Boundary Element Octahedral Fields in Volumes

JUSTIN SOLOMON
Massachusetts Institute of Technology
AMIR VAXMAN
Utrecht University
and
DAVID BOMMES
RWTH Aachen University

The computation of smooth fields of orthogonal directions within a volume is a critical step in hexahedral mesh generation, used to guide placement of edges and singularities. While this problem shares high-level structure with surface-based frame field problems, critical aspects are lost when extending to volumes, while new structure from the flat Euclidean metric emerges. Taking these considerations into account, this article presents an algorithm for computing such "octahedral" fields. Unlike existing approaches, our formulation achieves *infinite* resolution in the interior of the volume via the boundary element method (BEM), continuously assigning frames to points in the interior from only a triangle mesh discretization of the boundary. The end result is an orthogonal direction field that can be sampled anywhere inside the mesh, with smooth variation and singular structure in the interior, even with a coarse boundary. We illustrate our computed frames on a number of challenging test geometries. Since the octahedral frame field problem is relatively new, we also contribute a thorough discussion of theoretical and practical challenges unique to this problem.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Geometric algorithms & Systems*

General Terms: Algorithms, Theory, Experimentation

Additional Key Words and Phrases: Octahedral fields, boundary element method, frames, singularity graph

J. Solomon acknowledges the support of the NSF Mathematical Sciences Postdoctoral Research Fellowship (award number 1502435). D. Bommes was supported by the German Research Foundation (DFG, grant GSC 111, Aachen Institute for Advanced Study in Computational Engineering Science).
Authors' addresses: J. Solomon, 32 Vassar Street, room 32-D460, Cambridge, MA 02139 USA; email: jsolomon@mit.edu; A. Vaxman, Buys Ballot Laboratory, office 424, Princetonplein 5, De Uithof, 3584 CC Utrecht, The Netherlands; email: a.vaxman@uu.nl; D. Bommes, RWTH Aachen University, Schinkelstr. 2, 52062 Aachen Germany; email: bommes@aices.rwth-aachen.de.

DOI: http://dx.doi.org/10.1145/3065254

ACM Reference Format:
Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary element octahedral fields in volumes. ACM Trans. Graph. 36, 3, Article 28 (May 2017), 16 pages.
DOI: http://dx.doi.org/10.1145/3065254

1. INTRODUCTION

The design of direction fields in volumetric domains is a new problem that has gained momentum in recent years. Propelled by hexahedral (hex) mesh generation, most attention has been given to the computation of fields that assign three orthonormal directions to each point in a volume. These fields are designed to be agnostic to the ordering and sign of the individual directions at each point.

Hex meshes are desirable for applications in finite element (FEM) simulation. A hex mesh is essentially a warped three-dimensional grid covering a volumetric domain. The quality of a hex mesh depends on several parameters: the topological quality, reflected by a simple singularity graph; regularity, measured by evenly-sized and orthonormal hex elements; and alignment to the boundary of the domain. These qualities are inherited from those of the direction field used to guide the meshing process. Therefore, the design of smooth, orthonormal, boundary-aligned fields is a critical step in hex remeshing, and one that we pursue in this article.

We call such fields *octahedral direction fields* to emphasize their symmetric structure. This departure in terminology from the more generic term "frame fields" used in previous literature [Huang et al. 2011; Ray and Sokolov 2015] to refer to the same objects emphasizes that individual "xyz" labels are not assigned to the three directions, and in fact, some octahedral fields may not admit such a labeling globally due to topological restrictions. Rather, each set of directions has an octahedral symmetry group, isomorphic to the set of 90° rotations of a cube.

Common techniques to design surface-based tangent directional fields do not generalize easily to octahedral fields. Specifically, there are no clear generalizations of angle-based [Bommes et al. 2009] or complex [Knöppel et al. 2013] representations. The essential difficulty is that rotations are not commutative in three dimensions, and consequently, there is no trivial way to parameterize the set of rotations that morph one octahedral frame into another. In addition, classification of octahedral field singularities is still underexplored. Following Huang et al. [2011] and Ray and Sokolov [2015], we use a spherical harmonic parametrization of octahedral fields, with a few key differences, to help overcome some of these issues via a relaxation approach.

Octahedral fields are typically designed in volumes discretized by tetrahedral (tet) meshes; the boundary inherits a triangle mesh

discretization. Tet meshes facilitate the transfer of well-tested techniques from 2D symmetry field design on surfaces to this case. Nevertheless, working with tet meshes comes at a significant price. The complexity of tet meshes with fine elements can be prohibitive computationally, and tet meshes are themselves difficult to generate from a boundary mesh. Coarse tet meshes, on the other hand, induce noisy and ungainly field topology. This is unavoidable due to aliasing; singularity lines must follow the edges of the mesh.

Vector fields on surfaces require a mesh discretization to resolve curvature and non-trivial tangent bundles, and thus such disadvantages must be endured for surface-based problems. For the octahedral problem, however, volumes are flat and inherit the trivial Euclidean parallel transport, and tet mesh discretization is in a sense unnecessary for this problem. Simply put, tets do not convey meaningful geometric information for volumetric octahedral field design; they are simply a default numerical discretization of the problem.

In light of these observations, we offer a novel method to compute a *smooth* octahedral field aligned to prescribed boundary constraints on a triangle mesh. Our algorithm, which uses the boundary element method (BEM), uses variables sampled only at the boundary triangles. That is, we write our entire octahedral field pipeline in terms of a triangle mesh bounding the interior; the resulting field can then be queried at arbitrary points inside. Even with a coarse boundary, we obtain smooth fields, whose singularity graphs exhibit desirable structure.

2. RELATED WORK

2.1 Directional Field Design

Although far from a solved problem, the design of tangent directional fields on surfaces has been extensively studied in the literature; see Vaxman et al. [2016] for a recent survey. The most common paradigm for this design is to discretize a surface with a triangle mesh and compute a vertex-, edge-, or face-based directional field. Tangent planes are defined on these mesh elements, and directions are encoded according to a local basis. Differential notions of connection, parallel transport, and smoothness are discretized by defining maps between adjacent tangent spaces.

The use of triangle meshes means that the result can be sensitive to the discretization. Bad mesh elements affect the reliability of smoothness energies, and insufficient sampling affects the field topology considerably [Vaxman et al. 2016]. These artifacts can be somewhat alleviated by remeshing and refinement, but ultimately, there is no guarantee or reliability. Moreover, refining greatly increases the memory and time complexity of the computation. Methods that compute octahedral fields by discretizing volumes into finite elements [Huang et al. 2011; Ray and Sokolov 2015] inherit the same issues. As evidence, their singularity graphs appear jagged and noisy, and this effect does not vanish with refinement (see Figures 9 and 10).

Triangle meshes seem unavoidable for tangent directional field design, as there are not well-studied alternatives for computing and representing directional fields on surfaces that take curvature into account. When computing fields inside a 3D volume bounded by a surface, we do not have these limitations: the ambient space $\mathbb{R}^3$ is flat, and thus directions can be defined with a global coordinate system. It is then possible to work without volume discretization, as proposed in this article, using BEM.

2.2 Octahedral Frame Representation

For general N-directional fields on triangle meshes, where the field contains more than one vector per face, the definition of discrete differential properties requires *matching* individual vectors in adjacent tangent spaces. The matching can be done implicitly, by matching the vectors that are the most similar in some sense [Knöppel et al. 2013; Diamanti et al. 2014], or explicitly, by encoding all possible matching options as additional variables [Ray et al. 2008; Bommes et al. 2009]. The most common representation for explicit matching is angle-based [Li et al. 2006; Ray et al. 2008; Bommes et al. 2009], where an integer is used to determine the matching. Methods for implicit matching use Cartesian or complex representations [Ray et al. 2009; Knöppel et al. 2013], encoding a field with *N-rotational symmetry* by a single representative, using trigonometric (or complex exponential) functions.

Neither of these representations trivially extends to 3D fields with octahedral symmetry, as commutativity and other convenient structures from the rotation group SO(2) no longer apply. It is possible, however, to lift some ideas from representation by complex exponentials to SO(3) by use of the degree-four spherical harmonic basis, as introduced in Huang et al. [2011]. We choose to use this representation.

While on the plane there exists a natural identification between (non-zero) complex values and symmetric N-direction frames, for octahedral fields, there is a dimensionality mismatch. The basis of degree-four spherical harmonics is nine-dimensional, while the space of rotations is three-dimensional. For this reason, techniques like those used in Huang et al. [2011] and Ray and Sokolov [2015] can be considered *relaxations* of the octahedral field problem, requiring projection operators to return to the space of feasible frame functions. We employ a simple gradient descent iteration for this task, proposed in Ray and Sokolov [2015].

2.3 Octahedral Field Computation

Huang et al. [2011] discretize octahedral fields on tetrahedral mesh elements. This leads to a three-step algorithm consisting of (1) initialization by a loose convex relaxation to 9D spherical harmonic coefficients, (2) element-wise projection of the relaxed solution to a 3D rotation, and (3) a non-linear and non-convex refinement procedure.

Ray et al. [2015] propose some improvements to this three-step algorithm. Most importantly, the convex relaxation is tightened by modifying the boundary constraints, leading to seven linear conditions per boundary element compared to the one used in Huang et al. [2011]. The tighter set of constraints is important for geometric configurations where the surface normals do not naturally admit octahedral symmetry, e.g., near tetrahedral or spherical parts. Moreover, performance is significantly improved by a custom-tailored optimization algorithm.

A critical drawback of both previous algorithms is the requirement of a tetrahedral mesh, which not only requires effort to be generated but also strongly restricts the potential topologies of the field. Essentially, this leads to a chicken-and-egg problem. On the one hand, we need a tetrahedral mesh to compute the field, but on the other hand, to generate an appropriate tetrahedral mesh, carefully sampling the (unknown) singularity network of the field would be necessary. Another disadvantage of current approaches is that the magnitude of surface tangent field is completely unconstrained, which can lead to serious artifacts shown in Section 9; note that Li et al. [2012] presents a frame field construction without this issue but relies on local optimization and post-processing without a notion of global optimality.

2.4 Hexahedral Meshing

The most prominent important application of octahedral fields is in hex meshing. Similar to parametrization-based quad meshing

[Bommes et al. 2013b], the field can be used to construct an integer-grid map (IGM) [Bommes et al. 2013a], which maps integer isolines of a Cartesian grid to a conforming hexahedral mesh. The resulting hexahedral mesh can be robustly extracted, even if the map contains common local defects like degeneracies or orientation flips [Lyon et al. 2016].

Nieser et al. [2011] first pursued this direction by parameterizing manually designed octahedral fields. They observed that only a subset of octahedral field singularities are mappable. More specifically, line singularities can only be mapped if they correspond to 2D rotations and point singularities need to correspond to a triangulation of the sphere. Subsequent work proposes heuristics to obtain fields with valid line singularities based on mesh coarsening or refinement [Li et al. 2012; Jiang et al. 2014]. While important, these results are only a first step since mappability of an octahedral field additionally requires global consistency constraints that cannot be fulfilled by any of the known algorithms. One interesting observation is that local artifacts like non-mappable line singularities typically arise as an artifact of inappropriate sampling, i.e., a bad tetrahedral mesh.

Several strategies for initializing the non-linear optimization of octahedral fields have been applied to hexahedral meshing. Li et al. [2012] employ a surface cross-field to initialize volumetric elements in a nearest-neighbor manner. This requires the careful design of a surface field, since not every smooth surface field can be extended smoothly to the interior. A specialized initialization for CAD models with sharp features was proposed in Kowalski et al. [2014].

2.5 Boundary Element Methods

The boundary element method (BEM) [Pozrikidis 2002] is a technique for solving differential equations using calculations on the boundary of the domain. Most commonly, BEM is applied to interpolation of values from the boundary of a volume into its interior via the Dirichlet equation; variations of this technique are popular in scientific areas that use various interpolatory schemes. Using Green's theorem, BEM transforms differential problems in a domain into dense problems on the boundary. In the process, some complexity is pushed to preprocessing, yielding a smooth solution inside the domain.

In geometry processing and graphics, BEM has been used for barycentric coordinates for deformation [Lipman et al. 2008; Ben-Chen et al. 2009; Weber et al. 2012], simulation [Hahn and Wojtan 2015], and parametrization [Wang et al. 2013]. Beyond interpolation, BEM also can be incorporated into variational schemes, optimizing boundary conditions to achieve a desired effect in the interior; this approach was applied to deformation by Ben-Chen et al. [2009].

In this article, we use BEM discretizations of a *lower*-order than most works in the graphics literature. The accuracy of this discretization appears to be sufficient for our application, and the resulting per-triangle representation provides a natural setting for posing constraints involving tangents to the boundary of the volume.

3. PROBLEM OVERVIEW

The problem of representing and designing octahedral fields is relatively new. As such, we discuss its basic structure at some length to highlight the challenges that are unique to frame field computation in subsets of $\mathbb{R}^3$. Some of these basic principles were described in Huang et al. [2011] and Ray and Sokolov [2015], and we repeat them for clarity.

3.1 Field Representation

We work on a domain $\Omega \subset \mathbb{R}^3$, bounded by a set of smooth, closed surfaces $\partial\Omega$. Define the canonical axis set as

$$A := \{\pm e_1, \pm e_2, \pm e_3\},$$

where $e_i \in \mathbb{R}^3$ is the i-th standard basis vector. Then, an octahedral frame $A(p)$ at a point $p \in \Omega$ can be thought of as the set $R(p) \cdot A = \{R(p)x : x \in A\}$, where $R(p) \in \mathrm{SO}(3)$ is a rotation about the origin. More concretely, $R(p)$ transforms the three coordinate axes into the rotated frame. An *octahedral field* can be thought of as a (non-unique) choice of $R(p)$ for all points $p \in \Omega$. Referring to the taxonomy of Vaxman et al. [2016], an octahedral frame is a "three-dimensional 6-direction field," as its definition is agnostic to magnitude (although their taxonomy does not target frames in $\mathbb{R}^3$).

This representation is not unique: Multiple rotations can represent the same octahedral frame, as sign and order between the individual axes are irrelevant. For example, if $R_0 \in \mathrm{SO}(3)$ rotates 90° about any of the coordinate axes, then $R_0 \cdot A = A$, and hence R_0 represents the same frame as the identity matrix $I_{3\times3}$. We denote the group of all rotations that bring A to itself as the stabilizer group

$$O := \{R_0 \in \mathrm{SO}(3) : R_0 \cdot A = A\}.$$

In words, $O \subseteq \mathrm{SO}(3)$ contains all rotation matrices that bring the xyz axes to themselves, potentially with a change of order or sign. Formally, O is a subgroup of SO(3), isomorphic to the *octahedral group* of order 24, typically defined as the group of orientation-preserving symmetries of a cube. Thus, if we use rotations $R \in \mathrm{SO}(3)$ to parameterize frame fields in volumes, there are 24 different choices of R that all represent a given frame.

A natural idea might be to operate in the quotient space $\mathrm{SO}(3)/O$, identifying all elements of SO(3) that are the same up to rotation by an element of O. Cartesian approaches for 2D frame fields do exactly this: Every 2D frame can be thought of as a rotation of the xy axes by some 2×2 matrix $R \in \mathrm{SO}(2)$. The stabilizer group of the xy axes is isomorphic to $\frac{\mathbb{Z}}{4}$, generated by a 90° rotation about the origin. The corresponding quotient simply can be thought of as a set of angles repeating with a period of 90° rather than 360°; in the complex representation of 2D rotations, this quotient space can be identified with the complex plane through use of representatives $e^{4i\theta}$ [Ray et al. 2009; Knöppel et al. 2013].

Octahedral frames do not admit such a straightforward quotient representation, intuitively as a consequence of non-commutativity of rotations; more formally, SO(3) is a *simple group*, meaning it admits only trivial quotient groups. Nevertheless, one might attempt to build upon existing representations from 2D. For example, Diamanti et al. [2014] represents frames as roots of polynomials on $\mathbb{C}$. To avoid dealing with $\mathrm{SO}(3)/O$, one might attempt to identify SO(3) with the unit quaternions and represent frames as roots of quaternionic polynomials. Quaternionic polynomials $\mathbb{H}[x]$ form a non-commutative ring, however, with counterintuitive structure, allowing for polynomials with infinitely many roots [Hamilton 1866].

Huang et al. [2011] introduces a promising representation for octahedral frames. They design a specific canonical function $f_0 : S^2 \to \mathbb{R}$ on the unit sphere with critical points on the xyz axes (see Figure 1) and for ease of visualization, we take the suggestion of Ray and Sokolov [2015] to apply -1 to the original function proposed by Huang et al. [2011] so that the *peaks* of f_0 align with the xyz axes. Technicalities aside, one way of understanding f_0 is that it is the restriction of a polynomial over $\mathbb{R}^3$:

$$f_0(x, y, z) \propto x^4 + y^4 + z^4 + [\text{constant}].$$

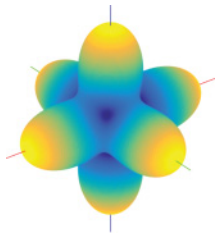


Fig. 1. Canonical function f_0 on the unit sphere S^2; radius from the origin and color both indicate function value in this plot.

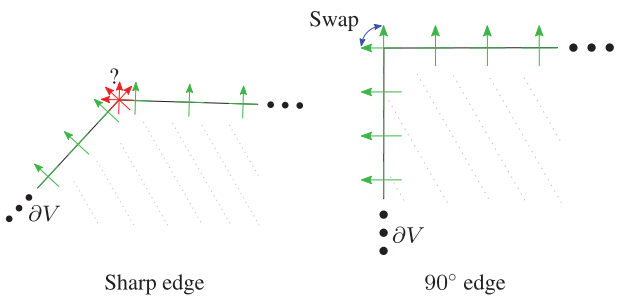


Fig. 2. Aligning the green frames with the surface normal leads to ambiguity at non-90° edges (left). This cannot be solved with orthogonal frame fields. At 90° edges, the constraint is satisfiable, but the direction that aligns with the normal must change.

This is the lowest-degree polynomial admitting non-trivial octahedral symmetry [Dunkl 1984].

Given a rotation matrix R, the function $f(x) = f_0(R^\top x)$ is a rotated version of the canonical function f_0 whose peaks align with the column vectors of R. For any matrix $R' \in O$, $f(x)$ has the property that $f(x) = f_0((RR')^\top x)$. That is, $f_0(R^\top x)$ *uniquely* represents the frame encoded by the columns of R, and it is invariant to the symmetries in O.

3.2 Boundary Constraints

The "classical" problem of octahedral field design is to assign a frame to every point inside a bounded volume $\Omega \subseteq \mathbb{R}^3$ that interpolates input frames on the boundary $\partial\Omega$. This problem is often relaxed to interpolating *partial* boundary information, most commonly allowing for any boundary frames that have one axis aligned to the surface normal.

Designing octahedral fields that align with surface normals is non-trivial for several reasons. The most straightforward issue is that surface normals might not be continuous at sharp corners. Then, the behavior of the octahedral field around a cusp is not obvious. This issue is particularly intricate since we wish *any* of the axes to align with the normal; so, normal discontinuities that exhibit rotations of 90° can lead to perfect alignments. See Figure 2 for examples.

The restriction of normal-aligned octahedral frames to the outer surface forms a tangent orthogonal frame field. Hence, topological and regularity issues like those discussed in Knöppel et al. [2013] are inherited, in addition to the issues above.

3.3 Singularity Graphs

Since frame fields are agnostic to sign and order of the individual directions, they are susceptible to inconsistencies in the transport of a frame around a closed loop. This is a direct result of the nature of the branched covering of 3D frames [Nieser et al. 2011].

Consider a closed 1D cycle $c(t) \in \Omega, 0 \leq t \leq 1, c(0) = c(1)$ within Ω, and let $F(t)$ encode the frame field restricted to this loop. We can take $F(t)$ to be a continuous function from t into the rotation matrices SO(3) even though this is a 24-cover of the space of octahedral frames, conceptually by differentially "matching" frames from $F(t)$ to $F(t + \mathrm{d}t)$.

A consequence of writing $F(t)$ as a continuous function from t into the rotation matrices is that $F(1) = F(0)R_0$, where R_0 encodes a symmetry of the canonical xyz frame. That is, $F(1)$ and $F(0)$ represent the same octahedral frame at $c(0) = c(1)$, but the xyz labeling of the individual directions might change to ensure continuity of $F(t)$. The cycle c is considered regular if $R_0 = I_{3\times3}$ and singular otherwise. We do not consider higher-order singularities, e.g., by enumerating the degree of singularities by the accumulated matching inconsistency alone. Such singularities are theoretically possible, but we did not empirically witness them in our examples. They are unlikely to appear for methods without explicit angle-based representation [Vaxman et al. 2016] interpolating fields on the surface, but more analysis is needed to establish this conjecture exactly.

It is beyond the scope of this article to give a full topological classification of singularities and indices for octahedral fields. In general, singularities in a 3D cross field come in two categories: singularity curves, around which there are singular cycles defined above, and singularity nodes, either connecting singularity curves to each other ("nodes") or connecting singularity curves to the boundary $\partial\Omega$ ("leaves"). The leaves also serve as singularities of the tangential frame field on the boundary surface, assuming the frames are aligned to the normal and that the normal field is smooth at the singularity. The set of singularity nodes and curves is the *singularity graph* of the frame field; see Figures 9 and 10 for examples of singularity graphs for frame fields from our method and others.

3.4 Smoothness

A secondary indicator of 3D frame field quality, beyond its singular topology, is its smoothness. We can think of a 3D frame field as a map $\tau : \Omega \to \mathrm{SO}(3)/O$ from the volume into the space of octahedral frames. Then, a classical measure of smoothness is the *Dirichlet energy*, defined as

$$E[\tau] := \int_\Omega \|d\tau\|_2^2 \,\mathrm{d}V,$$

where d denotes the differential of a map between manifolds and the norm $\|\cdot\|_2$ is induced by the metrics of $\mathbb{R}^3$ and SO(3). Minimizers of $E[\cdot]$ are known as harmonic maps.

If the field contains singular points, the Dirichlet energy diverges due to the rapid circulation about the singularity. Most topologies of ∂V imply that any smooth field aligning to the boundary normal *must* have a singular point, and hence computation of classical harmonic maps τ into $\mathrm{SO}(3)/O$ is meaningless. Discretely, this implies that the Dirichlet energy for unit frame fields, as an objective function for frame field optimization, grows to infinity in the limit of refinement; this makes the non-linear optimization steps of Huang et al. [2011] and Ray and Sokolov [2015] difficult to analyze. This problem was explored in Knöppel et al. [2013] for tangential vector fields, and the reasoning for 3D fields is similar. Hence, the straightforward resolution is similar as well: We compute smooth fields without a strict unit-norm constraint.

3.5 Field Sampling

Suppose that an octahedral field is computed on a tetrahedralization of Ω. Then, we have the field as samples on, e.g., the tet barycenters. This sampling removes much of the information in a smooth field, including the definition of a continuous curve c and the transportation upon it. Instead, the derivative $F'(t)$ is discretized on faces between two adjacent tets as the estimated rotation between the respective octahedral frames. Cycles are defined as cycles of tet centers around an interior edge, and the total rotation R_0 is the product of the rotations around the cycle. As a consequence, singular edges can only fall on tet edges, and singularity nodes must fall on tet vertices. The discrete singular network is smooth only if, by happenstance, the tet mesh vertices and edges are aligned with the field topology. Otherwise, this effect will not go away even with refinement. While Ray and Sokolov [2015] sample the field on vertices rather than tetrahedra, similar discrete reasoning applies. Other than adaptive remeshing and recomputation, the simplest solution is to compute the field as a smooth function, which is the approach we take.

4. SPHERICAL HARMONIC REPRESENTATION

Following the work of Huang et al. [2011] and Ray and Sokolov [2015], we represent octahedral frames as rotations of a canonical function $f_0(x)$ in the spherical harmonic basis. Other than serving as an efficient representation, this basis facilitates treatment of rotated frames by using Wigner D-matrices, explained below.

4.1 Canonical Function

From Section 3.1, we represent individual octahedral frames as rotations of the following function on the unit sphere:

$$f_0(x) := \sqrt{\frac{7}{12}}Y_{40}(x) + \sqrt{\frac{5}{12}}Y_{44}(x),$$

where $\{Y_{4m}(\cdot)\}_{m=-4}^{4}$ is the basis of degree-four real spherical harmonic functions and $x \in S^2$. This function, henceforth referred to as the *canonical axis function*, is peaked in the $\pm xyz$ directions; see Figure 1 for an illustration.

An advantage of the spherical harmonic representation is that rotations about the origin in this basis are well-understood theoretically. Suppose $R \in \mathbb{R}^{3\times 3}$ contains a set of orthonormal directions as its columns, so $R^\top R = I_{3\times 3}$; without loss of generality, we can assume $\det(R) = 1$. The spherical harmonic-based encoding of R will be the function $f(x) = f_0(R^\top x)$, illustrated in Figure 3. By the multi-way symmetry of f_0, changing the order of the columns in R does not affect $f(x)$ as a function of x.

4.2 Wigner D-Matrices

A consequence of representation theory for SO(3) is that $f(x) := f_0(R^\top x)$—like $f_0(x)$—is expressible within the $\{Y_{4m}\}_{m=-4}^{4}$ basis [Bröcker and Dieck 2003]. So, any rotation of f_0 can be encoded using coefficients in this basis.

Given a rotation R that transforms f_0 into f, the corresponding degree-four *Wigner-D matrix* of the rotation R transforms the coefficients of f_0 in the spherical harmonic basis into the coefficients of f [Kennedy and Sadeghi 2013]. Define $a_0 \in \mathbb{R}^9$ as the vector of coefficients of f_0 in the $\{Y_{4m}(\cdot)\}_{m=-4}^{4}$ basis:

$$a_0 := \left(0, 0, 0, 0, \sqrt{\frac{7}{12}}, 0, 0, 0, \sqrt{\frac{5}{12}}\right).$$

Then,

$$f(x) = f_0(R^\top x) = \sum_{m=-4}^{4} c_m Y_{4m}(x),$$

where the coefficients $c \in \mathbb{R}^9$ are given by

$$c = W_4(R)a_0,$$

and $W_4(R) \in \mathbb{R}^{9\times 9}$ is the degree-four Wigner-D matrix of R. The closed-form formulas for these matrices are somewhat involved. Nevertheless, Lehner et al. [2015] provides a library for computing them numerically in the complex spherical harmonic basis; a simple transformation converts to the real-valued version. Since rotating a function preserves its L_2 norm, $W_4(R)^\top W_4(R) = I_{9\times 9}$.

We note an intricacy of this representation of orthonormal frames. While $W_4(R)$ is a 9×9 orthogonal matrix, most 9×9 orthogonal matrices are *not* Wigner-D matrices. This is easy to see by counting degrees of freedom. Since $R \in \mathrm{SO}(3)$, R essentially has three degrees of freedom (e.g., axis in S^2 and rotation angle), while $\{W \in \mathbb{R}^{9\times 9} : W^\top W = I_{9\times 9}\}$ has 36 degrees of freedom.[1]

Define Γ to be the set of coefficients being rotations of f_0:

$$\Gamma := \{c \in \mathbb{R}^9 : c = W_4(R)a_0 \text{ for some } R \in \mathrm{SO}(3)\} \subseteq \mathbb{R}^9.$$

A corollary of the discussion in the previous paragraph is that $\Gamma \neq \{c \in \mathbb{R}^9 : \|c\|_2 = 1\}$. In fact, constraining c to be the coefficients of a rotation of f_0 is a highly non-linear, non-convex constraint, as we discuss in Section 5.2.

5. SMOOTH FIELD DESIGN

In this section, we describe the necessary ingredients for constructing smoothly-varying, boundary-aligned octahedral fields in volumes. We first describe our approach in the smooth setting and describe the actual algorithm with a discrete boundary in Section 7.

Suppose $\Omega \subseteq \mathbb{R}^3$ is a compact volume whose boundary $\partial\Omega$ is a smooth, connected surface. We wish to assign to every point in Ω an orthonormal frame that varies smoothly in the interior of Ω and is aligned with $\partial\Omega$ as a boundary condition. Our approach proceeds in two steps:

(1) We optimize for octahedral frames on the boundary $\partial\Omega$ whose interpolation to the interior of Ω via the boundary element method creates a smooth octahedral field.
(2) To evaluate the field at arbitrary points in the interior of Ω, we sample the field as a set of spherical harmonic coefficients and then project the interpolated coefficients onto Γ.

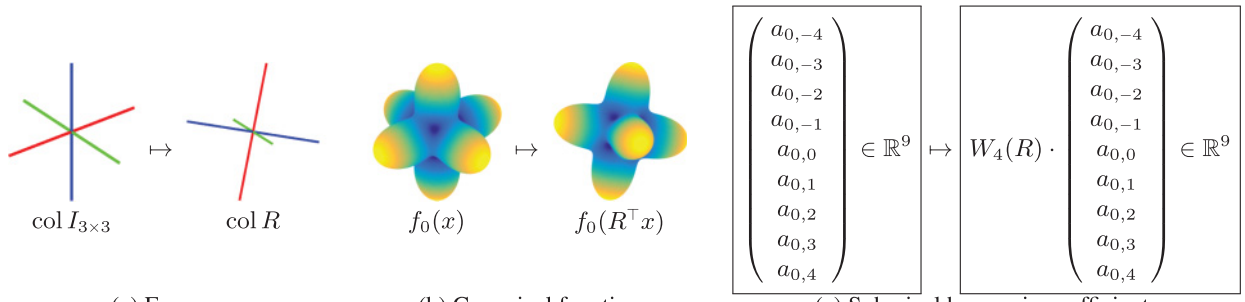
5.1 Global Completion

In the initial global step, we view our unknown as a function $u(x) := (u_{-4}(x), \ldots, u_4(x)) : \Omega \to \mathbb{R}^9$, assigning spherical harmonic coefficients to every point in Ω. These coefficients approximate rotations of the canonical function f_0. Below, we construct the global optimization for $u(x)$ term-by-term.

Smoothness in Y_{4m} basis. Recall that if $g(x) = \sum_{m=-4}^{4} c_m Y_{4m}$ and $h(x) = \sum_{m=-4}^{4} d_m Y_{4m}$, then

$$\|g - h\|_2^2 = \int_{S^2} |g(x) - h(x)|^2 \,\mathrm{d}x = \|c - d\|_2^2;$$

[1]Symmetry of $W^\top W = I_{9\times 9}$ induces 36 linearly dependent constraints.



(a) Frames (b) Canonical function (c) Spherical harmonic coefficients

Fig. 3. Representing rotated frames. Suppose (a) the columns of $R \in \mathbb{R}^{3\times3}$ represent a rotation of the xyz frame. To remove the xyz labels, we instead consider (b) rotating the canonical function $f_0(x) : S^2 \to \mathbb{R}$, which is peaked at the six axes without distinguishing them from each other. If (c) $a_0 \in \mathbb{R}^9$ contains the coefficients of $f_0(x)$ in the spherical harmonic basis, then the product $W_4(R) \cdot a_0 \in \mathbb{R}^9$ contains the coefficients of $f_0(R^\top x)$ in the same basis, where $W_4(\cdot) : \mathrm{SO}(3) \to \mathbb{R}^{9\times9}$ is the degree-four Wigner D-matrix function.

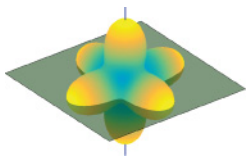


Fig. 4. Tangent frame.

that is, the distances in L^2 are equal to distances between spherical harmonic coefficients. This relationship allows us to measure the smoothness of $u(\cdot)$ using the classical Dirichlet energy (see Section 3.4) of u as a function $u : \Omega \to \mathbb{R}^9$ in the spherical harmonic basis:

$$E[u] := \int_\Omega \|\nabla u(x)\|_2^2 \, \mathrm{d}x. \tag{1}$$

Without boundary conditions, $E[\cdot]$ is minimized by $u \equiv$ const.

Relaxing the unit norm. As discussed in Section 3.4, for the Dirichlet energy to be finite and well-defined, we need to consider a relaxed space without the unit-norm constraint. Define the set of scalings of elements in Γ as

$$\overline{\Gamma} := \{\beta c \in \Gamma : \beta \in \mathbb{R}\} \subseteq \mathbb{R}^9.$$

Any element of $\overline{\Gamma}\backslash\{0\}$ unambiguously defines an orthonormal frame, since scaling does not affect the maximizers of f_0. Furthermore, this space corresponds to non-unit frame Dirichlet energies. Therefore, we opt to work in $\overline{\Gamma}$.

Unfortunately, as $\overline{\Gamma}$ is non-linear, the $u(x)$ resulting from this global step are allowed to leave the constraint space $\overline{\Gamma}$, and thus may not even encode true rotations of non-unit octahedral frames exactly. For this, we need to *project* them onto $\overline{\Gamma}$ again. Then, we can project them onto Γ by simple normalization. We address this in the local projection step (Section 5.2).

Normal alignment. Take $\hat{n}(\cdot) : \partial\Omega \to \mathbb{R}^3$ to be the outward-facing unit normal to $\partial\Omega$. For our frames to conform to the geometry of $\partial\Omega$, we wish for the boundary frames $u(x)$ to align with $\hat{n}(x)$ for all $x \in \partial\Omega$, as illustrated in Figure 4.

While the full space $\overline{\Gamma}$ is non-linear, Ray and Sokolov [2015] show that its restriction to those frames aligned with $\hat{n}(\cdot)$ forms a much simpler space. Define

$$v_c := \sqrt{\frac{5}{12}}(0, \ldots, 0, 1)$$
$$v_s := \sqrt{\frac{5}{12}}(1, 0, \ldots, 0)$$
$$v_n := \sqrt{\frac{7}{12}}(0, 0, 0, 0, 1, 0, 0, 0, 0).$$

The coefficients of any unit frame aligned with the $\pm z$ direction must take the form

$$u_{\pm z}(\theta) := v_n + v_c \cos\theta + v_s \sin\theta.$$

There is a single degree of freedom, expressed in θ, which results from rotation about the z-axis $W_4(R_z(\theta))a_0$. This formula is analogous to the complex representation of orthogonal tangent frames in Knöppel et al. [2013].

Applying a Wigner D-matrix $W_4(R_{\hat{n}(x)})$ to both sides parameterizes the frames aligned with $\hat{n}(x)$. Specifically, if $R_{\hat{n}}$ is an arbitrary rotation matrix taking $+z$ to $\hat{n}$, we write

$$u_{\hat{n}}(\theta) := W_4(R_{\hat{n}})[v_n + v_c \cos\theta + v_s \sin\theta].$$

Since this condition applies only to the boundary $\partial\Omega$, we can parameterize all boundary frames via two functions $c(\cdot) : \partial\Omega \to \mathbb{R}$ and $s(\cdot) : \partial\Omega \to \mathbb{R}$:

$$u(x) := W_4(R_{\hat{n}(x)})[v_n + v_c c(x) + v_s s(x)]. \tag{2}$$

This *linear* condition couples $u(\cdot)$, $c(\cdot)$, and $s(\cdot)$, since $W_4(R_{\hat{n}(x)})$ is a constant derived from $\hat{n}(x)$ alone. Note that the set of matrices that transform $\pm z$ into $\hat{n}(x)$ is non-empty; it carries a single degree of freedom, which simply changes the relative way to measure θ; this said, the matrix $R_{\hat{n}(x)}$ can be chosen arbitrarily. This corresponds with the local bases for tangential fields constructed in Knöppel et al. [2013]

The functions $c(x)$, $s(x)$ represent the tangential restriction of the octahedral field as a 4-direction field ("4-RoSy field"). Thus, they are not independent functions, and the true degree of freedom is a single angle θ. This is captured by a unit-norm constraint $c(x)^2 + s(x)^2 \equiv 1$, which Ray and Sokolov [2015] attempt to enforce in their non-linear optimization while their linear initializer ignores it altogether. This constraint, however, is topologically *unsatisfiable* for smooth fields on the surface, as explained in Section 3.4.

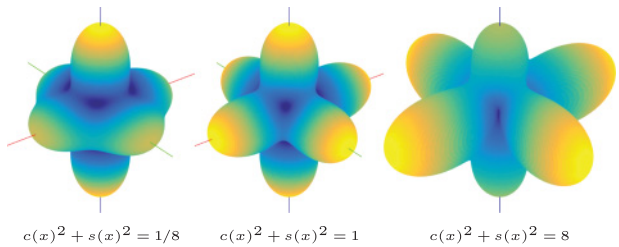


Fig. 5. Relaxing the constraint $c(x)^2 + s(x)^2 = 1$ still leads to frames that are well-aligned to the coordinate axes, although the scale in the xy plane differs from the scale in the z plane.

We propose a relaxation of the $c(x)^2 + s(x)^2 \equiv 1$ constraint that is still topologically satisfiable by smooth octahedral fields. Namely, we enforce that the *average* of $c(x)^2 + s(x)^2$ equals one over the outer surface:

$$\int_{\partial\Omega} [c(x)^2 + s(x)^2]\,\mathrm{d}x = A, \tag{3}$$

where A is the area of $\partial\Omega$. This global constraint is reminiscent of eigenvalue problems in differential geometry; see Mullen et al. [2008], Ben-Chen et al. [2010], and Knöppel et al. [2013] for examples. It forces the boundary frames to have non-trivial tangent plane directionality away from singular points. Even if $c(x)^2 + s(x)^2 \neq 1$ at some $x \in \partial\Omega$, the frame at x still has three clear orthogonal directions, as illustrated in Figure 5.

Full problem. Combining the terms above provides an optimization problem for u:

$$\boxed{\begin{aligned} \min_{u,c,s}\ & E[u] \\ \text{s.t. }\ & u(x) = W_4(R_{\hat{n}(x)})[v_n + v_c c(x) + v_s s(x)] \\ & \quad \forall x \in \partial\Omega \\ & \textstyle\int_{\partial\Omega}[c(x)^2 + s(x)^2]\,\mathrm{d}x = A. \end{aligned}} \tag{4}$$

As promised, this is a relaxation of the orthogonal frames problem in that we do not explicitly enforce $u(x) \in \overline{\Gamma}$. Counting degrees of freedom results in nine degrees of freedom per $x \in \Omega\backslash\partial\Omega$, representing the components of u; the constraint on the boundary implies only two degrees of freedom per $x \in \partial\Omega$.

Remark. Note that Equation (3) is *not* the only possible non-triviality constraint. We chose it after experimenting with several other possibilities. Constraining $\int_\Omega \|u(x)\|_2^2\,\mathrm{d}x = |\Omega|$ as an integral over the volume Ω can lead to degeneracies in which $u(x) \equiv 0$ on $\partial\Omega$. This happens when a large amount of rotation is required to align to the boundary. Another option is to let the coefficient of $v_n \in \mathbb{R}^9$ scale on the boundary with v_c and v_s. This is similar to our formulation, except letting the normal term vanish creates degeneracies for the case illustrated in Figure 2, in which the entire frame vanishes.

5.2 Local Projection Step

The interpolated $u(x)$ generally is not in the constraint set $\overline{\Gamma}$ when $x \notin \partial\Omega$, due to our relaxation in the global step. After solving Equation (4), however, we can project $u(x)$ pointwise onto the constraints. We use the gradient descent procedure in Ray and Sokolov [2015, Algorithm 5] for this purpose.

6. BOUNDARY OPTIMIZATION

The octahedral field problem involves a flat, Euclidean volumetric domain. While problems on surfaces must deal with curvature and parallel transport, these constructions are trivial for our problem. Hence, we view explicit tetrahedralization of the interior of Ω as a means for computation rather than a meaningful element of the problem. In this section, we show that indeed our problem can be solved *completely* on the boundary $\partial\Omega$, alleviating the need for any volumetric meshing to compute smooth frame fields.

If we fix $u \in \partial\Omega$, then the problem of computing u in the interior resembles a Laplace equation. Inspired by theBEM [Pozrikidis 2002], we replace the global optimization step with one that can be carried out on the outer surface.

The only part of Equation (4) that involves knowing u in the interior of Ω is $E[u]$. If we know $u|_{\partial\Omega}$, however, then expanding the form of $E[u]$ to recover the remainder provides a familiar Dirichlet problem:

$$\begin{aligned} \min_{u:\Omega\to\mathbb{R}^9}\ & \textstyle\int_\Omega \|\nabla u(x)\|_2^2\,\mathrm{d}x \\ \text{s.t. }\ & u|_{\partial\Omega} \text{ fixed.} \end{aligned} \tag{5}$$

Remaining terms from Equation (4) drop out when $u|_{\partial\Omega}$ is fixed. This is the weak form of the Laplace equation $\Delta u = 0$.

Suppose u satisfies $\Delta u = 0$ in the interior of Ω. Applying Green's first identity gives a way to compute the Dirichlet energy of u only given boundary information:

$$E[u] := \int_\Omega \|\nabla u\|_2^2 = \int_{\partial\Omega} u \frac{\partial u}{\partial \hat{n}}, \tag{6}$$

where $\hat{n}$ is the unit normal to $\partial\Omega$ and $\frac{\partial u}{\partial \hat{n}}$ indicates the directional derivative of u in the $\hat{n}$ direction. The key observation from this equation is that the optimization objective $E[u]$ in Equation (1) can be written in terms of only integrals and constraints on the *boundary* $\partial\Omega$, if we can recover $\frac{\partial u}{\partial \hat{n}}$ directly from $u|_{\partial\Omega}$.

To this end, for a fixed $x \in \partial\Omega$, a well-known formula from basic partial differential equations (PDEs) determines $u(x)$ given only $u|_{\partial\Omega}$ [Evans 2010]:

$$\frac{1}{2}u(x) = \int_{\partial\Omega} \left[G(y-x)\frac{\partial u(y)}{\partial \hat{n}} - u(y)\frac{\partial G(y-x)}{\partial \hat{n}} \right] \mathrm{d}y, \tag{7}$$

where G is the Green's function $G(x) := \frac{1}{(4\pi\|x\|_2)}$. If we know $u(x)$ on $\partial\Omega$, then this integral, when evaluated for all $x \in \partial\Omega$, can be viewed as a linear constraint determining the relationship between u and $\frac{\partial u}{\partial \hat{n}}$. Adding this integral equation to the optimization problem as a constraint, and using Equation (6) to evaluate $E[\cdot]$ alleviates the need to compute u in the interior of Ω:

$$\begin{aligned} \min_{u,\frac{\partial u}{\partial \hat{n}},c,s}\ & E[u, \tfrac{\partial u}{\partial \hat{n}}] \qquad \textit{(using Equation (6))} \\ \text{s.t. }\ & u(x) = W_4(R_{\hat{n}(x)})[v_n + v_c c(x) + v_s s(x)] \\ & \quad \forall x \in \partial\Omega \\ & \textstyle\int_{\partial\Omega}[c(x)^2 + s(x)^2]\,\mathrm{d}x = A \\ & \text{Equation (7) holds } \forall x \in \partial\Omega. \end{aligned} \tag{8}$$

7. ALGORITHM

We next describe the implementation of our technique for practical geometric models. After constructing the relevant vectors and matrices to describe our octahedral field problem, we parallel the local and global steps from Section 5 and show how they can be implemented using standard linear algebra and optimization machinery.

7.1 Discretization

We discretize the boundary $\partial\Omega \subseteq \mathbb{R}^3$ as a triangle mesh with n faces. Our goal is to find one vector of u values per triangle via a piecewise constant discretization of Equation (8).

Contrasting with some existing work in boundary elements for geometry processing [Lipman et al. 2008; Ben-Chen et al. 2009], we use piecewise-constant elements rather than piecewise-linear (per-vertex) elements because the triangles have well-defined tangent planes. Denote the set of coefficient variables as $u_k \in \mathbb{R}^9$, $k \in \{1, \ldots, n\}$. We use $\bar{u}_m \in \mathbb{R}^n$ to denote the value of the m-th coefficient of u along the entire mesh, where $m \in \{-4, \ldots, 4\}$, and $u \in \mathbb{R}^{9n}$ to denote the vector containing all the unknowns; these reshapings will simplify our notation later.

To discretize the objective $E[\cdot]$, we use BEM) to fill in boundary derivatives $\frac{\partial u}{\partial \hat{n}}$ from the per-triangle boundary values of u [Pozrikidis 2002, §5.1.4]. In particular, we use triangle midpoint-collocated BEM to obtain a matrix $B \in \mathbb{R}^{n\times n}$ such that $B\overline{u}_m \approx (\frac{\partial \overline{u}}{\partial \hat{n}})_m$; we employ closed-form calculations for integrals over triangle elements as outlined in Graglia [1993].

If $\bar{T} \in \mathbb{R}_+^{n\times n}$ is the diagonal matrix of triangle areas and $\bar{L} := \bar{T}B$, then the Dirichlet energy can be discretized from Equation (6) as

$$E[\{\bar{u}_m\}_{m=-4}^4] := \sum\nolimits_{m=-4}^{4} \bar{u}_m^\top \bar{L}\bar{u}_m = u^\top L u,$$

where $L \in \mathbb{R}^{9n\times 9n}$ contains blocks of $\bar{L}$'s. We assume L is symmetric, which can be enforced explicitly by taking $L \leftarrow \frac{1}{2}(L + L^\top)$ in case BEM-based approximation of L is slightly asymmetric. In practice, we find that L is positive semidefinite as would be expected from this Laplacian-style matrix; we leave a proof of this property or exploration of conditions when it holds as a topic for future research.

We additionally introduce variables $c, s \in \mathbb{R}^n$ for the boundary data in Equation (2). Denote by $u_0 \in \mathbb{R}^{9n}$ the vector with blocks $W_4(R_{\hat{n}_k})v_n$, where $\hat{n}_k$ is the normal of triangle k. Furthermore, denote by $H_c, H_s \in \mathbb{R}^{9n\times n}$ the matrices taking c, s to vectors with blocks $W_4(R_{\hat{n}_k})v_c c_k$ and $W_4(R_{\hat{n}_k})v_s s_k$, respectively. Then, Equation (2) becomes

$$u = u_0 + H_c c + H_s s.$$

Finally, we discretize the non-triviality of Constraint (3) as

$$c^\top \bar{T} c + s^\top \bar{T} s = A.$$

This leads to the following discretization of Equation (4):

$$\begin{aligned} \min_{\{u,c,s\}} \quad & u^\top L u \\ \text{s.t.} \quad & u = u_0 + H_c c + H_s s \\ & c^\top \bar{T} c + s^\top \bar{T} s = A. \end{aligned} \tag{9}$$

with $9n + 2n = 11n$ variables and $9n + 1$ constraints, resulting in exactly $2n - 1$ degrees of freedom.

7.2 Global Step: Optimization

Define

$$Q := \begin{pmatrix} H_c^\top L H_c & H_c^\top L H_s \\ H_s^\top L H_c & H_s^\top L H_s \end{pmatrix} \in \mathbb{R}^{2n\times 2n}$$

$$v := \begin{pmatrix} H_c^\top L u_0 \\ H_s^\top L u_0 \end{pmatrix} \in \mathbb{R}^{2n}$$

$$x := \begin{pmatrix} c \\ s \end{pmatrix} \in \mathbb{R}^{2n}$$

$$T := \operatorname{diag}(\bar{T}, \bar{T}) \in \mathbb{R}^{2n\times 2n}$$

Then, Equation (9) is equivalent to

$$\begin{aligned} \min_{x\in\mathbb{R}^{2n}} \quad & x^\top Q x + 2x^\top v \\ \text{s.t.} \quad & x^\top T x = A. \end{aligned} \tag{10}$$

When $v \neq 0$, this is *not* an eigenvalue problem. Nonetheless, in Appendix A, we derive a method for efficiently recovering the global optimum, inspired by the LSQI algorithm in Golub and Van Loan [2012]. Our final algorithm takes place in two steps:

(1) Computation of the smallest (algebraic) eigenvalue $\lambda_{\min}$ for the generalized problem $Qx = \lambda T x$. This can be carried out *without* applying Q^{-1}, e.g., using Matlab's `eigs` routine.
(2) Finding a unique root of $g(\lambda) := x(\lambda)^\top T x(\lambda) - A$ in the interval $(-\infty, \lambda_{\min})$, where $x(\lambda)$ solves the linear system $(Q - \lambda T)x = -v$. We use the bisection algorithm [Press et al. 2007] since $g(\lambda)$ is monotonic in this interval.

The globally optimal x is $x(\lambda_0)$, where $g(\lambda_0) = 0$.

The eigenvalue iteration is fast since it only requires multiplication by Q and T. Solution time is dominated by the second step; each iteration of bisection requires inversion of a dense linear system. The density is on the order of the number of boundary triangles rather than the number of tetrahedra in a volumetric mesh, however, implying a smaller size for Q than tet-based FEM.

7.3 Local Step: Interior Evaluation

The optimization above in Section 7.2 is a "precomputation" in the sense that once we have obtained per-triangle estimates of u and $\frac{\partial u}{\partial \hat{n}}$, we can evaluate $u(x)$ at *any* x in the interior of the mesh via a smooth formula. This sampling can be carried out in parallel for multiple x's, since the computation decouples given the boundary data.

In particular, in the smooth case, when $x \in \Omega\backslash\partial\Omega$, the formula of Equation (7) changes slightly to evaluate $u(x)$:

$$u(x) = \int_{\partial\Omega} \left[G(y-x)\frac{\partial u(y)}{\partial \hat{n}} - u(y)\frac{\partial G(y-x)}{\partial \hat{n}} \right] \mathrm{d}y. \tag{11}$$

The somewhat counterintuitive loss of a 1/2 factor is explained, e.g., in Pozrikidis [2002].

For x in the interior of a triangle mesh and triangle $k \in \{1, \ldots, n\}$, define

$$\sigma_k(x) := \int_{T_k} G(y-x)\,\mathrm{d}y$$

$$\rho_k(x) := \hat{n}_k \cdot \int_{T_k} \nabla G(y-x)\,\mathrm{d}y.$$

Graglia [1993] provides algorithms for computing these quantities in closed form. Then, we evaluate $u(x)$ as

$$u(x) = \sum_k [u_k \sigma_k(x) + [Bu]_k \rho_k(x)]. \tag{12}$$

This function is smooth in the interior of the mesh, regardless of its coarseness. After computing $u(x)$, the frame at $x \in \Omega$ is recovered using the recovery procedure outlined in Section 5.2.

Sharpening. We note one heuristic improvement in our pipeline. Before sampling u in the interior $\Omega\backslash\partial\Omega$ (Section 7.3), we replace $u(x)$ for $x \in \partial\Omega$ with $\frac{u(x)}{\|u(x)\|_2}$. While allowing $u(x)$ to take non-unit values makes sense during the global optimization stage, during interpolation, the larger values of $u(x)$ make the choices of frames in the interior near boundary singularities more clear.

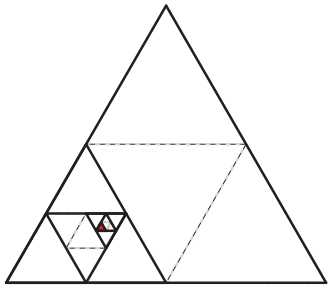


Fig. 6. We find singular points by iteratively "tightening" singular loops. In each step, a singular triangle is subdivided into four subtriangles, and the non-singular subtriangles are removed. This process is repeated recursively until the loop is sufficiently small.

7.4 Comparison to Existing Techniques

Having established the technical details of our algorithm, we briefly point out the main differences between our proposed method and previously published algorithms:

—Compared to both Huang et al. [2011] and Ray and Sokolov [2015], we use BEM rather than a tetrahedral mesh of the interior of Ω. This makes our fields and singularity curves smooth even when the boundary is discrete.
—Unlike both Huang et al. [2011] and Ray and Sokolov [2015], our global optimization step does not require a local, non-linear/non-convex refinement procedure. This allows for $u(x)$ to be sampled at multiple interior x's in a completely decoupled fashion.
—We use the improved boundary conditions of the linear step in Ray and Sokolov [2015] over Huang et al. [2011].
—Rather than enforcing the smoothly unsatisfiable pointwise $c(x)^2 + s(x)^2 \equiv 1$ constraint proposed in the non-linear step of Ray and Sokolov [2015] or dropping non-triviality on the boundary altogether like Huang et al. [2011], we use the relaxed Condition (3). This makes our optimization problem non-linear, but we provide an algorithm in Appendix A to solve the relaxed problem to global optimality.

8. SINGULARITY COMPUTATION

One challenge working with a smooth octahedral field rather than one sampled on a tetrahedral mesh is that there is no obvious way to extract its singular structure in closed form. A naïve way to approximate the singular graph would be to sample frames on a tetrahedral mesh inscribed in Ω and then use a discrete algorithm like that in Huang et al. [2011], but this forces the edges of the singular graph to align with mesh edges and requires dense sampling. Instead, to better visualize the singularity graphs of our smooth octahedral fields, we propose a random sampling technique that can find points on the singular graph without tetrahedral meshing. The end result is a cloud of points sampled from the singular graph, each of which is found by iteratively tightening a loop around a singularity as illustrated in Figure 6.

In more detail, we randomly sample a point $p \in \Omega$ with uniform probability and a random unit vector $\hat{n} \in S^2$. We construct an equilateral triangle in the plane orthogonal to $\hat{n}$ through p, rotated by a random angle about $\hat{n}$. The radius of the triangle is chosen to be at least $10\times$ the average distance between p and its closest neighbor in the collection of sampled p's; in practice, we find little dependence on the constant 10 so long as it is not too small. We sample our octahedral frames at the vertices and edge midpoints of the triangle (six points total), subdividing the triangle into four subtriangles. We test each subtriangle to see if it is a discrete singular loop, that is, if composing frame matchings from vertex to vertex yields the identity. Non-singular loops are discarded, and the remaining singular loops are again subdivided into four smaller subtriangles. This process continues until singular loops are below a fixed radius threshold; the barycenter of the refined triangle is taken as an approximation of a singular point.

The right columns in Figures 9 and 10 illustrate examples of our samplings. The refinement procedure is independent for each p, allowing for many loops to be subdivided in parallel. While we could draw these point clouds as graphs by, e.g., connecting these points to their nearest neighbors, we choose not to do so since mathematically, it is difficult to verify that the resulting graph is truly the singularity graph of the smooth field.

There is some under-sampling in the singular fields near the boundaries of the models, especially the ones in Figure 10. This is not reflective non-smoothness for our octahedral fields but rather the limited likelihood of sampling a p and surrounding triangle close to the boundary. Additional sampling near the boundary and/or biasing the triangle normal $\hat{n}$ to align with the surface normal potentially could resolve this issue if a dense sampling of the singular point cloud is desired.

9. EXPERIMENTS

In this section, we provide experiments demonstrating the behavior of our pipeline for computing 3D frame fields in volumes. We illustrate frames using a variety of means to provide intuition for their tangential and volumetric structure.

9.1 Boundary Conditions

Beyond our use of BEM to achieve *smooth* octahedral fields inside volumes, our formulation differs from previous work before discretization by the introduction of the non-triviality Constraint (3), which forces our fields to have non-vanishing tangential behavior on the boundary of the domain.

To isolate the effect of our boundary conditions from improvements due to use of BEM and other design decisions, we compare the behavior of our boundary conditions versus those from the linear step of Ray and Sokolov [2015] after implementing the latter using a BEM discretization. This is achieved simply by solving $Qx = -v$, to obtain the minimizer of Equation (10) without the non-triviality constraint.

Figure 7 compares these options by showing the restriction of our computed fields to the outer triangle mesh surfaces of different volumes V. We first compare values of the squared norm $c(x)^2 + s(x)^2$ as a function of $x \in \partial V$; our technique forces this value to average to one, while the weaker boundary conditions can decrease the tangential norm in favor of a smoother field. We also compare the field itself restricted to the boundary to demonstrate how this change has an effect on the singular structure of the resulting field.

9.2 Interior Frames

Figure 8 illustrates octahedral fields computed using our technique. For fair comparison, we show BEM analog of Ray and Sokolov [2015] discussed in the previous section. The images show the directionality and orientation of the frames using flows in one direction rendered as boxes aligned to the orthogonal component.

The smoothness of the flow lines illustrates the smoothness and lack of undersampling afforded by our procedure over tetrahedral

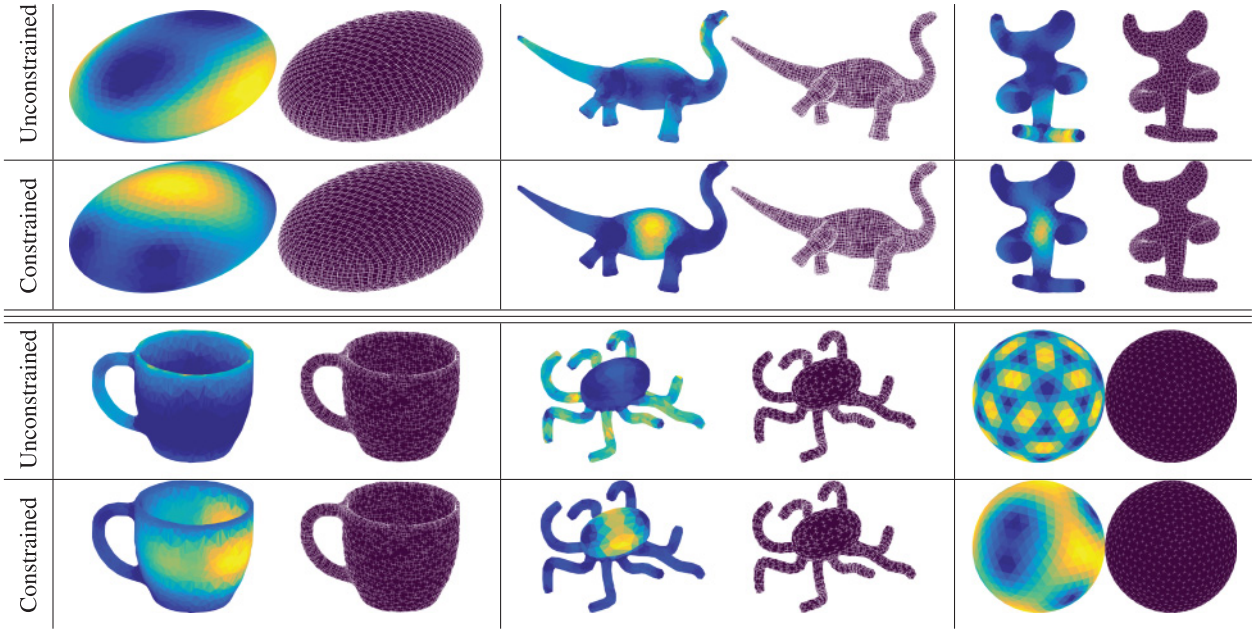


Fig. 7. Validation of boundary conditions. The first of each pair shows the norm $c(x)^2 + s(x)^2$ as a function along the boundary (scaled from blue to yellow), and the second in each pair shows the corresponding tangent frame field. Here, "unconstrained" refers to the boundary conditions of Ray and Sokolov [2015] adapted to BEM discretization, while "constrained" indicates addition of the non-triviality Constraint (3).

methods. Compared to the alternative boundary conditions, our method also yields better singular structure, e.g., near non-90° angles (boxed examples in first row) with lots of symmetry or non-orthogonal structure. This confirms our intuition that the boundary non-triviality Constraint (3) reduces catastrophic cancellation of frames when normal alignment forces $u(x)$ to change rapidly or repeatedly.

9.3 Comparison to Previous Work

Figures 9 and 10 illustrate a benchmark designed to evaluate our method in comparison to previous work [Huang et al. 2011; Ray and Sokolov 2015]. We also show intermediate output of Ray and Sokolov [2015] before non-linear refinement, since the linear stage of their pipeline is more closely related to our method (which does not require refinement). The authors of these two papers kindly agreed to participate in this benchmark and used their own implementations/hardware for these tests; this ensures fair representation, although we are unable to provide timing comparisons.

We compare on a set of tet meshes highlighting challenges in the octahedral frames pipeline. Each shape appears multiple times with different tet resolution filling the interior; note that the boundaries of different tet meshes are not identical even if they fill the same shape. Since previous methods operate on tet meshes, for comparison we extract the triangle mesh boundary from each test tet mesh, simplify it using Sacht et al. [2015] to no more than 6,000 triangular faces (a reasonable resolution for our BEM implementation), and run our method. This implies that our result can change between tests on the same shape; this resolution dependence is a byproduct of our testing procedure and *not* a drawback of our technique.

The output of previous techniques differs slightly between the two papers and with our own. Huang et al. [2011] generate one frame per tet, Ray and Sokolov [2015] generate one frame per vertex, and our method generates a smoothly-varying frame function. For this reason, the figures show the singular network corresponding to the discrete or smooth frame fields. See Section 8 for discussion of how we generate point clouds sampling from singular graphs in our frame fields.

There are several conclusions to draw from this experiment illustrating the advantages and drawbacks of our method. First, our singular point clouds are smooth even when the tetrahedral meshes are coarse; this contrasts with previous FEM-based methods, for which singular topology must be associated with mesh elements. Additionally, our method achieves simple topology even on fairly low resolution meshes. This topology agrees with results from denser meshings of the same surface without the need for non-linear refinement, a critical step in previous work (see, e.g., the sphere example for Ray and Sokolov [2015] without non-linear refinement for a failure of their linear initializer). Even when the initializer for Ray and Sokolov [2015] succeeds in generating a smooth field, the topology of this field can be more complicated, as illustrated in the torus example and highlighted in Figure 11.

This experiment shows some drawbacks as well. For instance, the sphere with a dense boundary shows spurious singularities near junctures of the singular graph. These represent failures of the frame projection procedure due to the relaxed constraints in the interior of the domain, which can be corrected using the non-linear refinement in Huang et al. [2011] and Ray and Sokolov [2015]. Additionally, the singular point clouds for the "real-world" meshes in Figure 10 are only sparsely sampled near the boundary, reflecting the fact that we can only sample points from the singular graph rather than constructing it explicitly; these samples are harder to obtain near the boundary surface because they must be contained within a sampled loop; this is of course a drawback of the simplistic singular graph reconstruction rather than the field itself.

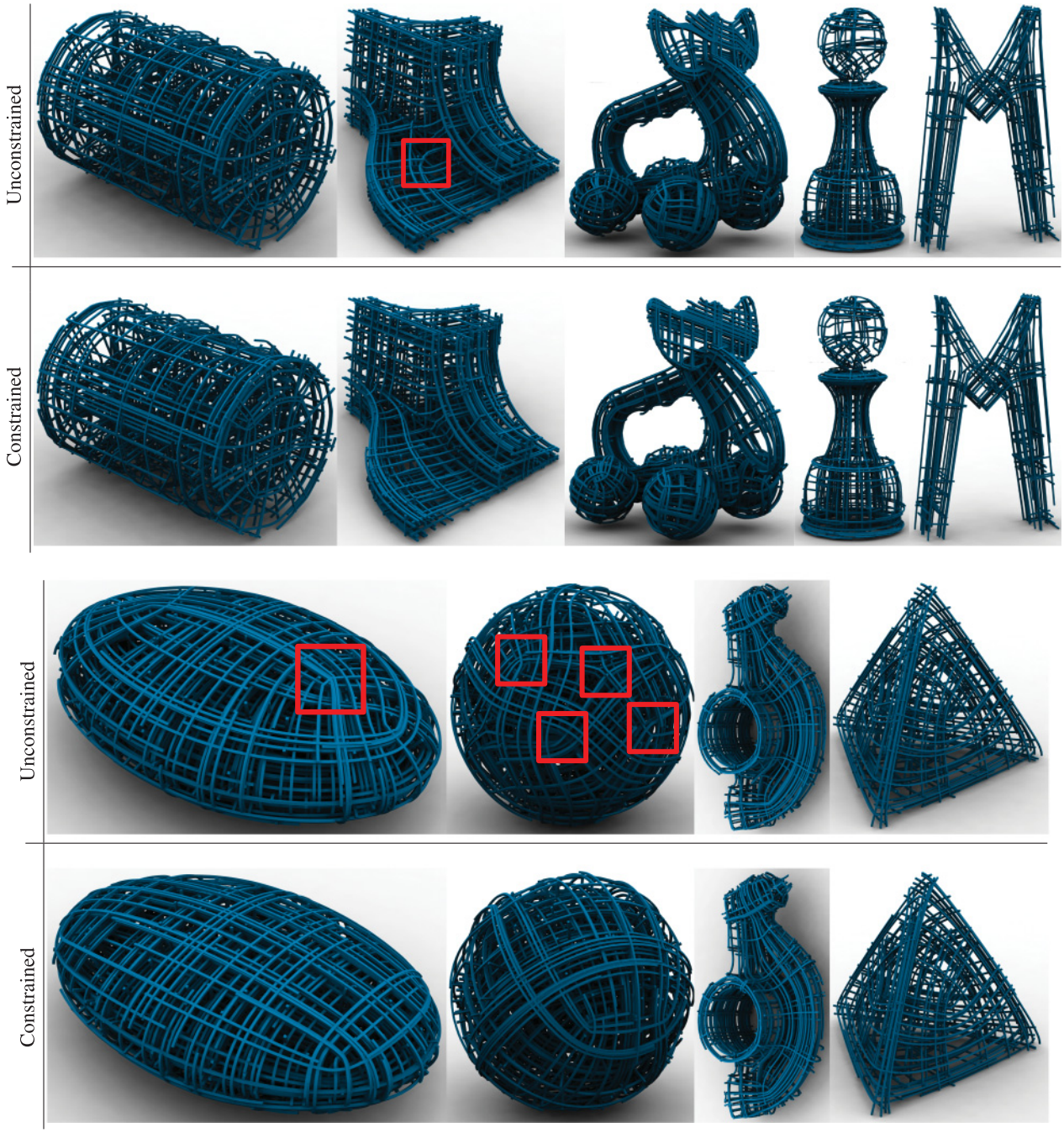


Fig. 8. Frame fields in the interior of V visualized using flows; red boxes highlight notable singularities and features. As in Figure 7, we compare our frame fields to those with the boundary constraints from Ray and Sokolov [2015] adapted using BEM for smoothness. Our fields have fewer singularities on the surface and interpolate smoothly to the interior with cube-like structure.

9.4 Boundary Density

Figure 12 tests our method under coarsening of the boundary. For this experiment, we consider the boundary of a tet mesh with 10,778 triangles. We then produce different coarsenings of the boundary using Sacht et al. [2015]. We choose this method because it produces "nested" meshes, each of which enclose the original tet mesh; hence, each can be used to sample the same interior. As illustrated in this experiment, our method is stable under coarsening of the boundary, a property derived from its expression in the well-understood boundary element basis.

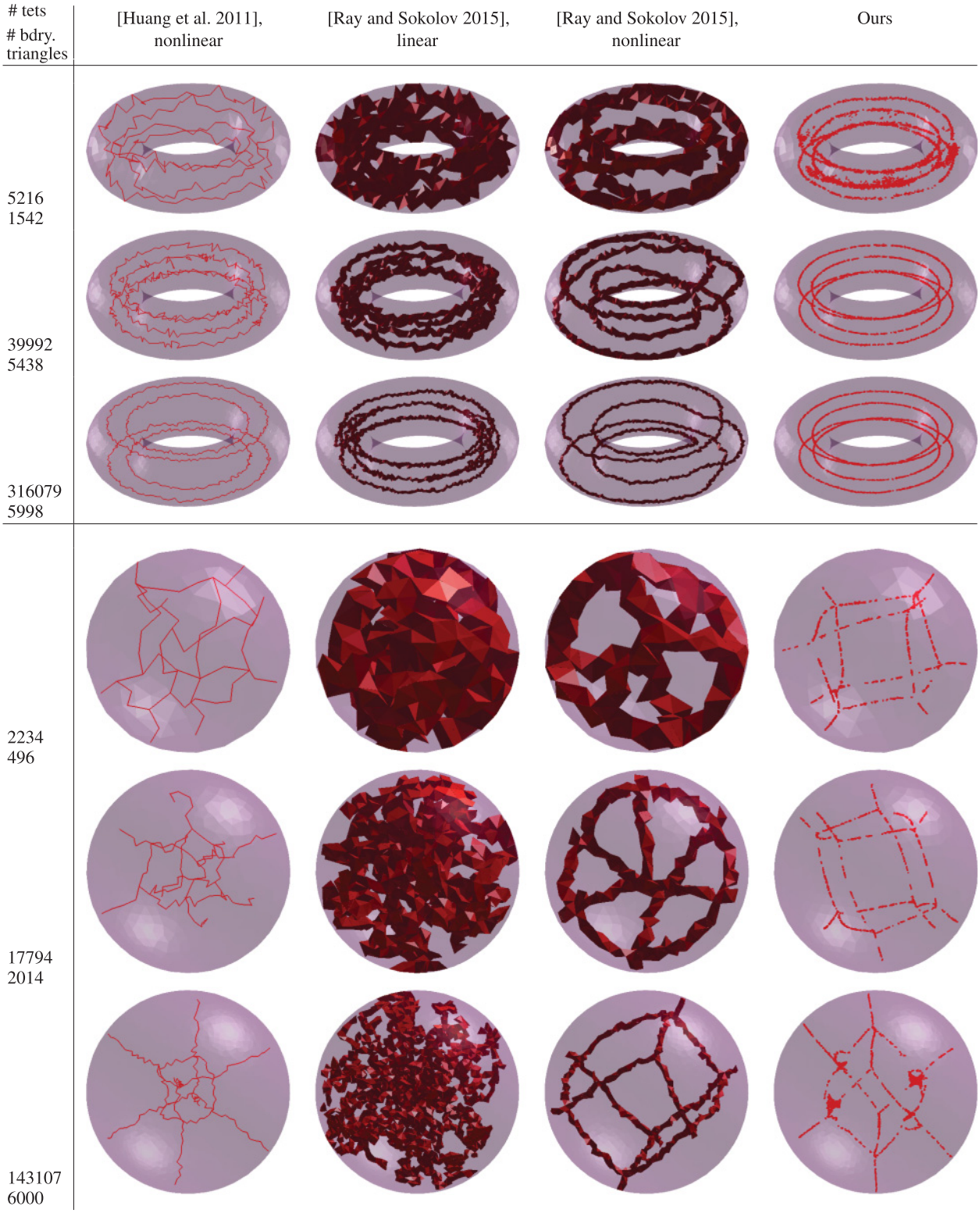


Fig. 9. Benchmark comparison to Huang et al. [2011] (final non-linearly-refined result) and Ray and Sokolov [2015] before and after non-linear refinement. Even with coarse boundaries, our singular graphs are smooth, whereas the singular graphs of these methods must lie on the tetrahedral mesh; our method also does not require local non-linear refinement. See §9.3 for a detailed discussion. Additional results are shown in Figure 10.

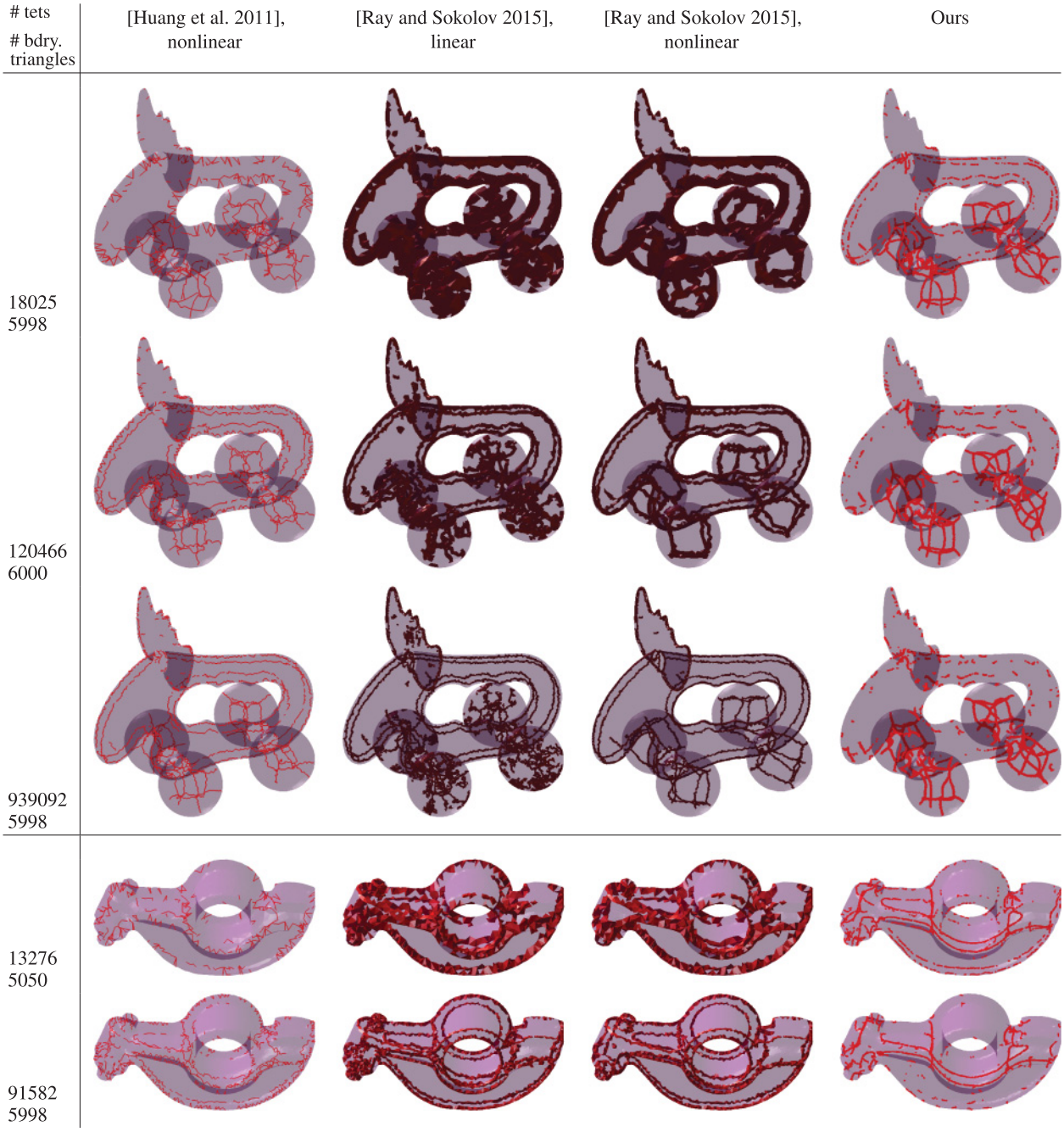


Fig. 10. Continuation of Figure 9.

This experiment suggests that we can squeeze efficiency out of our method by coarsening the boundary before sampling the interior frames; this reduces the number of terms in the sum of Equation (12). As an extreme example, Figure 13 shows streamlines of the frames from the coarsest example in Figure 12. This downsampled computation still produces smooth frames conforming to the outer geometry.

9.5 Hexahedral Meshing

Our octahedral frames have a practical impact on hex meshing pipelines. We experiment using a variant of Nieser et al. [2011], which relies upon octahedral frames for guidance. Figure 14 shows the results of hex remeshing on tet meshes guided by samples of our frames. Even without dense sampling or non-linear polishing, our

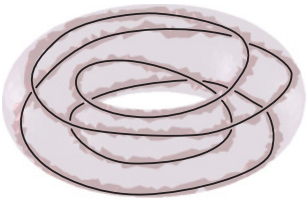


Fig. 11. Traced singularity graph for Ray and Sokolov [2015] on the torus with 316,079 tetrahedra after non-linear refinement; line breaks denote layering rather than a discontinuity in the singularity curve. Notice that the entire graph is a single connected component.

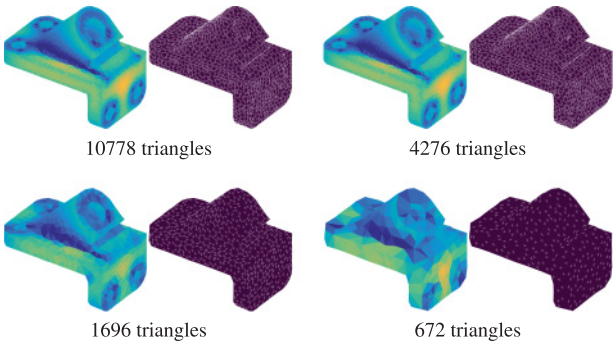


Fig. 12. Sensitivity to density of the boundary mesh; boundary fields visualized using the same method as Figure 7. Qualitatively, our result remains stable even after aggressive coarsening of the boundary.

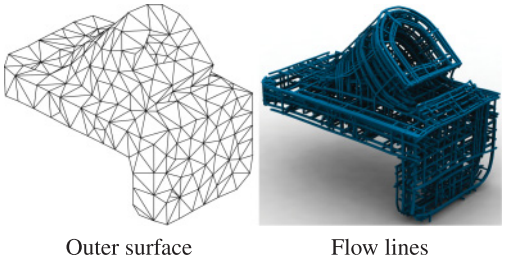


Fig. 13. Interior of the frame field from Figure 12 computed using only 672 triangles. The frames still vary smoothly and follow the contours of the boundary.

singular structure and smoothly-varying frames create a symmetric hex mesh. The hexahedral meshes are robustly extracted from the parametrization with the freely available HexEx library [Lyon et al. 2016].

9.6 Efficiency

Figure 15 contains timings for the tests in Figures 9 and 10 using our technique. We implement our method in Matlab with C++/OpenMP plugins for parallelizable tasks. Our computer has 5.3GB of RAM with 24 double-threaded 2.4GHz Intel CPUs.

For each test, timings are reported for separate stages of the pipeline. The global steps—construction of BEM matrices and optimization for boundary frames—are carried out on a single thread; times are reported in seconds. The local steps are easy to parallelize by sample point; since the field can be queried anywhere, we report

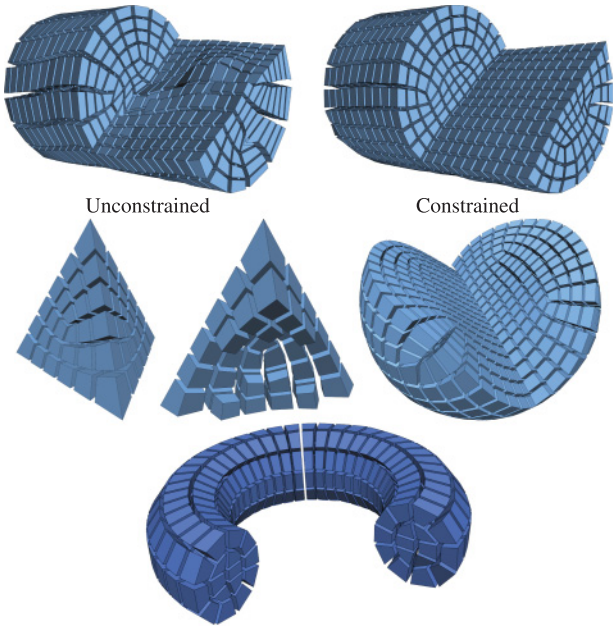


Fig. 14. (top) Comparison of hex meshes generated from octahedral frame alternatives sampled in a tet mesh of a cylinder; we show a cut through the two hex meshes to illustrate interior frames; (bottom) additional hex meshes (with cut-aways) generated from our frames.

Mesh	Triangles	Global		Local	
		BEM	*Optimization*	*Sampling*	*Projection*
Torus	1542	0.79	39.31	8499.9	4679.2
	5438	8.53	648.89	3565.2	11609.3
	5998	27.28	1185.27	2545.1	9449.4
Sphere	496	0.04	2.89	22654.2	19174.0
	2014	0.84	91.76	7636.4	31510.6
	6000	16.00	1467.86	2728.4	26769.2
Elk	5998	16.04	1114.02	2674.0	27360.0
	6000	18.65	1004.20	1747.0	31211.6
	5998	12.31	1231.54	2528.4	34926.5
Rocker arm	5050	9.43	698.84	3132.8	36548.9
	5998	10.81	785.87	3250.2	47338.2

Fig. 15. Timings for tests in Figures 9 and 10. The global steps of BEM construction and numerical optimization are single-threaded; times are reported in seconds. The local step has a parallel implementation; timings are in units of frames per second.

timings in terms of the average number of frames sampled/projected per second.

Note that in our benchmark, many triangle meshes for testing had 6,000 triangles. A drawback of our current implementation is that the $6,000 \times 6,000$ dense BEM matrices are stored in memory, which appeared to create many "cache misses" and related memory efficiency issues, as evidenced by the *much* faster computation times for the smaller torus/sphere meshes. Future work could consider application of the fast multipole method [Rokhlin 1985] to alleviate the need to store such dense matrices. Also note that the 6,000 limit was fairly arbitrary, and in fact, our method works similarly well with coarser boundary mesh approximations as evidenced by the experiments in Section 9.4.

10. DISCUSSION AND CONCLUSION

This work represents significant progress on several aspects of the octahedral field problem. We demonstrate a pipeline in which octahedral frames can be recovered with infinite resolution even after discretization of the boundary. Our pipeline, quadratic optimization algorithm, and robust projection also overcome several challenges related to optimization in the space of frames.

Some drawbacks of our approach and current implementation indicate avenues for future research. Since we do not use a meshing of the interior of the volume, posing constraints on alignment to interior features becomes more challenging. Note that the BEM does not require volumes to have connected boundaries, so this drawback likely could be addressed by augmenting the outer boundary surface with reversely-oriented interior elements. Additionally, our current implementation's efficiency is limited by use of dense matrices expressing interactions between every pair of boundary elements.

More broadly, octahedral frame field computation—and hex meshing—are by no means solved problems, and they will provide theoretical and practical challenges for years to come. A key question is whether it is possible to optimize directly in the space of octahedral frames without a relax-and-project pipeline; such an approach likely will make it more challenging to apply BEM tuned to classical PDEs. More immediately, it may be possible to generalize our pipeline to computation of other classes of symmetry fields in volumes and to apply fast multipole methods to further accelerate BEM [Liu 2009]. The specification of alignment constraints inside the volume, as for instance of interest for geological data [Ray and Sokolov 2015], seems to be a straightforward extension of the optimization problem that we plan to investigate in the future. Another challenge is the ability to detect and discover the singularity graph without the need for a sampling method (responsible for the artifacts in Figure 10). Finally, a discrete challenge might be to construct a hex mesh *directly* from a triangulated boundary and our octahedral field, eliminating the drawbacks of tet discretization altogether.

APPENDIX

A. QUADRATIC OPTIMIZATION

The Lagrangian optimality condition for Problem (10) is

$$(Q - \lambda T)x = -v \tag{13}$$

for dual variable $\lambda \in \mathbb{R}$. This system of equations shows that we can think of x as a function $x(\lambda)$ when the left hand side is non-singular.

Define $g(\lambda) := x(\lambda)^\top T x(\lambda) - A$. Since λ enforces $x(\lambda)^\top T x(\lambda) = A$, we seek the smallest root λ of the secular equation $g(\lambda) = 0$. Our remaining task is to bracket a root of $g(\lambda)$, and then it can be found by bisection.

Suppose the matrix of Equation (13) is not invertible. Then, there exists a non-trivial x that the left hand side takes to zero. Equivalently,

$$Qx = \lambda T x. \tag{14}$$

Computing λ from Q and T is a generalized eigenvalue problem, solvable using standard numerical machinery. Take $\lambda_{\min}$ to be the minimum such eigenvalue. Assuming the right hand side of Equation (13) is not orthogonal to the corresponding generalized eigenvector, we must have $\|x\| \to \infty$ as $\lambda \to \lambda_{\min}$ from the left. By equivalence of norms on $\mathbb{R}^{2n}$, this shows $\lim_{\lambda \to \lambda_{\min}^-} g(\lambda) = \infty$.

As $\lambda \to -\infty$, then $x \to 0$ assuming T is non-singular, since Equation (13) approaches the system $Tx = 0$. That is, $\lim_{\lambda \to -\infty} g(\lambda) = -A$ and $g(\lambda) < 0$ for sufficiently small λ.

We have shown that $g(\lambda)$ is continuous for $\lambda \in (-\infty, \lambda_{\min})$ and that this interval contains a root. Basic calculations show $g'(\lambda) > 0$ on this interval, so there is exactly one root.

ACKNOWLEDGMENTS

The authors thank N. Ray and D. Sokolov for correspondence while early versions of the manuscript were prepared, as well as all authors of Huang et al. [2011] and Ray and Sokolov [2015] for kindly running the benchmark tests in Figures 9 and 10. W. Leeb helped formalize early versions of this work. J. Solomon acknowledges support of an MIT Skoltech Seed Fund grant, titled "Boundary Element Methods for Shape Analysis."

REFERENCES

Mirela Ben-Chen, Adrian Butscher, Justin Solomon, and Leonidas Guibas. 2010. On discrete Killing vector fields and patterns on surfaces. In *CGF*, Vol. 29. 1701–1711.

Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. 2009. Variational harmonic maps for space deformation. *TOG* 28, 3 (July 2009), 34:1–34:11.

David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013a. Integer-grid maps for reliable quad meshing. *TOG* 32, 4 (July 2013), 98:1–98:12.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013b. Quad-mesh generation and processing: A survey. *CGF* 32, 6 (2013), 51–76.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), 77:1–77:10.

T. Bröcker and T. Dieck. 2003. *Representations of Compact Lie Groups*.

Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing N-polyvector fields with complex polynomials. *Proc. SGP* 33, 5 (2014), 1–11.

Charles Dunkl. 1984. Orthogonal polynomials on the sphere with octahedral symmetry. *Trans. AMS* 282, 2 (1984), 555–575.

L. C. Evans. 2010. *Partial Differential Equations*. AMS.

G. H. Golub and C. F. Van Loan. 2012. *Matrix Computations*. Johns Hopkins University Press.

R. D. Graglia. 1993. On the numerical integration of the linear shape functions times the 3-D green's function or its gradient on a plane triangle. *Trans. Antennas and Propagation* 41, 10 (Oct. 1993), 1448–1455.

David Hahn and Chris Wojtan. 2015. High-resolution brittle fracture simulation with boundary elements. *TOG* 34, 4 (2015), 151:1–151:12.

W. Hamilton. 1866. *Elements of Quaternions*. Longmans, Green, & Co.

Jin Huang, Yiying Tong, Hongyu Wei, and Hujun Bao. 2011. Boundary aligned smooth 3D cross-frame field. *TOG* 30, 6 (Dec. 2011), 143:1–143:8.

Tengfei Jiang, Jin Huang, Yuanzhen Wang, Yiying Tong, and Hujun Bao. 2014. Frame field singularity correction for automatic hexahedralization. *TVCG* 20, 8 (2014), 1189–1199.

R. A. Kennedy and P. Sadeghi. 2013. *Hilbert Space Methods in Signal Processing*. Cambridge University Press.

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2013. Globally optimal direction fields. *TOG* 32, 4 (July 2013), 59:1–59:10.

N. Kowalski, F. Ledoux, and P. Frey. 2014. Block-structured hexahedral meshes for CAD models using 3D frame fields. *Procedia Eng.* 82 (2014), 59–71.

Jeremy Lehner, Joscha Nehrkorn, Matthew Krzyaniak, Andrew Ho, and Stefan Stoll. 2015. EasySpin 5.0.15. Retrieved from http://easyspin.org/.

Wan Chiu Li, Bruno Vallet, Nicolas Ray, and Bruno Lévy. 2006. Representing higher-order singularities in vector fields on piecewise linear surfaces.

IEEE Transactions on Visualization and Computer Graphics 12, 5 (2006), 1315–1322.

Yufei Li, Yang Liu, Weiwei Xu, Wenping Wang, and Baining Guo. 2012. All-hex meshing using singularity-restricted field. *TOG* 31, 6 (Nov. 2012), 177:1–177:11.

Yaron Lipman, David Levin, and Daniel Cohen-Or. 2008. Green coordinates. *TOG* 27, 3 (Aug. 2008), 78:1–78:10.

Y. Liu. 2009. *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*. Cambridge University Press.

Max Lyon, David Bommes, and Leif Kobbelt. 2016. HexEx: Robust hexahedral mesh extraction. *ACM Trans. Graph.* 35, 4, Article 123 (July 2016), 11 pages. DOI:http://dx.doi.org/10.1145/2897824.2925976

Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. 2008. Spectral conformal parameterization. In *Proc. SGP*. 1487–1494.

Matthias Nieser, Ulrich Reitebuch, and Konrad Polthier. 2011. CubeCover: Parameterization of 3D volumes. In *CGF*, Vol. 30. 1397–1406.

C. Pozrikidis. 2002. *A Practical Guide to Boundary Element Methods with the Software Library BEMLIB*. CRC.

W. H. Press, Saul Teukolsky, William Vetterling, and Brian Flannery. 2007. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press.

Nicolas Ray and Dmitry Sokolov. 2015. On smooth 3D frame field design. *CoRR* abs/1507.03351 (2015).

Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Lévy. 2009. Geometry-aware direction field processing. *TOG* 29, 1 (2009).

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

Vladimir Rokhlin. 1985. Rapid solution of integral equations of classical potential theory. *J. Comput. Phys.* 60, 2 (1985), 187–207.

Leonardo Sacht, Etienne Vouga, and Alec Jacobson. 2015. Nested cages. *TOG* 34, 6 (Oct. 2015), 170:1–170:14.

Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional field synthesis, design, and processing. *Computer Graphics Forum* 35, 2 (2016), 545–572.

He Wang, Kirill A. Sidorov, Peter Sandilands, and Taku Komura. 2013. Harmonic parameterization by electrostatics. *TOG* 32, 5 (2013), 155:1–155:12.

Ofir Weber, Roi Poranne, and Craig Gotsman. 2012. Biharmonic coordinates. *CGF* 31, 8 (Dec. 2012), 2409–2422.

Received August 2016; revised December 2016; accepted February 2016