Optimal Transport-Based Polar Interpolation of Directional Fields

JUSTIN SOLOMON, Massachusetts Institute of Technology*
AMIR VAXMAN, Utrecht University*

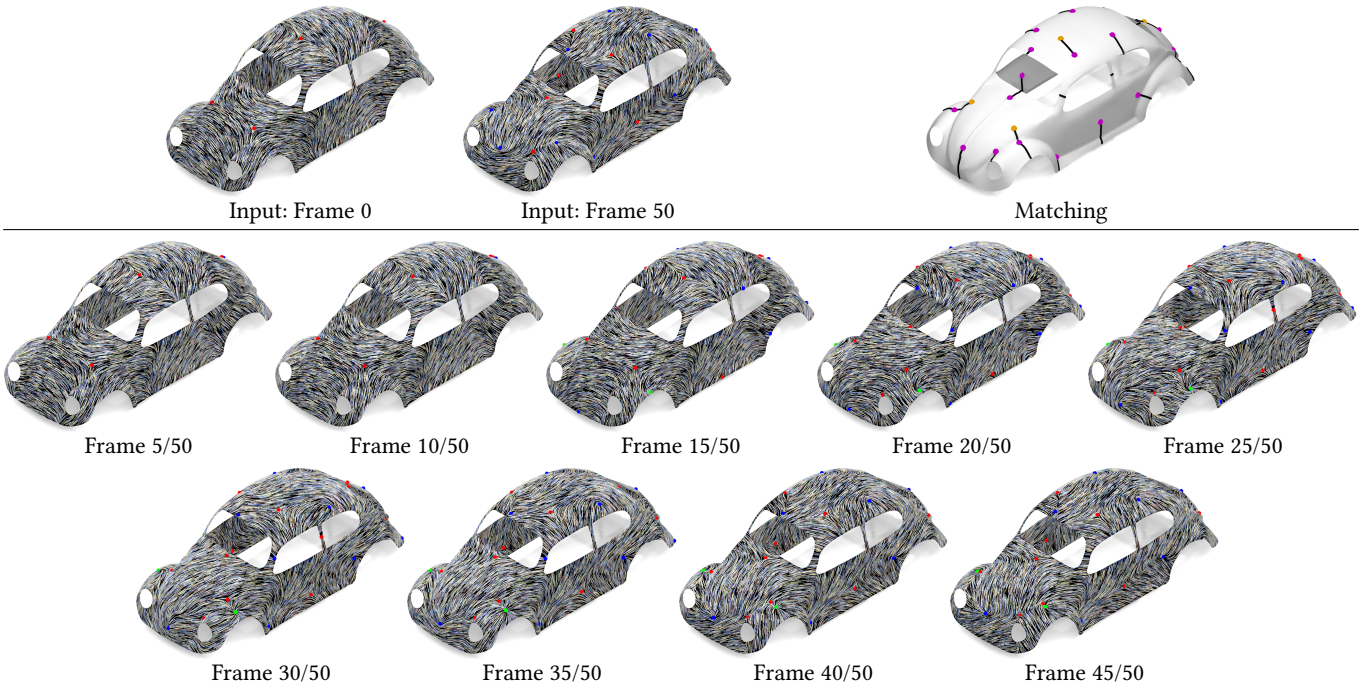

Fig. 1. Our algorithm uses optimal transport to match the singular points (upper right) of two input line fields (upper left) and uses this matching to generate a time-varying interpolant. Our technique interpolates the fields smoothly in time by sliding singularities along the domain rather than having them appear and disappear, merging them with boundary curves as needed. Singularities are color-coded according to index (green for $+2/N$, blue for $+1/N$, red for $-1/N$); purple and orange singularity colors in the matching image distinguish the two input frames.

We propose an algorithm that interpolates between vector and frame fields on triangulated surfaces, designed to complement field design methods in geometry processing and simulation. Our algorithm is based on a *polar* construction, leveraging a conservation law from the Hopf-Poincaré theorem to match singular points using ideas from optimal transport; the remaining detail of the field is interpolated using straightforward machinery. Our model is designed with topology in mind, sliding singular points along the surface rather than having them appear and disappear, and it caters to all surface topologies, including boundary and generator loops.

CCS Concepts: • **Computing methodologies** → **Shape analysis**; • **Mathematics of computing** → *Interpolation*.

Additional Key Words and Phrases: Optimal transport, frame fields

ACM Reference Format:
Justin Solomon and Amir Vaxman. 2019. Optimal Transport-Based Polar Interpolation of Directional Fields. *ACM Trans. Graph.* 38, 4, Article 88 (July 2019), 13 pages. https://doi.org/10.1145/3306346.3323005

* Joint first authors.
Authors' addresses: Justin Solomon, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology*, jsolomon@mit.edu; Amir Vaxman, Department of Information and Computing Sciences, Utrecht University*, A.Vaxman@uu.nl.

0730-0301/2019/7-ART88 $15.00
https://doi.org/10.1145/3306346.3323005

1 INTRODUCTION

Vector and frame fields are ubiquitous in geometry processing and physically-based animation. Used to guide physical motion, texture synthesis, meshing, and more exotic tasks like path planning, algorithms that design and process fields find application in most geometry-oriented stages of the graphics pipeline.

Field processing algorithms typically fall into two major categories. First, *field design* algorithms in geometry processing place fields on a domain given a sparse set of user constraints, such as placement of singular points or directionality. Second, *simulation* algorithms use vector fields to capture quantities like momentum and

velocity when simulating physics in an Eulerian fashion; these algorithms solve initial-value problems that approximate the evolution of fields describing physical quantities over time.

In the context of animation—either artist-guided or physically-based—the usual output of these methods is a sequence of fields at discrete points in time. For instance, fluid simulation algorithms generate velocities at discrete time steps from ODE/PDE integration. Similarly, artist-designed fields are only available at frames of an animation sequence where the artist intervened.

In this paper, we accompany the techniques above with an interpolation algorithm intended to serve as an in-betweening tool for fields. Our algorithm is built upon the premise that interpolation should be aware of the *topology* of the field at key frames. That is, we wish to design our interpolant to avoid generating spurious singular structures that are not present in the key frames.

Our algorithm is built on a link between field topology on surfaces and optimal transport. In particular, a consequence of the Hopf–Poincaré theorem is that the sum of singular indices for any field on a fixed surface has to do only with the topology of the underlying surface; a well-known analog holds for per-triangle fields on a manifold triangle mesh with or without boundary. This theorem provides a conservation law upon which we can formulate an optimal transport problem to match singular points to one another for interpolation. This transport-based matching is not only geometry-aware but also preserves integrality of indices in the matching, naturally avoiding inadmissible fractional indices.

Beyond matching singular points, interpolating the remaining components of the field can be carried out in a straightforward fashion while producing visually and numerically reasonable output. The end result is an interpolation technique with intuitive behavior; furthermore, the algorithm extends without change from vector fields to frame fields with multiple directions assigned to each point on the underlying surface.

While introducing our algorithm and its basic theoretical properties, we demonstrate its behavior on a variety of challenging test cases. Our method handles surfaces of varying genus and numbers of boundary loops, and it gracefully interpolates N-RoSy fields for arbitrary choices of N.

2 RELATED WORK

A complete review of vector and frame fields is out of the scope of our discussion; we refer the reader to the survey by Vaxman et al. [2016] for a detailed introduction and discussion. Here, we highlight the key work most relevant to our subsequent discussion.

Polar vector field design. Our work manipulates *polar* representations of vector fields, which give the direction of the field as an angle per face relative to some local basis [Bommes et al. 2009; Kälberer et al. 2007; Li et al. 2006; Ray et al. 2006, 2009]. The advantage of working with polar representations is direct control over topology, while the disadvantage is optimization with integer variables for automatic placement of singularities. When the singularities are prescribed in advance (or computed through some other means [Ben-Chen et al. 2008; Farchi and Ben-Chen 2018; Soliman et al. 2018]), the problem of reconstructing the field giving singular points and their degrees becomes linear. This problem is typically solved by recovering the rotation between neighboring frames [Ray et al. 2009], where the angles are chosen to sum up to the correct indices around regular and singular cycles. This is also akin to computing a new *connection* on the mesh [Crane et al. 2010]. The decomposition of the rotation angles into Hodge components reveals additional geometric structure in the field [de Goes and Crane 2010]; see §3.1 for additional discussion.

Simulation interpolation. A few papers consider problems related to temporal interpolation of vector field-valued data. While not directly comparable to our technique, these methods demonstrate the potential interest for vector field interpolation. We note that "vector field interpolation" is an overloaded term: A separate problem with the same name appearing in the visualization and meteorology literature is to interpolate sparse samples of a static field to the remainder of a domain.

Sato et al. [2018] interpolate fluid simulations to blend multiple sequences. Their method is specifically designed for velocity fields from solving the inviscid, incompressible Navier–Stokes equations. For that reason, their variational approach minimizes a (topology-unaware) Dirichlet energy that reduces divergence in the interpolated field; their method also takes multiple frames of animation sequences to be blended rather than two. An earlier method by the same author [2015] has similar drawbacks. Unlike our method, however, their algorithms naturally extend to volumetric fields, albeit without careful treatment of rotational symmetries.

Another place in fluid simulation where field interpolation appears is at the intersection of Lagrangian and Eulerian mechanics. In simulating fluid motion, it may be necessary e.g. to advect particles for shorter time steps than the time step for the Eulerian component. Interpolation methods applied to this task are rarely discussed in detail; the standard appears to be to use linear or polynomial coordinate-wise interpolation on a grid. See the tutorial by Fiedler [2018] and thesis by de Vries [2016] for explicit mention.

Some fluid simulation methods based on filaments like [Angelidis and Neyret 2005] explicitly track moving singular features as simulation variables, providing a straightforward means of singularity-aware interpolation if simulation is carried out in this fashion.

Optimal transport. Our algorithm for matching singular points is built from optimal transport (OT). [Lévy and Schwindt 2018; Peyré et al. 2019; Solomon 2018] survey recent developments in this active mathematical and applied discipline, including several graphics applications. While algorithms like those in [Lavenant et al. 2018; Lévy 2015; Mérigot 2011; Solomon et al. 2015, 2014] provide discretizations and optimization algorithms for OT relevant to graphics, in this paper our instance is so sparse we can solve it directly using a linear program.

One point of comparison to our work is the algorithm presented by Peyré et al. [2017], who interpolate between tensor fields using an extension of OT. Their algorithm fails to extend cleanly to curved surfaces, for which they rely on a fixed background matching between tangent spaces to formulate their problem; their algorithm also is not aware of polar field structure and cannot be applied to N-directional fields. In any event, tensor field processing is a distinct task from vector field processing, with different applications.

Vector field visualization. A related problem to field interpolation is *vector field visualization*, in which it may be desirable to detect and track topological changes as salient features in time-varying data. In 2D, singular features have long been used in this area to identify and simplify pertinent features of fields [Tricoche et al. 2000].

One point of comparison to our interpolation method is the work of Garth et al. [2004], which tracks singular points as they move in time-varying vector field data. While their task has some commonalities with our transport-based matching procedure (§5), their tracking algorithm is designed to follow differential changes in singular point positions in simulation data.

Other works use the Wasserstein distance from OT to match topological features globally; these works rely on OT as a means of solving a geometric matching problem but do not benefit from the conservation properties afforded in our case by the index theorem. Examples include OT between persistence diagrams [Soler et al. 2018] and feature sets [Ji and Shen 2006].

3 PRELIMINARIES

We begin our discussion of field interpolation by introducing notation and constructions needed to describe our technique.

3.1 Polar Hodge Decomposition

Suppose we are given a manifold orientable triangle mesh $M = (V, E, F)$, with vertices V, edges E, and triangular faces F. We consider the space of face-based tangent vector fields, that is, assignments of one vector to each triangle in F constrained to be in the triangle plane.

Suppose that each face $f \in F$ is parameterized by a local tangent basis $\{B_{1,f}, B_{2,f}\} \subset \mathbb{R}^3$. As a convenience, we can represent every tangent vector $v_f \in \text{span}\{B_{1,f}, B_{2,f}\}$ as a complex number whose real part is the component in the $B_{1,f}$ direction and whose complex part is the component in the $B_{2,f}$ direction.

Discrete connection. Two vectors in adjacent faces f and f' with a shared dual edge e can be compared by accounting for the difference between the bases $\{B_{1,f}, B_{2,f}\}$ and $\{B_{1,f'}, B_{2,f'}\}$. This is done by considering the representations of a vector along e in the complex bases for the two faces, yielding $e_f, e_{f'} \in \mathbb{C}$. Then, v_f and $v_{f'}$ are related by parallel transport in the discrete Levi-Civita connection [Knöppel et al. 2013; Polthier and Schmies 1998] when $v_f\overline{e_f} = v_{f'}\overline{e_{f'}}$, where bars indicate complex conjugation; deviation from this parallel transport is a typical means of comparing vectors in adjacent triangles with different tangent planes. We denote a closed path of faces and their adjacent edges as a *dual* cycle, around which vectors can be transported from one face to itself.

In the presence of Gaussian curvature, one cannot compute a vector field so that all adjacent vectors are parallel. Instead, parallel-transporting a vector around a cycle results in a rotation of the vector by the angle defect encoded in the cycle, the cycle's discrete curvature (*holonomy* is this curvature modulo 2π); for dual cycles around vertices v, the resulting angle defect is the discrete integrated Gaussian curvature $\overline{K}_v$ at $v \in V$. For a cycle of boundary vertices $b \in B$, where $B \subseteq 2^V$ is the set of boundary loops, parallel transport around a loop yields a rotation by the geodesic curvature of the dual cycle of boundary faces:

$$\overline{K}_b = \sum_{v \in b} \left(\pi - \sum_{t \in N(v)} \alpha_{t,v} \right),$$

where t denotes a triangle adjacent to boundary vertex $v \in b$ and $\alpha_{t,v}$ denotes the interior angle of t at v.

Meshes with higher genus g admit $2g$ dual homology-generating cycles, represented as *generator loops*. Each cycle $c \in G$ admits an angle defect $\overline{K}_c$ in a similar manner to boundary loops; here $G \subseteq 2^V$ denotes the set of such cycles. Note that the sign of $\overline{K}_c$ in this case depends on the orientation of the homology-generating cycle.

Following [Crane et al. 2010; Ray et al. 2009], in the polar representation paradigm, a unit vector field is expressed by measuring its deviation from parallel transport. This deviation is parameterized by a single angle θ_e per dual edge satisfying

$$\theta_e = \arg\left(\frac{v_{f'}\overline{e_{f'}}}{v_f\overline{e_f}} \right) + 2\pi k, k \in \mathbb{Z}.$$

The sum of θ_e over dual vertex, boundary, and generator cycles c of dual edges is always $2\pi I_c - \overline{K}_c$, where $I_c \in \mathbb{Z}$ is the *index* of the field on cycle c. To match a vector to itself around a cycle, we must negate the cycle's curvature, accounting for the term $-\overline{K}_c$ in the sum of rotations. This is equivalent to defining a discrete *trivial connection* [Crane et al. 2010]. The index of the cycle I_c then gives the number of full rotations by 2π the vector undergoes when traced around the cycle, and $K_c = 2\pi I_c$ is the (conic) curvature induced by the field.

A cycle c is considered regular if $I_c = 0$; otherwise it is singular. By the index theorem, we know

$$\sum_{v \in V} I_v + \sum_{b \in B} I_b = \chi = V - E + F = 2 - 2|G| - |B|. \quad (1)$$

The indices of generator cycles are in general unbounded and independent from one another. Here, we only consider isolated point singularities (as opposed to, e.g., line singularities), since to the best of our knowledge the general case has not been explored in the context of polar representations.

We encode the cycles algebraically by assigning to each primal edge on the mesh an orientation, expressed as an ordering of its two vertices; this primal orientation induces a dual orientation for each edge, i.e., the primal edge's left and right faces. Then, every dual cycle c is a sequence of edges, represented as a row vector $c \in \{-1, 0, 1\}^{1 \times |E|}$, where c_e is ± 1 according to dual edge e's orientation along the cycle if $e \in c$ and 0 otherwise. The columns of the discrete exterior derivative operator $d_0 \in \{-1, 0, 1\}^{|E| \times |V|}$ that pertain to inner vertices are exactly the oriented dual cycles around inner vertices. A column of d_0 corresponding to a boundary vertex encodes the rotation between two boundary faces in an open cycle.

Slightly abusing notation, we can collect all boundary cycles in matrix $B \in \{-1, 0, 1\}^{|B| \times |E|}$ and all generator cycles in matrix $G \in \{-1, 0, 1\}^{2g \times |E|}$. In fact, a single row $b \in B$ is simply the sum of rows in $d_0^\top$ corresponding to all boundary vertices in that loop.

Then, for every trivial connection parameterized by θ_e we have

$$\begin{pmatrix} d_0^\top \\ B \\ G \end{pmatrix} \theta_e = K_c - \overline{K}_c. \tag{2}$$

The dual 1-form θ_e fully defines any vector field v up to face-based magnitude $l_f = |v_f|$ and a global rotation $\theta_0 \in \mathbb{R}$ acting identically on the field in each face. The pair (θ_e, θ_0) provides a *polar representation* of the field with magnitude l_f.

Boundary cycle indices. A boundary cycle can be considered as a generalized case of a point singularity. Consider again the cycle matrix B. Each cycle's index is defined as the change in angle of the vector field integrated over the boundary curve. For instance, a field that is always orthogonal to the boundary has index 0. In the case of a unit disc in the plane, $\chi = 2 - 2g - b = 1$, and therefore in this case there must be internal singularities whose indices add up to 1 in the interior. For instance, a radial vector field with a single source singularity in the center of the disc satisfies this constraint.

Hodge decomposition and smoothest connection. Given an admissible set of indices I_c, it is possible to create a field that minimizes $\sum_{e \in E} |\theta_e|^2$ and obeys the prescribed singularities. First consider a closed mesh without boundary. As θ_e is a dual 1-form, we consider its Hodge decomposition [Crane et al. 2013]:

$$\theta_e = d_1^\top \psi_f + \star d_0 \psi_v + h,$$

where ψ_f is a dual 0-form (scalar per face) that we call the *detail function* and ψ_v is a primal 0-form (scalar per vertex) that encodes the curvature change at every vertex. Pre-multiplying the two sides of this expression by $d_0^\top$, we find

$$d_0^\top \theta_e = 0 + \Delta_0 \psi_v + 0 = K_v - \overline{K}_v,$$

where $\Delta_0 = d_0^\top \star_1 d_0$ is the integrated 0-form Laplace-Beltrami operator. Similarly, for generator indices, we have

$$G(h + \star_1 d_0 \psi_f) = K_c - \overline{K}_c.$$

If we reduce the harmonic component h to a $2g$-dimensional representation $h = H\alpha$, where the columns of $H^{|E| \times 2g}$ are a basis for harmonic dual 1-forms, and set $P = GH$, we obtain a compact representation:

$$P\alpha = K_c - \overline{K}_c - G \star_1 d_0 \psi_f.$$

Here, P is the *period matrix* of the mesh [Tong et al. 2006].

With these definitions in place, the connection parameterized by θ_e can be decomposed into two essential parts:

(1) $\star d_0 \psi_v + h$, a smooth field generated by the indices of vertex cycles and generators, and
(2) a detail function ψ_f modulating the smooth field in each face f without affecting its topology, since $d_0^\top d_1^\top \psi_f = 0$ and $G d_1^\top \psi_f = 0$ by definition.

We call (ψ_v, h, ψ_f) *polar Hodge decomposition* of the connection θ_e associated with v. This defines v up to global rotation, defined by the arbitrary alignment of the field in a single face. The smoothest field with prescribed indices satisfies $\psi_f = 0$, which is the one computed by Crane et al. [2010].

Hodge decomposition with boundary. In the presence of boundary vertices, the Hodge decomposition described above is nonunique. Additional assumptions are needed to make the spaces of exact, coexact, and harmonic dual 1-forms orthogonal, leading to a unique Hodge decomposition [Poelke and Polthier 2016].

The consistent choice we make is to assume that $\theta_e = 0$ on boundary edges e and that ψ_f is constant along every boundary cycle but not necessarily equal between independent boundary loops. That implies that $d_0^\top \theta_e$ on boundary edges is only $\star_1 d_0 \psi_v$ without any coexact contribution. ψ_v is then defined on all vertices, and $\Delta_0 \psi_v = K_v - \overline{K}_v$ still holds for every vertex, including boundary vertices. The Laplace–Beltrami operator of ψ_v on the boundary is defined to simply ignore boundary dual edges.

N-RoSy fields. The theory of polar Hodge decompositions readily extends to N-RoSy fields [Palacios and Zhang 2007; Ray et al. 2008], which encode N symmetric unit fields per face. The only difference is to use fractional indices I_c/N, where N denotes the number of direction vectors coupled together at each point. Curvature-based matchings around cycles are accordingly integer multiples of $2\pi/N$, and the index theorem holds without change with this new curvature. For example, an vector field on a sphere can have 2 singularities of index 1, and a 5-RoSy can have 10 singularities of index $1/5$. The index I_c is an integer.

3.2 Optimal Transport

To give intuition for our discussion later on, we briefly provide a few simple notions from optimal transport. Suppose we are given two sets of points, $\{p_i\}_{i=1}^n \subseteq S$ and $\{p'_j\}_{j=1}^k \subseteq S$, on a domain S. To each p_i, we associate a quantity (or *mass*) $m_i \geq 0$, and to each p'_j we associate a mass $m'_j \geq 0$. We assume $\sum_i m_i = \sum_j m_j$. These two objects can be interpreted as distributions on S, sums of weighted δ-functions $\mu_0 := \sum_i m_i \delta_{p_i}$ and $\mu_1 := \sum_j m'_j \delta_{p'_j}$, respectively.

The *two-Wasserstein* distance between these two discrete distributions is computed as follows:

$$\mathcal{W}_2^2(\mu_0, \mu_1) := \begin{cases} \min_{T \in \mathbb{R}^{n \times k}} & \sum_{ij} d(p_i, p'_j)^2 T_{ij} \\ \text{subject to} & T_{ij} \geq 0 \ \forall i, j \\ & \sum_j T_{ij} = m_i \\ & \sum_i T_{ij} = m'_j. \end{cases} \tag{3}$$

Here, $d(\cdot, \cdot)$ denotes distance along S. (3) can be interpreted as a minimum-cost matching problem, where the cost of moving a unit of mass from p_i to p'_j equals squared distance $d(p_i, p'_j)^2$. The entry T_{ij} represents the amount of mass moved from p_i to p'_j in the matching. Here, the *square* of distance in the cost is included so that $\mathcal{W}_2$ can be interpreted as a geodesic distance with a shortest path in the space of distributions; see [Santambrogio 2015; Villani 2003] for discussion.

4 ALGORITHM OVERVIEW

Our algorithm takes as input the polar representations of two pointwise unit-norm fields $(\psi_v^0, h^0, \psi_f^0)$ and $(\psi_v^1, h^1, \psi_f^1)$ on a fixed mesh and outputs an interpolant $(\psi_v^t, h^t, \psi_f^t)$ sampled at a set of $n + 1$ evenly-spaced values $\{t_0, \ldots, t_n\} \subset [0, 1]$ with $t_0 = 0$ and $t_n = 1$. It

then reconstructs the full field v^t by interpolating the global rotation of the field.

Our method takes place in several steps, decoupling over the various pieces of the polar Hodge decomposition. We summarize them here and then provide detail on each in the subsequent sections.

Preprocessing. Depending on the application, our field may be provided in *Cartesian* (xyz coordinates per face) rather than polar form. In that case, our first step is to convert the field to polar form. This is done by first carrying out *principal matching* [Diamanti et al. 2014] to recover the connection θ_e and consequently the singularities by computing shortest rotation angles along dual edges, and then extracting the rest of the polar decomposition using linear algebra. Principal matching for an N-RoSy field takes θ_e as the smallest rotation between two adjacent field elements.

Singular matching (§5). Our first and most critical step considers the singular topology of the input fields. We define a version of (3) that can cope with negative mass and prove a lemma showing that the matching T_{ij} contains integers if the matched distributions m_i, m'_j are integer. This lemma allows us to solve a matching problem between the field indices encoded in K^0_v and those in K^1_v (superscript denotes time t), which must sum to the same value as discussed in §3.1.

Curvature interpolation (§6). Our next step is to interpolate between the 0-forms ψ^0 and ψ^1. Our matching from the previous step yields a matrix showing how much of the curvature of each singular point and boundary loop in the first field gets mapped to the corresponding objects in the second field. We synthesize a time-varying potential ψ^t by solving a linear problem: We *prescribe* $K^t_v = \Delta_0 \psi^t$ at the interior vertices by moving singular indices along geodesics with constant speed, and boundary curvature is chosen to make $\partial \psi^t / \partial t$ as small as possible.

Generator interpolation (§7). We need to adjust the generator indices through time to make the reconstructed field as smooth as possible. As we demonstrate, this step is essential even if the generator indices are the same in both source and target.

Detail function interpolation (§8). Our remaining angular task is to interpolate the detail function ψ_f. Since we have taken care of the critical topological features in previous steps, we find that a simplistic method suffices.

Reconstruction (§9). At this point, we are ready to convert our field to xyz coordinates. We solve a linear problem to find the Euclidean embedding of our field in a numerically stable fashion and recover the global rotation by solving a small synchronization problem.

5 SINGULAR MATCHING

Before producing interpolated components of the polar representation, we first solve a small but critical linear program determining the structure of our interpolant, illustrated in Figure 2.

Recall that the field curvature K satisfies a number of index-related properties. Most importantly, K for any N-field equals an integer multiple of $2\pi/N$ at any interior vertex and when it is summed around any boundary loop; this sum determines the index of the field

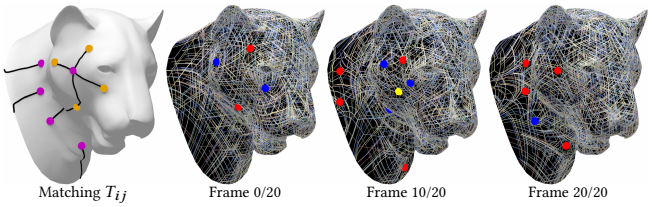

Fig. 2. Given two (3-RoSy) fields on a surface (frames 0, 20), we match the singular points (left; orange for the first field, purple for the second) using the signed matching problem (4). Our interpolant is designed to slide singularity indices along the geodesics; at $t = 0.5$ (frame 10) the singular points have split their indices according to the matching and moved along the geodesics between the start and end points.

at these locations. Note in this section that we disregard homology-generating loops since they do not appear in the index formula (1).

With these observations in mind, we collect vectors q^0 and q^1—not necessarily of the same length—containing the nonzero index of each singular point and boundary loop in the two input fields. Thanks to the topological properties above, we know that $\sum_i q^0_i = \sum_j q^1_j$; hence, in some sense we can think of the indices as "mass" to be transported along the mesh during the course of the interpolation, matched using an optimal transport problem (§3.2).

Unlike the Wasserstein distance (3), however, q^0 and q^1 can have negative entries. Hence, we propose solving the following extension of (3) to generate a matching:

$$\begin{cases} \min_T & \sum_{ij} D^2_{ij} |T_{ij}| \\ \text{subject to} & \sum_j T_{ij} = q^0_i \\ & \sum_i T_{ij} = q^1_j. \end{cases} \tag{4}$$

This extension of transport allows us to transport negative mass $T_{ij} < 0$ at the same cost as transporting its positive counterpart; after introducing some slack variables, this problem can be solved as a linear program. Here, D_{ij} denotes geodesic distance along the mesh; if entry i or j corresponds to a boundary loop rather than a singular point, we take D_{ij} to be the minimum distance to any point on that loop. We use fast marching to approximate these distances along the mesh [Kimmel and Sethian 1998]. Note that this optimization problem does not "match" in the sense of moving whole singularities, but rather splits and merges indices according to the objective function. We pose no restriction on the numbers or locations of indices at the end times (except compliance with the index theorem).

A hypothetical concern about (4) is that it does not explicitly constrain T_{ij} to be integer. For example, if we split the mass at an index-1 vertex of a vector field in half and transport the two halves to different destinations, it would be unclear how to construct an interpolant respecting this matching: Singular points of vector fields do not have fractional indices. Somewhat surprisingly—and further supporting the suitability of transport for vector field interpolation—the following lemma shows that this situation never occurs:

Proposition 5.1. *The optimization problem* (4) *always admits a solution with* $T_{ij} \in \mathbb{Z}$ *for all* i, j *when* $q^0_i, q^1_j \in \mathbb{Z}$ *for all* i, j.

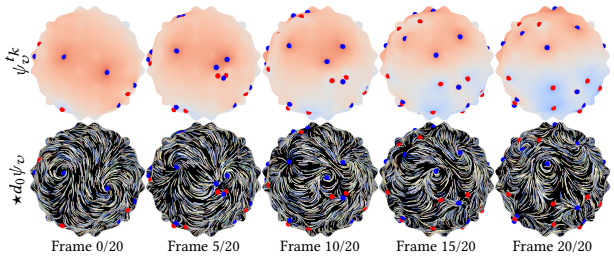

Fig. 3. Interpolation of the scalar potential ψ_v using the method described in §6 on a model without boundary curves; in this case it is not necessary to solve (6), and instead K_v is prescribed using the "splatting" procedure directly.

Proof. Decompose T into positive and negative parts $T = R - S$ with $R, S \geq 0$. Then, (4) can be rewritten as a linear program:

$$\begin{cases} \min_T & \sum_{ij} D_{ij}^2 (R_{ij} + S_{ij}) \\ \text{subject to} & \sum_j (R_{ij} - S_{ij}) = q_i^0 \\ & \sum_i (S_{ij} - R_{ij}) = -q_j^1 \\ & R, S \geq 0. \end{cases} \tag{5}$$

In this form, each R_{ij} and S_{ij} appears in two constraints with opposite-signed coefficients ± 1; hence, the constraints of (5) are totally unimodular by [Allaire 2007, Proposition 11.3.15]. Then, by [Allaire 2007, Corollary 11.3.11] the extremal points of the constraint polyhedron are integer, implying the desired result. □

We use MOSEK [2017] to solve (4) efficiently as a linear program; there are relatively few variables since the indices i, j are over singular points and boundary loops rather than the entire set of mesh vertices. Since MOSEK uses an interior point solver, we communicate an integer constraint explicitly to account for the rare case where the solution of the matching problem is nonunique. Due to Proposition 5.1, this integer constraint has no effect on the optimal value, and in practice we find it does not affect efficiency of the solve. Note the integer constraint is not strictly necessary, in the sense that an algorithm like the simplex method would find the integer solution by construction; in practice, integrality constraints are handled by MOSEK by switching to simplex or using the interior point method plus branch-and-bound rounding.

6 CURVATURE INTERPOLATION

Our next task is to use the matching T_{ij} to generate a time-varying potential function ψ_v, equivalent to interpolating curvature K_v in time. We accomplish this task in two stages. First, a "splatting" stage places nonzero values in T_{ij} on the corresponding geodesic from i to j, effectively dragging singular points along geodesics with constant speed in time. Second, when the surface has a boundary, we distribute curvature along boundary loops; this second step is omitted when the surface lacks a boundary.

6.1 Boundary-Free Case

The key idea of the curvature interpolation step is to drag singular points along the surface, with indices determined by the matching T_{ij}. As illustrated in Figure 5, when the mesh has no boundary, this step is *prescriptive*: No optimization problem is needed. A sample result of this procedure is shown in Figure 3.

When the underlying mesh has no boundary, curvature interpolation takes place as follows. After solving (4) for the matching T, we iterate over every nonzero element of T_{ij}; since the mesh has no boundary, i and j correspond to vertices of the mesh. For every time $t \in (0, 1)$, we place ("splat") a singularity of index T_{ij} at a vertex on the geodesic from vertex i to the vertex j; the time t determines the distance traveled along the geodesic. This index (scaled by $2\pi/N$) is stored as a nonzero element of the curvature vector K_v^t; the remaining elements of K_v^t are zero by default. Then, the scalar potential ψ_v^t at time t can be obtained by solving a Poisson equation: $\Delta_0 \psi_v^t = K_v^t - \overline{K}_v$. The end result is a sequence of scalar potentials ψ_v^t determining the topology of the field at each intermediate time t; in subsequent steps, the detail function ψ_f is used to modify the local geometry of the field without affecting its topology.

6.2 Interpolation with Boundary

The general case for curvature interpolation including boundary loops is more challenging, since when a singularity merges with (or splits from) a boundary, its index jumps. We need to "spread" the curvature gradually along all vertices in the affected boundary loop to make the change seamless in time. Hence, in the presence of boundary loops, our second stage uses a linear system to account for the case when the geodesic from i to j touches the boundary, allowing for curvature to be distributed to more than one boundary vertex. In particular, we find the smoothest ψ_v^t in time t respecting prescribed curvature on boundary loops and at interior vertices.

In this case, the splatting stage computes *two* quantities at each time t_k: A prescribed curvature $K_v^{t_k}$ for each interior vertex v and a "singular budget" $r_b^{t_k}$ for each boundary loop $b \in B$; the latter prescribes the total curvature in loop b but does not determine its distribution among b's vertices. These two together account for all the curvature K_v in the input fields, prescribing its location at each time step as either on an interior vertex or somewhere along a boundary loop. For each nonzero element of T_{ij}, we accumulate $K_v^{t_k}$ for interior vertices v and $r_b^{t_k}$ over the time steps t_k as follows:

- If i and j both correspond to interior vertices, we "splat" $2\pi T_{ij}/N$ along the geodesic from i to j as in §6.1. If the geodesic touches a boundary vertex in loop $b \in B$ at time t_k, in that time step we instead add $2\pi T_{ij}/N$ to the singular budget $r_b^{t_k}$.
- If either i or j is a boundary loop, we do the same but use the shortest path to any vertex in the loop as our path. To account for the time it takes to gather boundary curvature at a single vertex before splitting off/merging a singular point from/to the boundary, we start the interior path at $t \in \{1/4, 3/4\}$ and add to the singular budget of the loop before or after this point in time.
- If $i = j$ and both correspond to a boundary loop, we add $2\pi T_{ij}/N$ to the loop's singular budget for all times t_k.

Our implementation uses the fast marching method to approximate geodesic curves [Kimmel and Sethian 1998]. For simplicity, if our mesh does not contain a vertex exactly on a geodesic at a time point t_k, we snap to the closest vertex; our algorithm is fast enough that we can afford to use fairly dense meshes. If this snapping is unacceptable, one simply could subdivide the mesh *a priori* so that

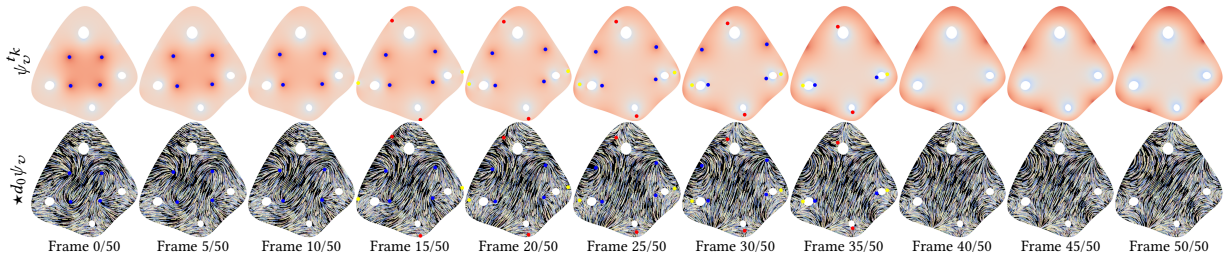

Fig. 4. Interpolation of the scalar potential ψ_v for a 2-RoSy, using the method described in §6 on a model with boundary curves. Note the splitting and merging of singularities into the boundaries, and how the curvature of the boundary vertices adapt accordingly.

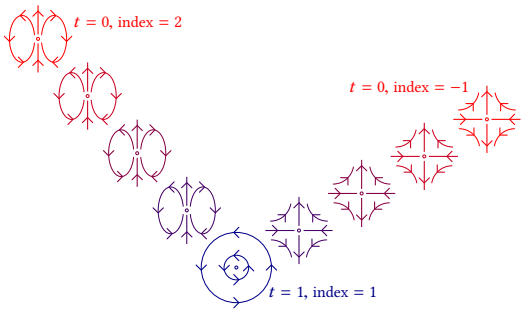

Fig. 5. Intuition for "curvature interpolation" (§6). Here, we wish to interpolate from a field with an index-2 singularity as well as an index-(-1) singularity to a field with a single index-1 singularity. The curvature interpolation step prescribes the locations and indices of the singular points at intermediate times by moving singular points with index T_{ij} along the geodesic from singularity i at $t = 0$ to singularity j at $t = 1$ with constant speed. No optimization is needed unless the domain has a boundary.

all sampled geodesic points appear as vertices; this would have little effect on the efficiency of the algorithm.

Next, we distribute the boundary singular budget to the individual boundary vertices in each time step using a variational problem; an example of the output is shown in Figure 4. Recall that $\Delta_0\psi_v = K_v - \overline{K}_v$, where Δ_0 is the cotangent Laplacian defined in §3.1. The procedure above prescribes K in the interior vertices at all time steps. All that is left is to find K on the boundary vertices with the constraint that the sum over each loop b at time t_k is prescribed as $r_b^{t_k}$. We do so by solving the following optimization problem:

$$\begin{cases} \min_{K,\psi_v} \ \frac{1}{2}\sum_k \|\psi_v^{t_k} - \psi_v^{t_{k-1}}\|_2^2 \\ \text{subject to } \Delta_0\psi_v^{t_k} = K_v^{t_k} - \overline{K}_v \ \forall v \in V, k \in \{0,\dots,n\} \\ \qquad\qquad K_v^0, K_v^1 \text{ prescribed } \forall v \in V \\ \qquad\qquad K^{t_k} \text{ prescribed } \forall v \in V\backslash B, k \in \{0,\dots,n\} \\ \qquad\qquad \sum_{v\in b} K_v^{t_k} = r_b^{t_k} \ \forall b \in B. \end{cases} \tag{6}$$

Again, solving this optimization problem is only necessary when the mesh contains boundary loops; otherwise the splatting procedure above completely determines K.

If we eliminate redundant variables by substituting $\psi_v = \Delta_0^+(K_v - \overline{K}_v)$ and removing prescribed elements of K, the true number of unknowns in (6) equals the number of boundary vertices. The optimization is a linear least-squares problem subject to linear constraints, which we solve using the linearly-constrained variation of conjugate gradients (CG) in [Shariff 1995]; Appendix A derives the iterations. The number of variables is the number of time steps times the number of boundary edges; we use CG because the system involves a Kronecker product of easily-inverted small matrices.

The rationale behind (6) is that we wish the time-varying potential ψ_v to be as smooth as possible a function of time. We choose to regularize temporal smoothness of ψ_v rather than K since K is *not* smooth in the L_2 sense: It is a sequence of moving δ-functions representing singularities moving along the surface.

Solving (6) yields a time-varying ψ_v that satisfies all the topological properties needed to reconstruct a field at intermediate time steps. Specifically, we have the following properties:

- $K_v^{t_k} = \Delta_0\psi_v^{t_k}$ is $2\pi/N$ times an integer at all steps at internal vertices $v \in V\backslash B$ by construction,
- around boundary loops c, $K_v^{t_k}$ also sums to $2\pi/N$ times an integer $I_c^{t_k}$ giving the index we will need to construct the exact part of the interpolated field, and
- $K_v^{t_k}$ sums to the same value over $v \in V$ in each time step.

7 GENERATOR INTERPOLATION

The interpolation of generator indices I_c for $c \in G$ is more intricate than moving singular vertices, since they cannot simply exchange curvature or merge with moving singular vertices. The main challenge is that generators do not appear in the index theorem (1) and hence are not easily included in our modified transport problem (4).

Even if the two input fields have matching generator indices $I_c^0 = I_c^1$, it is sometimes necessary to change the index of a generator at some intermediate time $t \in (0,1)$ to obtain a smooth field. An example is shown in Figure 6. Suppose that the interpolant leaves the generator indices fixed and only moves singularities. Consider a singular point with nonzero curvature K_v that crosses a generator loop; the effect is similar to sliding the generator over the singularity by adding the singularity cycle around v (a row in $d_0^\top$) to the generator cycle (a row in G). Next, consider the edges shared between the two vertices v_i, v_{i+1} along which the singularity traveled. In the subsequent time step, we would immediately add or subtract K_v to/from sum of θ_e on these edges. If we keep the generator index I_c fixed and consequently fix the sum of θ_e over the generator dual edges, then the K_v change over the shared edge between v_i and v_{i+1} would force all the other θ_e values in the generator cycle to change abruptly by $-K_v$ to compensate. For this case, a reasonable solution is to change the generator index to negate this behavior.

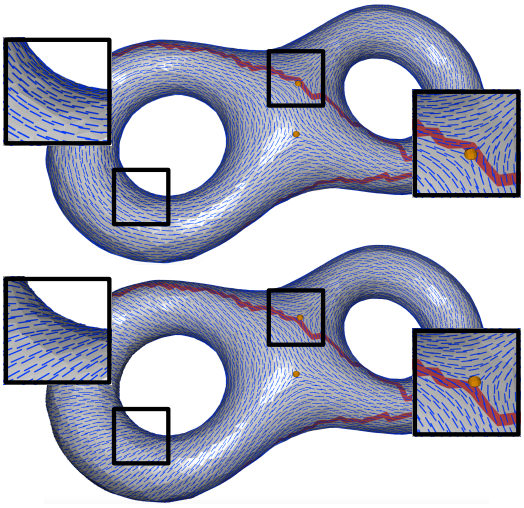

Fig. 6. When a prescribed singular point crosses over a homology generator, the resulting trivial connection field can change considerably. Here, the two rows show fields generated after moving a singular vertex (right) a small distance from one side of a generator to the other; the field undergoes an extreme change on the opposite handle (left).

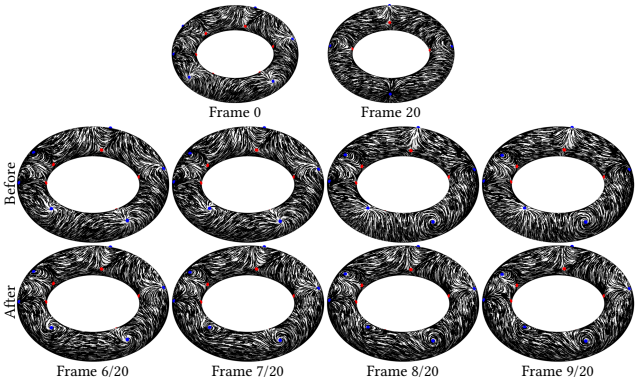

Fig. 7. Interpolated fields without (top) and with (bottom) generator interpolation. Note how in frames 7 and 8 our procedure automatically removed the generator index artifact.

It is not obvious how to generalize this example to multiple singularities that do not all pass through generators on their paths. This is further complicated when we need to interpolate different generator indices from the end times $t \in \{0, 1\}$. Even more (see Figure 19), it is likely impossible to change generator indices smoothly in all cases without "spawning" singularities as for boundary loops.

To avoid these issues when possible, we compute the interpolated curvatures of the generators through a variational formulation that seeks the smoothest field in time. In particular, we solve the following problem for $\alpha \in \mathbb{R}^{2g \times n}$:

$$\begin{cases} \min_{\alpha, I_c} & \frac{1}{2} \sum_k \|\theta_e^{t_k} - \theta_e^{t_{k-1}}\|_{\text{Fro}}^2 \\ \text{subject to } P\alpha = \frac{2\pi I_c}{N} - \overline{K}_c - G\theta_{\text{exact}} \\ & I_c^0, I_c^1 \text{ prescribed.} \\ & \theta_e = H\alpha^{t_k} + \theta_{\text{exact}}^{t_k} \end{cases} \tag{7}$$

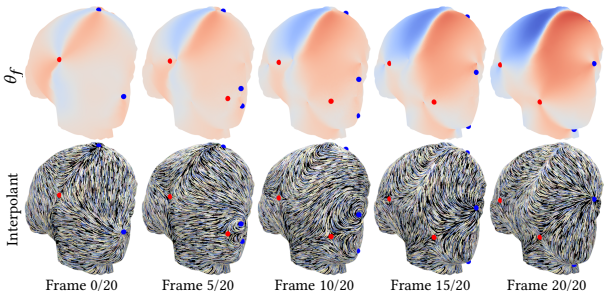

Fig. 8. Interpolation of detail function ψ_f.

where P is the period matrix as in §3.1, α is the set of linear coefficients in the harmonic basis represented in the columns of H, and $\theta_{\text{exact}} = \star_1 d_0 \psi_v$ is computed from our singularity interpolation in §6. I_c is the (integral) index of the generator c. Our current connection is then $\theta_e = H\alpha + \theta_{\text{exact}}$

Although this problem is an integer program with $2gn$ unknowns, we found that a greedy procedure of solving the relaxed problem without an integer constraint using a direct linear solve, rounding a single integer variable, fixing its value, and iterating produced acceptable solutions in all our examples, and that a globally-optimal integer solution did not produce better results.

8 DETAIL FUNCTION INTERPOLATION

The input fields may not be purely the result of an as-parallel-as-possible (smoothest) field design method, and thus contain some extent of detail function ψ_f^0, ψ_f^1 that we must interpolate. We do so with the same rationale as for generator correction: Find the ψ_f^t for which is field is smoothest temporally. Hence, we pose the following problem for the detail function ψ_f:

$$\begin{cases} \min_{\psi_f, \theta_e} & \frac{1}{2} \sum_k \|\theta_e^{t_k} - \theta_e^{t_{k-1}}\|_{\text{Fro}}^2 \\ \text{subject to } \theta_e^{t_k} = \star_1 d_0 \psi_v^{t_k} + P\alpha^{t_k} + d_1^\top \psi_f^{t_k}, \end{cases} \tag{8}$$

while interpolating the end conditions. The value of $\psi_f^{t_k}$ is defined only up to a constant per time frame. This constant does not matter in our formulation, since we only use the resulting $d_1^\top \psi_f$ to modulate our connection from the previous section. For simplicity, we keep it fixed at zero in an arbitrary face per time frame to remove the constant null space. After eliminating the variable θ_e, we solve this quadratic minimization using the conjugate gradient algorithm; an example result is shown in Figure 8.

9 RECONSTRUCTION

At this point in our algorithm, we have all the components we need to construct our time-varying field up to global rotation. Hence, we begin by reconstructing a unit complex vector $v_f^{t_k}$ per triangular face f and time t_k by simply fixing one vector in an arbitrary fixed face with some orientation and minimizing $\sum_{f,g} \left| v_f \exp(i\theta_e) - v_g \right|^2$, summed over all neighboring faces f, g on edge e. Since our interpolant satisfies all the criteria needed to be a polar Hodge decomposition, the minimum energy is zero; least-squares is used only for numerical stability. That is, reconstruction does not technically

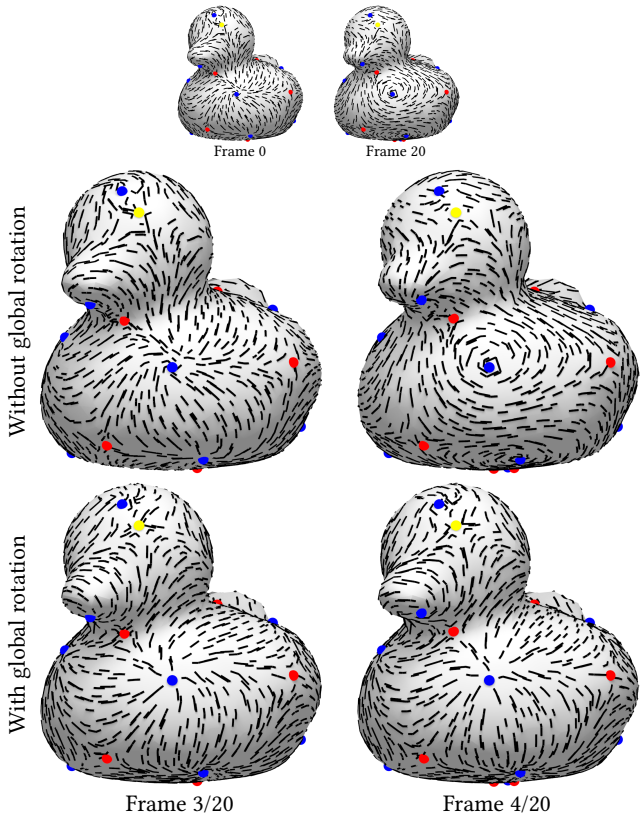

Fig. 9. Two frames of our interpolation sequence before (top) and after (bottom) the rotation synchronization. The fields in the bottom row are temporally coherent, while the top row snaps between adjacent frames.

require a linear system, but we find this approach to be stabler than breadth-first traversal without a significant change in run time.

We solve the reconstruction problem above individually per time step, so we still need to align all fields temporally. Recall that if we rotate a field by the same amount θ in each face, the polar representation does not change except the global rotation. Hence, our frame-by-frame reconstruction procedure in the previous paragraph does not produce temporally-coherent frames until we choose a global rotation, that we parameterize by a single $\theta_0^{t_k}$ for each frame.

Define $z_k := e^{i\theta_0^{t_k}}$. Conceptually, we might wish to minimize $\sum_k \|z_k v_f^{t_k} - z_{k-1} v_f^{t_{k-1}}\|_2^2$ with the constraint that z_1, z_n are prescribed to align with the input fields and that $\|z_k\|_2 = 1$ for all k. This resembles the well-known *angular synchronization* problem, which admits a convex relaxation [Singer 2011]:

$$\begin{cases} \min_{z_k, Z} & \langle Z, M \rangle \\ \text{subject to} & z_1, z_n \text{ prescribed from input fields} \\ & \operatorname{diag}(Z) = 1 \\ & \left(\begin{array}{c|ccc} 1 & z_1 & \cdots & z_n \\ \hline \overline{z}_1 & & & \\ \vdots & & Z & \\ \overline{z}_n & & & \end{array}\right) \succeq 0, \end{cases} \tag{9}$$

where M is the matrix of our quadratic form objective. This relaxation agrees with our original problem if we add a constraint that $Z_{ij} = \overline{z}_i z_j$; because we relaxed this constraint, after we solve this small convex semidefinite program whose size scales only with the number of time steps n, we simply normalize the resulting z_k's. Our final temporally-coherent field is given by $\overline{u}_f^{t_k} := z_k v_f^{t_k}$.

Figure 9 shows an example of the global rotation procedure above acting on a sequence of fields. After the rotation step that the fields in two adjacent frames are extremely close to one another, while before they differ considerably.

10 EXPERIMENTS

For the most part, experiments showing the output of our algorithm on challenging models are interspersed with our discussion above. Our examples include meshes with boundary (e.g., Figures 1 and 4) and nontrivial topology (e.g., Figure 7). The video accompanying this paper also shows some more densely-sampled interpolations in time. Below we discuss a few more experimental results relevant to evaluation of our technique.

Figure 10 illustrates resilience of our algorithm to time discretization. Here, we show interpolants between the same pair of vector fields with varying numbers of time steps n. While solving our optimization problem with larger n slightly improves temporal coherence, the main role of n is simply to control the number of desired interpolation frames.

Figure 11 shows an example of our algorithm interpolating between fields of extremely different singular structures as a stress test. Our transport problem still is able to recover a reasonable matching, leading to a collection of singularities splitting off from the few source points in frame 0.

Figure 12 shows an example where the singular points of the two input fields are on either side of a hole in the model. Despite this obstruction to sliding the singular point directly, our curvature interpolation method described in §6 handles this case reasonably: The singular points exchange curvature through merging and splitting with the boundary loop.

Another related challenging case is shown in Figure 13, in which the indices of the singularities are opposite in sign yet are placed in the same positions in both end time frames; this constitutes a large deformation in the field. In this case, the singularities "exchange" their curvature with the boundary loop.

Figure 14 shows a stress test for the generator interpolation method in §7; Figure 15 shows the corresponding matching of singular points generated using the transport problem in §5. Our integer search strategy correctly identifies shifts in the indices of the homology generators, maintaining smoothness of the interpolant through the sequence.

Figure 16 gives timings for the various stages of our interpolation algorithm on some test models shown in other figures. Given the large number of outputs produced by our algorithm for each time step (a ψ_v per vertex v, a ψ_f per triangular face f, and smaller variables α and θ_0), our algorithm is fairly efficient; steps other than generator integer rounding are either prescriptive, linear solves, or small convex programs. Note one outlier in our timings is the amount of time that it takes to compute geodesic curves between

Fig. 10. Stability of our interpolant to time discretization. Here, we run the same procedure for $n \in \{5, 10, 20, 50\}$ and show frames aligned at the same times t.

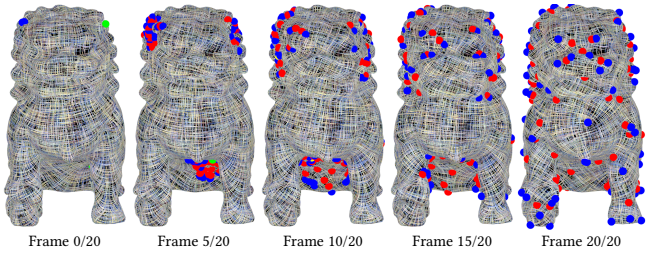

Fig. 11. Interpolation stress test with a 4-RoSy field: A simple field with relatively few singular points is interpolated to a dense field with many singular points.

singular points and boundary loops; we are extremely confident that an efficient implementation of fast marching would make this contribution to our timings negligible. Our entire algorithm is coded in MATLAB, where the convex problems use MOSEK called from CVX; the timings are based on experiments carried out on a 2015 iMac with 4GHz Intel i7 core and 32Gb of memory.

Figure 17 shows the result of using our interpolated fields to guide seamless parameterization, using the implementation of [Bommes et al. 2009] by Vaxman et al. [2017]. Our interpolant slides singular points along the surface, leading the resulting quad pattern to have singularities at exactly these locations.

While there are not many obvious points of comparison for our algorithm in geometry processing, we can compare against an extremely simple interpolant. In particular, if our input fields are are specified in xyz coordinates as u^0 and u^1, a algebraic interpolant at time t can be written as $tu^1 + (1-t)u^0$. This simpler interpolation strategy is unaware of singular topology, and hence singular points can appear and disappear as t progresses. Figure 18 shows an example comparing our method to this approach.

11 DISCUSSION AND CONCLUSION

Interpolating vector fields in time is a relatively new problem. While our algorithm incorporates geometric semantics, like interpolating the topology of the field, which cannot be obtained by naïve algebraic averaging of field elements, there is still room for extension and improvement in future work.

Probably the most critical future work will involve extension from two-dimensional fields to three-dimensional volumes, a critical extension for applications in visualization and simulation. While the polar representation of fields used in this paper and many others in geometry processing provides critical insight into field topology on surfaces, an equally well-understood analog has yet to be developed for volumes and may be challenging to derive given noncommutativity in SO(3). Even in the theory of smooth vector field topology, the definition of index and other topological invariants for volumetric fields becomes fairly complex to state and manipulate; see [Liu et al. 2018] as an example of the complexity when considering topological invariants of higher-dimensional fields.

Another limitation of our work, that we exemplified in the double-torus and torus cases, is that interpolating the field on generator handles is done somewhat heuristically; it is the only nonconvex step in our process. It is not obvious that our interpolation is always smooth in this case. A simple example is in Figure 19, where there is a difference in index around a single generator handle, which does not have any singularities with which to blend indices smoothly in time. We conjecture that a process of merging and splitting singularities into the generator may be possible and leave a theoretically sound formulation for future work.

Other aspects of our algorithm could benefit from expansion or improvement. For instance, we currently prescribe that singularities split from the boundary at times $t \in \{1/4, 3/4\}$, but it may be possible to formulate a better heuristic or variational problem that accounts for the length of the geodesic to the boundary relative to the time needed to gather curvature at a single boundary point. Also, since we consider interpolation of polar fields, an additional step is needed

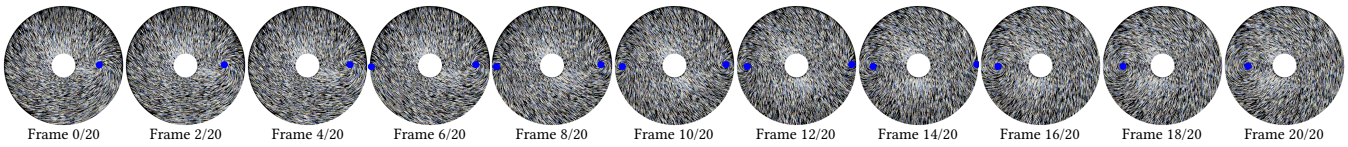

Fig. 12. Interpolation of fields on an annulus in which one singular point merges with the boundary, while another splits from it.

Fig. 13. At $t = 0$, there are four $\frac{1}{4}$ singularities, that deform into $-\frac{1}{4}$ singularities at $t = 1$. This is made possible by exchanging curvature with the boundary loop through singularities splitting and merging.

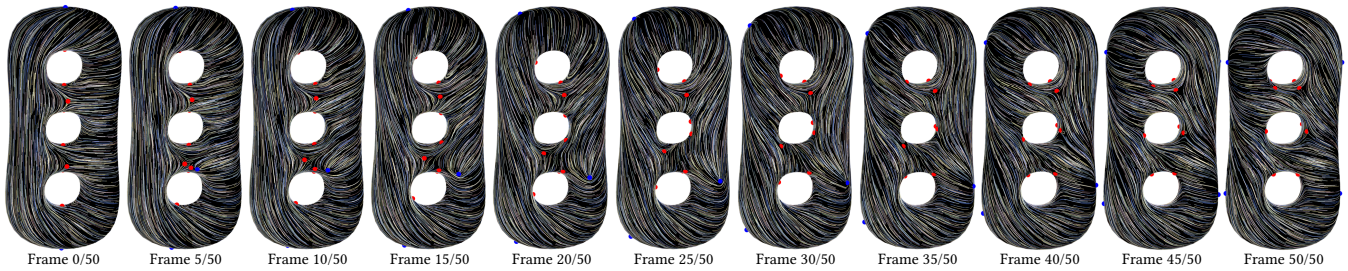

Fig. 14. Stress test of our technique for handling surfaces with nontrivial genus.

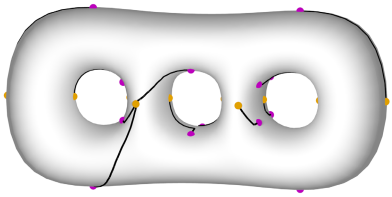

Fig. 15. Singular matching for the example in Figure 14.

to interpolate the norm of the field per face; while linear techniques with nonnegativity constraints and/or conserved total norm often suffice, one more complex possibility might be to push forward the magnitude function under a self map of the surface taking singular points from their sources to the target in the interpolant, e.g. using coparameterization methods like [Aigerman and Lipman 2016].

Beyond these mathematical extensions, additional challenges appear in extending our general method to cope with demands of different applications. For instance, in fluid simulation it may be desirable to promote divergence-free properties as suggested by Sato et al. [2018]. For field design applications, incorporating rough user guidance in design of the interpolant in an intuitive and effective manner has nontrivial interplay with our topological constructions. We also can attempt to link this work to machine learning, e.g. interpolating time series of vector fields gathered at sparse collections of points in space and time.

Even without these improvements, our method linking vector field structure to optimal transport for interpolation shows how these two disciplines can provide mutual insight to a practical end. Interpolating vector fields using this machinery yields expected qualitative behavior using well-posed machinery understandable in both the discrete and smooth contexts.

ACKNOWLEDGMENTS

We thank Mirela Ben-Chen, Nicolas Bonneel, and the MIT Geometric Data Processing group for discussion during the design of this paper. J. Solomon acknowledges the support of Army Research Office grant W911NF-12-R-0011, of National Science Foundation grant IIS-1838071, from the MIT Research Support Committee, from an Amazon Research Award, from the MIT–IBM Watson AI Laboratory, and from the Skoltech–MIT Next Generation Program.

A CONJUGATE GRADIENTS FOR ψ_v

Take $X \in \mathbb{R}^{n_b \times (n-2)}$ to be the restriction of the matrix K to just boundary vertices for $t_k \in \{t_2, \ldots, t_{n-1}\}$, where n_b is the number of boundary vertices; this is the true unknown of our problem since K is prescribed at all time steps in interior vertices and ψ_v is related to K through L. In particular, we can write $K = \overline{K} + HX$, where $H \in \{0,1\}^{|V| \times n_b}$ lifts boundary values to a function over the mesh with zeros in non-boundary locations and $\overline{K}$ contains the prescribed curvatures in interior vertices and at $t \in \{0,1\}$. If $D \in \{-1,0,1\}^{(n-1)\times n}$ is the finite time-differencing operator and $T \in \{0,1\}^{n\times(n-2)}$ inserts a zero as the first and last element of a

Name	Fig.	$\|V\|$	$\|E\|$	$\|F\|$	g	$\|V_B\|$	$\|B\|$	Preprocess	Geodesic	Matching (§5)	Splat (§6)	K_v optim. (§6)	Generator (§7)	Detail (§8)	Recon. (§9)
beetle	1	17908	52645	34728	0	1106	11	0.92	103.02	2.20	0.43	367.07	0.01	16.61	14.72
lion	2	8356	25029	16674	0	36	1	0.38	2.88	2.54	0.07	0.90	0.00	2.86	3.57
bumpy sphere	3	5724	17166	11444	0	0	0	0.25	10.81	1.99	0.18	0.06	0.01	4.78	5.09
flatmesh	4	3759	11054	7292	0	232	5	0.15	2.23	1.91	0.04	23.57	0.01	2.25	2.96
torus	7	20494	61482	40988	1	0	0	0.83	19.75	2.10	0.27	0.10	1.09	6.99	6.70
igea	8	25282	75840	50560	0	0	0	1.28	16.82	2.53	1.11	0.15	0.01	10.61	10.14
duck	9	901	2697	1798	0	0	0	0.06	4.70	2.50	0.11	0.02	0.01	0.56	1.58
moomoo	10	1045	3129	2086	0	0	0	0.06	4.08	1.91	0.10	0.02	0.01	0.58	1.58
chinese lion	11	25194	75576	50384	0	0	0	1.21	279.13	2.65	4.30	0.15	0.01	10.55	10.10
annulus	12	20800	61760	40960	0	640	2	0.89	2.45	1.93	0.08	26.05	0.01	6.27	6.30
lilium	13	3389	9978	6590	0	186	1	0.17	1.31	2.70	0.04	12.37	0.01	2.37	3.17
hole3	14	5884	17664	11776	3	0	0	0.27	10.85	1.84	0.15	0.05	1.44	5.23	5.43
sphere	17	642	1920	1280	0	0	0	0.05	1.44	1.82	0.09	0.02	0.00	0.18	1.14

Fig. 16. Timings in seconds for examples in this paper. Here, $|V_B|$ denotes the number of boundary vertices and $|B|$ denotes the number of boundary loops.

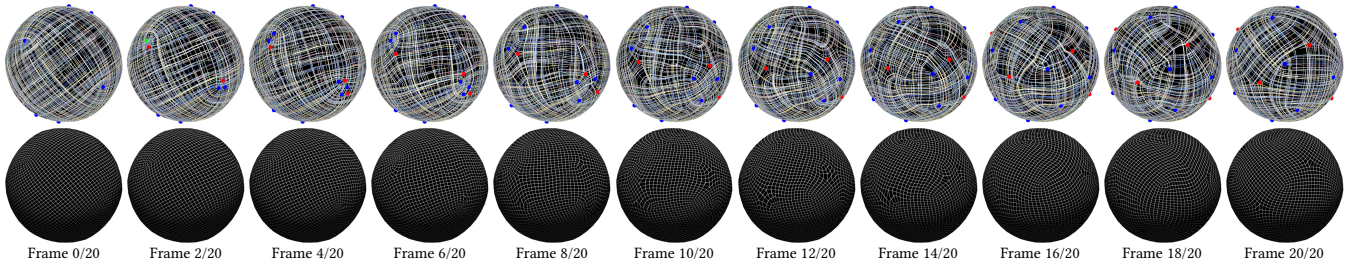

Fig. 17. Our interpolation between 4-RoSy fields (top) can be integrated into a gradual progression of seamless parameterizations of a sphere (bottom).

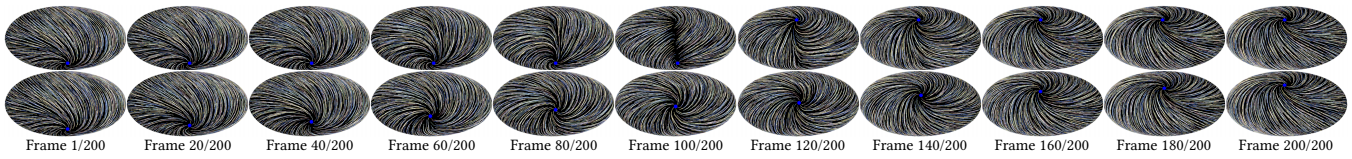

Fig. 18. Linear interpolation of vector field coefficients in standard coordinates does not move singular points (top row), while our method slides singular points in an intuitive fashion (bottom row). Furthermore, linear interpolation results in "hairline" singularity line artifacts (Frame 100).

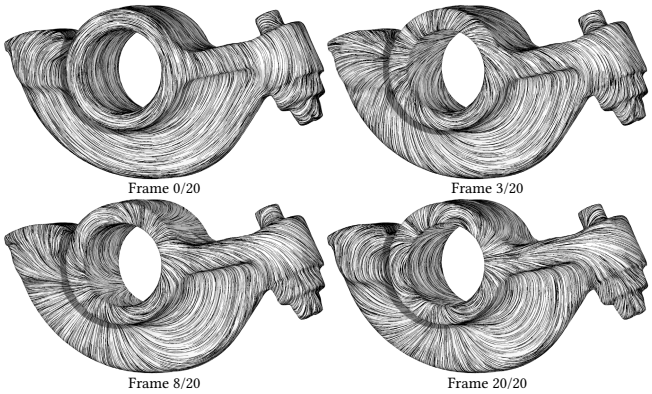

Fig. 19. The interpolation between the genus 0 rockerarm, with index 0 in time $t = 0$ and index -3 in time $t = 1$ cannot be made smoothly in our framework. Instead, our system introduces "jumps" in indices in the given time frames, and little progress beyond.

vector, we can rewrite (6) as

$$\begin{cases} \min_X & \frac{1}{2}\|L^+(\overline{K} + HXT^\top)D^\top)\|^2_{\text{Fro}} \\ \text{subject to } SX = B. \end{cases} \tag{10}$$

Here, $B \in \mathbb{R}^{n_\ell \times n}$ is the singular budget, given as one real value per time step per boundary loop, and $S \in \{0,1\}^{n_\ell \times |V|}$ sums over boundary loops. In this form, we have reduced our number of unknowns to one value per boundary vertex per time step, a considerable savings over the original formulation (6).

Even though it has a matrix as an unknown, the formulation (10) is a quadratic least-squares problem with a linear constraint. Hence, we use the variant of conjugate gradients derived in [Shariff 1995] solve this problem iteratively. Pseudocode is provided in Algorithm 1. It is simple to obtain a feasible initializer for X using standard linear algebra techniques since the relationship $SX = B$ is underdetermined. Our implementation of Algorithm 1 parenthesizes matrix-matrix products to avoid ever creating a $|V| \times |V|$ matrix. We use a tolerance of $\varepsilon = 10^{-5}$.

REFERENCES

Noam Aigerman and Yaron Lipman. 2016. Hyperbolic orbifold Tutte embeddings. *ACM Trans. Graph.* 35, 6 (2016), 217–1.

Grégoire Allaire. 2007. *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation.* Oxford University Press.

Alexis Angelidis and Fabrice Neyret. 2005. Simulation of smoke based on vortex filament primitives. In *Proc. Symposium on Computer Animation.* ACM, 87–96.

Mirela Ben-Chen, Craig Gotsman, and Guy Bunin. 2008. Conformal flattening by curvature prescription and metric scaling. In *Computer Graphics Forum*, Vol. 27.

Algorithm 1 Constrained conjugate gradients for ∇_v

function ConjugateGradients

// Convert to $min_X \|MXN + P\|_{\text{Fro}}$ *s.t.* $SX = B$

$M \leftarrow L^+ H$

$N \leftarrow (DT)^\top$

$P \leftarrow -L^+ \overline{K} D^\top$

// Begin [Shariff 1995]

Initialize X with any feasible point

Define $f(Y) : Y \mapsto M^\top MYNN^\top$

$G \leftarrow f(X) + M^\top PN^\top$

// Kernel projection operator

$Q \leftarrow (SS^\top)^{-1}$

Define $\text{proj}(Y) : Y \mapsto Y - F^\top QFY$

$Z \leftarrow \text{proj}(G)$

$E \leftarrow -Z$

// Conjugate gradient main loop

$d \leftarrow \langle E, E \rangle$

while $\langle E, E \rangle > \varepsilon d$ **do**

$O \leftarrow f(E)$

$\alpha \leftarrow -\langle G, E \rangle / \langle O, E \rangle$

$X \leftarrow X + \alpha E$

$\delta \leftarrow \langle G, Z \rangle$

$G \leftarrow G + \alpha O$

$Z = \text{proj}(G)$

$\beta \leftarrow \langle G, Z \rangle / \delta$

$E \leftarrow -Z + \beta E$

end while

return $MXN + P$

end function

Wiley Online Library, 449–458.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. In *ACM Transactions On Graphics (TOG)*, Vol. 28. ACM, 77.

Keenan Crane, Fernando De Goes, Mathieu Desbrun, and Peter Schröder. 2013. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses*. ACM, 7.

Keenan Crane, Mathieu Desbrun, and Peter Schröder. 2010. Trivial connections on discrete surfaces. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 1525–1533.

Fernando de Goes and Keenan Crane. 2010. *Trivial connections on discrete surfaces revisited: A simplified algorithm for simply-connected surfaces.* Technical Report.

Stefan de Vries. 2016. *Modeling sediment transport pathways in the mouth of the Scheldt estuary.* Master's thesis. University of Twente.

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

Nahum Farchi and Mirela Ben-Chen. 2018. Integer-only cross field computation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 91.

Simon Fiedler. 2018. Flip Subframe Interpolation. https://vimeo.com/251487780

Christoph Garth, Xavier Tricoche, and Gerik Scheuermann. 2004. Tracking of vector field singularities in unstructured 3D time-dependent datasets. In *Proceedings of the Conference on Visualization*. IEEE Computer Society, 329–336.

Guangfeng Ji and Han-Wei Shen. 2006. Feature tracking using earth mover's distance and global optimization. In *Pacific Graphics*, Vol. 2.

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

Ron Kimmel and James A Sethian. 1998. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences* 95, 15 (1998), 8431–8435.

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2013. Globally optimal direction fields. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 59.

Hugo Lavenant, Sebastian Claici, Edward Chien, and Justin Solomon. 2018. Dynamical Optimal Transport on Discrete Surfaces. *ACM Transactions on Graphics (TOG)* 37, 6 (Dec. 2018), 250:1–250:16.

Bruno Lévy. 2015. A numerical algorithm for L_2 semi-discrete optimal transport in 3D. *ESAIM: Mathematical Modelling and Numerical Analysis* 49, 6 (2015), 1693–1715.

Bruno Lévy and Erica L Schwindt. 2018. Notions of optimal transport theory and how to implement them on a computer. *Computers & Graphics* 72 (2018), 135–148.

Wan-Chiu Li, Bruno Vallet, Nicolas Ray, and Bruno Levy. 2006. Representing higher-order singularities in vector fields on piecewise linear surfaces. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006).

Heng Liu, Paul Zhang, Edward Chien, Justin Solomon, and David Bommes. 2018. Singularity-constrained octahedral fields for hexahedral meshing. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 93.

Quentin Mérigot. 2011. A multiscale approach to optimal transport. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 1583–1592.

MOSEK. 2017. *The MOSEK optimization toolbox for MATLAB manual (Version 8.1).* http://docs.mosek.com/8.1/toolbox/index.html

Jonathan Palacios and Eugene Zhang. 2007. Rotational symmetry field design on surfaces. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 55:1–55:10.

Gabriel Peyré, Lénaïc Chizat, François-Xavier Vialard, and Justin Solomon. 2017. Quantum entropic regularization of matrix-valued optimal transport. *European Journal of Applied Mathematics* (2017), 1–24.

Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport. *Foundations and Trends in Machine Learning* 11, 5-6 (2019), 355–607.

Konstantin Poelke and Konrad Polthier. 2016. Boundary-aware Hodge decompositions for piecewise constant vector fields. *Computer-Aided Design* 78 (2016), 126–136.

Konrad Polthier and Markus Schmies. 1998. *Straightest Geodesics on Polyhedral Surfaces.* Springer Berlin Heidelberg, 135–150.

Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. 2006. Periodic global parameterization. *ACM Transactions on Graphics (TOG)* 25, 4 (2006), 1460–1485.

Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Levy. 2009. Geometry-aware direction field processing. *ACM Transactions on Graphics (TOG)* 29, 1 (2009), 1.

Nicolas Ray, Bruno Vallet, Wan Chiu Li, and Bruno Lévy. 2008. N-symmetry direction field design. *ACM Transactions on Graphics (TOG)* 27, 2 (2008), 10.

Filippo Santambrogio. 2015. Optimal transport for applied mathematicians. *Birkäuser, NY* (2015), 99–102.

Syuhei Sato, Yoshinori Dobashi, and Tomoyuki Nishita. 2018. Editing Fluid Animation Using Flow Interpolation. *ACM Transactions on Graphics (TOG)* 37, 5 (2018), 173.

Syuhei Sato, Yoshinori Dobashi, Yonghao Yue, Kei Iwasaki, and Tomoyuki Nishita. 2015. Incompressibility-preserving deformation for fluid flows using vector potentials. *The Visual Computer* 31, 6-8 (2015), 959–965.

MHBM Shariff. 1995. A constrained conjugate gradient method and the solution of linear equations. *Computers & Mathematics with Applications* 30, 11 (1995), 25–37.

Amit Singer. 2011. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis* 30, 1 (2011), 20.

Maxime Soler, Melanie Plainchault, Bruno Conche, and Julien Tierny. 2018. Lifted Wasserstein Matcher for Fast and Robust Topology Tracking. *Proc. IEEE Symposium on Large Data Analysis and Visualization* (2018).

Yousuf Soliman, Dejan Slepčev, and Keenan Crane. 2018. Optimal cone singularities for conformal flattening. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 105.

Justin Solomon. 2018. Optimal Transport on Discrete Domains. *Proceedings of Symposia in Pure Mathematics* (2018).

Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. 2015. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 66.

Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. 2014. Earth mover's distances on discrete surfaces. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 67.

Yiying Tong, Pierre Alliez, David Cohen-Steiner, and Mathieu Desbrun. 2006. Designing quadrangulations with discrete harmonic forms. In *Eurographics Symposium on Geometry Processing*. 1–10.

Xavier Tricoche, Gerik Scheuermann, and Hans Hagen. 2000. A topology simplification method for 2D vector fields. In *Visualization*. IEEE, 359–366.

Amir Vaxman et al. 2017. Directional: directional field synthesis, design, and processing. https://github.com/avaxman/Directional.

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

Cédric Villani. 2003. *Topics in optimal transportation.* Number 58. American Mathematical Soc.