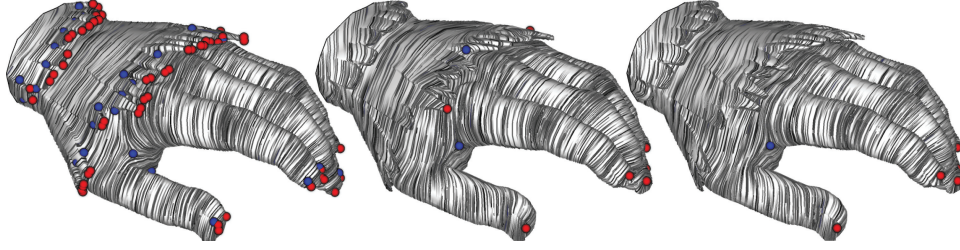


Fig. 3. Smoothing a 2-symmetry direction field on a surface with small geometric details. Left: Local smoothing creates many singularities. Middle: The singularity-merging strategy proposed in N-RoSy reduces the number of singularities but does not ensure that a global optimum will be found. Right: Our method allows to ignore all the details while being still sensitive to more global features (fingers).

Table I. Main Features of Direction Field Smoothing and Design

| Method        | Local | DTVF             | N-RoSy           | NSDF             | GADF             |
|---------------|-------|------------------|------------------|------------------|------------------|
| Normalized    | both  | no               | no               | yes              | yes              |
| Symmetries    | 1,2,4 | 1                | $\mathbb{N}^*$   | $\mathbb{N}^*$   | $\mathbb{N}^*$   |
| Geometry ctrl | yes   | yes <sup>1</sup> | yes              | yes <sup>3</sup> | yes              |
| Topology ctrl | no    | no <sup>1</sup>  | yes <sup>2</sup> | yes <sup>2</sup> | yes <sup>2</sup> |

<sup>1</sup>The Design of Tangent Vector Field (DTVF) approach ensures topology control only in case where there are no constraints (no geometry control).

<sup>2</sup>In N-RoSy the topology is controlled a posteriori (after the smoothing) whereas in N-Symmetry Direction Field (NSDF) it is controlled a priori. In this work (GADF), topology is controlled implicitly by minimizing the number of singularities.

<sup>3</sup>Geometric constraints can only be applied a posteriori in NSDF by modifying a given field so it cannot be used for smoothing an existing field.

**Contributions.** Our main contribution is a direction field processing framework that combines all the aforementioned features. It provides a trade-off between *geometric* algorithms that only aim at maximizing the smoothness of a direction field without control over the topology and *topological* algorithms that smooth the direction field with explicit control over its topology. *Geometric* algorithms usually generate too many singularities when the geometry is complex (high frequencies) while *topological* algorithms require the user to manually control all the topology. In contrast, our algorithm allows to significantly reduce the number of singularities due to high geometric frequencies while letting a global topology emerge from the smoothing.

The *geometric* limit of our algorithm is a classical local smoothing algorithm similar to the one used in Li et al. [2006a]. The *topological* limit sacrifices smoothness to guarantee a simple topology. In particular, on topological disks and without directional constraints, the algorithm controls the position and type of singularities (see Section 3.1 for a proof).

**Previous Work.** We will now present existing works and their position with respect to the desired features we mentioned (see Table I). Note that in most works, generating a direction field was only considered as a preprocessing step for a more specific application.

Most previous approaches rely on local smoothing. In texture synthesis [Praun et al. 2000; Turk 2001], extrapolation of a tangent vector field is used to give a coherent orientation of the texture synthesized on the surface. In real-time hatching [Praun et al. 2001] and anisotropic remeshing [Alliez et al. 2003], the orientation of strokes (respectively quad edges) is defined by the principal axis of the curvature tensor. This tensor defines a direction modulo a rotation of  $\pi$ , that can be smoothed to remove meaningless singularities and avoid jitter effects in the application. More recently, Fisher et al. [2007] proposed a Discrete Exterior Calculus (DEC)-based

method allowing for Hodge decomposition to generate smooth 1-forms (equivalent to vector fields) used for texture synthesis. In quad remeshing, periodic global parameterization [Ray et al. 2006] introduces direction fields defined with a modulo of  $\pi/2$  to take into account the quad orientation invariance by rotation of  $\pi/2$ . In fact they can smooth a direction fields with rotational invariance of  $2\pi/N$ ,  $N \in \mathbb{N}$ , such as direction fields ( $N = 1$ ), line fields ( $N = 2$ ), or cross fields ( $N = 4$ ). This first family of algorithms does not provide fine control over the field topology, but smoothing and extrapolation are quite easy to achieve.

In most applications, the field singularities play an important role. For instance, a source in quad remeshing will generate a pole, that is, a vertex with valence  $\neq 4$  or a nonquad polygon. The same singularity in hatching will generate a point of convergence of strokes that has an important visual impact. Zhang et al. [2006] propose a vector field design able to repair a field after smoothing by moving singularities and removing pairs of singularities. They extend it [Palacios and Zhang 2007] to  $N$ -rotational symmetry (N-RoSy) field design. As the Poincaré Hopf theorem makes it impossible to introduce a single singularity without introducing another one with opposite index, their interface is based on moving singularities and placing directional constraints. As illustrated in Figure 3, the iterative pair cancellation strategy may not lead to an optimal topology simplification. Finally, Ray et al. [2008] present an  $N$ -Symmetry Direction Field (NSDF) design that solely focuses on the direction (does not take the vector norm into account) and provides exact control of the singularities.

The method introduced in this article is *geometry aware* in the sense that a precomputation step estimates the field distortion created by the geometry. It is then integrated into the objective function of a *geometric* algorithm that prevents high geometric frequencies from generating singularities.

This feature is critical for global parameterization applications as singularities should only capture the global shape of the object. Some algorithms [Tong et al. 2006; Kalberer et al. 2007; Ray et al. 2006] suggest to smooth the curvature tensor to automatically place the singularities. Another approach [Kharevych et al. 2006] uses the distortion of a mapping as heuristic to place singularities. This idea was extended in Ben-Chen et al. [2008] and Springborn et al. [2008] to work with an isotropic metric and to iteratively introduce singularities to reduce its distortion. As illustrated in Figure 4, such an iterative process does not lead to an optimal placement of singularities.

The rest of the article is organized as follows: Section 1 introduces the theory and structure of direction fields on triangulated surfaces and restates the objectives in this new formalism. Sections 2 and 3 present algorithms that, respectively, consider every geometric feature as meaningful (Figure 1, left) or as meaningless (Figure 1,



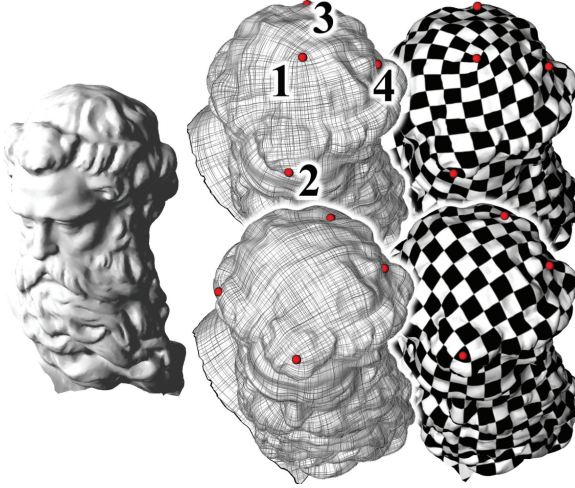


Fig. 4. To compute a global parameterization of Neptune head, cone singularities can be iteratively introduced (up), or computed simultaneously (down). Note how the second option better distributes them.

right). From these two extreme cases, we derive an intermediate algorithm in Section 4 (Figure 1, middle) that is able to consider geometric features larger than a user-defined size as meaningful.

## 1. DIRECTION FIELD ON A TRIANGULATED SURFACE

An  $N$ -symmetry direction field is the definition, for each point of the surface, of a set of  $N$  unit vectors of the tangent plane which is invariant by rotation of  $2\pi/N$ . As both its topology and geometric variations are defined by the angle deviation relative to parallel transport, it can be nicely represented on triangulated surfaces using the discrete exterior calculus formalism. In this section, we first recall discrete exterior calculus notations (Section 1.1) and explain how a direction field is sampled by a 0-form in the DEC formalism (Section 1.2). This allows us to define the field curvature as a 1-form using exterior derivation (Section 1.3) and to define the index of singularities as a 2-form using  $2^{nd}$ -order exterior derivatives (Section 1.4). The problem is then restated in this formalism (Section 1.5).

### 1.1 Discrete Exterior Calculus Notations

We assume that our mesh is oriented, that is, facets have coherent normals and each edge has an orientation. This allows us to define a unique orientation for dual edges. We use notations from Discrete Exterior Calculus (DEC) [Desbrun et al. 2005a] on the dual mesh  $M^* = (\mathcal{F}^*, \mathcal{E}^*, \mathcal{V}^*)$  as it makes the exposition clearer and the proofs easier. In this setting, 0-forms are scalars on dual vertices ( $\mathcal{F}^*$ ), 1-forms are scalars on oriented dual edges ( $\mathcal{E}^*$ ), and 2-forms are scalars on dual facets ( $\mathcal{V}^*$ ). Throughout the article, we will use the convention that a quantity indexed by 0, 1 or 2 is a 0, 1 or 2-form. Indices  $i, j, \dots$  will refer to vertices of the dual mesh (triangles of the primal mesh), and  $ij$  denotes the dual edge between primal triangles  $i$  and  $j$ . Finally, we will make use of the DEC norm for dual 1-forms defined as

$$\|f_1\|^2 = \sum_{ij \in \mathcal{E}^*} w_{ij}^{-1} f_1(ij)^2, \quad (1)$$

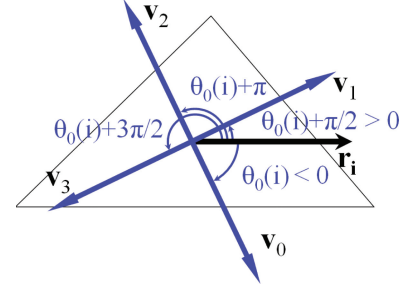


Fig. 5. A  $N$ -symmetry direction (here  $N = 4$ ) on a triangle  $T_i$  is a set of vectors  $\mathbf{v}_k, k \in \{0..N-1\}$  defined as the images of a reference vector  $\mathbf{r}_i$  by rotations of  $\theta_0(i) + k(2\pi/N)$ . By convention, negative angles correspond to clockwise orientation (here  $\theta_0(i) < 0$  and  $\theta_0(i) + \pi/2 > 0$ ).

where  $w_{ij} = \cot(\beta) + \cot(\beta')$  are the cotan weights [Pinkall and Polthier 1993] and  $\beta$  and  $\beta'$  are the two angles facing  $ij$ .

### 1.2 Direction Field Sampling

As shown in Figure 5, an  $N$ -symmetry direction  $\mathbf{v}^N$  on a smooth surface is defined as a set of  $N$  unit vectors lying in the tangent plane of the surface that is preserved by rotation of  $2\pi/N$  around the normal. We define our  $N$ -symmetry directions on the triangles of a mesh where the tangent space is well defined. In each triangle  $T_i$ , an oriented edge  $\mathbf{r}_i$  is used as the reference vector. A 0-form  $\theta_0$  then defines the  $N$ -symmetry direction  $\mathbf{v}_i^N$  on  $T_i$  as the set of images of  $\mathbf{r}_i$  by rotations (around the triangle normal) of  $\theta_0(i) + 2\pi k/N$ ,  $k \in \mathbb{Z}$ .

### 1.3 Direction Field Curvature

In continuous settings, the “curvature” of a direction field is a one form that for neighbor points  $A$  and  $B$  gives the angle difference between the field direction at point  $B$  and the field direction at point  $A$  parallel transported to point  $B$ . This notion, introduced in Ray et al. [2008] using covariant derivatives, is used to define the “smoothness” of a field as the squared norm of the curvature.

In discrete settings, let  $i$  and  $j$  be two adjacent triangles such that the dual edge  $ij$  is oriented from  $i$  to  $j$ . We can isometrically bring  $i$  and  $j$  to be in the same plane, and define the *curvature* 1-form  $C_1$  of the direction field  $\mathbf{v}^N$  along  $ij$  as the angle of rotation that brings  $\mathbf{v}_i^N$  to  $\mathbf{v}_j^N$ . Simple geometry (see Figure 6) shows that  $\theta_0$  only defines the curvature 1-form  $C_1$  up to integer multiples of  $2\pi/N$ . We have

$$C_1(\theta_0, p_1) = r_1 + d_0\theta_0 + 2\pi p_1/N, \quad (2)$$

where:

- $r_1(ij)$  is the angle of a rotation that brings  $\mathbf{r}_i$  to  $\mathbf{r}_j$ , given by:  $r_1(ij) = \angle(\mathbf{r}_i, ij) + \angle(ij, \mathbf{r}_j)$  where  $\angle$  is the angle oriented by the triangle normal. Defining  $r_1(ji) = -r_1(ij)$  makes  $r_1$  a 1-form;
- $d_0$  is the exterior derivative for 0-forms given by  $(d_0\theta_0)(ij) = \theta_0(j) - \theta_0(i)$ ;
- $p_1$  is an 1-form such that  $p_1(ij)$  is an integer for each  $ij$ . This property will be used further to prove the validity of our method (see Section 3.1). This variable is equivalent to the “period jumps” [Ray et al. 2008]: it determines how directions are interpolated by removing the modulo  $2\pi/N$  (see Figure 6).

The couple  $(\theta_0, p_1)$  along with the choice of reference vectors defines a unique direction field with a well-defined curvature. This allows us to define the indices of its singularities.



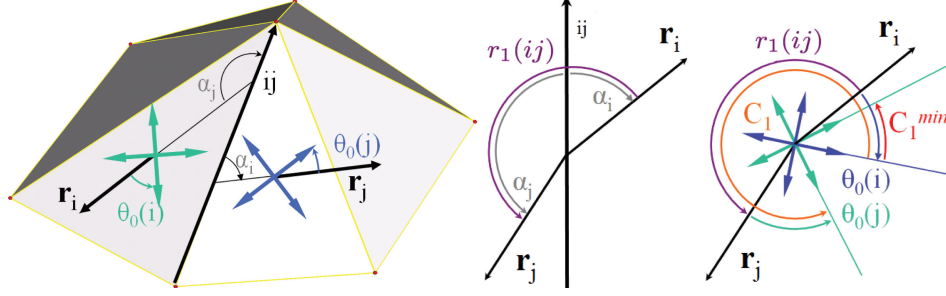


Fig. 6. Left: 4-symmetry directions on adjacent facets with common edge  $e^{ij}$ . Middle: The rotation angle between reference vectors  $\mathbf{r}_i$  and  $\mathbf{r}_j$  is  $r_1(ij) = \alpha_j - \alpha_i$ . Right:  $C_1 = r_1(ij) + \theta_0(j) - \theta_0(i)$  is a rotation angle between the directions on  $t_i$  and  $t_j$ , and so is any angle  $C_1 + p_1(ij)(2\pi/N)$ . We represent here  $C_1$  and the minimum angle  $C_1^{\min}$  corresponding to  $p_1(ij) = 0$  and  $p_1^{\min}(ij) = -3$ .

#### 1.4 Index

The singularities of a  $N$ -symmetry direction field can be classified by their indices defined as a 2-form [Ray et al. 2008]. We have

$$I(\theta_0, p_1) = \frac{d_1 C_1(\theta_0, p_1) + K_2}{2\pi} = \frac{d_1 r_1 + K_2}{2\pi} + \frac{d_1 p_1}{N}, \quad (3)$$

where:

- $d_1$  is the exterior derivative for 1-forms given by  $(d_1 f_1)(v^*) = \sum_{e^* \in \partial v^*} f_1(e^*)$  where  $\partial v^*$  denotes the oriented boundary of the dual cell relative to vertex  $v$ ;
- $K_2$  is the angle defect 2-form.  $K_2(v^*)$  is  $2\pi$  minus the sum of angles of triangle corners adjacent to  $v$ , which corresponds to the integrated Gaussian curvature over the dual cell  $v^*$ . The average Gaussian curvature over  $v^*$  is then given by  $K_2^{av}(v^*) = K_2(v^*)/|v^*|$  where  $|v^*|$  is the area of  $v^*$ , approximated by (one-third of) the 1-ring area of  $v^*$ .

Indices are multiples of  $1/N$  and a zero index corresponds to the absence of singularity. On the figures, singularities with positive (respectively negative) indices are marked by small red (respectively blue) spheres. Note that the indices of singularities are controlled by  $p_1$  alone, whatever the choice for  $\theta$ . The term  $(d_1 r_1 + K_2)/2\pi$  is necessarily an integer that depends only on the choice of the reference vectors.

#### 1.5 Problem Statement in Our Formalism

The problem statement given in the Introduction includes a list of desirable features for a direction field processing algorithm. Now we can translate them into our formalism.

- (1) *Direction smoothness.* A natural smoothness criterion for a direction field depending only on the direction (not on a vector norm) will be the objective  $C_1(\theta_0, p_1) = 0$ . Note that this is different from the usual vector field smoothness criterion, but coherent with Hertzmann's definition of direction field smoothness [Hertzmann and Zorin 2000]. This definition of the direction field smoothness both quantifies the curvature of the streamlines of the field and the parallelism of streamlines. One can notice that these two quantities are switched for the orthogonal direction.
- (2) *Rotational symmetry.* The angle-based representation of direction fields trivially meets this requirement.
- (3) *Geometry control.* Directional constraints can always be achieved by fixing the corresponding values of  $\theta_0$  (modulo

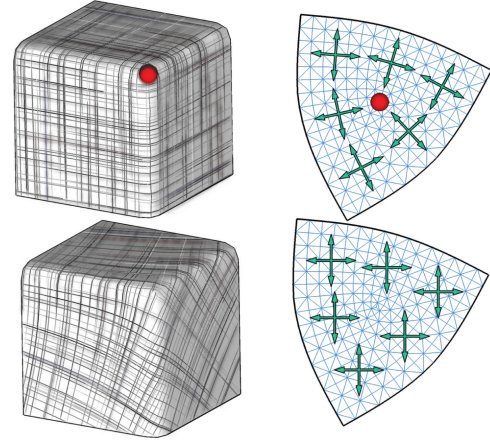


Fig. 7. 4-symmetry direction field on a cube corner. Top row: a singularity of index  $1/4$  placed at the corner exactly balances the angle defect, so the field is not distorted on the cube (left) but distorted in the map (right). Bottom row: removing the singularity allows to define a direction field that is constant in the map (right) but necessarily distorted on the surface (left).

$2\pi/N$ ). For smoothing an existing field, geometric algorithms [Hertzmann and Zorin 2000; Li et al. 2006a] just need a data-fitting term (like  $|\theta_0 - \theta_0^{\text{init}} + 2k\pi|^2$ ) while doing it with algorithms that exactly control the topology is very difficult. Indeed, the modulo defined by  $2k\pi$  must be explicitly given as it controls the topology (see Ray et al. [2008]).

- (4) *Topology control.* The control of the geometric influence will then be based on the relation between index, direction field curvature, and surface angle defect (see Eq. (3)). If we generate a smooth field, it will have a small curvature  $C_1(\theta_0, p_1)$ . Thus  $d_1 C_1(\theta_0, p_1)$  will be small too as it sums  $C_1(\theta_0, p_1)$  over one rings. In this case, the index will be close to  $K_2/2\pi$ . As a consequence, smoothing algorithms introduce singularities in a way that balances the angle defects. For instance, in Figure 7, introducing a singularity of index  $1/4$  on a cube corner (of angle defect  $\pi/2$ ) allows to cancel the curvature  $C_1(\theta_0, p_1)$ . Conversely, removing singularities ( $I_2(\theta_0, p_1) = 0$ ) leads to generate an amount of curvature that is proportional to the angle defect. We capture this curvature in a (minimal norm) target 1-form  $C_1'$  such that  $d_1 C_1' = -K_2$ . We can then modify our objective to  $C_1(\theta_0, p_1) = C_1'$ . Indeed, if  $C_1(\theta_0, p_1)$  is close to  $C_1'$ , the index will tend to be zero even if the angle defect is high.

Choosing  $C_1^t$  to have minimal norm keeps the new objective as close as possible to the original smoothness criterion.

The rest of the article is organized as follows: Section 2 presents a “naive” approach to smooth a direction field by simply minimizing its curvature, Section 3.1 shows that it is possible to modify the objective function such that the same algorithm produces a direction field that respects a prescribed topology (on a topological disk), then Section 4 explains how to perturb the objective function in order to avoid “meaningless geometric details” generating too many singularities.

## 2. FULL GEOMETRIC INFLUENCE

When no control over the field topology is required, an existing *geometric* smoothing algorithm [Li et al. 2006a] can be used to smooth direction fields given a set of soft and hard constraints. Here, we just change the weighting coefficients to make them compatible with the DEC theory. The objective function is derived from the curvature definition (Eq. (2)). We have

$$C_1(\theta_0, p_1)(ij) = r_1(ij) + \theta_0(j) - \theta_0(i) + 2\pi p_1(ij)/N = 0 \quad (4)$$

for each dual edge  $ij$ . We remove the integer variables  $p_1$  by taking the cosine and sine of Eq. (4) and by choosing new variables  $V_i = (\cos(N\theta_0(i)), \sin(N\theta_0(i)))$

$$V_i - R(Nr_1(ij))V_j = 0, \quad (5)$$

where  $R(\beta)$  is the matrix of rotation of angle  $\beta$ .

$$R(\beta) = \begin{pmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{pmatrix}$$

We weight our objectives according to the 1-form norm (Eq. (1)) by building a smoothness energy.

$$E_{\text{smooth}} = \sum_{ij \in \mathcal{E}^*} w_{ij}^{-1} (V_i - R(Nr_1(ij))V_j)^2$$

For field smoothing, we can add a fitting term  $E_{\text{fit}} = \sum_i |T_i| (V_i - V_i^{\text{init}})^2$ , where  $|T_i|$  is the area of triangle  $T_i$ , and  $V_i^{\text{init}}$  is the direction of the field to be smoothed on triangle  $T_i$ . This leads to minimize  $E_{\text{smooth}} + \lambda E_{\text{fit}}$  where  $\lambda$  balances between smoothness and data fitting. Hard constraints are easy to introduce in the system by locking the corresponding  $V_i$  as explained in Levy [2005].

This energy is a sum of square of linear equations that can be minimized easily. However, the new variables need to respect  $\|V_i\| = 1$  to be valid. As this constraint is nonlinear, we do not enforce it explicitly but iterate with the normalized solution chosen as smooth constraints for the next step.

The angles  $\theta_0(i)$  are then given by  $\theta_0(i) = \text{atan}(V_i \cdot (0, 1) / V_i \cdot (1, 0)) / N$  which gets incremented by  $\pi/N$  when  $V_i \cdot (1, 0) < 0$ . Finally,  $p_1$  is defined to minimize the field curvature as the closest integer to  $N(-r_1 - d_0\theta)/2\pi$ . Notice that setting  $p_1$  is important as it defines the indices (see Eq. (3)).

## 3. NO GEOMETRIC INFLUENCE

The aim of this section is to show that the *geometric* algorithm presented in the previous section can be used to generate a direction field that will not make the shape of the surface generate additional singularities. This is achieved by removing from the objective function (Eq. (2)) the portion of the field curvature that is a direct consequence of the surface geometry.

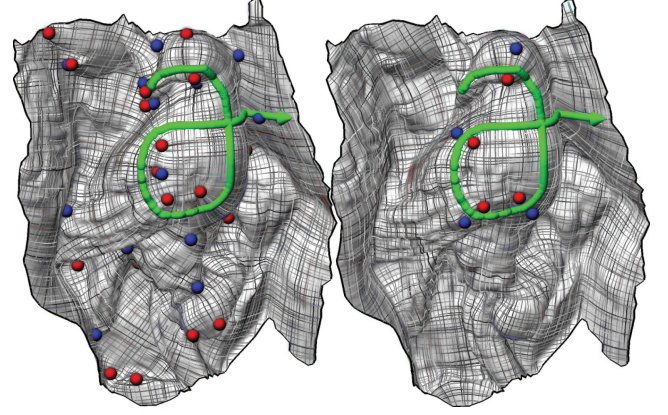


Fig. 8. Directional constraints (green arrow) with filtered geometric influence (left) and without geometric influence (right). Note the smaller number of singularities.

—First, we consider the case of a topological disk, where the new objective function is defined to be zero for direction fields without singularities and with minimal curvature. We can then prove that the singularities can be exactly controlled by simply modifying the objective function;

—then we extend this idea on arbitrary topologies and see that in practice, the minimum number of singularities required to satisfy the Poincaré Hopf theorem will emerge naturally.

### 3.1 Topological Disks

On a topological disk, we can exactly control the singularities by choosing a 2-form  $I_2^t$  that constrains the desired index.

**THEOREM 3.1 (EXACT CONTROL).** *Let  $\mathcal{D}$  be a mesh with disk topology, and  $I_2^t$  be a target 2-form with values multiple of  $1/N$  on  $\mathcal{D}$ . Then for any 1-form  $C_1^t$  such that*

$$d_1 C_1^t = -K_2 + 2\pi I_2^t, \quad (6)$$

*there exists a discrete direction field  $(\theta_0, p_1)$  such that  $I_2(\theta_0, p_1) = I_2^t$  and  $C_1(\theta_0, p_1) = C_1^t$ .*

**PROOF.** The discrete Poincaré lemma [Desbrun et al. 2005b] implies that for a  $k$ -form  $\omega_k$  on a topological disk,  $d_k \omega_k = 0$  iff it exists a  $(k-1)$ -form  $\sigma_{k-1}$  such that  $\omega_k = d_{k-1} \sigma_{k-1}$ . We will invoke this argument twice.

- (1)  $I_2^t = I_2(0, 0) = (d_1 r_1 + K_2)/2\pi$  is the index of the singularities of the field defined by the reference vectors with curvature  $r_1$ , such that  $N(I_2^t - I_2^t)$  is an integer 2-form.  $d_2 N(I_2^t - I_2^t) = 0$  by definition of  $d_2$  so there exists an integer 1-form  $p_1$  such that  $N(I_2^t - I_2^t) = d_1 p_1$ . Then by definition of the index (Eq. (3)),  $I(\theta_0, p_1) = (d_1 r_1 + K_2)/2\pi + d_1 p_1/N = I_2^t + (I_2^t - I_2^t) = I_2^t$ .
- (2) We have  $d_1(C_1^t - r_1 - 2\pi p_1/N) = -K_2 + 2\pi I_2^t - d_1 r_1 - 2\pi(I_2^t - I_2^t) = 0$  so there exists a 0-form  $\theta_0$  such that  $d_0 \theta_0 = C_1^t - r_1 + 2\pi p_1/N$ . Inserting these expressions into the definition of curvature (2) we have  $C_1(\theta_0, p_1) = C_1^t$ .  $\square$

The first part of the theorem ( $I_2(\theta_0, p_1) = I_2^t$ ) shows that we can exactly control singularity indices by setting an appropriate  $p_1$  (independently of  $\theta_0$ ). In particular, we can remove all singularities by setting  $I_2^t = 0$ . The second part shows that we can control the curvature of the field as long as the target curvature  $C_1^t$  satisfies  $d_1 C_1^t = -K_2 + 2\pi I_2^t$ . Hence if we want to smooth a field with

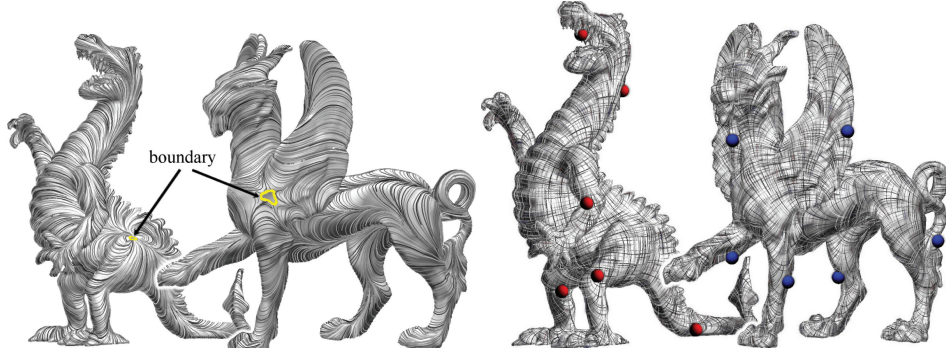


Fig. 9. Left: Removing geometric influence on a topological disk (left) and a genus 2 surface with a boundary (right). Right: If no geometric influence is wanted, the algorithm evenly distributes singularities on the surface: 8 singularities of index  $1/4$  on a topological sphere (left), and 8 singularities of index  $-1/4$  on a genus-two surface (right).

topology control, we cannot ask for  $C_1^t = 0$  anymore, but only for the  $C_1^t$  of minimal norm (defined in Eq. (1)) under this constraint. The discrete Poincaré lemma also implies that there exists at least one exact solution to  $d_1 C_1^t = -K_2 + 2\pi I_2^t$ , and we select the one of minimal norm. These constraints are linear and can be exactly fulfilled, so we can enforce them with Lagrange multipliers. However, in practice we do not need to exactly enforce these constraints, and it is sufficient to add a strong penalty term in our energy. This leads to a smaller system and gives similar results.

Once the optimal  $C_1^t$  has been computed, the smoothing algorithm (Section 2) can be adapted by replacing the objective (5) with the new objective  $C_1(\theta_0, p_1) = C_1^t$ , which by the same transformation becomes

$$V_i = R(N(r_1(ij) - C_1^t(ij)))V_j. \quad (7)$$

If a unique hard constraint on  $V_i$  is given, the system admits an exact solution that can be computed by an exact solver, and corresponds to the smoothest field with no singularity (see left model in Figure 9). In other terms, this corresponds to smoothing the surface with virtually zero Gaussian curvature.

### 3.2 Extension to Surfaces of Arbitrary Genus

For open surfaces of arbitrary genus, the same algorithm will not generate any singularity in practice (see right model in Figure 9). For closed surfaces of arbitrary genus, we must have  $\sum_{v \in S} d_1 C_1^t(v^*) = 0$  and  $\sum_v K_2(v^*) = 2\pi \chi$ , where  $\chi$  is the Euler characteristic of the surface. As a consequence, it is impossible to enforce  $d_1 C_1^t = -K_2$ . The solution proposed in this section, and improved in Section 4 is to choose a  $C_1^t$  satisfying  $d_1 C_1^t = \bar{K}_2 - K_2$  where  $\bar{K}_2$  is defined such that  $\bar{K}_2/|v^*|$  is constant. We have

$$\bar{K}_2(v^*) = \frac{|v^*| \sum_{v'^*} K_2(v'^*)}{\sum_{v'^*} |v'^*|} = \frac{2\pi \chi |v^*|}{\sum_{v'^*} |v'^*|},$$

where  $|v^*|$  is the dual cell area of  $v$ . With this new  $C_1^t$ , Eq. (7) does not have an exact solution so we apply the iterative process described in Section 2 with objective curvature  $C_1^t$  such that  $d_1 C_1^t = \bar{K}_2 - K_2$ . This smoothing behaves as if the surface had a constant Gaussian curvature, so it evenly distributes singularities over the surface (see Figure 9). Obviously this is an extreme solution where the singularities appear independently of the geometry. The next section presents a trade-off where only the influence of high geometric frequencies is removed.

*Note on singularities indices.* Computing the indices requires  $p_1$  to be defined (see Eq. (3)). This integer one form  $p_1$  is now defined as the integer minimizer of  $(C_1(\theta_0, p_1) - C_1^t)^2$ , which by Eq. (2) is the closest integer to  $N(C_1^t - r_1 - d_0\theta)/2\pi$ . Notice that this expression differs from the usual way to compute the index because we take into account the geometric correction term  $C_1^t$ .

## 4. FINAL ALGORITHM: FILTERED GEOMETRIC INFLUENCE

We have presented a smoothing direction field algorithm (Section 2) that generates singularities to capture the surface geometry. By changing only its objective function (Section 3), the same algorithm can also generate direction fields where the surface geometry has a minimal influence on this apparition of singularities. In this section, we explore how to modify the objective function such that only “meaningful features” will be captured by the singularities.

A straightforward solution would be to weight the part of the objective function that captures the influence of the geometry ( $C_1^t$ ). This would remove a part of the geometry influence, but in a way that is not related to the geometric feature size. For this reason, we prefer restating the objective function by filtering the angle defect influence.

We have seen in Section 3 that the influence of the surface’s geometry on the smoothing can be canceled by changing the objective function to  $C_1(\theta_0, p_1) = C_1^t$  such that  $d_1 C_1^t = \bar{K}_2 - K_2$ . In this case, the smoothing behaves as if the surface has a constant Gaussian curvature  $K_2^{av} = \bar{K}_2/|v^*|$ . More generally, changing the objective to  $d_1 C_1^t = K_2^{corr} - K_2$  will make the smoothing behave as if the surface had angle defects  $K_2^{corr}$  instead of  $K_2$ . What most applications need is to remove only those singularities due to high geometric frequencies while placing singularities according to the global shape of the surface. To achieve this, we chose as  $K_2^{corr}$  a low-pass filtered version of  $K_2$ . As it must still be an admissible angle defect,  $K_2^{corr}$  needs to satisfy  $\sum_{v^*} K_2^{corr}(v^*) = 2\pi \chi$ .

Section 4.1 explains how to obtain  $K_2^{corr}$  by low-pass filtering  $K_2$ , then Section 4.2 presents a practical solution to edit the field topology.

### 4.1 Filtering $K_2$

We will smooth the density of curvature  $K_2^{av}$  defined in Section 1.4 (to be mesh independent) with a Gaussian smoothing algorithm



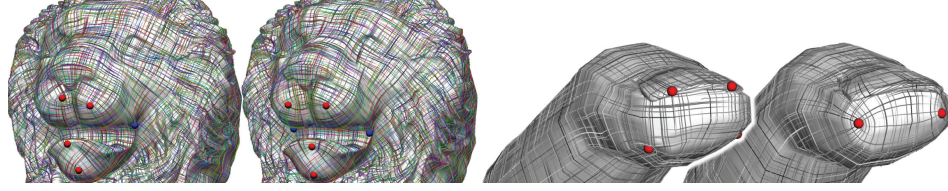


Fig. 10. Left: Topological editing is useful to adjust the topology, especially when the surface exhibits local symmetries. Right: Merging 1/4 index singularities into 1/2 index singularities can simplify the field topology.

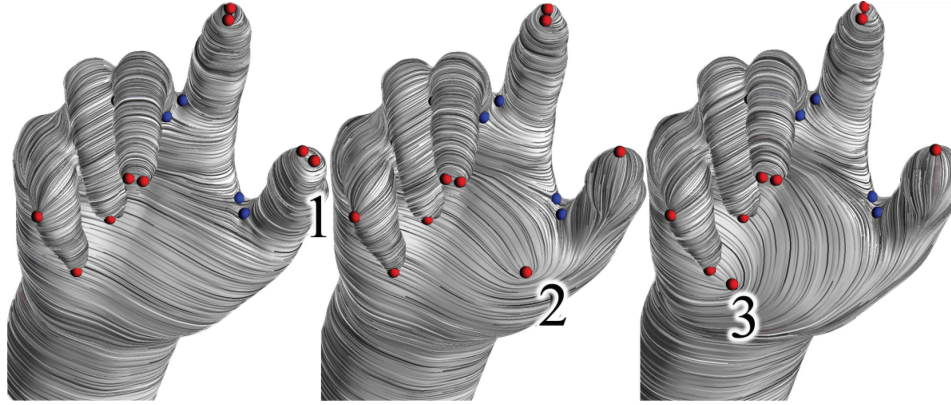


Fig. 11. Moving an index from (1) to new positions (2) then (3), by a modification of  $C_1^t$  is robust enough in practice.

based on a raw estimation of the geodesic distance using the shortest path of edges between two points. Note that the algorithm acts on  $K_2$  directly but it smooths  $K_2^{gv}$ . We choose a Gaussian radius  $\sigma$ , then for each vertex  $v_i$ , we use a Dijkstra algorithm to compute the distance  $D_{ij}$  from vertex  $v_i$  to vertex  $v_j$ , stopping when  $D_{ij} > 2\sigma$ . The filter can then be computed as

$$K_2^{corr}(v_j^*) = \sum_{v_i^*} \frac{c_{ij} K_2(v_i^*)}{\sum_{v_k^*} c_{ik}} \quad \text{with} \quad c_{ij} = |v_j^*| e^{-\left(\frac{D_{ij}}{\sigma}\right)^2}.$$

Notice that the smoothing ensures

$$\sum_{v^*} K_2(v^*) = \sum_{v^*} K_2^{corr}(v^*)$$

as  $\sum_j c_{ij} K_2(v_i^*) / \sum_k c_{ik} = K_2(v_i^*)$ . Large values of  $\sigma$  will correspond to smoother  $K_2^{corr}$  thus remove larger geometric details, but will require longer computation time. The geometry canceling smoothing algorithm of Section 2 corresponds to the limit  $\sigma \rightarrow 0$ , where  $c_{ij} \rightarrow 0$  except  $c_{ii} \rightarrow v^*(i)$  and  $K_2^{corr} \rightarrow K_2$ . The geometry canceling smoothing algorithm of Section 3 corresponds to the limit  $\sigma \rightarrow \infty$ , where  $c_{ij} \rightarrow v^*(j)$  and  $K_2^{corr} \rightarrow \bar{K}_2$ . Once  $K_2^{corr}$  is computed, we get  $C_1^t$  as in Section 3 except that the constraint is replaced by  $d_1 C_1^t = K_2^{corr} - K_2$ .

## 4.2 Editing the Direction Field Topology

The field topology obtained automatically by our algorithm can be improved by the user by merging singularities (see Figure 10), or to moving them to semantically meaningful positions, for instance to respect local symmetries (see Figure 10). Such topological editing operations could be guaranteed using previous work [Ray et al. 2008], but this approach makes it impossible to continue editing the field geometry as before. For this reason, we prefer another approach

based on editing  $K_2^{corr}$  that has no theoretical guarantees but never fails in practice.

Our method is based on updating  $C_1^t$  accordingly to the desired topology. Canceling a pair of singularities of indices  $+I$  at  $v^*$  and  $-I$  at  $v'^*$ , or equivalently moving an index  $+I$  singularity from  $v^*$  to  $v'^*$  is done by adding  $I$  to  $K_2^{corr}(v'^*)$  and subtracting  $I$  to  $K_2^{corr}(v^*)$ . Then  $C_1^t$  is computed as usual, and the user can continue processing the field as before (adding directional constraints, smoothing the field, etc.).

## 5. RESULTS AND APPLICATIONS

We provide some insights on the parameter tuning and timings for the Michelangelo's *David* statue at a resolution 100K triangles. Here are the three main steps of our final algorithm applied to this model.

- (1) Smoothing the angle defect  $K_2$  is done by a Gaussian filter and the result is called  $K_2^{corr}$  (Section 4.1). The only tuning parameter is the Gaussian radius  $\sigma$ . Increasing  $\sigma$  makes it possible to “trade” some field smoothness against a simpler field topology. It takes respectively 20s, 1min 15s, and 5 min 30s to smooth with  $\sigma = 0.05h$ ,  $0.1h$ , and  $0.2h$ , where  $h$  denotes the height of the statue.
- (2) Computing the target curvature  $C_1^t$ : minimize its squared norm under constraint  $d_1 C_1^t = K_2^{corr} - K_2$  (Section 3). Computing  $C_1^t$  from  $K_2^{corr}$  requires 2 seconds with an efficient linear solver (CHOLMOD [Davis and Hager 2005]).
- (3) Enforcing  $C_1(\theta_0, p_1) = C_1^t$  in the least square sense after a change of variables as in Li et al. [2006a, Section 2]. It requires a series of quadratic minimizations. The tuning parameters are the trade-off  $\lambda$  between smoothness and data fitting and the

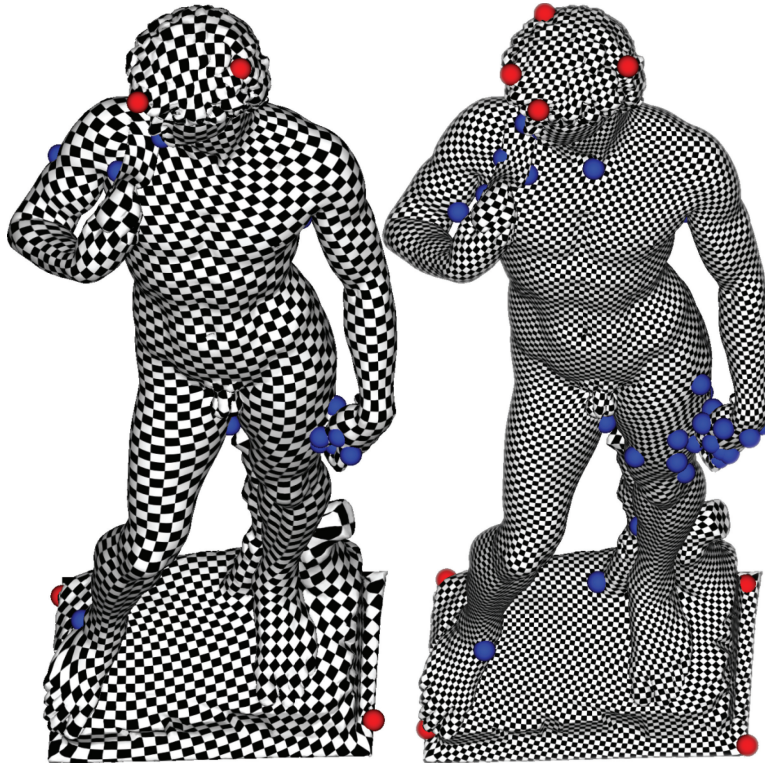


Fig. 12. 2-symmetry (left) and 4-symmetry (right) direction fields generated by our method are used as input for a global parameterization (using quad cover).

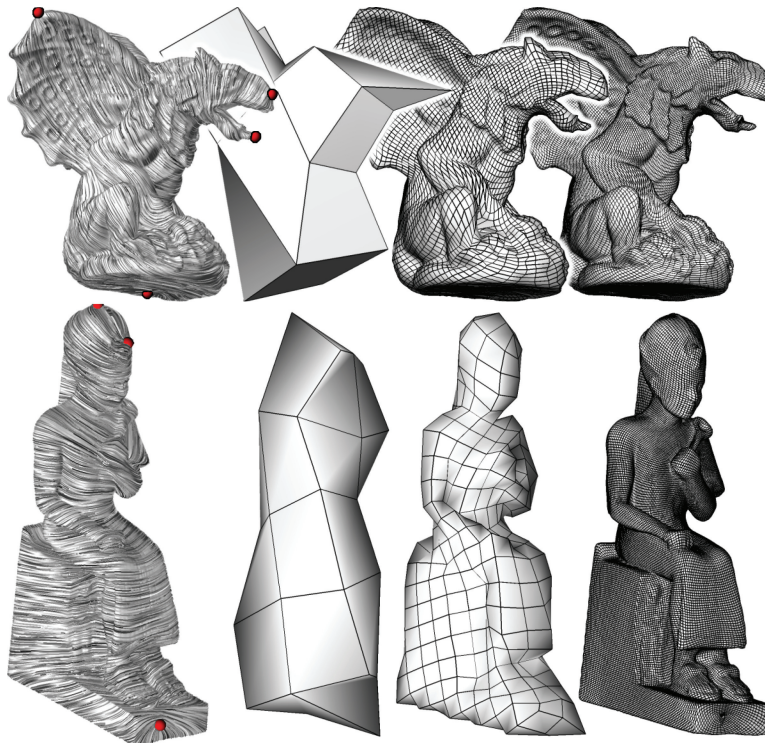


Fig. 13. Producing few singularities allows to create very coarse quad mesh (right) with quad cover. This mesh can be used as base domain for geometric images with different resolution (middle).



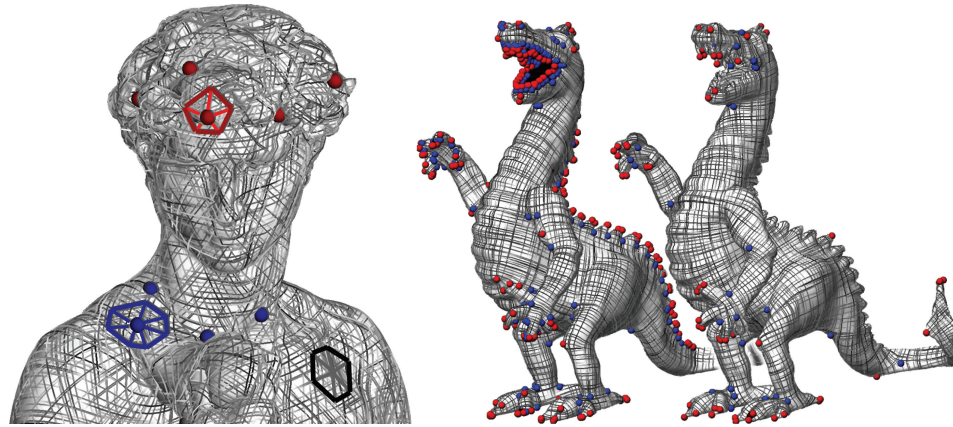


Fig. 14. Left: A 6-symmetry direction fields generated on the head of the Michelangelo’s *David* statue. Notice that for triangular remeshing applications,  $1/6$  indices will become valence 5 vertices (red),  $-1/6$  indices will become valence 7 vertices (blue), and zero indices become regular (valence 6) vertices (black). Right: Our method (right) allows to remove singularities that are due to geometric high frequencies (left).

number of iterations. 1 to 5 iterations are usually sufficient, and each step costs 2.5 seconds with CHOLMOD on the *David*.

As the filtering of the influence of geometric details is performed in precomputation steps (1 and 2), it is then possible to apply many direction field processing algorithms such as field generation, smoothing, or direction extrapolation (using step 3) that will always preserve the property of ignoring geometric details for the generation of singularities. Compared with previous works, this framework makes it much simpler to design direction fields.

—N-RoSy [Palacios and Zhang 2007] does not allow to preserve a simple topology while editing the direction field because it uses topology simplification as a postprocessing step.

—NSDF [Ray et al. 2008] requires manually placing singularities. It may be tedious for high genus models such as the Michelangelo’s *David* statue (genus 8) that requires at least 56 singularities of index  $-1/4$  for a 4-symmetry direction field (see Figure 12).

For interactive design of direction fields, our algorithm allows controlling the field by simply “painting” hard constraints on the surface while preserving the trade-off between simple topology and smoothness. Since new constraints are usually introduced to locally modify the field geometry, the smoothing is also performed locally. This allows real-time feedback while painting new constraints.

It has been shown in Palacios and Zhang [2007] that complex field topology may affect the quality in texturing / hatching applications. For global parameterization based on cone singularities, that topological complexity becomes critical as it determines the type of parametric domain. Our method provides direction fields suitable for this application as they have the minimal number of singularities to capture the global shape of the object. As illustrated in Figure 13, the global parameterization (very coarse quad mesh) is simple enough to be used as a parametric domain for geometry images [Gu et al. 2002] (increasing the geometry image resolution gives finer meshes).

Finally, the generality of the  $N$ -symmetry framework makes it eligible for triangular remeshing based on 6-symmetry directions fields. As explained in Section 1.5, increasing the number of symmetries will also increase the number of singularities generated by high geometric frequencies. Figure 14 shows that our framework allows to create 6-symmetry directions fields with few singularities even in the presence of high geometric frequencies.

**Conclusion.** In the past few years, the research topics of quad remeshing and mesh parameterization have been converging toward what can be called quad parameterization. We believe that computer graphics and modeling applications would greatly benefit from quad parameterization, as it allows for both seamless texture mapping and automatic conversions between surface representations. However, quad parameterization is still a theoretic concept as existing methods either fail on complex models or require too much user interaction. This work removes some limitations by providing a simple yet efficient tool to define quad orientation with a clean global topology. We hope this will help quad parameterization to become tomorrow’s standard in the computer graphics industry.

## REFERENCES

- ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LEVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Trans. Graph.* Special issue for SIGGRAPH Conference, 485–493.
- BEN-CHEN, M., GOTSMAN, C., AND BUNIN, G. 2008. Conformal flattening by curvature prescription and metric scaling. In *Proceedings of the IEEE Eurographics. Eurographics*.
- DAVIS, T. A. AND HAGER, W. W. 2005. Row modifications of a sparse Cholesky factorization. *J. Matrix Anal. Appl.* 26, 3, 621–639.
- DESBRUN, M., KANZO, E., AND TONG, Y. 2005a. Discrete differential forms for computational modeling. *Siggraph’05 Course Notes on Discrete Differential Geometry*. Chapter 7.
- DESBRUN, M., LEOK, M., AND MARSDEN, J. 2005b. Discrete Poincaré lemma. *Appl. Numer. Math.* 53, 2–4, 231–248.
- FISHER, M., SCHRÖDER, P., DESBRUN, M., AND HOPPE, H. 2007. Design of tangent vector fields. In *Proceedings of the SIGGRAPH’07 Conference*. ACM, New York, 56.
- GU, X., GORTLER, S., AND HOPPE, H. 2002. Geometry images. In *Proceedings of the SIGGRAPH’02 Conference*. 355–361.
- HERTZMANN, A. AND ZORIN, D. 2000. Illustrating smooth surfaces. In *Proceedings of the ACM SIGGRAPH Conference*. 517–526.
- KALBERER, F., NIESER, M., AND POLTHIER, K. 2007. Quadcover—Surface parameterization using branched coverings. *Comput. Graph. Forum* 26, 3, 375–384.
- KHAREVYCH, L., SPRINGBORN, B., AND SCHRÖDER, P. 2006. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.* 25, 2, 412–438.



- LEFEBVRE, S. AND HOPPE, H. 2006. Appearance-space texture synthesis. In *Proceedings of the SIGGRAPH'06 Conference*. ACM Press, New York, 541–548.
- LEVY, B. 2005. Numerical methods for digital geometry processing. In *Proceedings of Israel Korea Bi-National Conference*.
- LI, W.-C., RAY, N., AND LEVY, B. 2006a. Automatic and interactive mesh to T-spline conversion. In *Proceedings of EG/ACM Symposium on Geometry Processing*.
- LI, W.-C., VALLET, B., RAY, N., AND LEVY, B. 2006b. Representing higher-order singularities in vector fields on piecewise linear surfaces. *IEEE Trans. Visual. Comput. Graph.*
- PALACIOS, J. AND ZHANG, E. 2007. Rotational symmetry field design on surfaces. In *Proceedings of the SIGGRAPH'07 Conference*. ACM, New York, 55.
- PINKALL, U. AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experiment. Math.* 2, 1, 15–36.
- PRAUN, E., FINKELSTEIN, A., AND HOPPE, H. 2000. Lapped textures. In *Proceedings of the SIGGRAPH'00 Conference*, K. Akeley, Ed. 465–470.
- PRAUN, E., HOPPE, H., WEBB, M., AND FINKELSTEIN, A. 2001. Real-Time hatching. In *Proceedings of the SIGGRAPH'01 Conference*, E. Fiume, Ed. 579–584.
- RAY, N., LI, W. C., LEVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph.* 25, 4, 1460–1485.
- RAY, N., VALLET, B., LI, W.-C., AND LEVY, B. 2008. N-symmetry direction fields on surfaces of arbitrary genus. *ACM Trans. Graph.*
- SPRINGBORN, B., SCHRÖDER, P., AND PINKALL, U. 2008. Conformal equivalence of triangle meshes. *ACM Trans. Graph.* 27, 3, 1–11.
- TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *Proceedings of the Symposium on Geometry Processing*. Eurographics, 201–210.
- TURK, G. 2001. Texture synthesis on surfaces. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01)*. ACM Press, New York, 347–354.
- ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2006. Vector field design on surfaces. *ACM Trans. Graph.* 25, 4, 1294–1326.

Received December 2008; accepted July 2009