Mathematics and Implementation of
Computer Graphics Techniques 2015

Boundary Aligned Smooth 3D Cross-Frame Field

Jin Huang, Yiying Tong, Hongyu Wei, Hujun Bao
SIGGRAPH Asia 2011

Kenshi Takayama

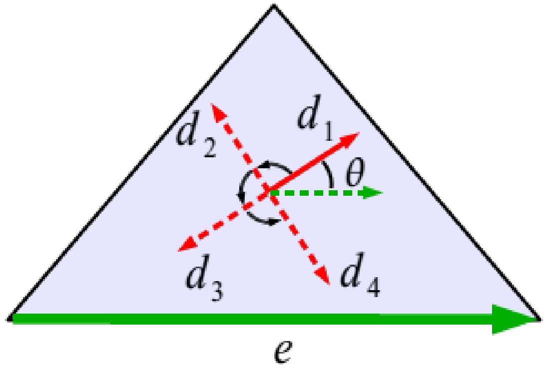
Assistant Prof @ National Insitute of Informatics

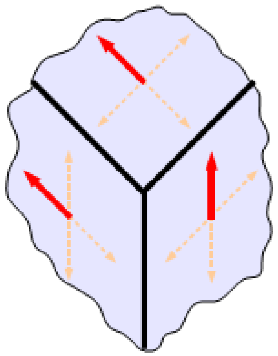
takayama@nii.ac.jp

1st round

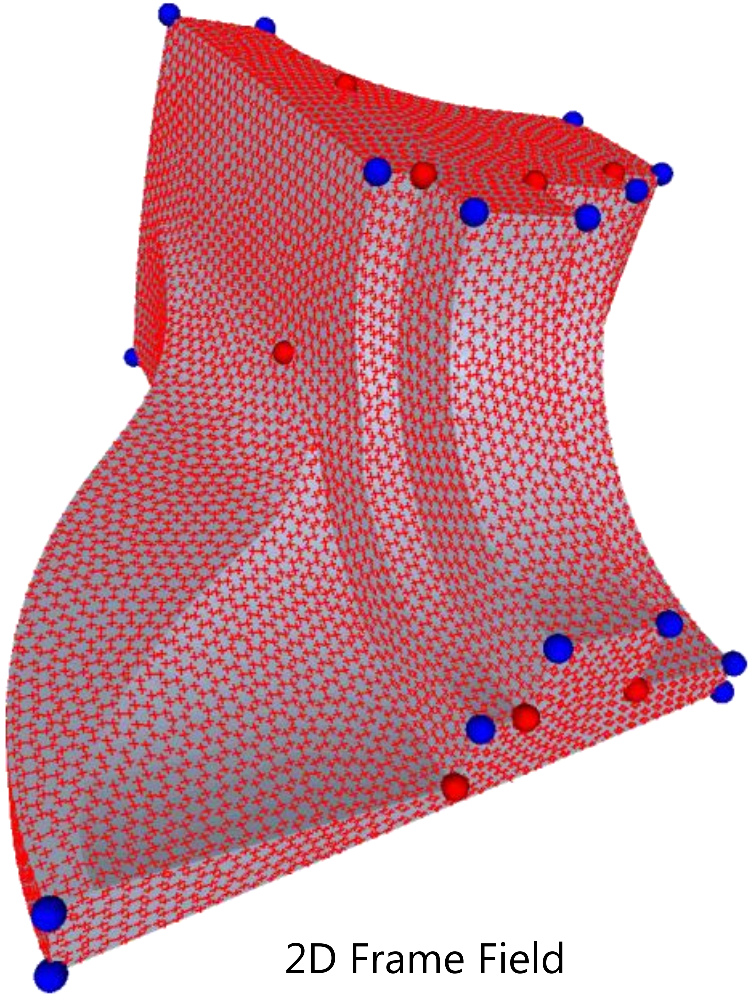
25 July 2015

Background: 2D Frame Field & Quad Meshing

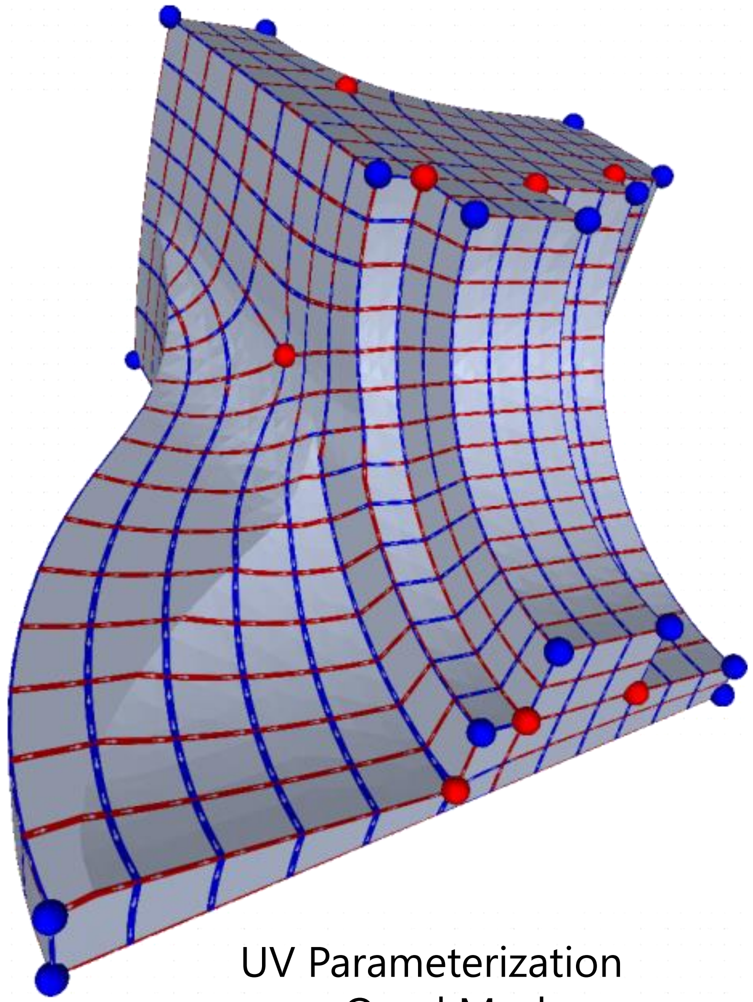




- 2D Frame Field
  ← Auto-computed
- UV Parameterization
  ← Auto-computed



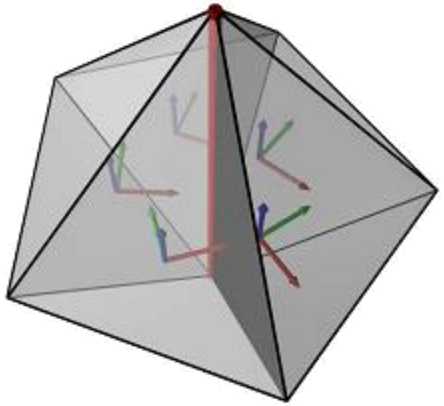
2D Frame Field



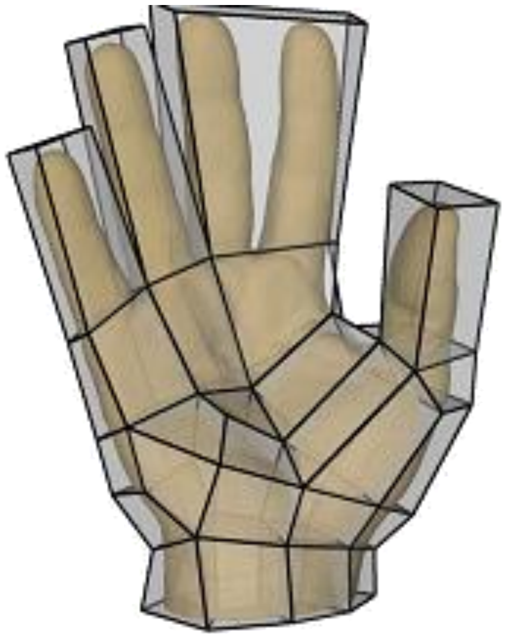
UV Parameterization
= Quad Mesh

Mixed-Integer Quadrangulation [Bommes,Zimmer,Kobbelt,SIGGRAPH09]

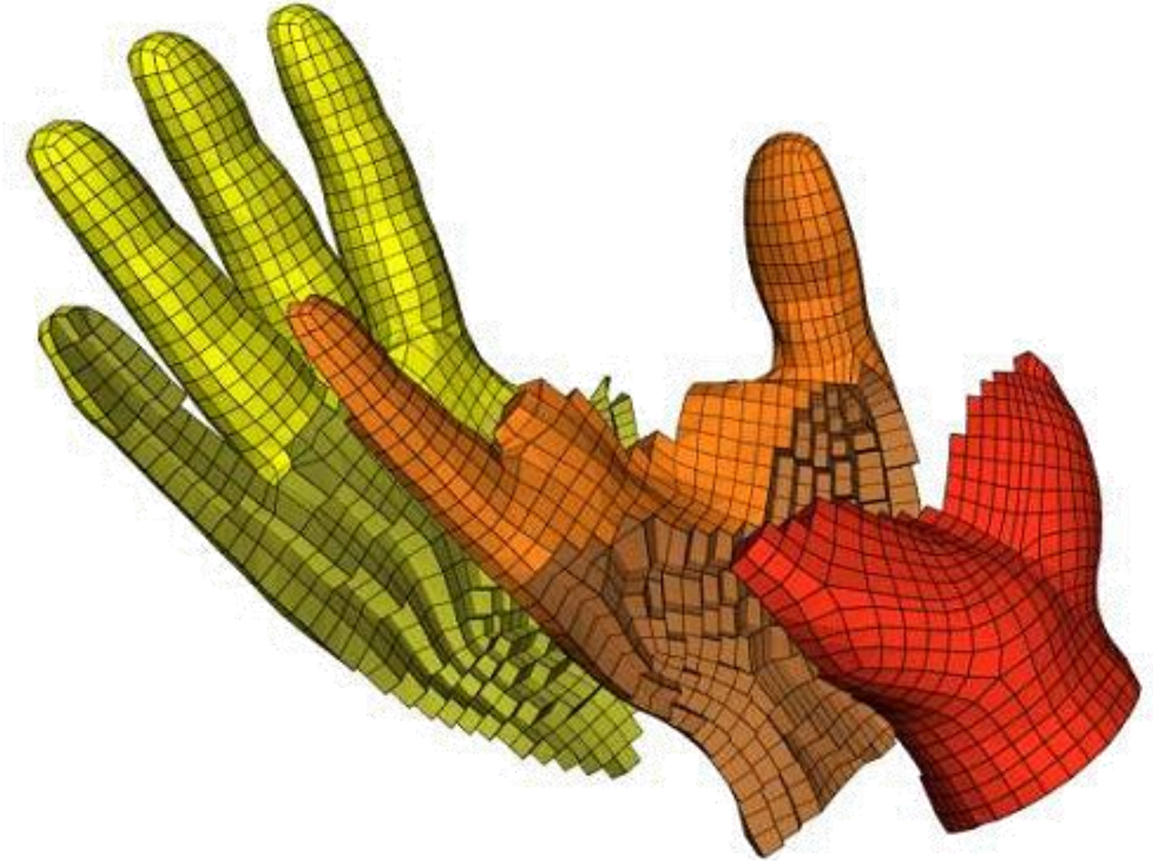
Background: 3D Frame Field & Hex Meshing



- 3D Frame Field
  - 🡐 Heuristic
- UVW Parameterization
  - 🡐 Auto-computed



"Meta-Mesh" to define
3D Frame Field

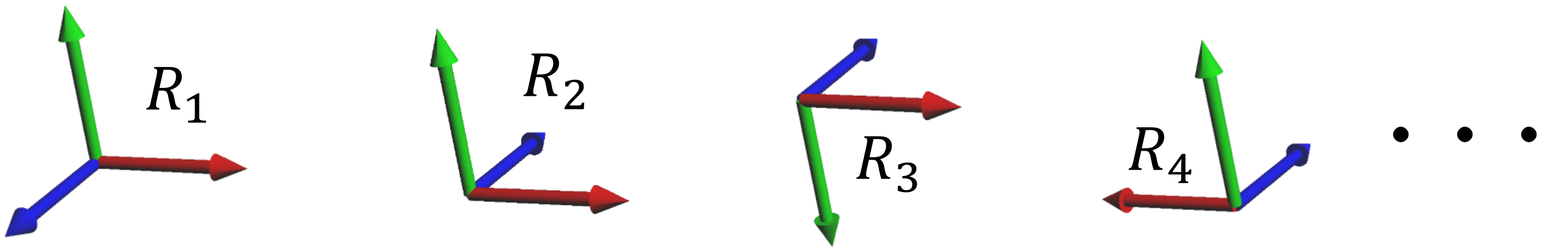


UVW Parameterization
= Hex Mesh

CubeCover - Parameterization of 3D Volumes [Nieser,Reitebuch,Polthier,SGP11]

Definition of 3D Frame

- Don't care about orientation / ordering of axes

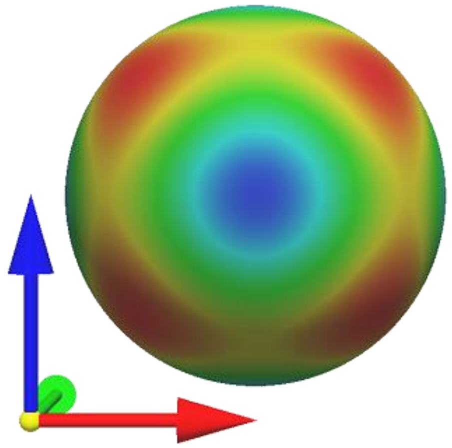


  - $[R_1] = [R_2] = \cdots = \{R_1, R_2, \ldots, R_{24}\}$

- Question: How distant is $[R_a]$ from $[R_b]$?

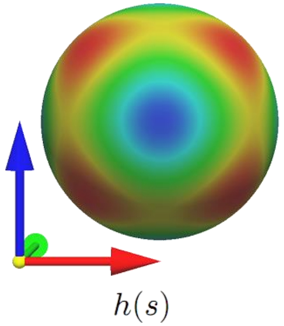
- Key insight: $h(s) \coloneqq s_x^2 s_y^2 + s_y^2 s_z^2 + s_z^2 s_x^2, \quad s \in S^2$

  - Invariant under sign flip / axis reordering!

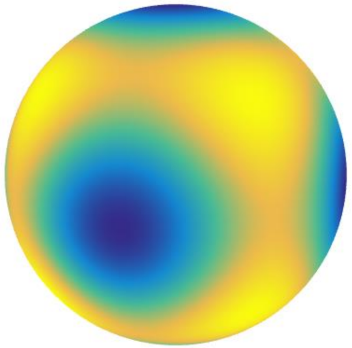


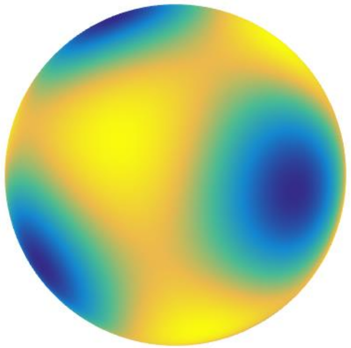
$h(s)$

Distance between 3D Frames



$$h(s) \coloneqq s_x^2 s_y^2 + s_y^2 s_z^2 + s_z^2 s_x^2$$





$$d(R_a, R_b) \coloneqq \int_{s \in S^2} \left(h(R_a^\top s) - h(R_b^\top s) \right)^2 ds$$

- Integral over an entire sphere ➔ **S**pherical **H**armonics!

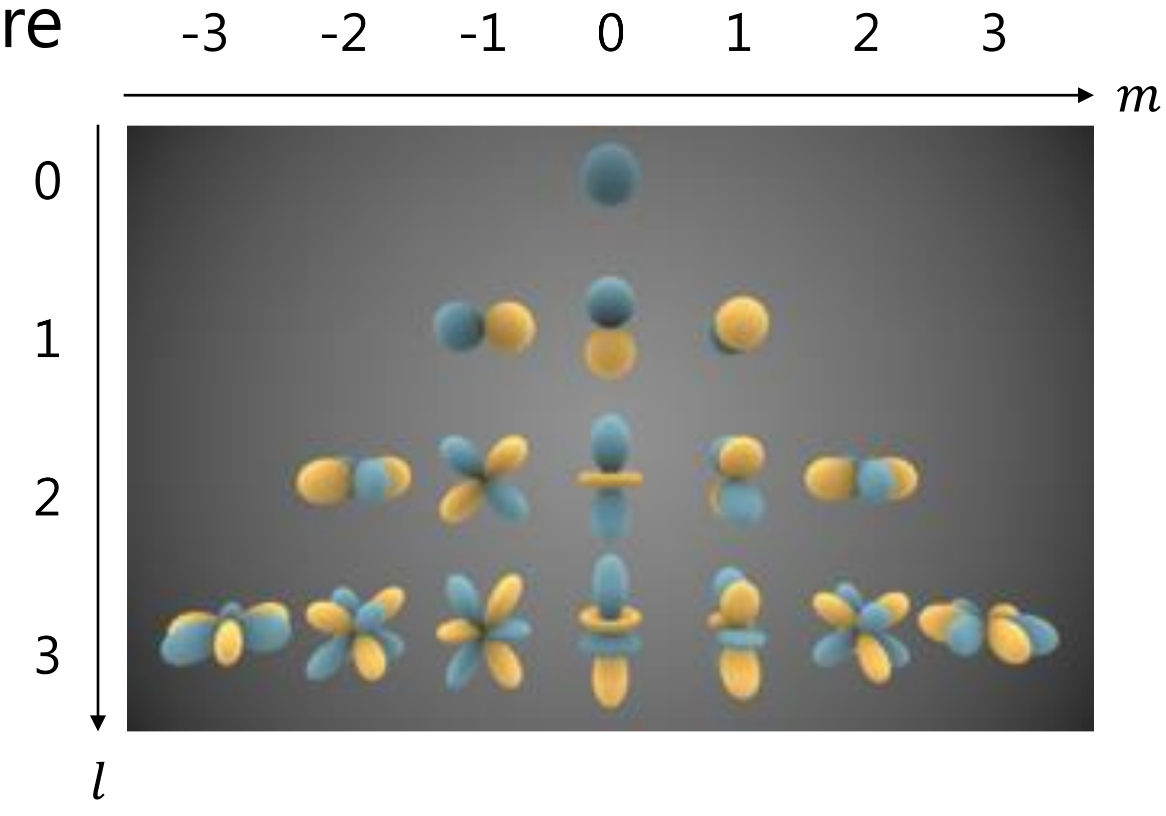
Basics of Spherical Harmonics

- Something like Fourier series on sphere

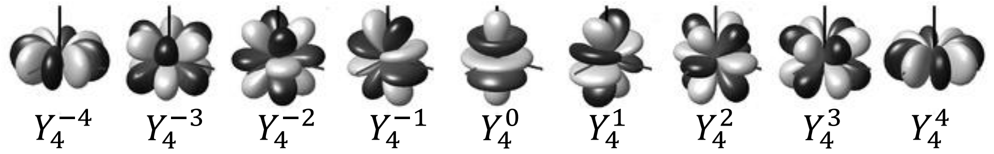
$$f(\theta,\phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \hat{f}_l^m \, Y_l^m(\theta,\phi)$$

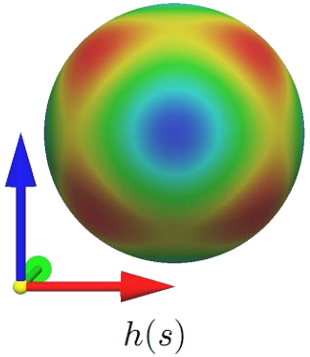
- Orthonormality:

$$\int_{s \in S^2} Y_l^{m_1}(s)\, Y_l^{m_2}(s)\, ds = \begin{cases} 1 & \text{if } \; m_1 = m_2 \\ 0 & \text{otherwise} \end{cases}$$



Frame represented by SH

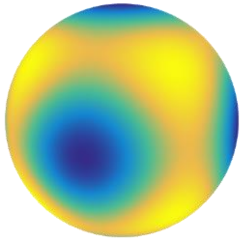




$$h(s) = -\frac{2\sqrt{\pi}}{15}\left(Y_4^0(s) + \sqrt{\frac{5}{7}}Y_4^4(s) + 16\sqrt{\pi}Y_0^0(s) \right)$$

simplify ➡ $h(s) \coloneqq \sqrt{7}Y_4^0(s) + \sqrt{5}Y_4^4(s)$

- "Frequency" unaffected by rotation:



$$h(R^\top s) =: f_{[R]}(s) = \sum_{m=-4}^{4} \lambda_m Y_4^m(s)$$

- Frame represented as SH coeffs for band $l = 4$ (i.e. 9-vector) :

$$\hat{f}_{[R]} \coloneqq (\lambda_{-4}, \lambda_{-3}, \lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1, \lambda_2, , \lambda_3, \lambda_4)$$
$$\hat{h} = \hat{f}_{[I]} = (0,0,0,0,\sqrt{7},0,0,0,\sqrt{5})$$

- Coeffs mapped by *some* 9x9 matrix $\hat{R}$:

$$\hat{f}_{[R]} = \hat{R}\,\hat{h}$$

- Distance between R_a & R_b :

$$d(R_a, R_b) = \left\| \hat{R}_a\hat{h} - \hat{R}_b\hat{h} \right\|^2$$

Computing 9x9 $\hat{R}$ from 3D rotation R

- Not immediately obvious

- Insight: Obvious for certain cases: R_Z^{θ} & $R_X^{\pi/2}$
  - (Not sure…)

Rotation about Z axis by θ

➔ Represent rotation by ZYZ Euler angle

- $R(\alpha,\beta,\gamma) \coloneqq R_Z^{\gamma}\left(R_Y^{\beta}\right)R_Z^{\alpha} = R_Z^{\gamma}\left(R_X^{-\frac{\pi}{2}}\ R_Z^{\beta}\ R_X^{\frac{\pi}{2}}\right)R_Z^{\alpha}$

- $\hat{R}(\alpha,\beta,\gamma) = \hat{R}_Z^{\gamma}\left(\hat{R}_X^{-\frac{\pi}{2}}\ \hat{R}_Z^{\beta}\ \hat{R}_X^{\frac{\pi}{2}}\right)\hat{R}_Z^{\alpha}$

Frame that aligns with boundary surface

- Frame $[R]$ aligns with:
  - Z axis iff $\hat{f}_{[R]}(0) = \sqrt{7}$ (Proof in Appendix)

    Coeff for Y_4^0

  - Surface normal n iff $\left(\hat{R}_{n \to \mathrm{Z}}\, \hat{f}_{[R]}\right)(0) = \sqrt{7}$
    - $R_{n \to \mathrm{Z}}$: Rotation that brings n to Z axis

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

Discretization & Objective

- Tetrahedral mesh over domain Ω
- Frame var $\hat{f}_{p_i}$ at center p_i of every (interior/exterior) triangle TRI_i
- Piecewise-linear frame field $\hat{f}$
  - Gradient $\nabla \hat{f}_{\mathrm{TET}_j}$ constant within each tetrahedron TET_j
- Objective to be minimized:

$$E_{\text{smooth}} \coloneqq \sum_{\mathrm{TET}_j} \mathrm{volume}(\mathrm{TET}_j) \sum_{m=-4}^{4} \left\| \nabla \hat{f}_{\mathrm{TET}_j}(m) \right\|^2$$

$$E_{\text{align}} \coloneqq \sum_{\mathrm{TRI}_i \in \partial\Omega} \mathrm{area}(\mathrm{TRI}_i) \left\| \left(\hat{R}_{n_i \to \mathrm{Z}}\, \hat{f}_{p_i} \right)(0) - \sqrt{7} \right\|^2$$

$$E_{\text{full}} \coloneqq \frac{E_{\text{smooth}}}{\mathrm{volume}(\Omega)^{1/3}} + w_{\text{align}} \frac{E_{\text{align}}}{\mathrm{area}(\partial\Omega)}$$

Optimization

- Energy quadratic in $\{\hat{f}_i\}$ ➔ Simple Laplace-like least squares <Step 1>

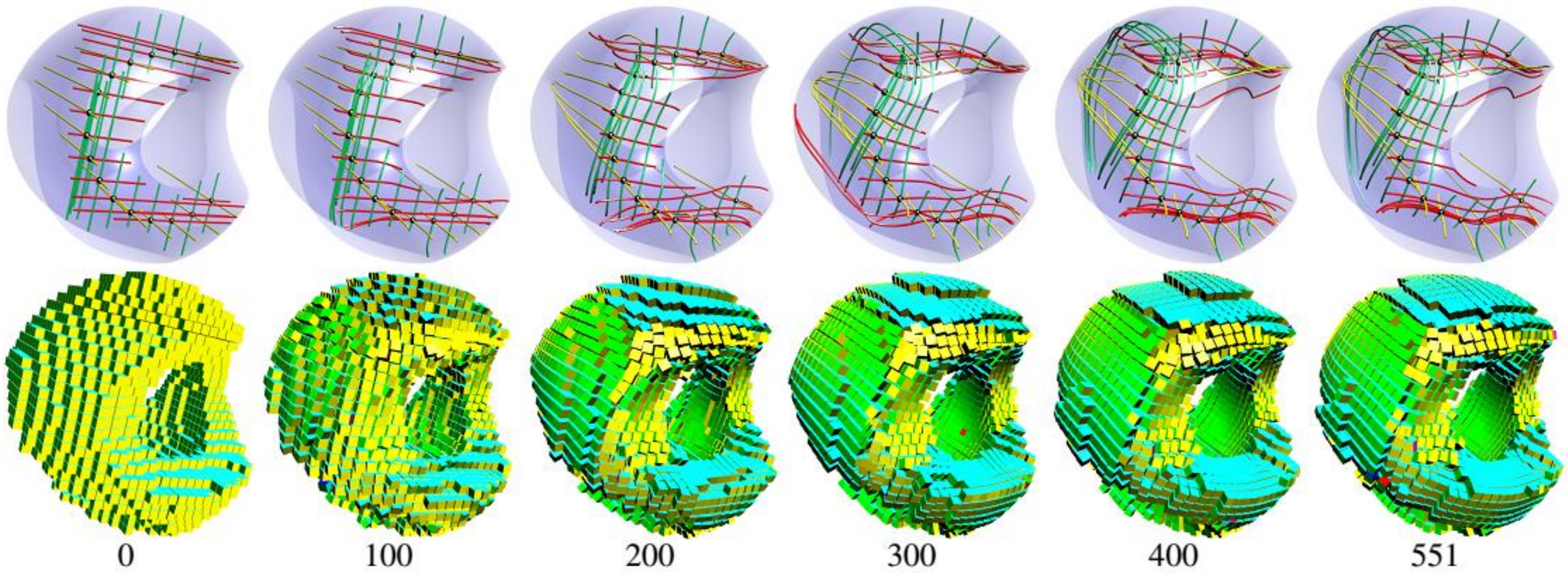
- Problem: Arbitrary $\hat{f}_i$ doesn't represent rotation!

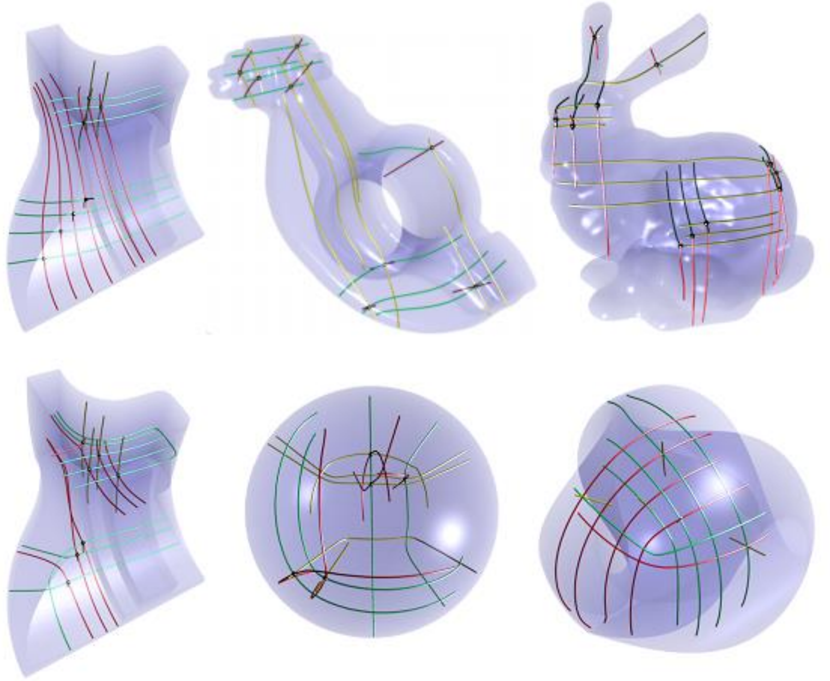
  ➔ <Step 2> *Project* $\hat{f}_i$ to its closest rotation $R(\alpha_i, \beta_i, \gamma_i)$
  - (Not sure how to do it...)

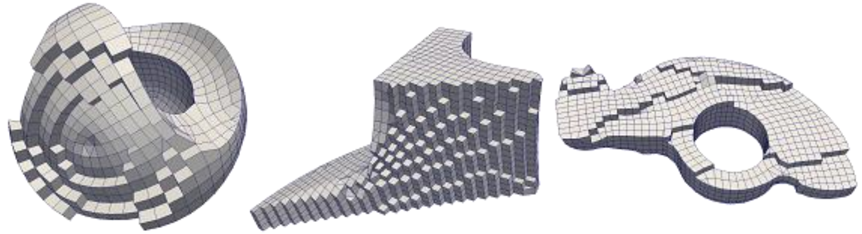
- <Step 3> Using $\Phi_i \coloneqq (\alpha_i, \beta_i, \gamma_i)$ as initial guess, run *nonlinear optimization* over $\{\Phi_i\}$
  - L-BFGS (solver: ALGLIB, dlib, etc)
  - (Not sure about analytic form of derivative...)

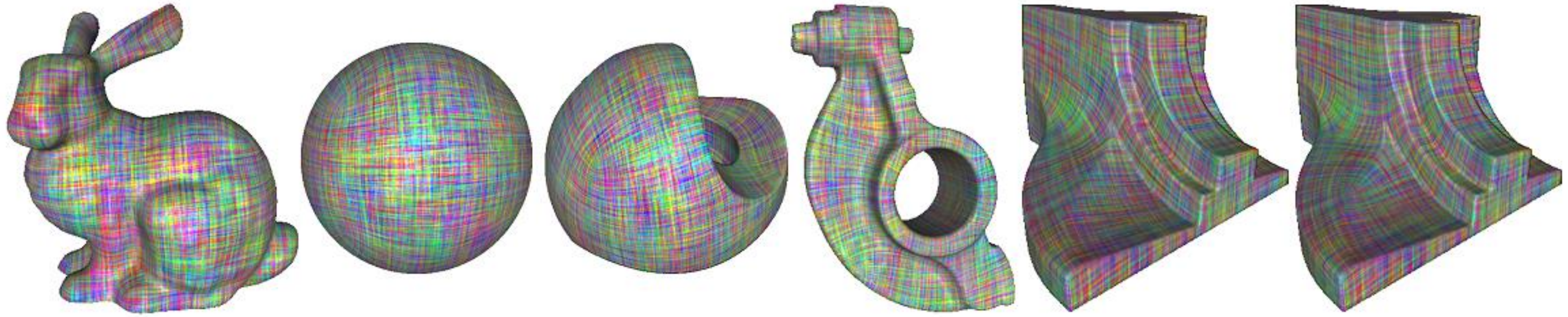
$\hat{f}_0 \leftarrow \arg\min_{\hat{f}} E_f(\hat{f})$
for all rotation $\Phi_i = (\alpha_i, \beta_i, \gamma_i)$ **do**
  $\Phi_{0,i} \leftarrow \arg\min_{\Phi_i} \|\hat{f}_{0,i} - \hat{R}(\Phi_i)\hat{h}\|^2$
end for
repeat
  L-BFGS iteration for $\arg\min_{\Phi} E_f(\hat{f}_{[R(\Phi)]})$
until $-\frac{\Delta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

Results & Performance









model	tetrahedron	memory	iteration	time
sphere	194k	1.1G	838	21.8m
fandisk	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

Questions

- Regarding implementation:
  - Expressions for $\hat{R}_{\mathrm{Z}}^{\theta}$ & $\hat{R}_{\mathrm{X}}^{\pi/2}$
  - Projection of $\hat{f}_i$ to its closest rotation $R(\alpha_i, \beta_i, \gamma_i)$
  - Analytic derivative of E_{full} w.r.t. $\{\Phi_i\}$

- Possible idea for improvement:
  - Can we sidestep nonlinear optimization by alternating Laplace smoothing and “normalization”?