

Boundary Aligned Smooth 3D Cross-Frame Field

Jin Huang[†] Yiyong Tong[‡] Hongyu Wei[†] Hujun Bao[†]
[†] State Key Lab of CAD&CG, Zhejiang University [‡] Michigan State University

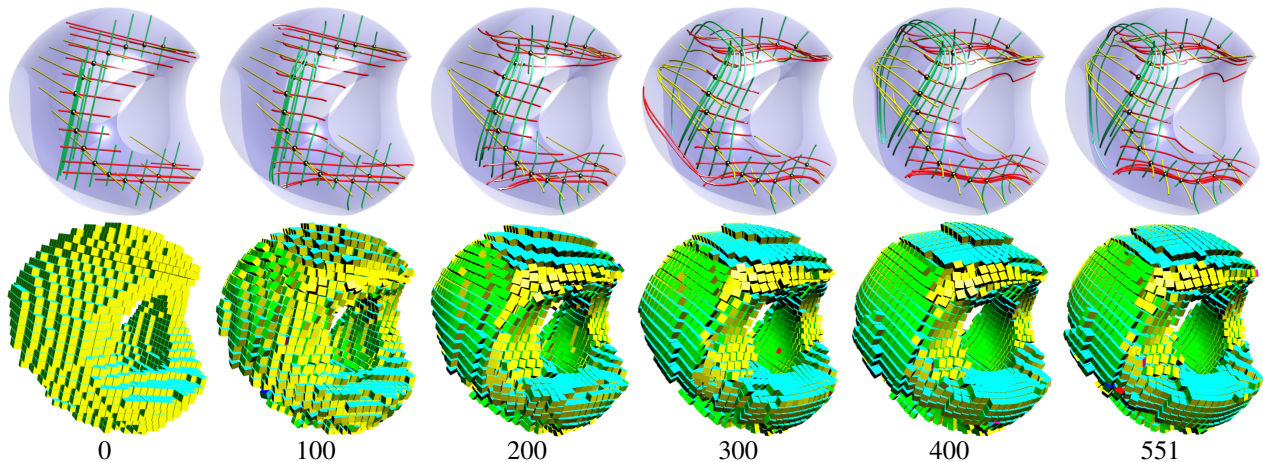


Figure 1: Snapshots of the optimization procedure to construct a boundary aligned 3D cross-frame field. The top row shows the internal streamlines. The next row contains another visualization with cubes spread by a parameterization along the current cross-frame field and rotated by the current local frame $R(\Phi)$. The corresponding number of iteration is shown at the bottom.

Abstract

In this paper, we present a method for constructing a 3D *cross-frame field*, a 3D extension of the 2D cross-frame field as applied to surfaces in applications such as quadrangulation and texture synthesis. In contrast to the surface cross-frame field (equivalent to a 4-Way Rotational-Symmetry vector field), symmetry for 3D cross-frame fields cannot be formulated by simple one-parameter 2D rotations in the tangent planes. To address this critical issue, we represent the 3D frames by spherical harmonics, in a manner invariant to combinations of rotations around any axis by multiples of $\pi/2$. With such a representation, we can formulate an efficient smoothness measure of the cross-frame field. Through minimization of this measure under certain boundary conditions, we can construct a smooth 3D cross-frame field that is aligned with the surface normal at the boundary. We visualize the resulting cross-frame field through restrictions to the boundary surface, streamline tracing in the volume, and singularities. We also demonstrate the application of the 3D cross-frame field to producing hexahedron-dominant meshes for given volumes, and discuss its potential in high-quality hexahedralization, much as its 2D counterpart has shown in quadrangulation.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: hexahedral, spherical harmonics, N-RoSy frame field

Links: [DL](#) [PDF](#)

1 Introduction

Many recent quadrangulation methods start by constructing a smooth field of orientations determined up to a rotation of π or $\pi/2$. Substantial progress has been made towards the generation of quadrilateral meshes with controlled element sizes and edge directions by optimizing such fields. However, many applications require discretization of 3D volumes rather than just their boundary surfaces. Applications such as simulated elasticity of 3D volumetric objects, computational electromagnetics, and computational fluid dynamics require Finite Element, Finite Volume, or Finite Difference methods on a discretized domain. These methods benefit from a high-quality hexahedral mesh, since hexahedral meshes offer several numerical advantages over tetrahedral meshes due to their tensor product nature. They are also desirable for applications such as geometric design and B-spline fitting, and amenable to applications such as 3D texture atlases. In addition, hexahedral meshes often capture the symmetries of 3D objects and domains better than tetrahedral meshes, thus making the model more intuitive to designers or animators. However, the automatic generation of a hexahedral mesh for a given curved 2D boundary with feature alignment, sizing, and regularity control remains far more challenging than automatic tetrahedralization.

In this paper, we focus on frame field construction, a critical step towards automatic generation of hexahedral meshes that meet the aforementioned requirements. We propose constructing a 3D local *frame field* akin to the cross field in curved 2D surfaces. We use a spherical harmonics-based optimization to obtain a smooth frame field, which can be seen as three vector fields that are continuous up to a simultaneous 3D rotation of all three fields composed of $\pi/2$ rotations around the orthonormal axes. We show as an example that a 3D parameterization guided by the frame field can be constructed and used to generate the final boundary- and feature-aligned mesh with well-shaped hexahedral elements.

1.1 Related Work

To generate high-quality feature-aligned and boundary-sensitive hexahedral meshes for domains with irregular boundaries, either explicit or implicit smooth and boundary-aligned orientation fields for the edges of the cube-like cells are necessary. For 2D meshes, it has been shown that it is often convenient to obtain a smooth field of orientations determined up to a rotation of π or $\pi/2$. For example, Quadrangulation methods such as [Bommes et al. 2009; Zhang et al. 2010b] usually have the quadrilateral edges aligned to such fields. Recently, this concept has been generalized to N-Way Rotational-Symmetry (N-RoSy) fields [Palacios and Zhang 2007; Ray et al. 2008; Lai et al. 2010], with those used in quadrangulation classified as 4-RoSy vector fields. For hexahedral meshing, we extend this notion to 3D by allowing irregular edges so as to obtain better element shapes while conforming to an irregular boundary. However, the extension of such symmetry to 3D is nontrivial. Although a few works have touched on the topic of 3D vector field or frame field (represented as a tensor field) generation [Takayama et al. 2008; Zhang et al. 2010a; Vyas and Shimada 2009; Yamakawa and Shimada 2003], none of them specifically address a symmetry property. 3D rotational-symmetry frame fields remain hitherto an unexplored area. In this paper we constructed a description of local frames using the spherical harmonic basis. Spherical Harmonics (SH) have been used for 3D-rotation estimation in [Makadia and Daniilidis 2003] and for optimal rotation alignment of 3D models in [Kazhdan 2007], which we adapt for the purpose of 3D rotational-symmetry.

In curved surface quadrangulation, the quad edges are often extracted as isocurves of a piecewise or periodic parameterization given a cross field [Ray et al. 2006; Kälberer et al. 2007; Bommes et al. 2009; Zhang et al. 2010b]. Our 3D parameterization can be seen as an extension of [Bommes et al. 2009] and [Zhang et al. 2010b].

Theoretically, given a quadrilateral mesh with an even number of quadrilaterals representing the boundary of a domain homeomorphic to a sphere, it is known that a topological hexahedral mesh exists [Eppstein 1996]. It is further proved by [Carbonera and Shepherd 2010] that the quadrilateral mesh can be turned into the boundary of a hexahedral mesh with all edges being line segments, although the faces are not guaranteed to be planar. Given recent advances in quadrilateral meshing (e.g., [Bommes et al. 2009; Zhang et al. 2010b]), it is conceivable that we can turn any triangle mesh that bounds a simply connected domain into a hexahedral mesh by first constructing a quadrangulation of the boundary. However, to get high quality hexahedral meshes with feature alignment, well-shaped hexahedra, and sizing control, experienced engineering may prefer user-guided or semi-automatic decomposition of the 3D shape into simple sub-volumes, which can then be automatically decomposed into hexahedral meshes [Roca 2009]. Simple sub-volumes generated by divide-and-conquer can be decomposed by algorithms such as mapping [Cook and Oakes 1982], submapping [Ruiz-Gironés and Sarrate 2010], and sweeping [Scott et al. 2005], which work only on specific topological and geometric types of volumes. Multi-block approaches rely on user-input (e.g. [Pointwise 2009]) to create a very coarse hexahedral mesh, with each hexahedron subdivided into a structured hex mesh subsequently. Fully automatic hexahedral meshing methods, such as grid-based methods, decomposition-based approaches, and outside-in algorithms, generally cannot achieve high mesh quality and boundary sensitivity with reasonable speed (cf. surveys such as [Tautges 2001; Shepherd 2007]). Our hexahedron-dominant meshing uses the Morse-Smale Complex to construct the surface quadrangulation, then follows the framework presented in [Meshkat and Talmor 2000].

2 Cross-Frame Field in SH Representation

As in recent approaches to surface quadrangulation, we explicitly construct a smooth frame field for orientation of the edges in the target mesh. N-RoSy surface vector fields and cross fields have been successfully used as the frame fields on surfaces. It is a natural idea to extend the cross frame representation used in the method by [Bommes et al. 2009] to 3D. However, the integer problem involved in the 3D case is much more complicated than in the 2D case. For the extension of a 4-Symmetry field, there are 24 possible transformations in the symmetry group. It also requires at least two vector fields (orthogonal to each other) to determine the directions of the three axes instead of the single vector field in 2D. Unlike rotations in 2D, 3D rotations form a *noncommutative* 3 dimensional group ($SO(3)$), rendering the optimization procedure much more difficult than rounding floating point numbers in 2D. Thus, instead of the explicit representation, we propose a spherical harmonics-based representation, which handles the desired symmetry implicitly.

2.1 Cross-Frame in the Volume

Analogous to the 2D cross-frames (4-RoSy vectors) used for quadrangulation of surfaces, a 3D cross-frame field should possess the following properties:

1. Cubic symmetry: the frame produced by rotating around any of its axes by multiples of $\pi/2$ is treated as an equivalent frame.
2. Smoothness: The frame fields have small spatial gradients, which implies that the resulting edge orientation would be smooth.
3. Normal alignment: at the boundary, one of the frame’s axes is parallel to the normal of the boundary, which forces the cells of the hexahedral mesh to align properly with the given boundary surfaces.

Cubic symmetry is the key to frame field optimization. A 3×3 rotation matrix $R = [x, y, z]$ applied to the standard global reference frame is a straightforward choice for the representation of a local reference frame, where $x, y, z \in \mathbb{R}^3$ are the axes of its local frame. We define a 3D cross-frame to be the equivalence class of frames which can be obtained from R through rotations in the rotation symmetry group of a cube,

$$F(R, r_1, r_2, r_3) = RR_X^{r_1 \pi/2} R_Y^{r_2 \pi/2} R_Z^{r_3 \pi/2} \quad (1)$$

where R_*^r denotes the matrix for the rotation around Euclidean axis $*$ by r (i.e., a permutation matrix with possible changes to the signs when r is a multiple of $\pi/2$). We denote all of the rotation matrices associated with R by this symmetry group as a set $[R] = \{F(R, r_1, r_2, r_3) | r_1, r_2, r_3 \in \{0, 1, 2, 3\}\}$, and define a cross-frame by such a set. Therefore, $[R] = R[I]$, where I denotes the identity matrix and $[I]$ the rotation matrices associated with I by this symmetry group. Thus, each cross-frame contains the same number of frames as $[I]$. For each $R \in [I]$, we can choose x from positive and negative directions along axes X, Y , or Z , and choose y from any four of those orthogonal to x . As z is completely determined by $z = x \times y$, we have $6 \times 4 = 24$ possibilities in each cross-frame.

2.2 Descriptor Function

To avoid mixed integer programming for the optimization in the above representation, we need an equivalent descriptor for all the elements in $[R]$. Such a descriptor leads to a proper difference measurement between two cross-frames for optimization of the

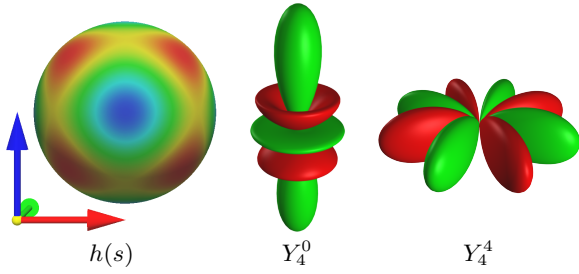


Figure 2: The visualization of the spherical function defined in Equation 2 and the spherical basis functions involved in its SH representation.

smoothness. For efficiency and the simplicity in handling boundary conditions as we discuss in Section 2.4, the descriptor that we designed for $[R]$ is a function $f_{[R]}(s)$ defined on the unit sphere $S^2 = \{s \in \mathbb{R}^3 \mid \|s\| = 1\}$:

$$f_{[R]}(s) = h(R^T s), \quad (2)$$

where $h(t) = t_x^2 t_y^2 + t_y^2 t_z^2 + t_z^2 t_x^2$. For any $M \in [I]$, $h(M^T t) = h(t)$, because M only changes the orders and signs of t_x , t_y , and t_z . Given any $R_1, R_2 \in [R]$, we have $M = R_2^{-1} R_1 \in [I]$. Thus $h(R_1^T s) = h(M^T R_2^T s) = h(R_2^T s)$, and the above definition does not depend on the choice of R . Through examining the first and second derivatives, it can be observed that the maxima are located in directions towards the eight corners of the axis aligned cube centered at the origin, and the minima are located in directions towards the centers of the faces of the same cube. As shown in Figure 2, it resembles a cube with a smooth chamfer angle.

The difference between two cross-frames $[R_a]$ and $[R_b]$ can be formulated naturally as an integration over the sphere,

$$\int_{S^2} (f_{[R_a]}(s) - f_{[R_b]}(s))^2 ds. \quad (3)$$

The difference is 0 if and only if $[R_a] = [R_b]$, since the maxima of the two functions will not be aligned if $[R_a] \neq [R_b]$, resulting in a non-zero integral.

2.3 A Primer on Spherical Harmonics

The above function $f_{[R]}(s)$ can be converted to a spherical harmonic representation for efficiency. Before we discuss the actual representation, we briefly go over the relevant basics of spherical harmonics. For a more detailed exposition, see [Green 2003].

In the classical Fourier transformation of periodic 1D functions (or equivalently functions defined on the unit circle), one represents a smooth function by a linear combination of basis functions that are eigenfunctions of the Laplace operator Δ (i.e., the second derivative), i.e., solutions to $\Delta f + \lambda f = 0$. If we parameterize the unit circle using angle θ , the basis functions are simply $\sin(l\theta)$'s and $\cos(l\theta)$'s, where l is a non-negative integer, and l^2 is the eigenvalue λ associated to $\sin(l\theta)$ and $\cos(l\theta)$.

Extending to the unit sphere S^2 , we can construct an orthogonal basis for spherical functions using eigenfunctions of the Laplace-Beltrami operator Δ_{S^2} (which is, again, a mixture of second derivatives). It can be shown that the eigenvalues are $l(l+1)$, where l is a non-negative integer. The eigenspace associated with band l is spanned by the mutually orthogonal eigenfunctions Y_l^m (called the spherical harmonics), where $m = -l, \dots, 0, \dots, l$. In the following,

we use the real number version of the spherical harmonics Y_l^m . See [Green 2003] for the actual expression of Y_l^m .

We may regard l as the ‘‘frequency’’ of the basis function, and regard the representation of a function f by its spherical harmonics coefficient \hat{f} as decomposing the signal into components of different frequencies. Since Δ_{S^2} is rotation-invariant, we will not change the frequencies by rotating any spherical function. Thus, a spherical function rotated by R has its spherical harmonics coefficients mapped to another set of spherical harmonics coefficients by a linear transformation denoted by \hat{R} , which is block diagonal, as band l coefficients are mapped to band l coefficients.

In practice, \hat{R} for an arbitrary rotation R is often performed through its ZYZ Euler angles representation for the following reasons. First, a rotation around the Z axis by an arbitrary angle R_Z^θ assumes a simple form due to the symmetry of Y_l^m . Second, a rotation around the Y axis R_Y^θ can be achieved by $R_X^{-\pi/2} R_Z^\theta R_X^{\pi/2}$, and the rotation around the X axis by $\pm\pi/2$ is also of a simple form.

2.4 Spherical Harmonics Representation

Given a rotation matrix $R(\alpha, \beta, \gamma) = R_Z^\gamma R_Y^\beta R_Z^\alpha$ represented in ZYZ Euler angles, we can compute the corresponding spherical harmonic coefficients for $f_{[R]}(s)$ as follows,

$$\begin{aligned} \hat{f}_{[R]} &= \left(\hat{R}_Z^\gamma \hat{R}_X^{-\pi/2} \hat{R}_Z^\beta \hat{R}_X^{\pi/2} \hat{R}_Z^\alpha \right) \hat{h} \\ &= \hat{R}(\alpha, \beta, \gamma) \hat{h}, \end{aligned} \quad (4)$$

where $\hat{f}_{[R]}(\hat{h}, \text{ resp.})$ denotes the SH transformation of $f_{[R]}(h, \text{ resp.})$. Thus, we obtain the following easy-to-compute equivalent difference measurement,

$$\int_{S^2} (f_{[R_a]}(s) - f_{[R_b]}(s))^2 ds = \|\hat{R}_a \hat{h} - \hat{R}_b \hat{h}\|^2 \quad (5)$$

Note that the function $f_{[I]}(s) = h(s)$ can be losslessly projected onto three spherical harmonics bases in band 0 and band 4:

$$f_{[I]} = -\frac{2\sqrt{\pi}}{15} \left(Y_4^0 + \sqrt{\frac{5}{7}} Y_4^4 + 16\sqrt{\pi} Y_0^0 \right). \quad (6)$$

The shape of Y_4^0 and Y_4^4 is shown in Figure 2. If we omit the band 0 component Y_0^0 , which is constant on the sphere and has zero contribution to the difference measurement, the above equation implies that any function obtained from a rotation also only has coefficients in band 4. Thus, we only need the band 4 coefficients in Equation 5, which means that the dimension of the vector \hat{f} to store the coefficients is 9, and \hat{R} can be stored as a 9×9 matrix. After scaling it for simplicity, we can replace the original $f_{[I]}$ by the following form without changing the difference measurement (except for a constant factor),

$$f_{[I]} = \sqrt{7} Y_4^0 + \sqrt{5} Y_4^4. \quad (7)$$

This simplification further reduces the computational cost significantly, because we only need to evaluate two columns in \hat{R} for the smoothness optimization procedure based on the difference measurement.

The requirement of normal alignment leads to the following simple equation using our spherical harmonic representation.

Lemma. The corresponding frame is aligned with z direction, iff $\hat{f}(4)$, i.e. the coefficient for Y_4^0 , is $\sqrt{7}$.

Here, we use $\hat{f}(4+m)$ to denote the coefficient in front of Y_4^m . The proof of the Lemma is provided in the Appendix.

Thus, the alignment of the cross-frame F with normal n can be expressed as

$$\hat{f}_{[R_{n \rightarrow z}]}(4) = (\hat{R}_{n \rightarrow z} \hat{f}_{[I]})(4) = \sqrt{7} \quad (8)$$

where $R_{n \rightarrow z}$ is a rotation which rotates n to axis z , and the ZYZ angle (α, β, γ) can be found from the equation $n = Z(-\alpha)Y(-\beta)Z(-\gamma)z$, with one of solutions being

$$\alpha = -\text{atan2}(n_y, n_x), \quad \beta = -\text{acos}(n_z), \quad \gamma = 0 \quad (9)$$

2.5 Discretization and Numerical Method

The given volume Ω is first tessellated into tetrahedra $\{\text{TET}_j\}$ using NETGEN [Schöberl 1997]. To precisely define the surface alignment constraints, especially when there are sharp surface features, we sample the fields at the (bary)center of each face in this tetrahedral mesh. In such a discretization scheme, we denote the center of face i as a face node p_i , and associate it with ZYZ angles $\Phi_i = (\alpha_i, \beta_i, \gamma_i)$ to represent the cross-frame at p_i .

Simply summing up the cross-frame difference for each pair of immediately adjacent variable nodes into the global smoothness measurement may lead to improper weighting, especially for meshes with non-uniform sampling densities. We instead use the L_2 -integral of the gradient of the spherical harmonic coefficient over the whole volume as follows, similar to the Dirichlet energy $\int_{\Omega} |\nabla \psi|^2$ of scalar field ψ .

For each tetrahedron TET_j with nodes $\{p_i\}, i = 1, \dots, 4$ on its four faces, we evaluate their spherical harmonic representation of the cross-frame from $\{\Phi_i\}$. We use a piecewise-linear spherical harmonic coefficients field \hat{f} , linear within each TET_j . Thus, $\nabla \hat{f}[k]$, the gradient of its k th coefficient, is constant within each tetrahedron. Given the values at the barycenter of each face, we can obtain the same constant gradient using the piecewise-linear (Crouzeix-Raviart) basis w_i associated to p_i ,

$$\nabla \hat{f}(k) = \sum_{i=1}^4 \hat{f}_{p_i}(k) \nabla w_i. \quad (10)$$

The point-wise smoothness measurement can be evaluated as

$$E_s(\text{TET}_j) = E_s(p) = \sum_{k=0}^8 \left\| \nabla \hat{f}(k) \right\|^2, \quad p \in \text{TET}_j, \quad (11)$$

which is constant within each tetrahedron.

The global smoothness measurement is therefore

$$E_s = \int_{\Omega} E_s(p) dp = \sum_{\text{TET}_j} \text{volume}(\text{TET}_j) E_s(\text{TET}_j). \quad (12)$$

In this discretization scheme, the alignment energy can be formulated as

$$\begin{aligned} E_a &= \int_{\partial\Omega} \|(\hat{R}_{n_p \rightarrow z} \hat{f}_p)(4) - \sqrt{7}\|^2 dp \\ &\approx \sum_{\text{TRI}_i \in \partial\Omega} \text{area}(\text{TRI}_i) \|(\hat{R}_{n_i \rightarrow z} \hat{f}_{p_i})(4) - \sqrt{7}\|^2, \end{aligned} \quad (13)$$

where TRI_i is the i th surface triangle, and n_i and p_i are its normal and center position respectively.

Since E_s and E_a are integrated over the volume and surface respectively, to make the result independent of scaling, we balance them by a weighted average: (Note that the volume integral of the gradient squared is of the unit of length)

$$E_f = \frac{E_s}{\text{volume}(\Omega)^{1/3}} + w_a \frac{E_a}{\text{area}(\partial\Omega)}, \quad (14)$$

where w_a is the relative weighting for the alignment constraint.

As an initialization to the non-linear optimization, we first optimize Equation 14 with respect to arbitrary \hat{f} (i.e., the band 4 SH coefficients). This amounts to solving a linear system akin to a Laplace equation of \hat{f} . We then compute the rotation generating the SH coefficients closest to the solution of the linear system for each node. The choice on the relative weight for surface (w_a) can lead to less/no internal singularities (polycube-like topological structure) at around 10^{-2} , or more internal singularities at around 10^2 . With this initial cross-frame field, we employ the L-BFGS method (an implementation based on `alglib` [Bochkanov n.d] and `dlib` [dlib n.d]) to minimize the above energy ($w_a = 10^3$) with the variables $\{\Phi_i\}$. The solver converges to a minimum in about 1000 iterations in most cases (see Figure 3). We illustrate the convergence behavior in Figure 1 with the example on the sculpture model with a simple uniform initial frame field. In the top row, we use the ZYZ angles at each node to rotate the standard cube with one constant color per face. Although adjacent cross-frames are similar, the representative frame chosen out of the 24 possible frames may result in very abrupt changes in color in such a visualization. This shows that our cross-frame representation in spherical harmonics is superior to common local frame representation, because the latter representation requires one to align adjacent local frames before measuring their differences. It could also be noticed in this figure that the boundary constraints for normal alignment properly guide smoothing of the internal 3D cross-frame. Note that the residual will not be reduced to zero for smoothly changing frame field, since it is the integral of the squared gradient field.

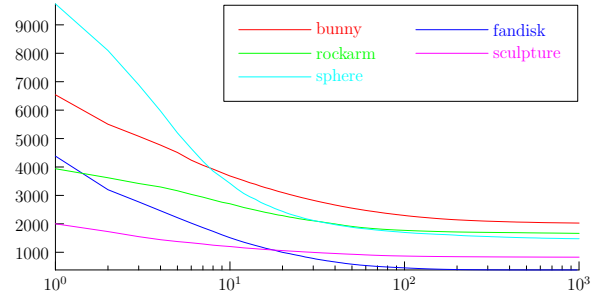


Figure 3: 1000 L-BFGS iterations are enough for our results.

Algorithm 1 Pseudocode for constructing the cross-frame field.

```

 $\hat{f}_0 \leftarrow \arg \min_{\hat{f}} E_f(\hat{f})$ 
for all rotation  $\Phi_i = (\alpha_i, \beta_i, \gamma_i)$  do
   $\Phi_{0,i} \leftarrow \arg \min_{\Phi_i} \|\hat{f}_{0,i} - \hat{R}(\Phi_i)\hat{h}\|^2$ 
end for
repeat
  L-BFGS iteration for  $\arg \min_{\Phi} E_f(\hat{f}_{[R(\Phi)]})$ 
until  $-\frac{\Delta E_f}{E_f} < 10^{-5}$ 

```

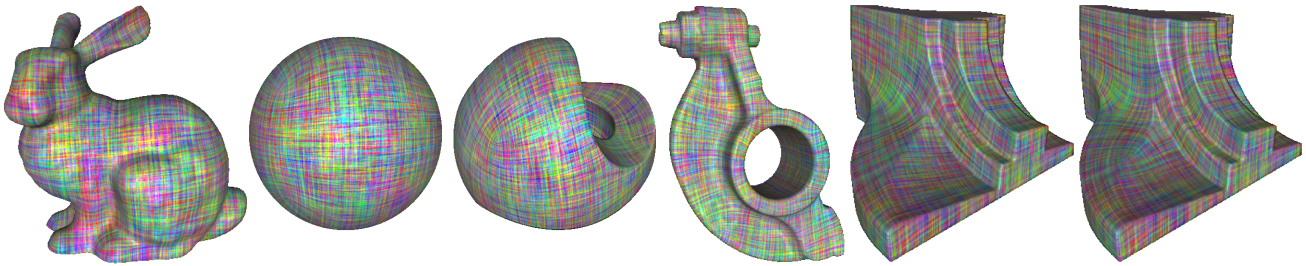


Figure 4: The 3D cross-frames restricted to the surface in Line Integral Convolution visualization. Only the normal alignment constraints are used for the results except the last one, which is constructed according to the surface 2D cross-frame field for comparison ([Ray et al. 2006]).

3 Results and Discussion

In this section, we first visualize the 3D cross-frame field, then demonstrate its application to hexahedron-dominant mesh generation. The performance statistics for our algorithm can be found in Table 1. Note that the memory and the time complexity for each iteration is near linear in the number of the tetrahedrons.

model	tetrahedron	memory	iteration	time
sphere	194k	1.1G	838	21.8m
fan disk	301k	1.8G	536	27.3m
bunny	363k	2.3G	811	37.3m
sculpture	507k	3.0G	667	76.0m
rockarm	947k	5.8G	928	155.9m

Table 1: Performance statistics for our algorithm. The time is measured on a computer with Intel Core i7 920 processor and 18GB memory using two threads.

3.1 Field Visualization

Visualization of a cross-frame field in 3D is much more challenging than in 2D. Therefore, we convert our field into a 2D cross-frame on the surface by restricting the 3D cross-frame to the local tangent plane at each boundary point. Then the method in [Palacios and Zhang 2011] is used to visualize the resulting 2D cross-frame field in Figure 4 and the companion video. It can be observed that the surface component is of high quality. This also indicates that the normal alignment works well, otherwise the tensor field would have been skewed or not aligned with the features.

In the last two fan disk results shown in Figure 4, we compare our results to those of [Ray et al. 2006]. Since in our case the surface cross-frame is also constrained inside of the volume, there are some differences between our method and the results of existing 2D cross-frame field generation methods. If required by the application, we may enhance the alignment by setting a desired surface 2D cross-frame field (generated by, e.g., an automatic quad meshing method) as an additional constraint on the surface. We did not apply the additional 2D cross-frame boundary condition on any other results in this paper.

To get a clear picture of the cross-frame field in the interior of the object, we also visualize it by tracing the streamlines along six directions from a seed point until they reach the boundary or exceed a given length. In general, it is impossible to align the cross-frame globally, therefore we only align the nearby cross-frame locally when tracing the line. As shown in Figure 5, the internal cross-frame is quite smooth, and intersects with the boundary surface either perpendicularly or tangentially. Some streamlines may turn

through a large angle when in a singularity region. However, as there are few singularities in our 3D cross-frame fields, this case rarely occurs.

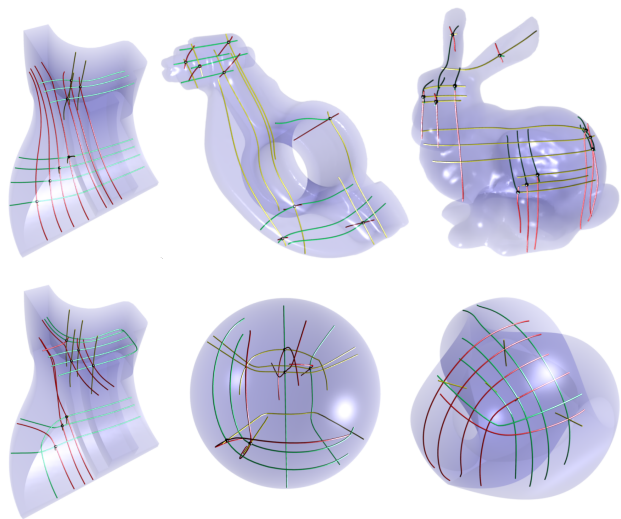


Figure 5: Streamline tracing of the 3D cross-frame field. The seed points are indicated by small black spheres. The fan disk result in the upper row is applied with normal alignment constraints, and the bottom one is created by fixing the cross-frame field on boundary according to a surface 2D cross-frame field.

Similar to the definition in CubeCover [Nieser et al. 2011], we call an edge singular if tracking a frame around a small closed loop around the edge results in a nontrivial transformation. More precisely, we pick an arbitrary orientation for each cross-frame, find the transition rotation in $[I]$ that aligns the frames at each pair of adjacent nodes best, evaluate the product of the transition rotations along a closed loop around each edge formed by the nodes on the faces adjacent to that edge, and mark the edge as a singularity if the product is not the identity matrix. We visualize the singularities as black edges in Figure 6. In some cases, the resulting cross-frame field contains no internal singularity lines, which means that the topology is similar to that of a volumetric polycube.

3.2 Hexahedral Dominant Meshing

Given the boundary aligned 3D cross-frame field from the optimization procedure, one may attempt to generate hexahedral meshes by extending surface quadrangulation techniques. Considering the large number of cells in a volumetric mesh, we choose

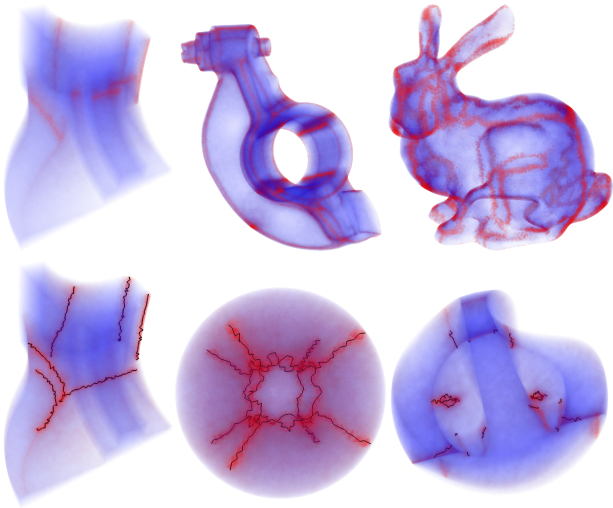


Figure 6: The singularity edges in the volume are rendered black. The pseudo-color in the volumetric rendering indicates the local residual of E_s (increasing from blue to red). The bottom left fundisk result is constructed under the additional boundary constraint of aligning to a 2D cross-frame field.

the mix-integer programming method [Bommes et al. 2009] over the wave-based method [Zhang et al. 2010b] to avoid large-scale non-linear optimization.

As in the mixed integer quadrangulation [Bommes et al. 2009], the first step of the hexahedral remeshing is cutting the tetrahedral mesh into a topological ball with all singularities located on the new boundary. For this purpose, we interpolate the cross-frames defined at the centers of triangles to cross-frames at the center of the tetrahedron by simple averaging. Then, we construct a dual minimal spanning tree (MST) with the cross-frame difference measure between the adjacent tetrahedra as edge weights. After that, we start from a random tetrahedron, and choose a representative frame $R \in [R]$ in each tetrahedra by choosing the one closest to the adjacent traversed tetrahedron in a breadth first search. For each face not in the dual MST, we find the best rotation to align the two local frames in its adjacent tetrahedra, and setup integer constraints for the globally smooth parameterization. If the rotation is an identity matrix, we glue the adjacent tetrahedra as long as it keeps the Euler characteristic to 2, which reduces the number of integer constraints as in [Bommes et al. 2009]. Integer constraints are also introduced for the surface nodes to avoid incomplete cells near the boundary. There may be conflicts in the constraints since we do not perform a thorough topological analysis. So, the integer constraints for the translational jump of parameter u after aligning the rotations in tetrahedra p and q , $u_p - u_q = k_{pq}$, $k_{pq} \in \mathbf{Z}$ is implemented as a penalty added to the target function for the alignment of the frame field and parameter gradients, in the form of $w(u_p - u_q - k_{pq})^2$, where w is the penalty weight. We iteratively solve for the parameterization with mixed-integer programming similar to [Bommes et al. 2009]. Note that in order to make the cubes in Figure 1 accurately spread along the cross-frame field, we do not enforce any of the afore-mentioned integer constraints.

Following [Lévy and Liu 2010], we use the method by [Meshkat and Talmor 2000] to construct a hexahedron-dominant mesh. Given the above volumetric parameterization, we use Morse-Smale Complex (MSC) [Edelsbrunner et al. 2001; Dong et al. 2006] to construct the surface mesh. To ensure the continuity of the scalar field $\cos(\pi u) \cos(\pi v) \cos(\pi w)$ on the surface, we multiply the param-

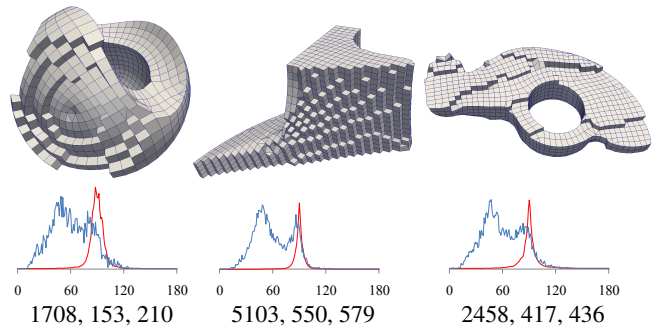


Figure 7: The sectional views of the hexahedral dominant meshes generated from the 3D cross-frame field. The blue and red curves indicate the angle distributions for all the triangular and quadrilateral faces respectively. We list the numbers of hexahedra, pentahedra and tetrahedra separately at the bottom.

eterization (u, v, w) by 2. Then we extract internal grid points with integer parameterization coordinates, construct an intermediate tetrahedral mesh from the surface mesh and these interior points, and finally merge sets of adjacent tetrahedra matching specific patterns into a hexahedron-dominant mesh. The results are illustrated in Figure 7.

4 Conclusion

We present a 3D cross-frame representation and formulate a target function for the optimization of a boundary-aligned smooth 3D cross-frame field. Our work extends the 4-RoS vector field, or cross-frame, on 2D surfaces to 3D through a novel spherical harmonic representation. With such a representation, we can measure the difference between two 3D cross-frames, each regarded as an equivalence class containing 24 different frames. Furthermore, we have proven that the boundary alignment constraint, which is rotationally invariant in the normal direction, can be simply written into a linear equation for the SH representation. With these constraints of object surface normal alignment alone, our method can generate high quality 3D cross-frame fields in the interior of a 3D object with surface feature alignment. Additional constraints can certainly be applied to further control the field generation, such as specifying the orientation at certain locations.

We demonstrate the resulting cross-frame field using several visualization methods. In addition, we extended [Bommes et al. 2009] to apply the cross-frame field in automatic hexahedron-dominant mesh generation. There can be conflicting integer constraints in the parameterization equations due to the lack of a thorough topological analysis. However, with just the cross-frame field, the mesh contains non-hexahedron elements in a small percentage of the volume, and preserves the features well.

A major limitation of this work is a possible high computational cost of the 3D cross-frame field optimization. There is no guarantee for finding the global minimum. We will explore alternative optimization and initialization schemes for better performance and result quality.

A relevant application worthy of further exploration is the automatic generation of pure hexahedral meshes based on our 3D cross-frame field. Most of the latest quadrangulation methods start from a surface cross-frame field, and the quality of the field is crucial for the final result. Our work provides a solution for the first step in extending these methods to 3D. We will explore modifications to the optimization target function, to automatically gen-

erate frame field applicable to the CubeCover hexahedralization method [Nieser et al. 2011], or to automatically generate the orientation field with all singularities restricted to the boundary for the volumetric polycube method [Gregson et al. 2011].

In addition to the application to hexahedral remeshing, the 3D cross-frame can also be used for solid texture synthesis and other applications. As with the numerous applications of symmetric vector fields on surfaces, this new methodology may open a valuable line of research into ways to control the cross-frame field with constraints designed for application-specific requirements.

Acknowledgements

This work was partially supported by China 973 Program (No. 2009CB320801), NSFC (No. 60933007 and No. 61170139) and NSF grants (IIS 0953096, CCF 0936830, CMMI 0757123, and CCF 0811313). We wish to thank Mathieu Desbrun and Charles Owen for helpful discussions. We would also like to thank the anonymous reviewers for their valuable comments and suggestions.

References

- BOCHKANOV, S., n.d. Alglib. <http://www.alglib.net/>.
- BOMMES, D., ZIMMER, H., AND KOBELT, L. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph. (SIGGRAPH)* 28, 3 (July), 77:1–77:10.
- CARBONERA, C. D., AND SHEPHERD, J. F. 2010. A constructive approach to constrained hexahedral mesh generation. *Eng. with Comput.* 26, 4 (August), 341–350.
- CHENG, Y. 1995. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 8 (August), 790–799.
- COMANICIU, D., MEER, P., AND MEMBER, S. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 5, 603–619.
- COOK, W., AND OAKES, W. 1982. Mapping methods for generating three-dimensional meshes. *Computers In Mechanical Engineering*, pp.67–72.
- DLIB, n.d. dlib c++ library. <http://dlib.net/>.
- DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. *ACM Trans. Graph. (SIGGRAPH)* 25, 3, 1057–1066.
- EDELSBRUNNER, H., HARER, J., AND ZOMORODIAN, A. 2001. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry (SoCG)* 30, 1 (July), 87–107.
- EPPSTEIN, D. 1996. Linear complexity hexahedral mesh generation. In *Proc. 12th Symp. Computational Geometry*, ACM, 58–67.
- GREEN, R. 2003. Spherical Harmonic Lighting: The Gritty Details. *Archives of the Game Developers Conference* (Mar.).
- GREGSON, J., SHEFFER, A., AND ZHANG, E. 2011. All-hex mesh generation via volumetric polycube deformation. *Computer Graphics Forum (SGP)* 30:5, 1407–1416.
- KÄLBERER, F., NIESER, M., AND POLTHIER, K. 2007. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum* 26, 3 (Sept.), 375–384.
- KAZHDAN, M. 2007. An approximate and efficient method for optimal rotation alignment of 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 7, 1221–1229.
- LAI, Y.-K., JIN, M., XIE, X., HE, Y., PALACIOS, J., ZHANG, E., HU, S.-M., AND GU, X. 2010. Metric-driven rosy field design and remeshing. *IEEE Transactions on Visualization and Computer Graphics* 16, 1, 95–108.
- LÉVY, B., AND LIU, Y. 2010. Lp centroidal Voronoi tessellation and its applications. *ACM Trans. Graph. (SIGGRAPH)* 29, 4 (July), 119:1–119:11.
- MAKADIA, A., AND DANILIDIS, K. 2003. Direct 3d-rotation estimation from spherical images via a generalized shift theorem. In *IEEE Conference on Computer Vision and Pattern Recognition*, 217–226.
- MANN, S., AND ROCKWOOD, A. 2002. Computing singularities of 3d vector fields with geometric algebra. In *Proceedings of the Conference on Visualization '02*, IEEE Computer Society, Washington, DC, USA, VIS '02, 283–290.
- MESHKAT, S., AND TALMOR, D. 2000. Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh. *International Journal for Numerical Methods in Engineering* 49, 1-2, 17–30.
- NIESER, M., REITEBUCH, U., AND POLTHIER, K. 2011. CUBE-COVER - parameterization of 3d volumes. *Computer Graphics Forum (SGP)* 30:5, 1397–1406.
- PALACIOS, J., AND ZHANG, E. 2007. Rotational symmetry field design on surfaces. *ACM Trans. Graph. (SIGGRAPH)* 26:3, 55.
- PALACIOS, J., AND ZHANG, E. 2011. Interactive visualization of rotational symmetry fields on surfaces. *IEEE Transactions on Visualization and Computer Graphics* 17, 7, 947–955.
- POINTWISE, 2009. Gridgen - reliable cfd meshing. <http://www.pointwise.com/gridgen/>.
- RAY, N., LI, W. C., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph.* 25, 4, 1460–1485.
- RAY, N., VALLET, B., LI, W.-C., AND LÉVY, B. 2008. N-symmetry direction field design. *ACM Transactions on Graphics* 27:2, 10:1–10:13.
- ROCA, X. 2009. *Paving the path towards automatic hexahedral mesh generation*. PhD thesis, Applied Mathematics, UPC, Barcelona.
- RUIZ-GIRONÉS, E., AND SARRATE, J. 2010. Generation of structured hexahedral meshes in volumes with holes. *Finite Elem. Anal. Des.* 46, 10 (October), 792–804.
- SCHÖBERL, J. 1997. Netgen - an advancing front 2d/3d mesh generator based on abstract rules. *Comput and Vis in Science* 1, 1, 41–52.
- SCOTT, M. A., EARP, M. N., BENZLEY, S. E., AND STEPHENSON, M. B. 2005. Adaptive sweeping techniques. In *Proceedings, 14th International Meshing Roundtable*.
- SHEPHERD, J. 2007. *Topologic and geometric constraint-based hexahedral mesh generation*. PhD thesis, University of Utah.
- TAKAYAMA, K., OKABE, M., IJIRI, T., AND IGARASHI, T. 2008. Lapped solid textures: filling a model with anisotropic textures. *ACM Trans. Graph. (SIGGRAPH)* 27, 3 (August), 53:1–53:9.

TAUTGES, T. J. 2001. The generation of hexahedral meshes for assembly geometry: survey and progress. *International Journal for Numerical Methods in Engineering* 50:12, pp. 2617–2642.

VYAS, V., AND SHIMADA, K. 2009. Tensor-guided hex-dominant mesh generation with targeted all-hex regions. In *Proceedings of the 18th International Meshing Roundtable*, B. W. Clark, Ed. Springer Berlin Heidelberg, 377–396.

YAMAKAWA, S., AND SHIMADA, K. 2003. Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells. *International Journal for Numerical Methods in Engineering* 57, 15, 2099–2129.

ZHANG, G.-X., DU, S.-P., LAI, Y.-K., NI, T., AND HU, S.-M. 2010. Sketch guided solid texturing. *Graphical Models* 73:3, 59–73.

ZHANG, M., HUANG, J., LIU, X., AND BAO, H. 2010. A wave-based anisotropic quadrangulation method. *ACM Trans. Graph. (SIGGRAPH)* 29, 4 (July), 118:1–118:8.

Appendix

Proof to the Lemma Given any rotation $R(\alpha, \beta, \gamma)$, using the computer algebra system Maxima, it can be easily verified that $\hat{f}[4] = \sqrt{7}$ is equivalent to:

$$\begin{aligned} & (\cos(4\alpha) + 7) \cos(4\beta) + \\ & (4 - 4 \cos(4\alpha)) \cos(2\beta) + 3 \cos(4\alpha) - 11 = 0 \end{aligned} \quad (15)$$

From Equation 15, we have

$$\begin{aligned} & (p + 7)(2q^2 - 1) + (4 - 4p)q + 3p - 11 = 0 \\ \Rightarrow & p(2q^2 - 4q + 2) + (14q^2 + 4q - 18) = 0 \\ \Rightarrow & p(q^2 - 2q + 1) = -(7q^2 + 2q - 9) \\ \Rightarrow & p(q - 1)^2 = -(q - 1)(7q + 9), \end{aligned} \quad (16)$$

where $p = \cos(4\alpha)$, $q = \cos(2\beta)$.

One solution to the above equation is $q = 1$, i.e. $\beta = n\pi$, rotating around y axis by $n\pi$. In this case, the frame is aligned with z axis.

If $q \neq 1$, we have,

$$p = -\frac{7q + 9}{q - 1}. \quad (17)$$

Since $q \in [-1, 1)$, the only possible solution is $q = -1$, $p = 1$, i.e. $\beta = (n + 0.5)\pi$, $\alpha = 0.5n\pi$. In this case, the frame is also aligned with z axis.

Therefore, if $\hat{f}[4] = \sqrt{7}$, the frame is aligned with z . The converse is trivial.