Cubic Stylization & Polycube

USTC, 2024 Spring

Qing Fang, fq1208@mail.ustc.edu.cn

https://qingfang1208.github.io/

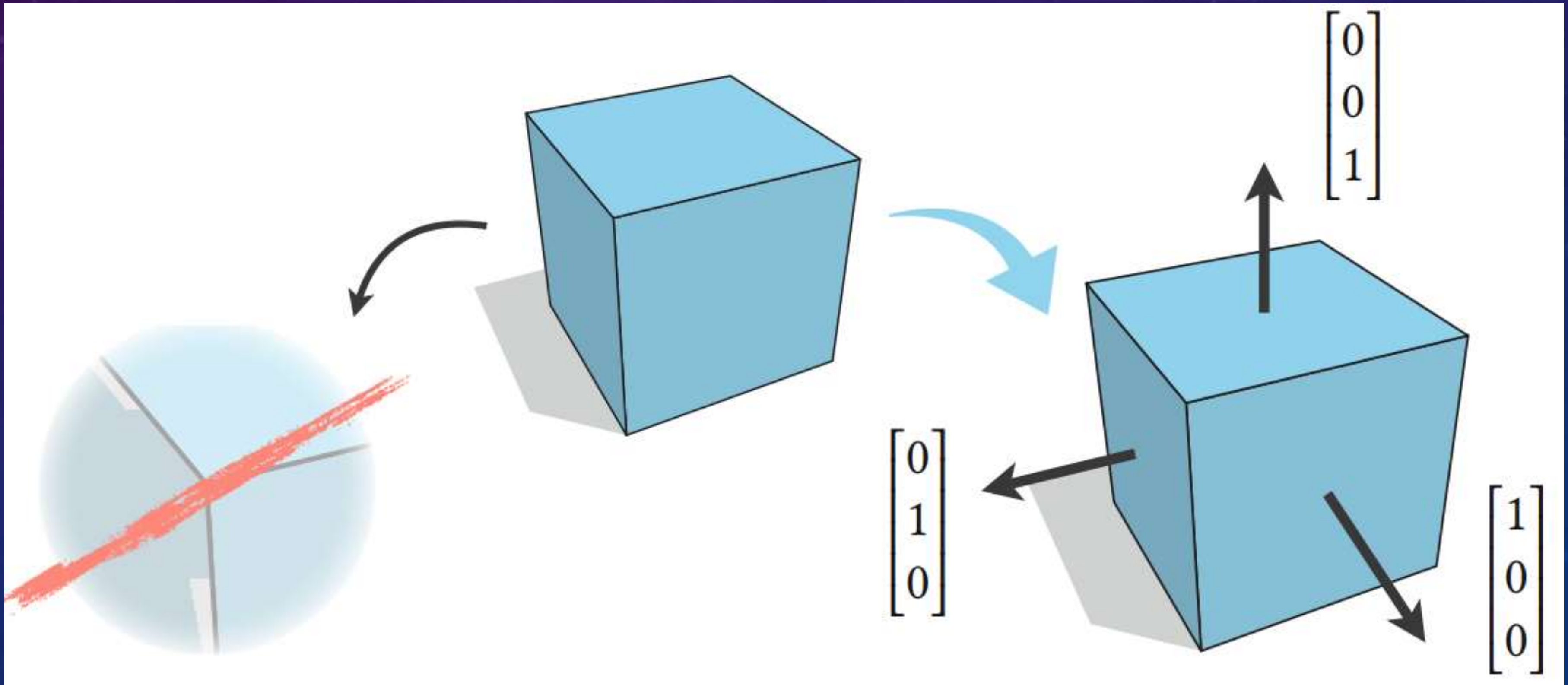
Cubic stylization

Cubic stylization

How to characterize a cube?

Cubic stylization

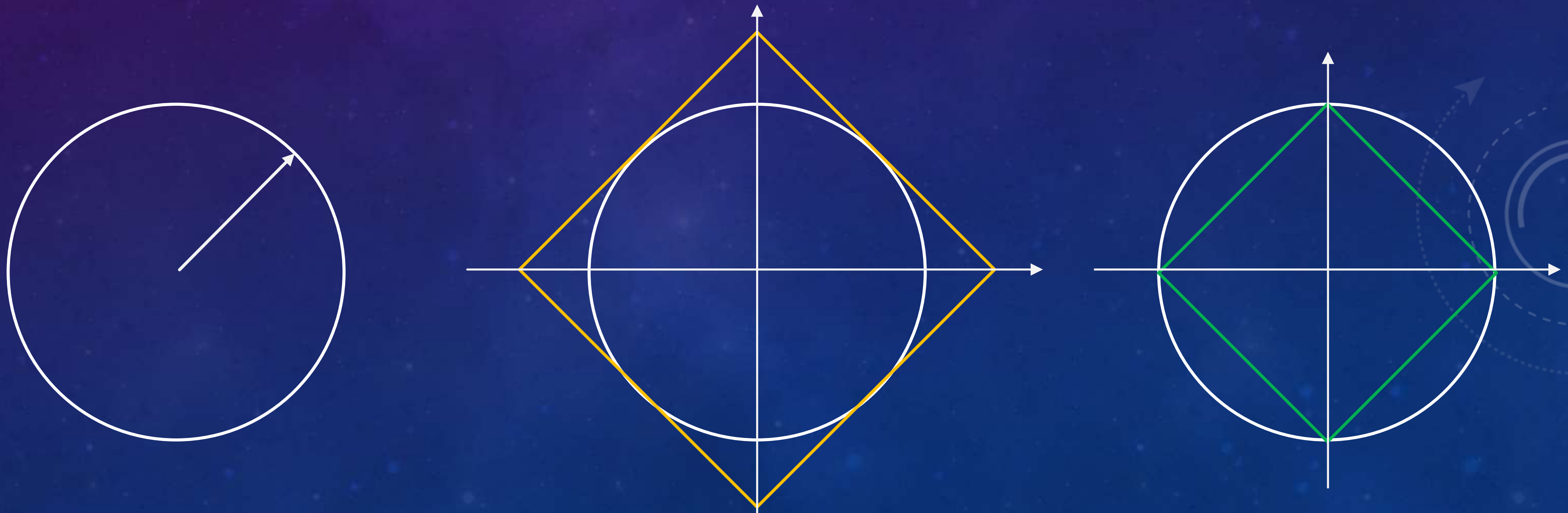
- Cubic geometry has axis-aligned surface normals



Cubic stylization

- Minimizing L1-norm

$$\|n\|_1 = |n_x| + |n_y|$$



Cubic stylization

- As-rigid-as-possible deformation

$$E(R,p') = \sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2$$

- Vertex normal of deformed mesh $n_i' = R_i n_i$

$$E_{cubic} = \sum_i a_i \|R_i n_i\|_1$$

Cubic stylization

$$E(R, p') + \lambda E_{cubic}$$

Optimization

$$\sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2 + \lambda \sum_i a_i \|z_i\|_1, s.t. z_i - R_i n_i = 0$$

ADMM updates - penalty functions $\frac{\rho}{2}\|z_i - R_i n_i + u_i\|_2^2$

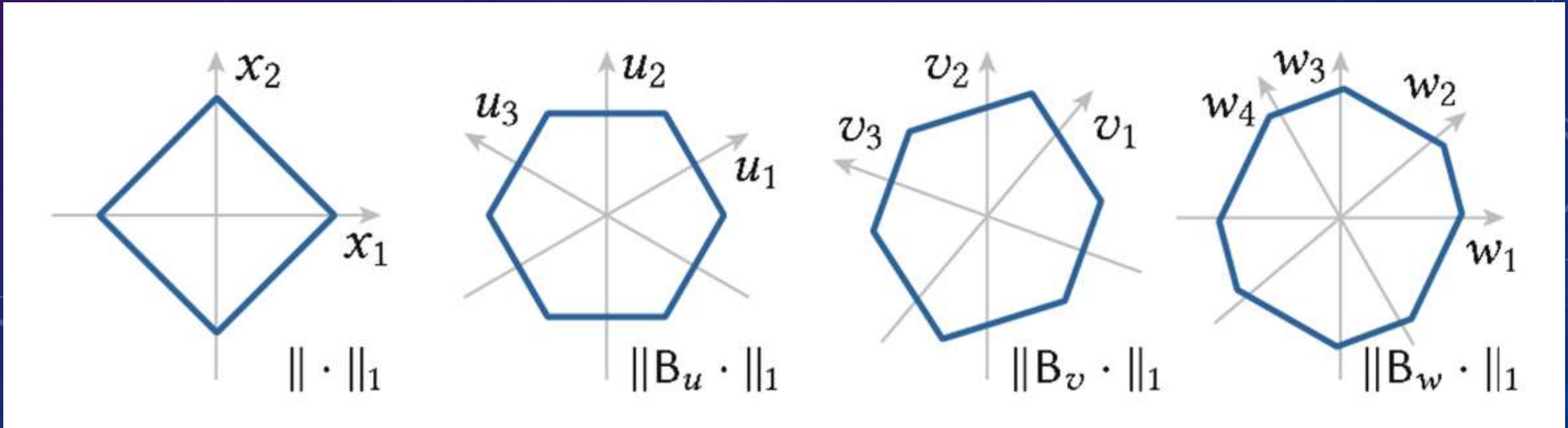
1. Local update R_i
2. Local update z_i
3. Update u_i and ρ

Orientation Dependent

Polygonal Boxes Stylization

Polygonal Boxes Stylization

$$\sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2 + \lambda \sum_i a_i \| B R_i n_i \|_1$$



Assignment requirements

- Cubic stylization algorithm
- Email: ID_name_homework#1.zip
  - Pdf : Input + parameter + output
  - Source code (no exe)
- Deadline: 2024.04.17, 23:59

Extension

- Normal-Driven Spherical Shape Analogies

Normal-Driven Spherical Shape Analogies

- Spherical Shape Analogies

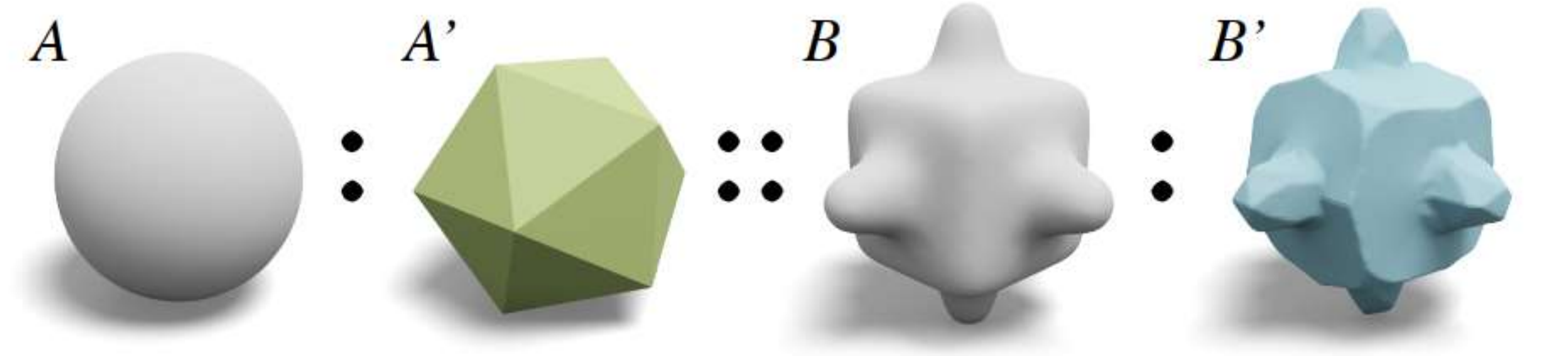
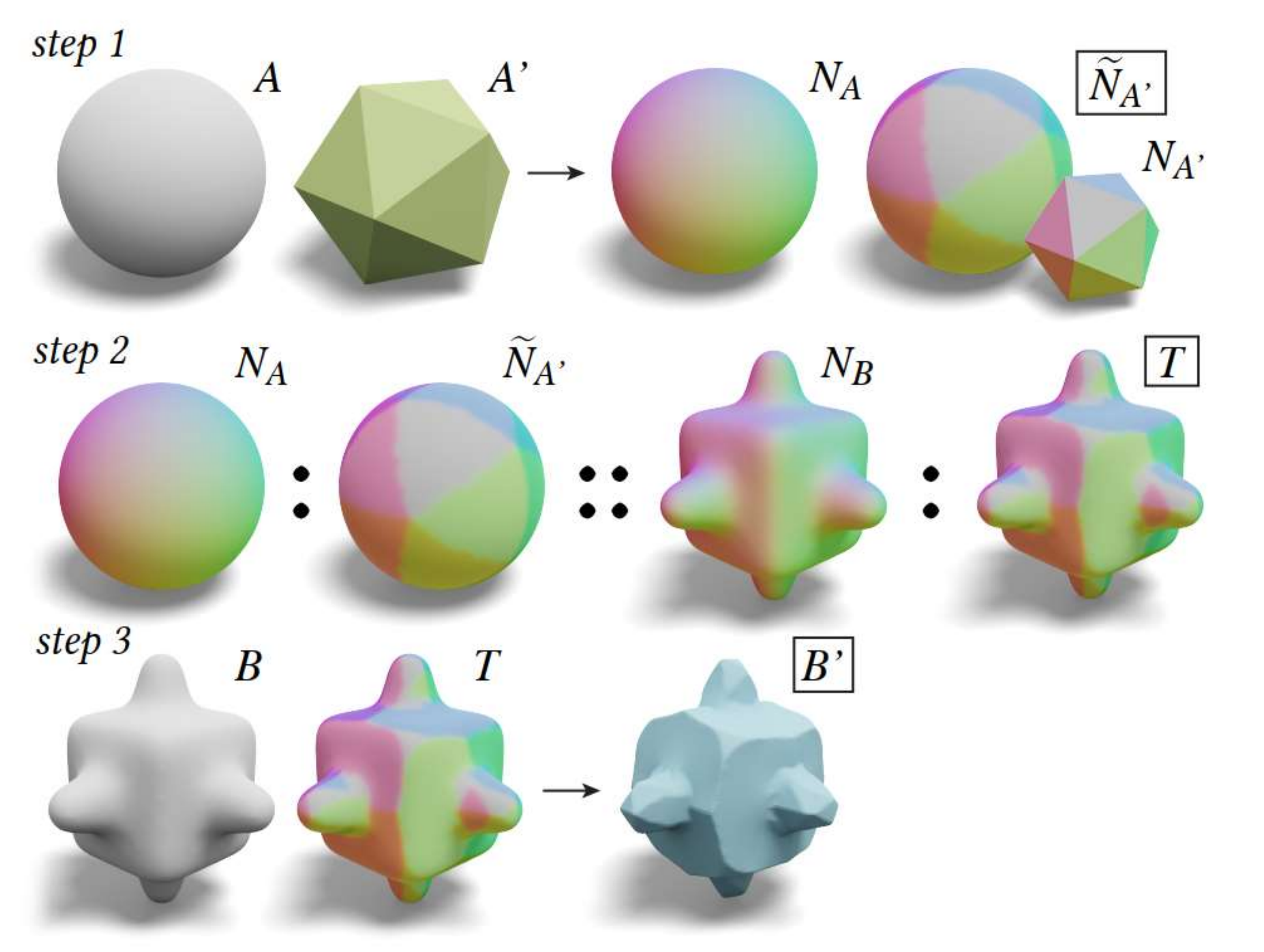


Figure 6: *We generate an output shape B' that relates to the input B in the same way as how the surface normal of a given primitive A' relates to the surface normal of a sphere A.*

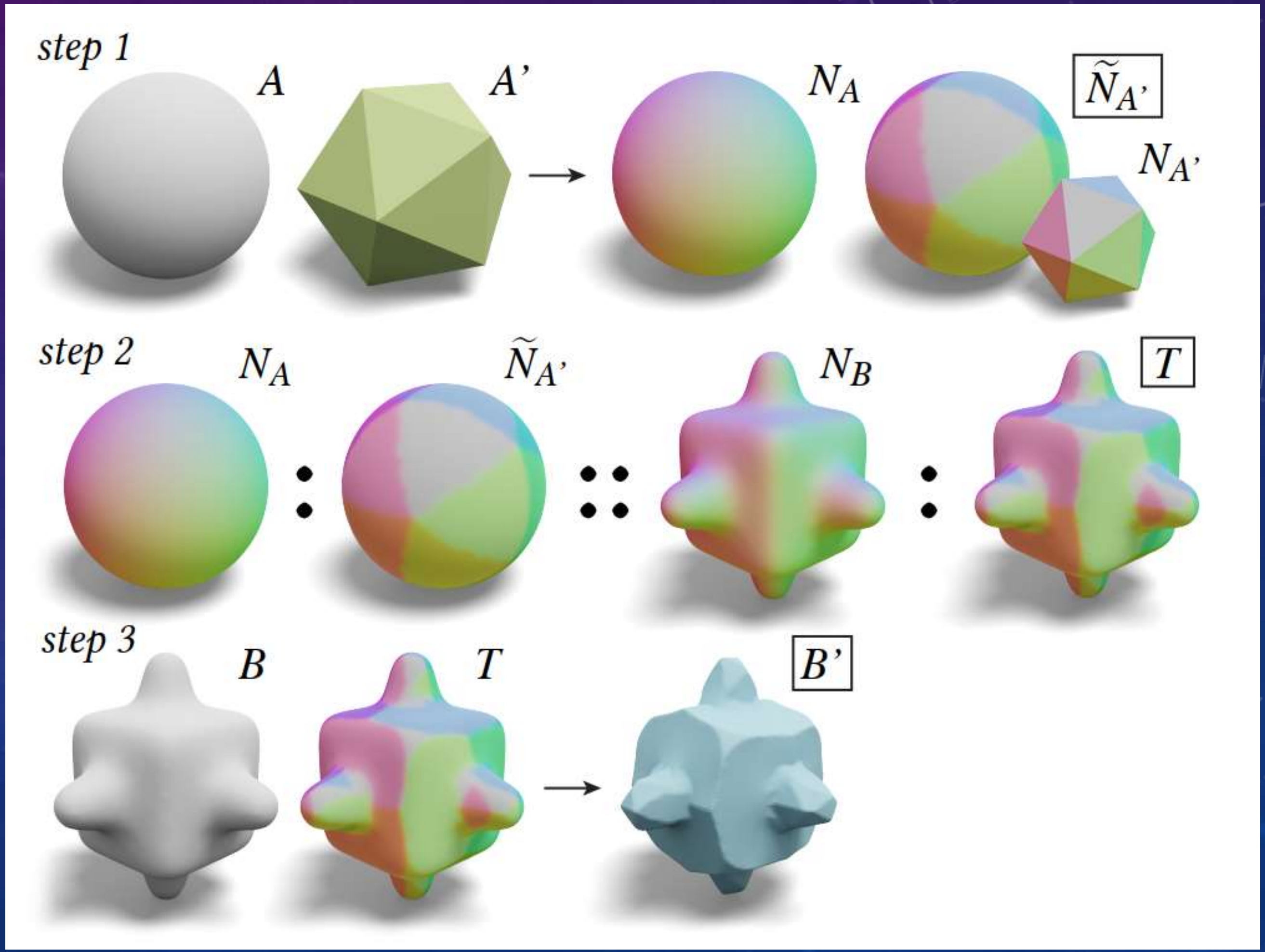
Normal-Driven Spherical Shape Analogies

- Map the normals of the style shape $N_{A'}$ to a unit sphere to obtain $\widetilde{N}_{A'}$ (top row)
- Transfer the relationship between N_A and $\widetilde{N}_{A'}$ to the input shape to obtain the target normal T (middle row)
- Optimize the input shape B so that the actual output normals are aligned with the target normal T (bottom row)



Generating $\widetilde{N}_{A'}$

- Closest normals
- Spherical parameterization
- User-provided $\widetilde{N}_{A'}$



Normal-Driven Optimization

- As-rigid-as-possible deformation

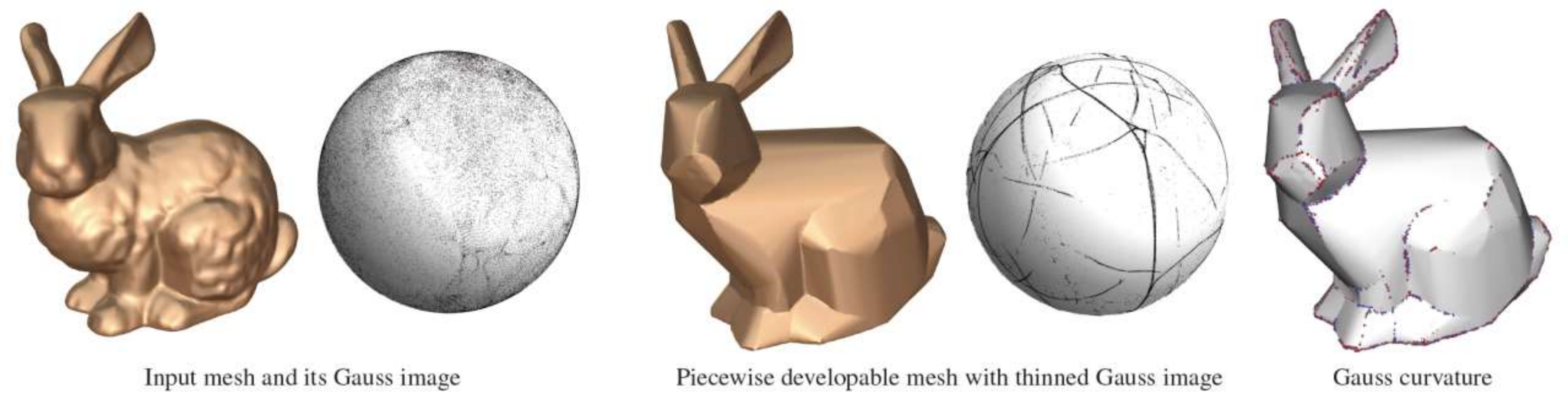
$$E(R,p') = \sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2$$

- Vertex normal of deformed mesh $n_i' = R_i n_i$

$$E_{cubic} = \sum_i a_i \| R_i n_i - t_i \|_2^2$$

Extension

- Developable Approximation via Gauss Image Thinning



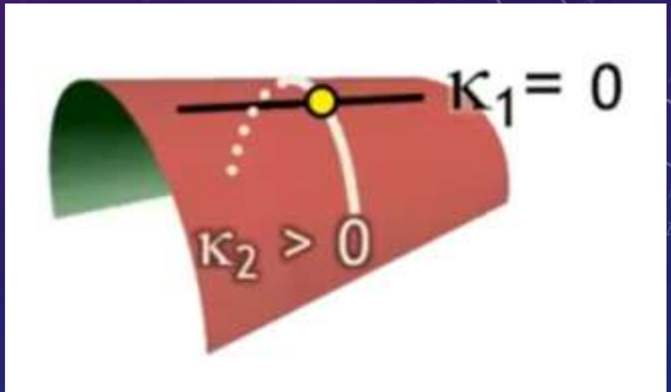
Smooth developable surface

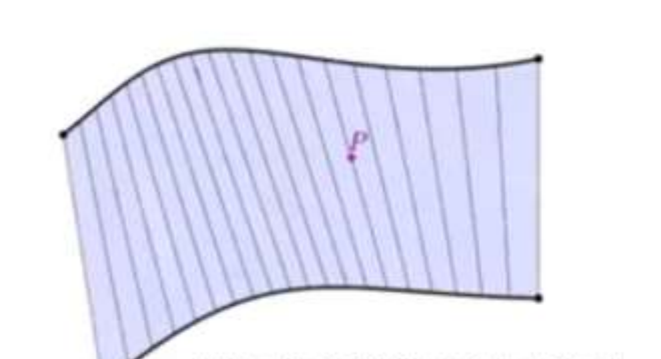
- Zero Gaussian curvature $K = \kappa_1 \kappa_2 = 0$
- Special ruled surface

$$\boldsymbol{x}(u, v) = (1 - v)\boldsymbol{a}(u) + v\boldsymbol{b}(u)$$

$$\Longrightarrow \det(\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{a} - \boldsymbol{b}) = 0$$

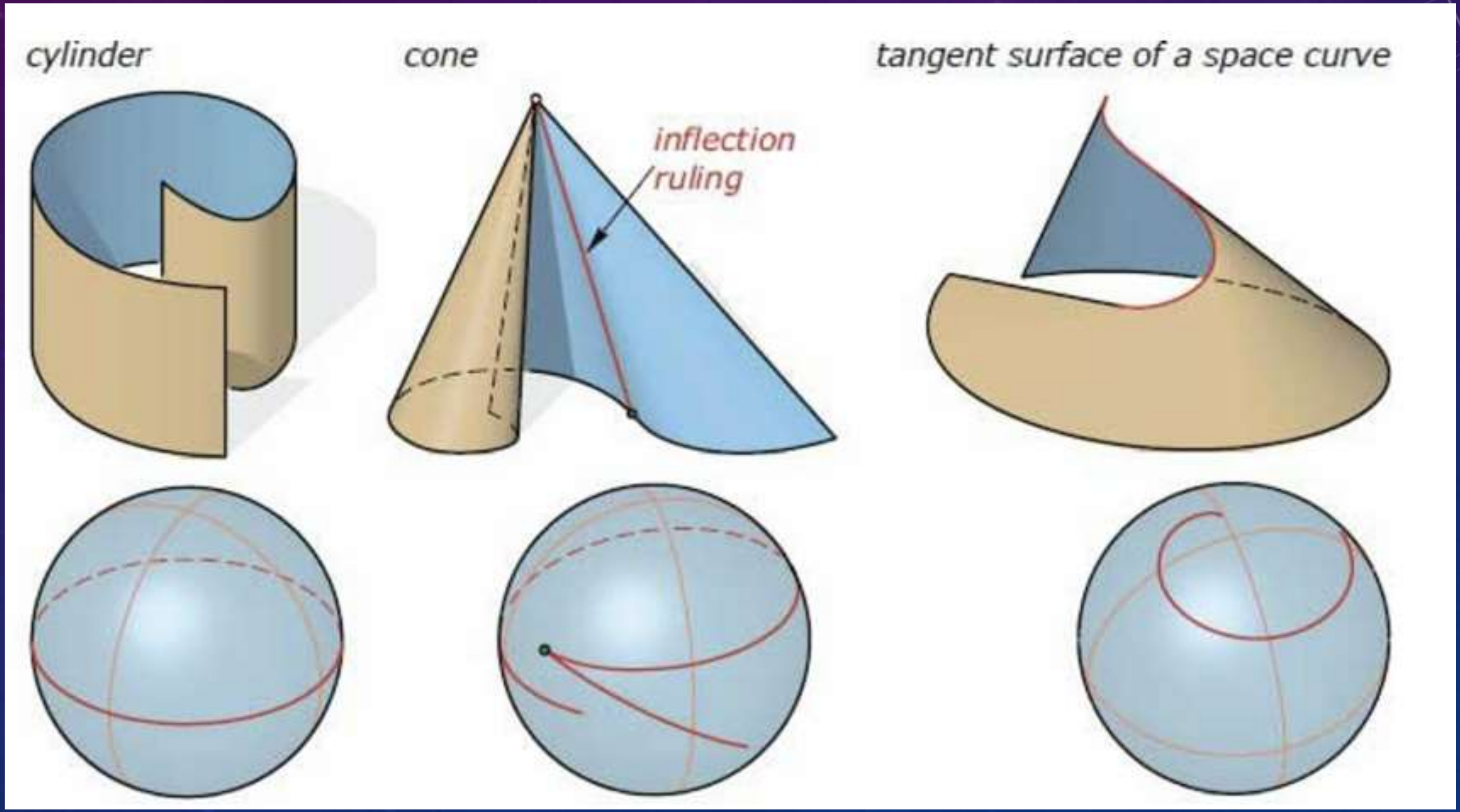
- 1-dimensional Gauss image



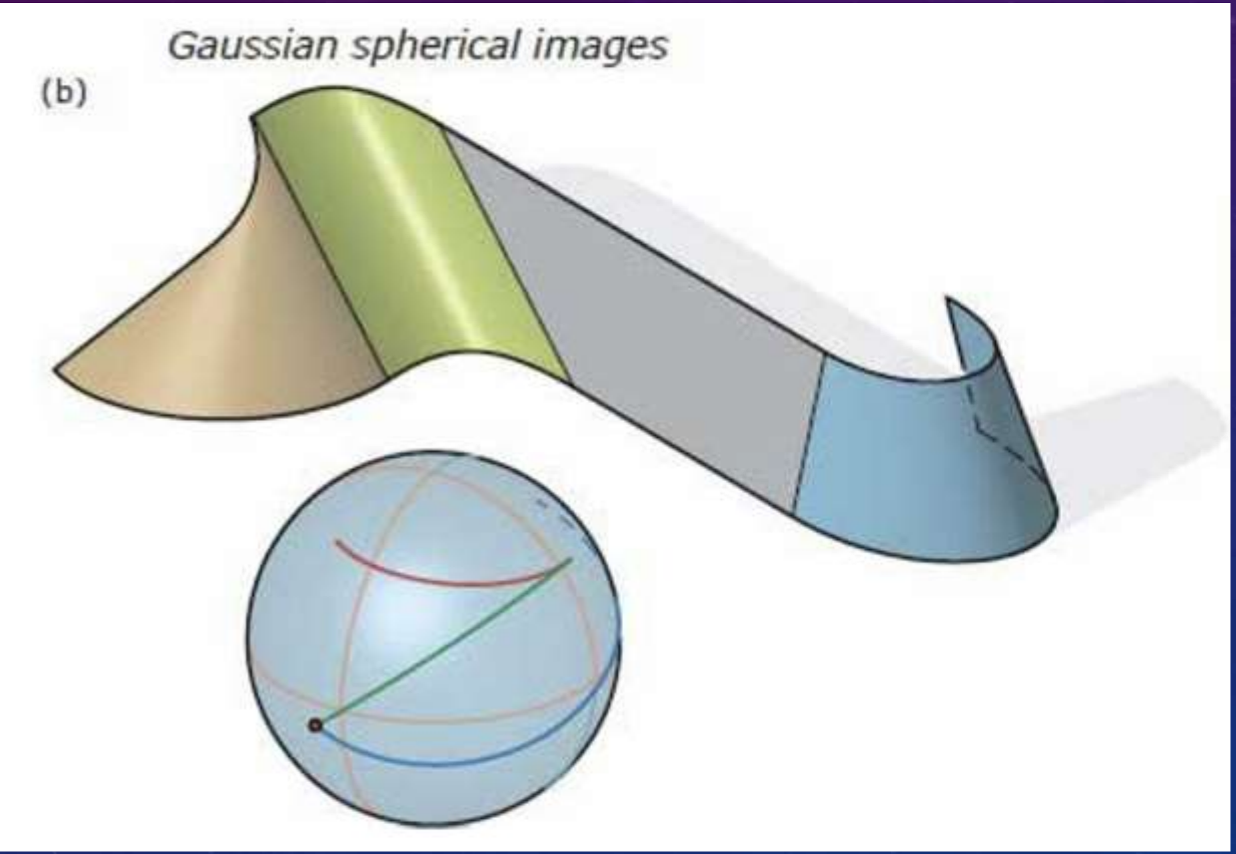


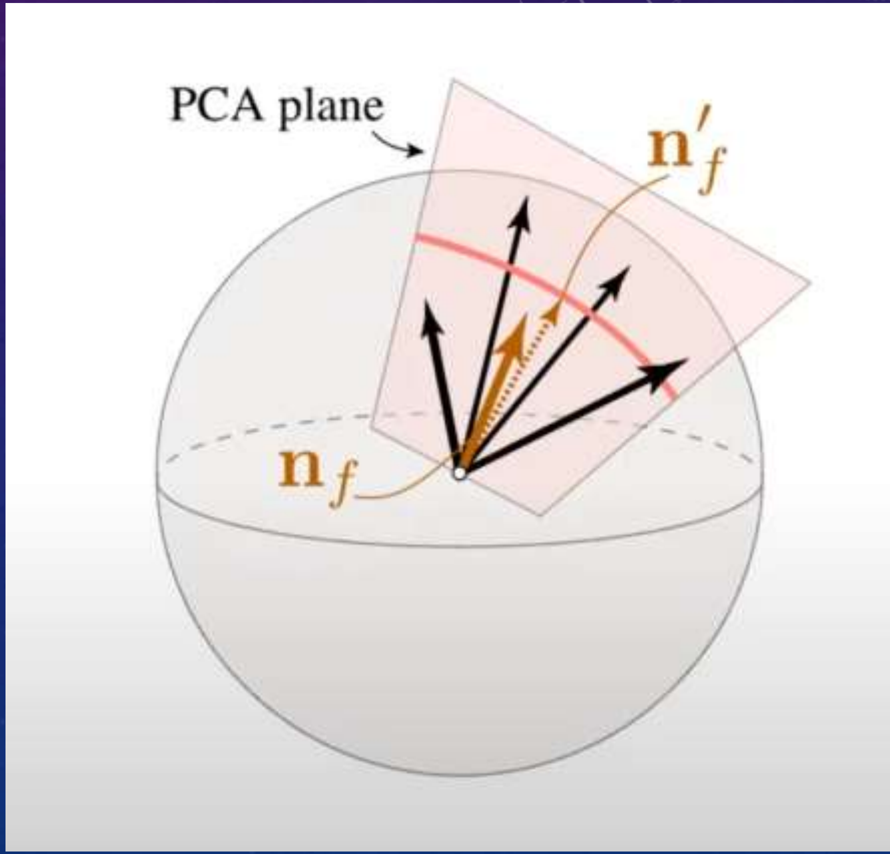
Wikipedia (CC BY-SA 4.0) by Ag2gaeh

1-dimensional Gauss image



1-dimensional Gauss image

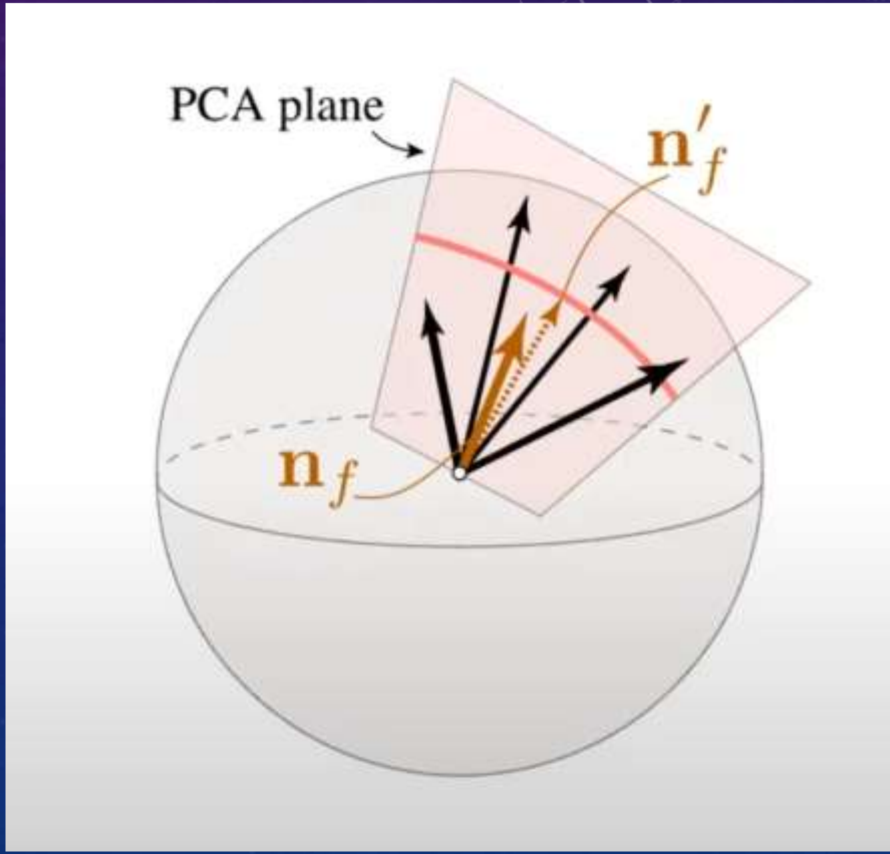




1-dimensional Gauss image

$$A_f = \sum_{g \in \mathcal{N}_f} w_g n_g n_g^T$$

- First two right singular vectors span the plane.
- Project n_g to the plane to get target normals



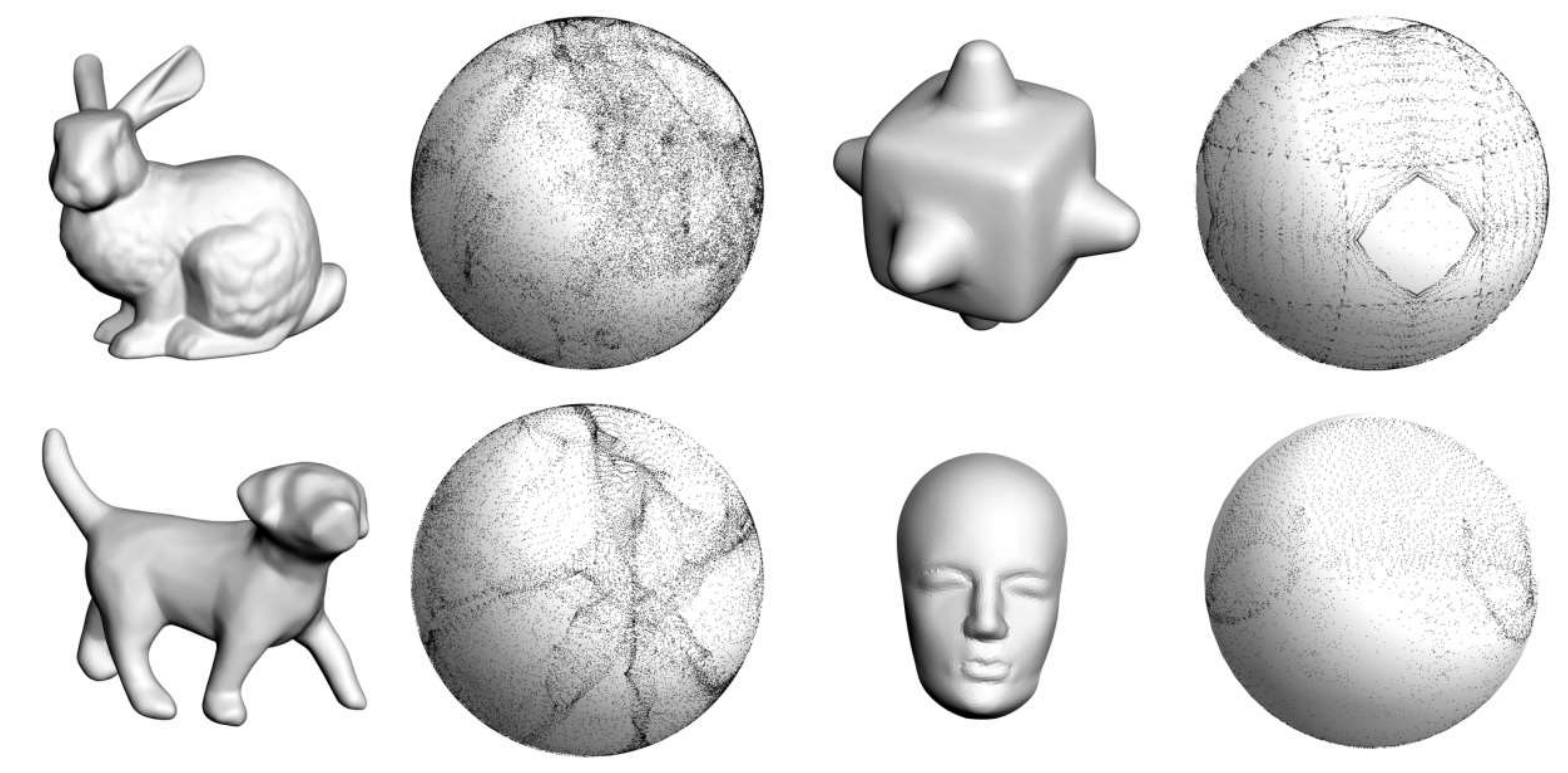
Normal-Driven Optimization

- As-rigid-as-possible deformation

$$E(R,p') = \sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p'_i - p'_j) - R_i(p_i - p_j) \right\|^2$$

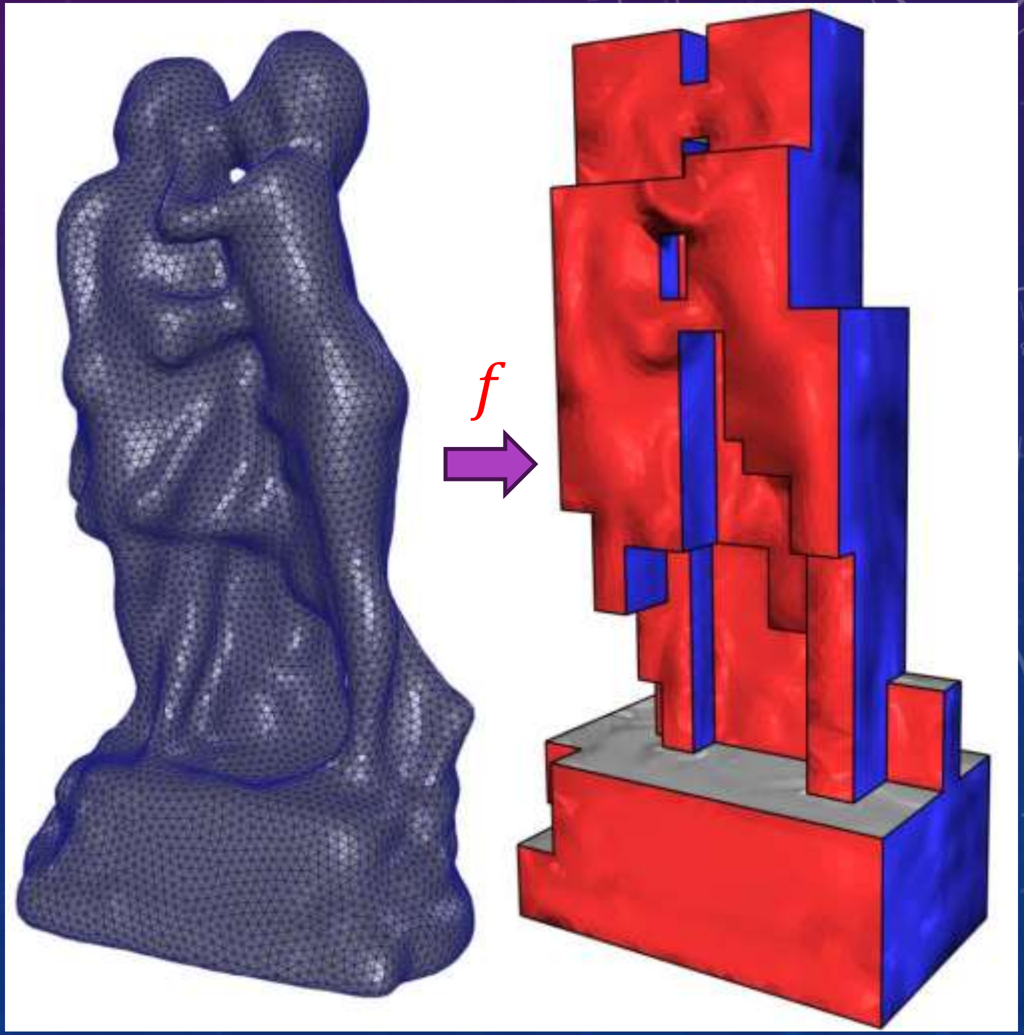
- Vertex normal of deformed mesh $n'_i = R_i n_i$

$$E_{cubic} = \sum_i a_i \| R_i n_i - t_i \|_2^2$$



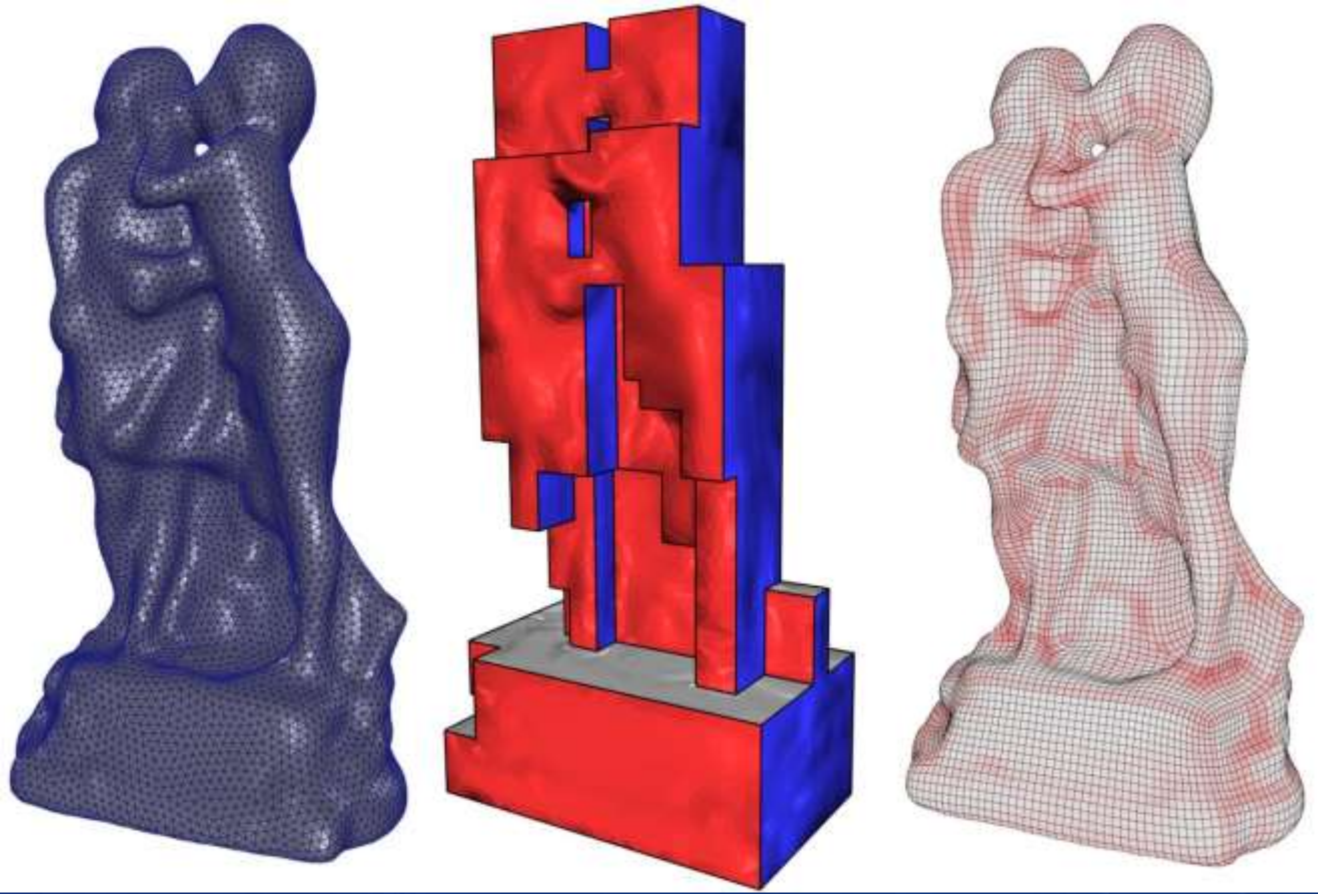
PolyCube

- Definition:
  - Compact representations for closed complex shapes
  - Boundary normal aligns to the axes.
  - Axes: $(\pm1,0,0),\ (0,\pm1,0),\ (0,0,\pm1)$
- PolyCube-map f
  - A mesh-based map.
  - Foldover-free and low distortion.



Applications

- All-hex meshing



Applications

- All-hex meshing
- Texture Mapping

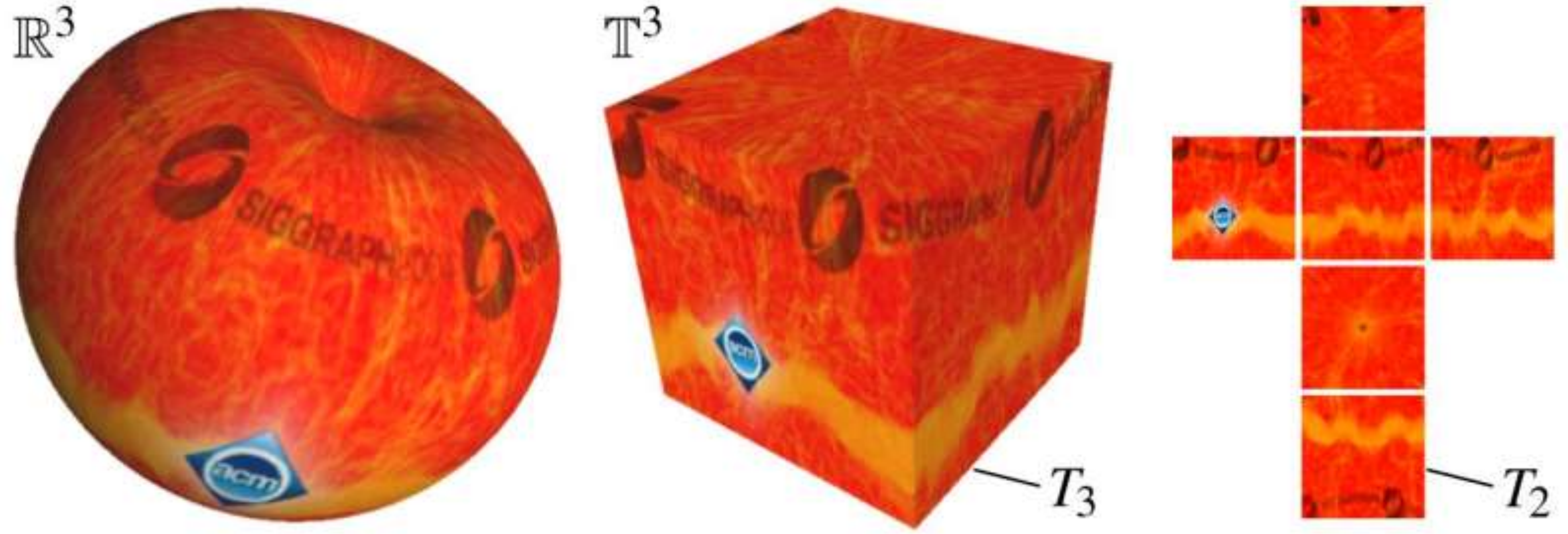
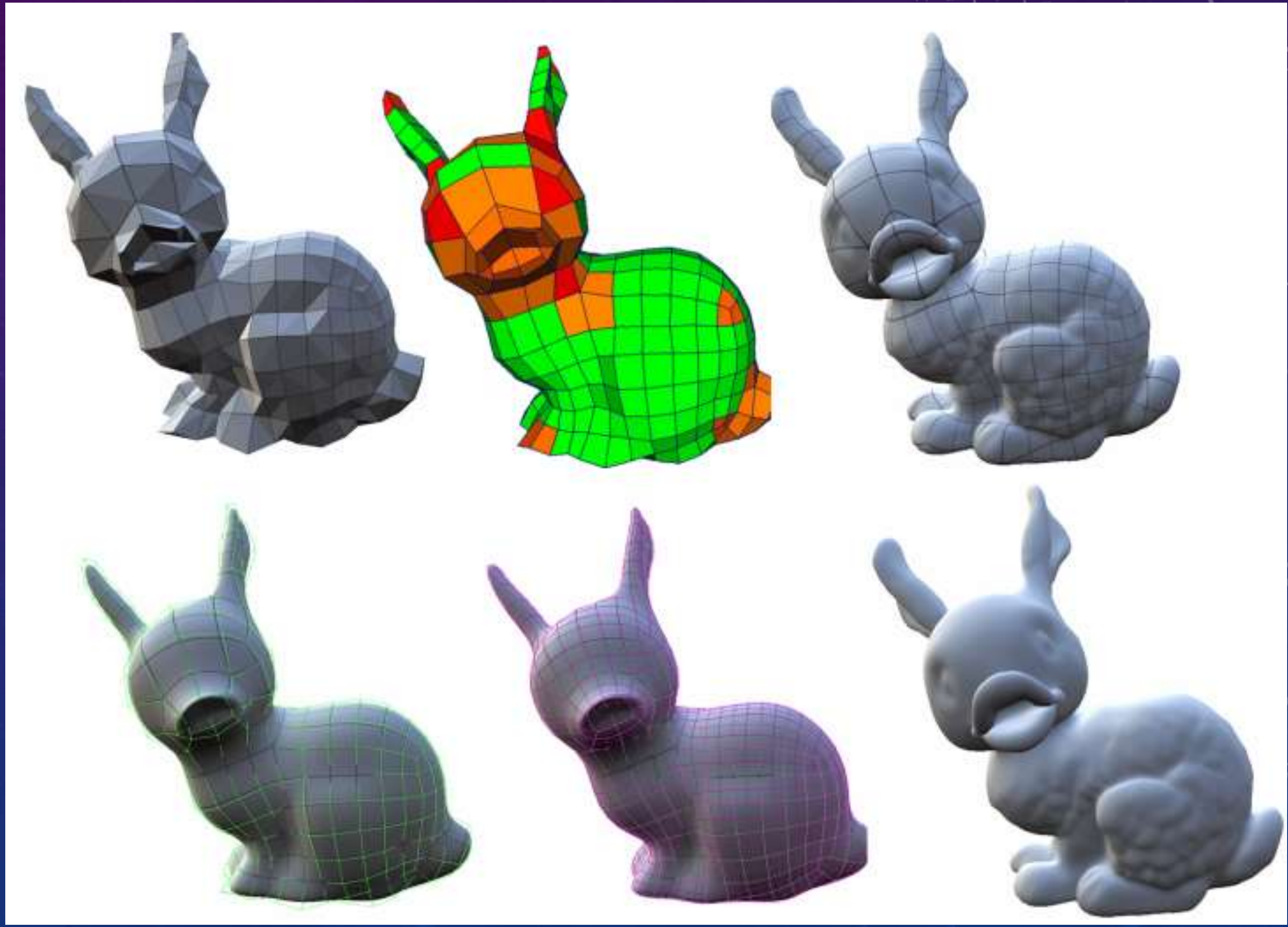


Figure 1: Cube maps can be used to seamlessly texture map an apple (left). In this case, the 3D texture domain T_3 is the surface of a single cube that is immersed in the 3D texture space $\mathbb{T}^3$ (middle) and corresponds to a 2D texture domain T_2 that consists of six square images (right).

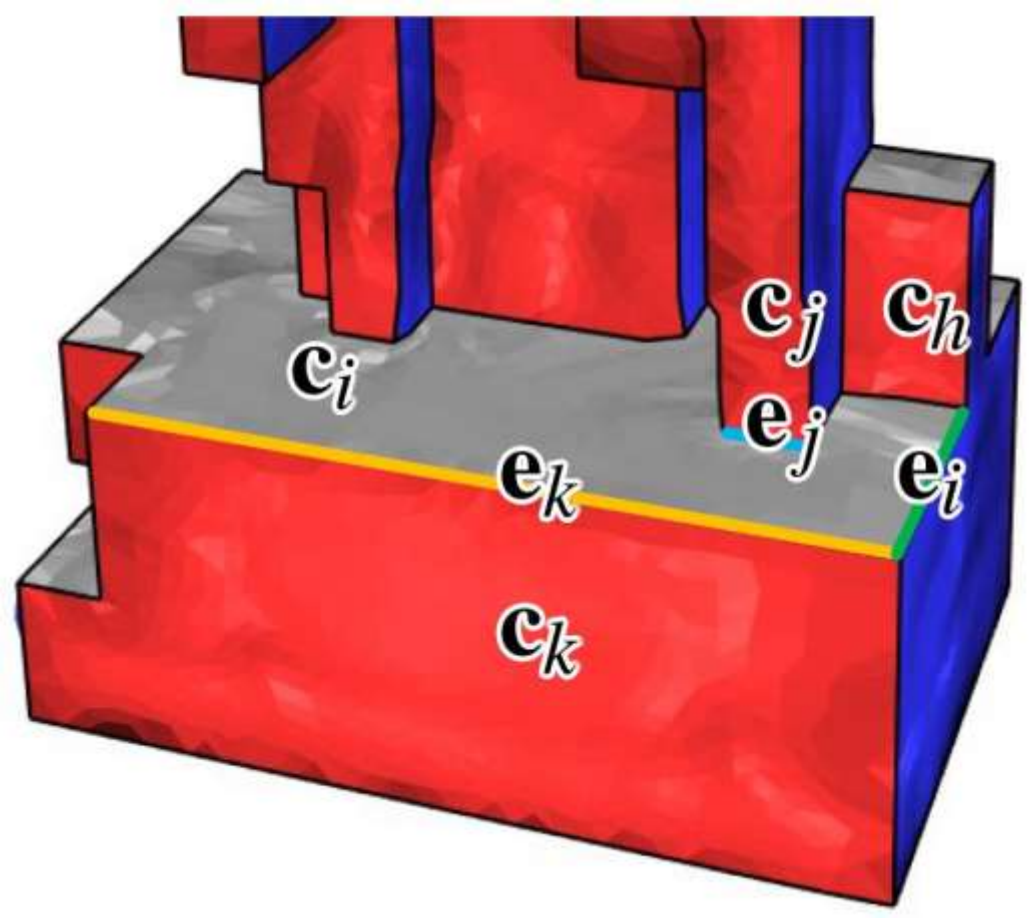
Applications

- All-hex meshing
- Texture Mapping
- GPU-based subdivision



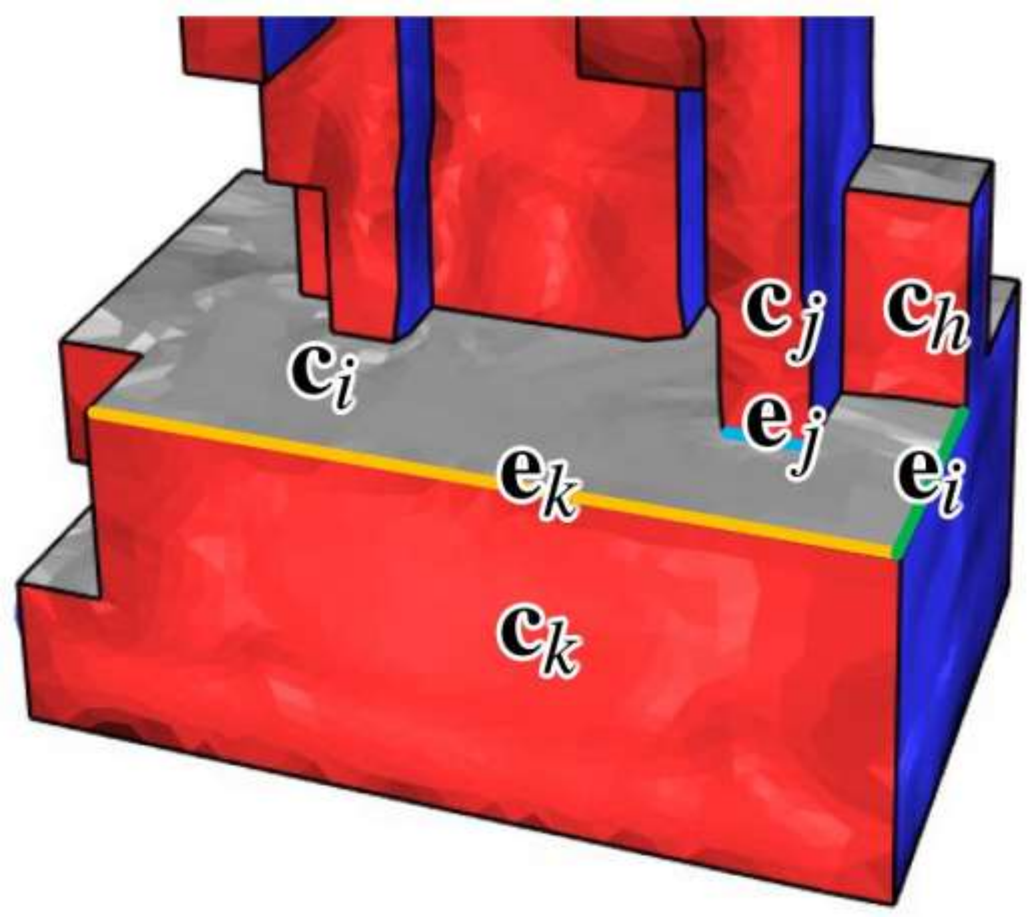
PolyCube facet, edge, and vertex

- PolyCube facet: share the same label
- PolyCube edge: the edges between facets
- PolyCube vertex: sharing by at least three charts



Sufficient topological conditions

- Any PolyCube facet should have at least four neighboring PolyCube facets.
- Any two neighboring PolyCube facets should not have opposite labels such as $+X$ and $-X$.
- The valence of each PolyCube vertex is three.

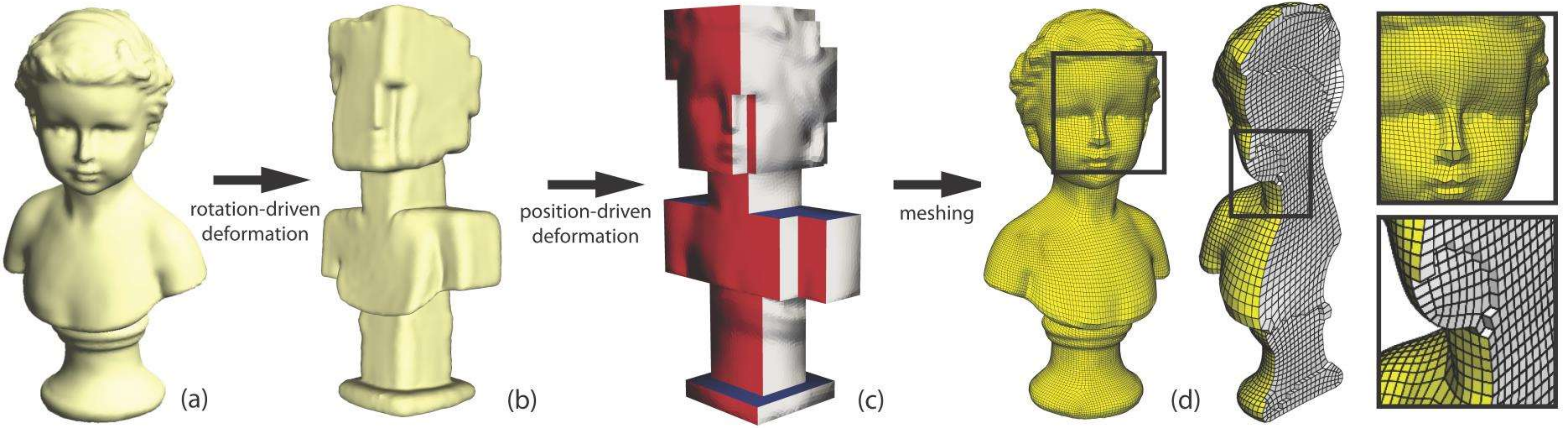


Methods

- Deformation-based method
  - All-Hex Mesh Generation via Volumetric PolyCube Deformation
- Cluster-based method
  - PolyCut: Monotone Graph-Cuts for PolyCube Base-Complex Construction
- Voxel-based method
  - Optimizing PolyCube domain construction for hexahedral remeshing

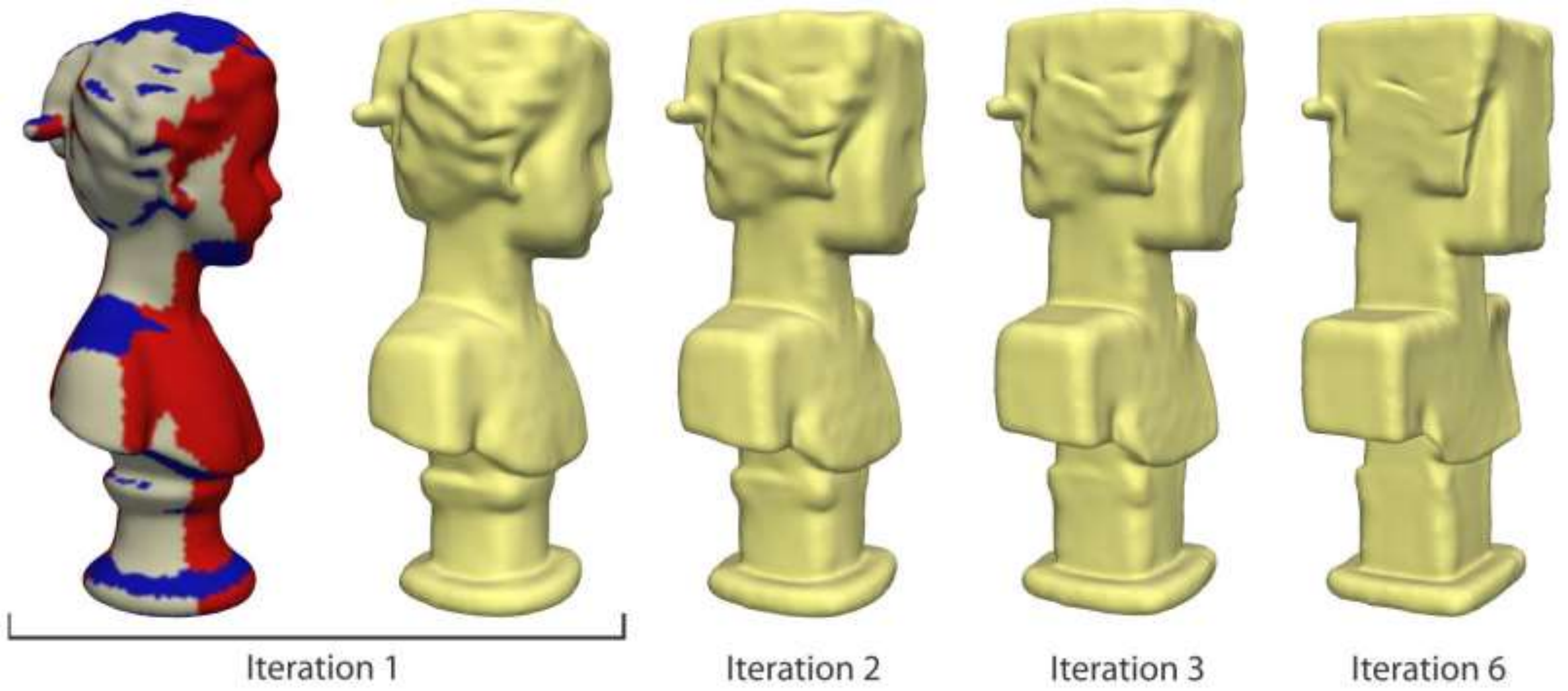
Deformation-based method

- All-Hex Mesh Generation via Volumetric PolyCube Deformation



Rotation-driven deformation

- Goal: gradually aligns the model's surface normals with one of the six global axes, preserving shape as much as possible.



Rotation-driven deformation

- As-Rigid-As-Possible deformation

$$E(R, p') = \sum_i w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i (p_i - p_j) \right\|^2$$

- How to determine R_i?
  - No local step
  - Rotations are determined by axis-alignment constraints

Determine R_i

- For every surface vertex (except those on sharp features), the minimal rotation necessary to align each surface vertex normal with one of $\pm X, \pm Y, \pm Z$.
- Smoothly propagate to feature and interior vertices. Laplace equation per quaternion component.
- Solve E by least squares.

Labeling

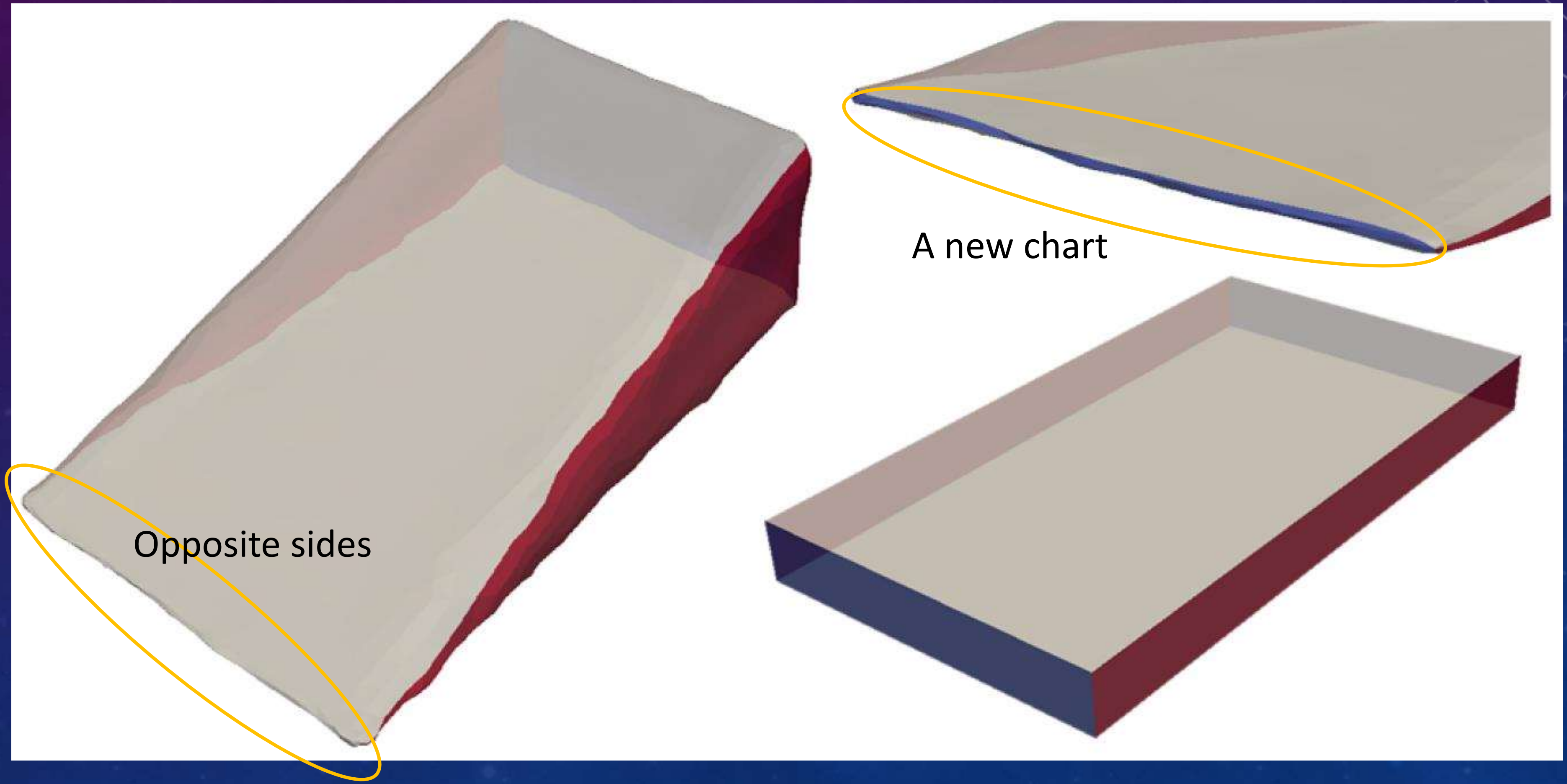
1. Label surface triangles according to the closest axis

2. Group similarly labeled triangles into charts.

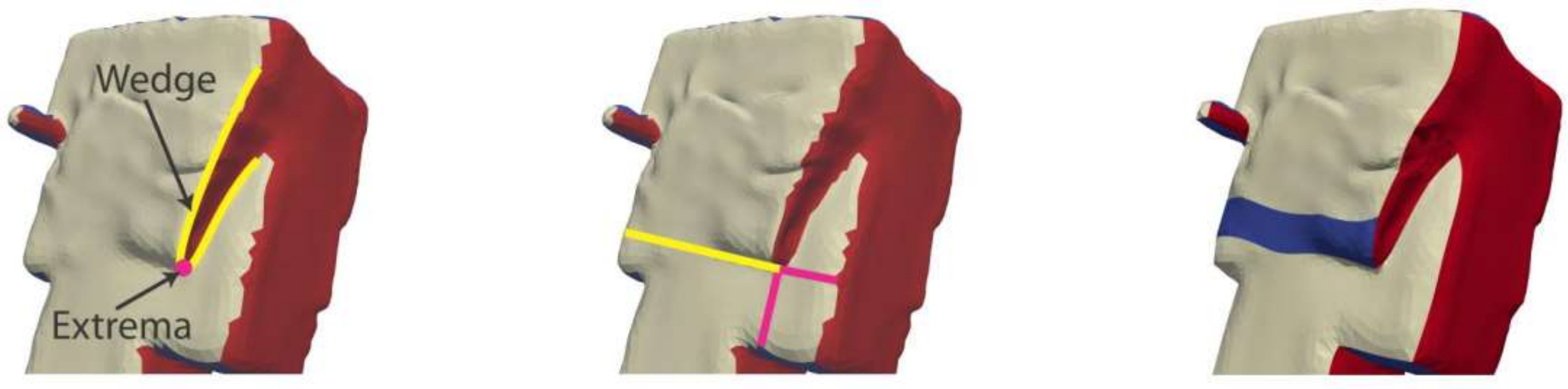
3. Straighten chart boundaries.

4. Remove small, spurious charts bounded by at most two edges

Multi-orientation chart



Highly non-planar chart



Detect extrema along the chart boundary

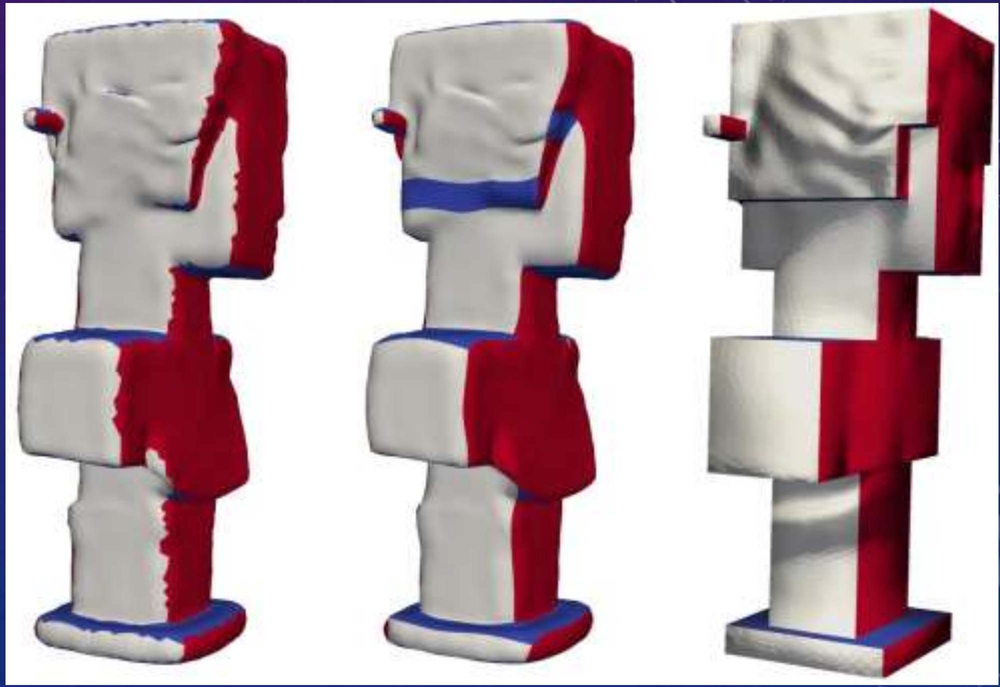
Three possible axis-aligned cut options

Valid cuts are defined as those that would not introduce new charts with three or fewer neighbors

Position-driven deformation

$$\sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2$$

- Constrain each chart to an axis-aligned plane
- Soft distance preservation energy

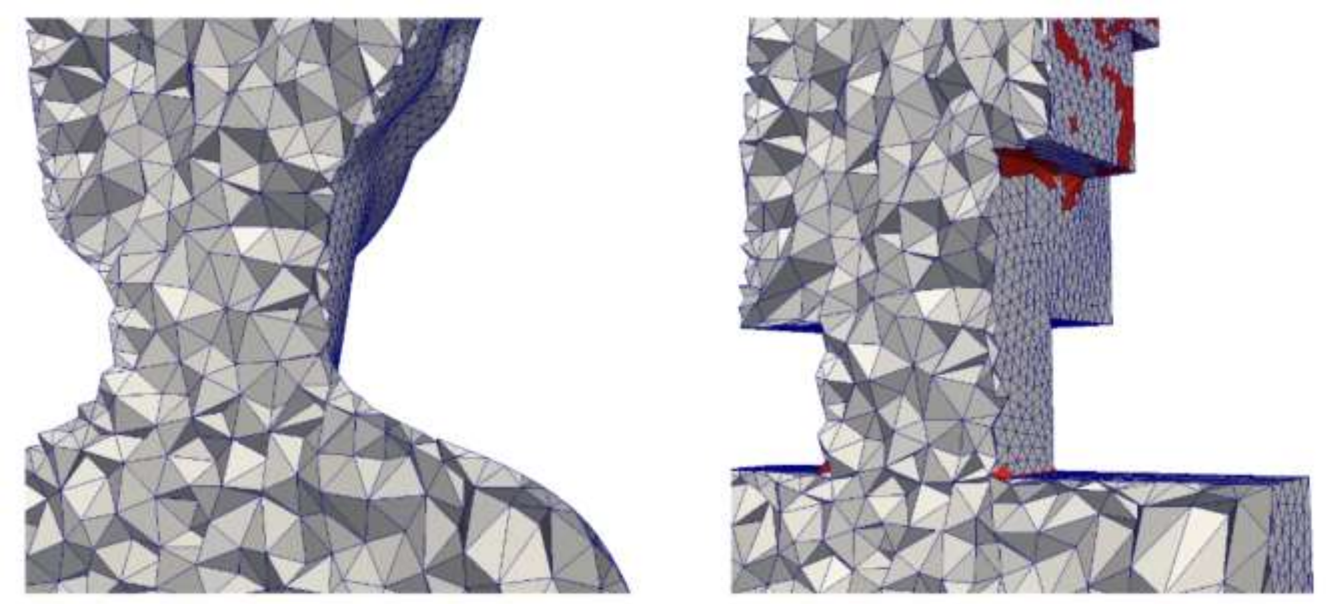


Discussions

- Inverted tet
- More papers

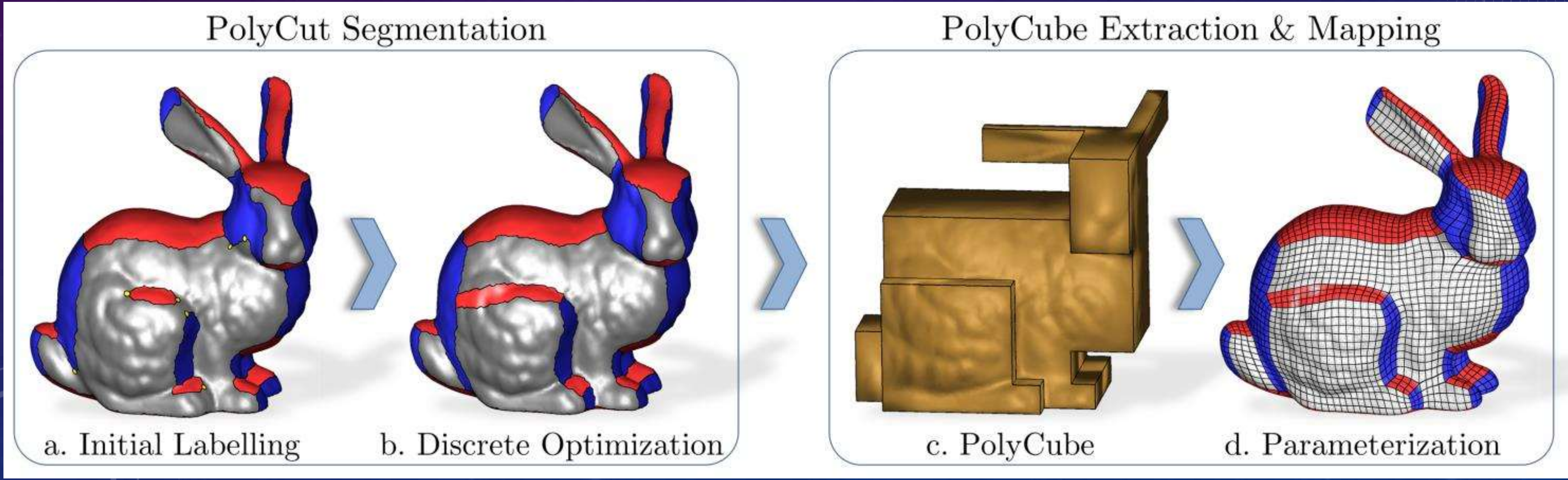
$L1$ -based Construction of Polycube Maps from Complex Shapes (2014)

Efficient Volumetric PolyCube-Map Construction (2016)



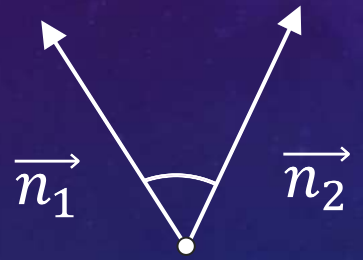
Cluster-based method

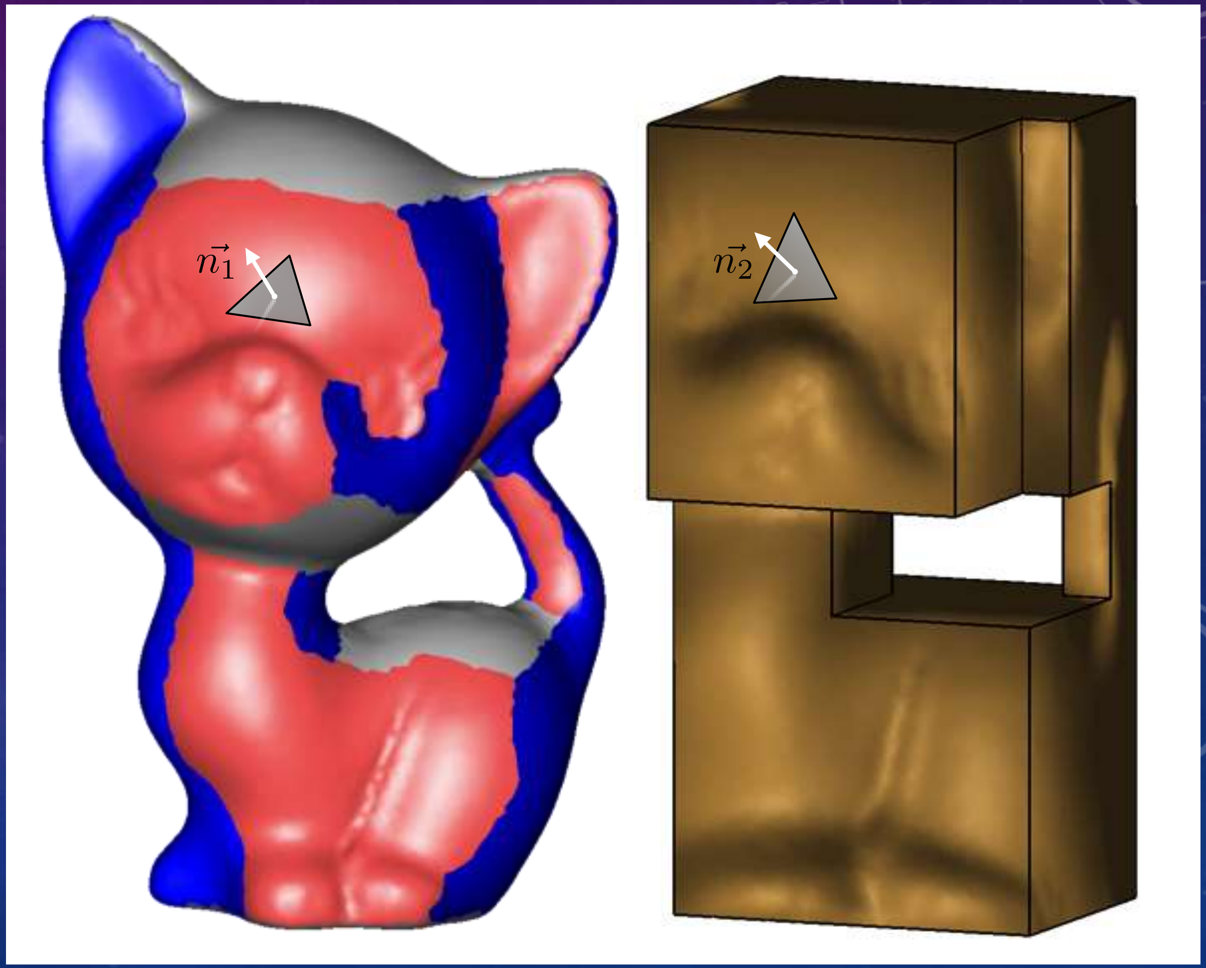
- PolyCut: Monotone Graph-Cuts for PolyCube Base-Complex Construction



Cluster

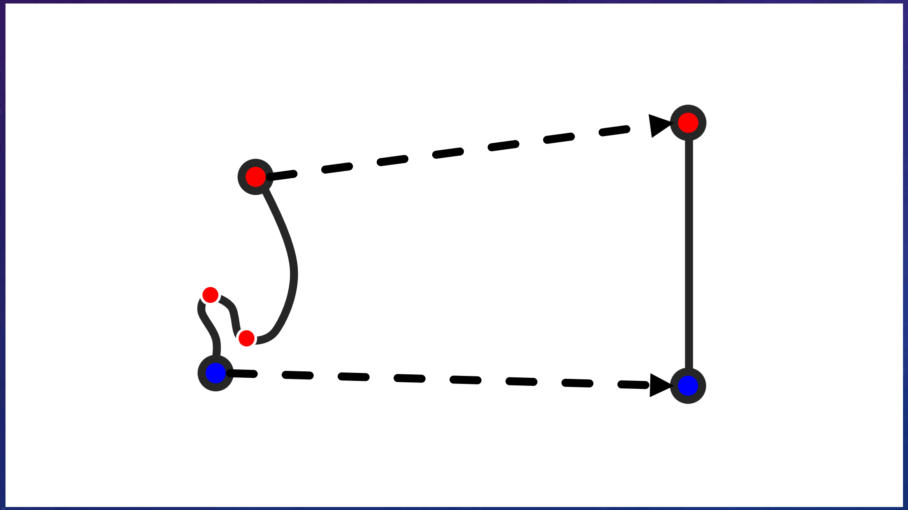
- Angular distance - distortion

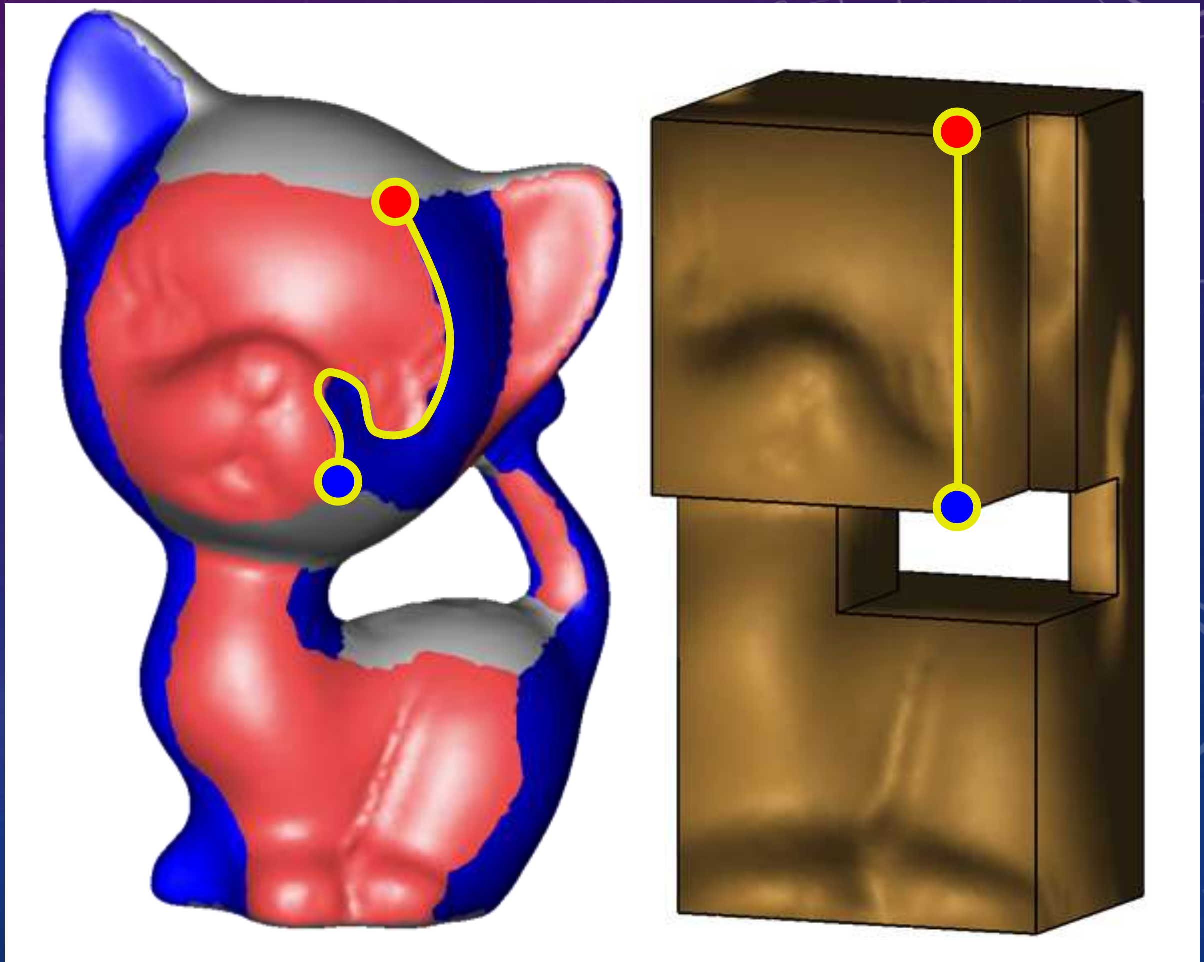




Cluster

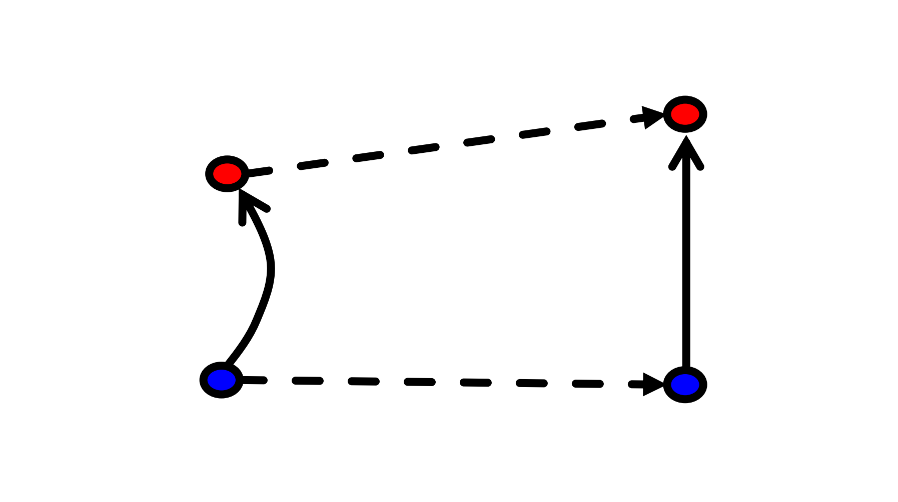
- Angular distance – distortion
- Monotonicity

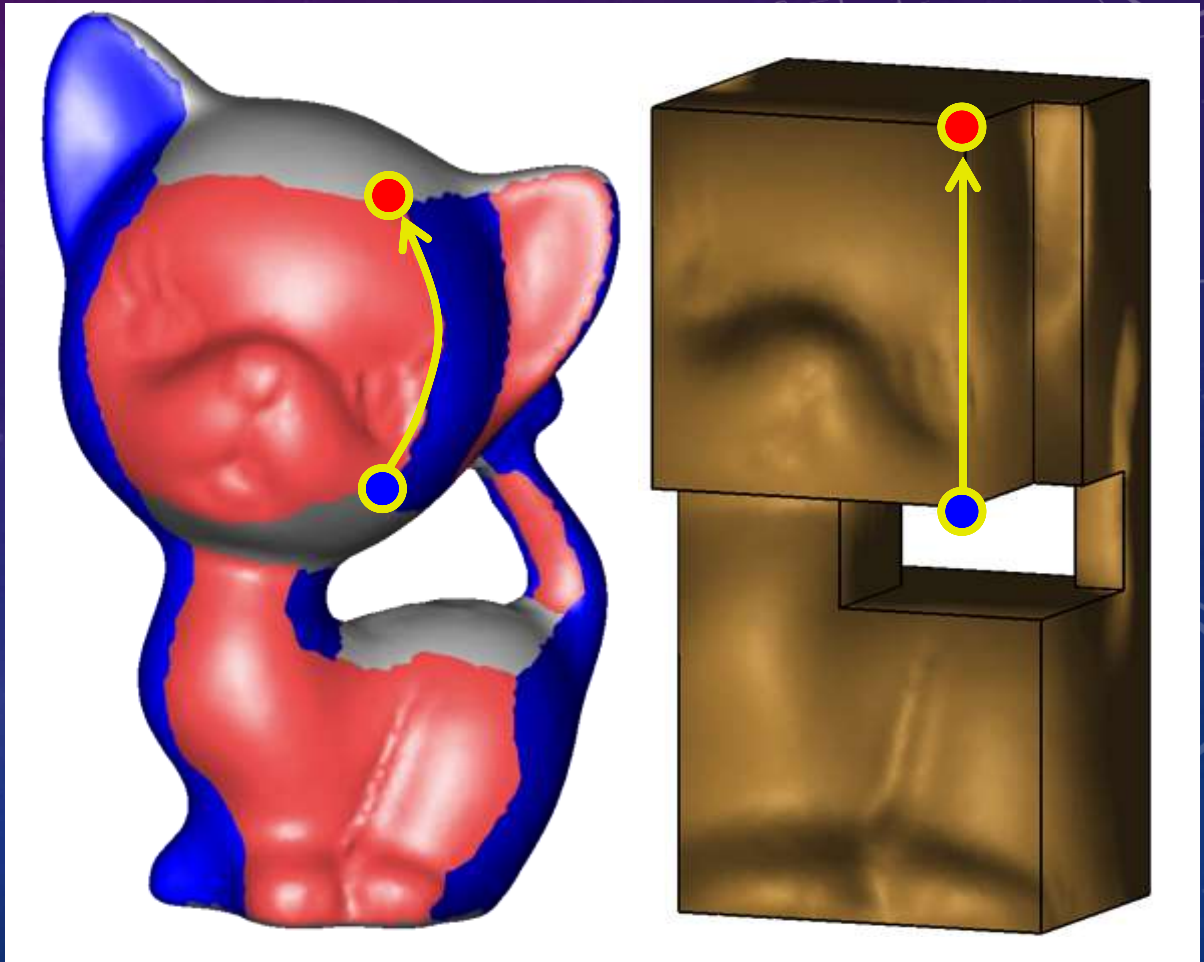




Cluster

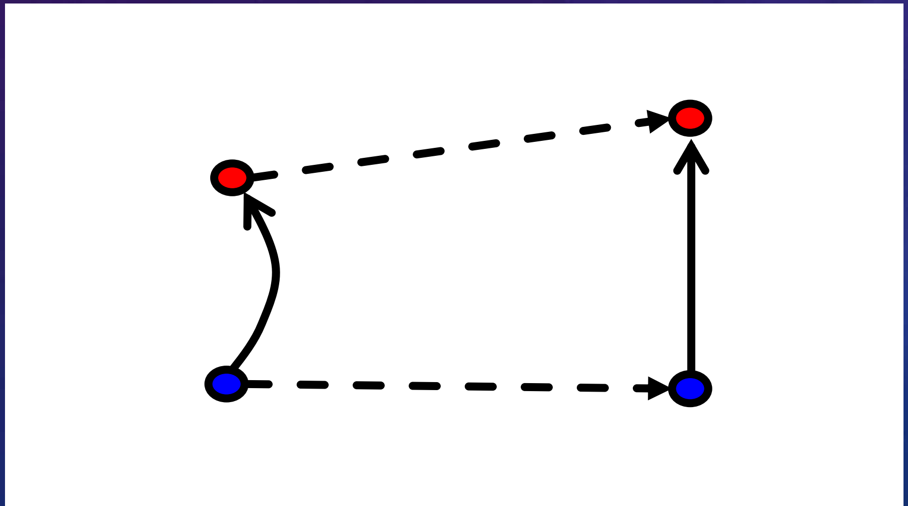
- Angular distance – distortion
- Monotonicity

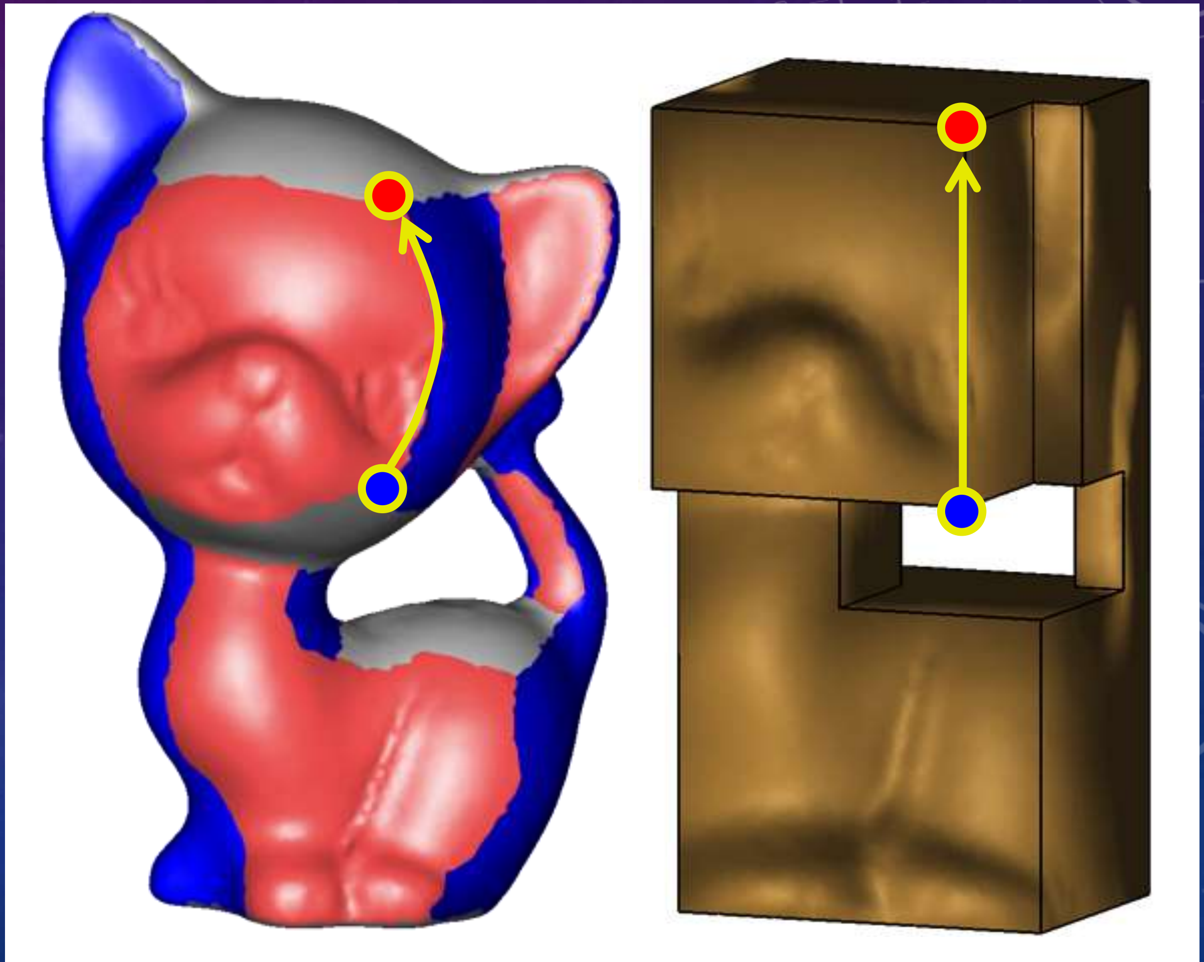




Cluster

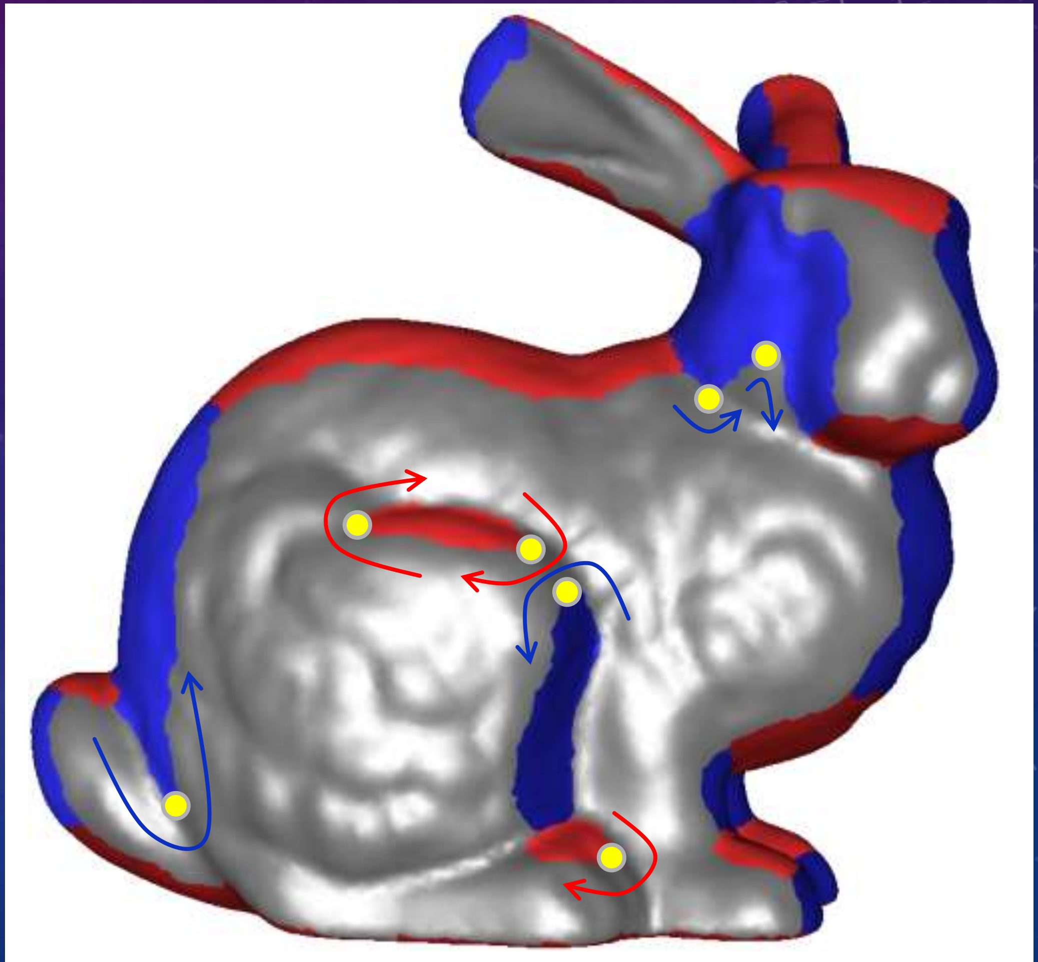
- Angular distance – distortion
- Monotonicity





Monotonicity

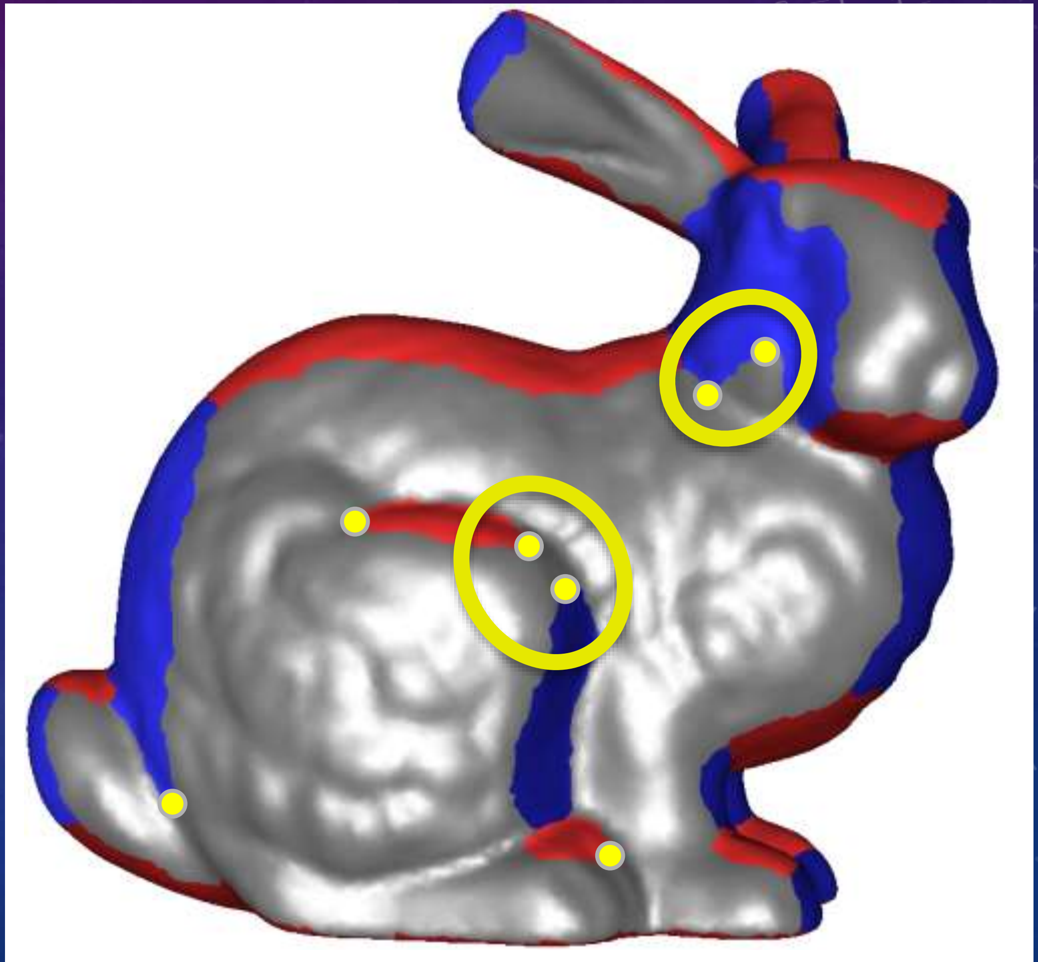
- Monotonicity requires global constraints that we cannot plug into our energy term…



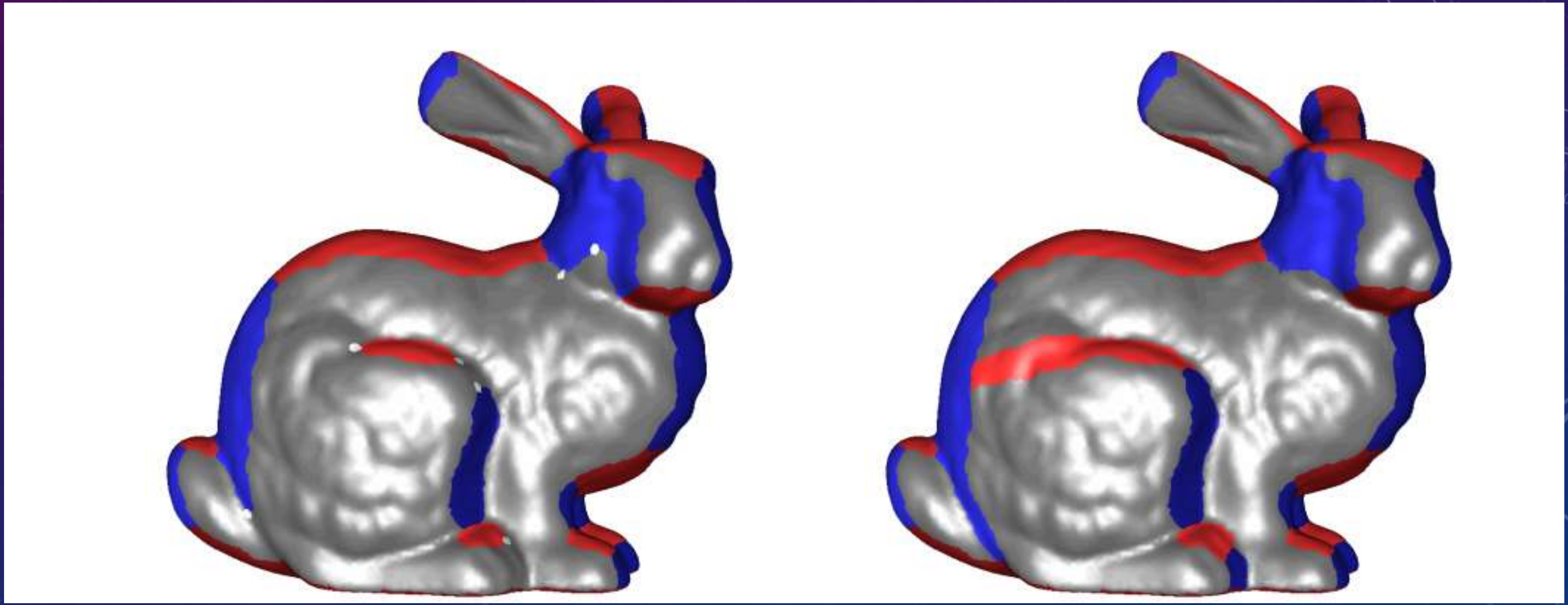
Monotonicity

- Monotonicity requires global constraints that we cannot plug into our energy term…
- Hill Climbing

Explore the space of segmentations to find the closest fully monotone labeling…



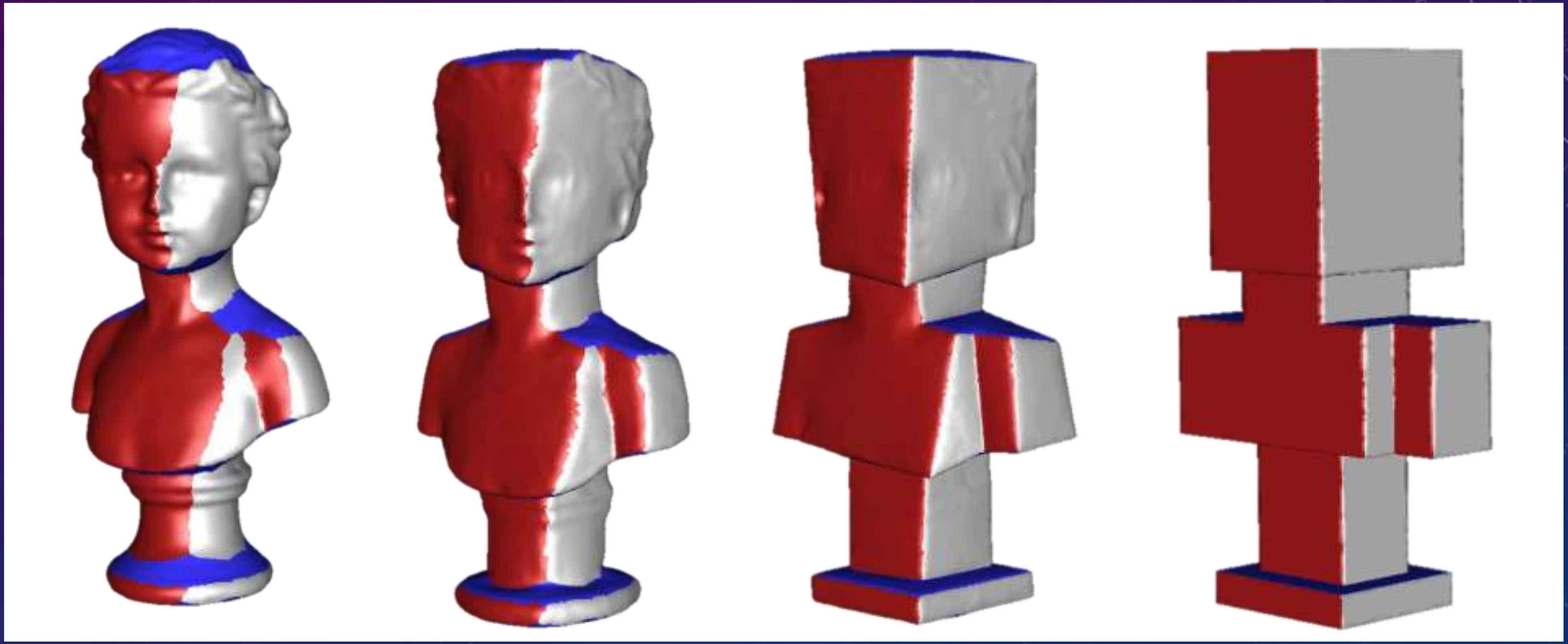
Monotonicity



Before
Hill Climbing

After
Hill Climbing

PolyCube deformation



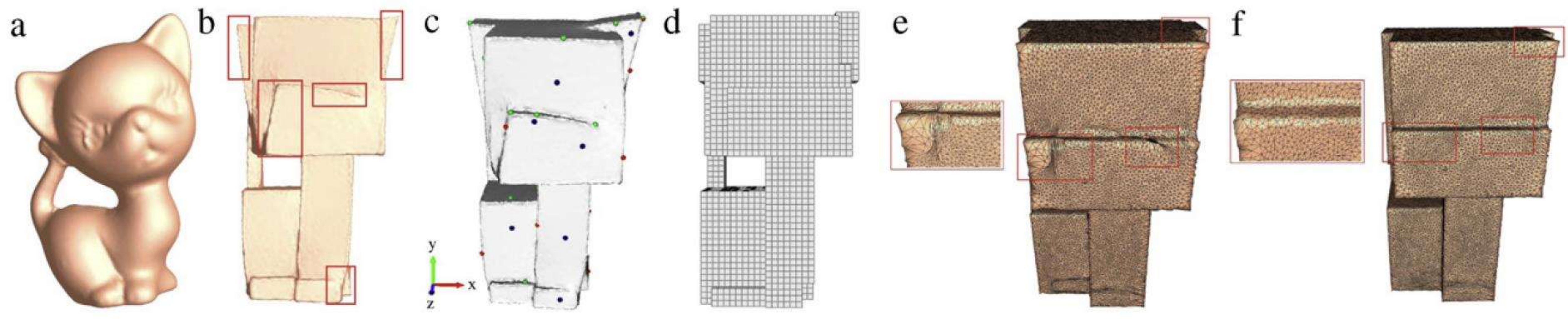
Initial Deformation

Gradual Deformation

Final PolyCube

Voxel-based method

- Optimizing PolyCube domain construction for hexahedral remeshing



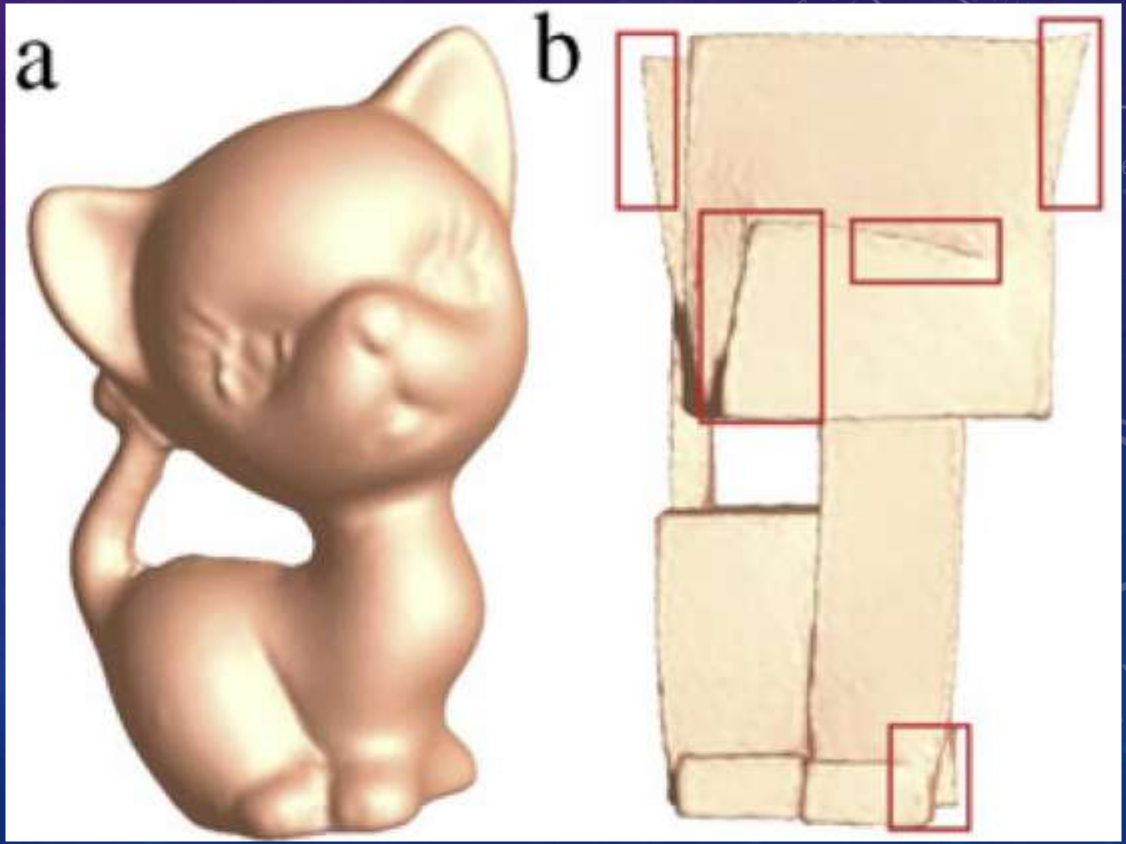
Pre-deformation

PolyCube construction and optimization

Mapping computation

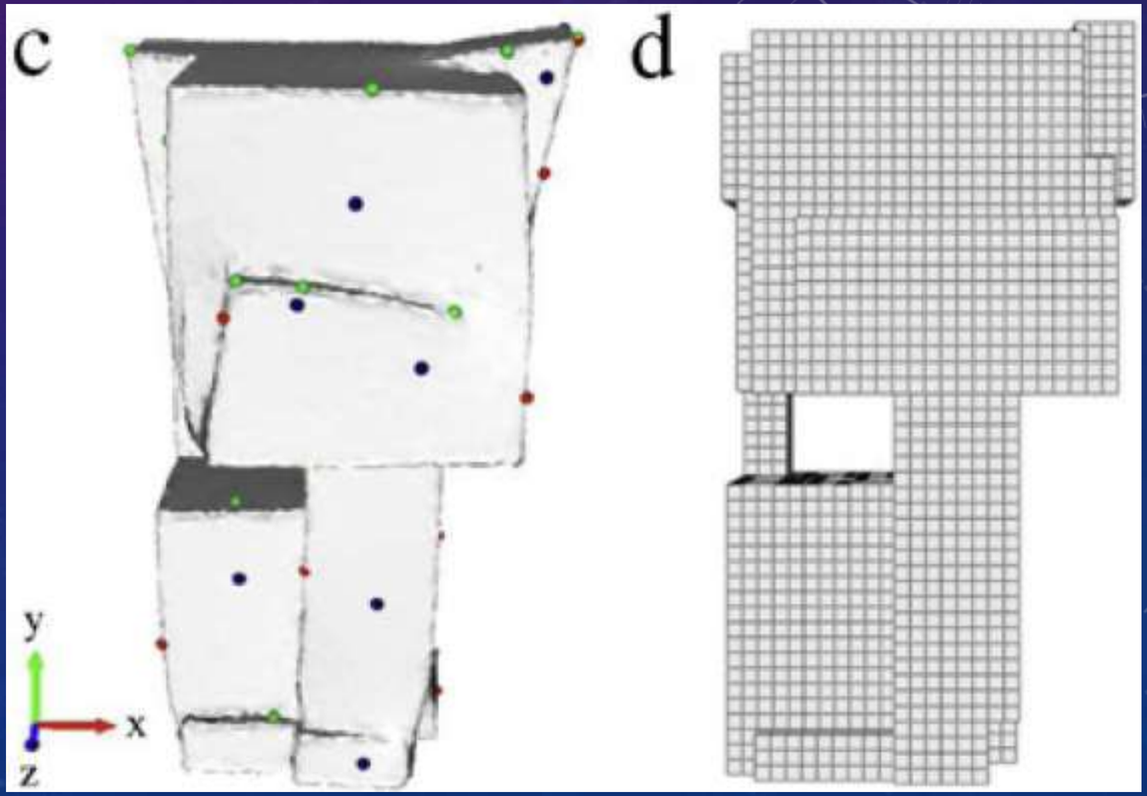
Motivation

- Wedge regions are hard to avoid



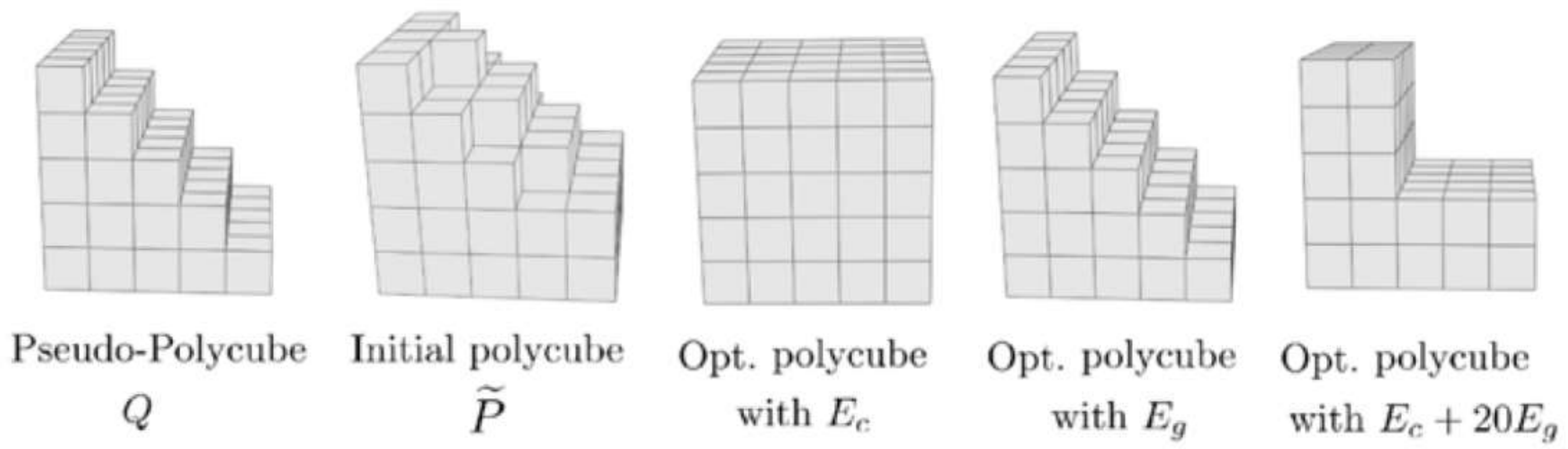
Motivation

- Wedge regions are hard to avoid
- PolyCube construction – length of cube
  - Topology preservation
  - Close to deformed shape



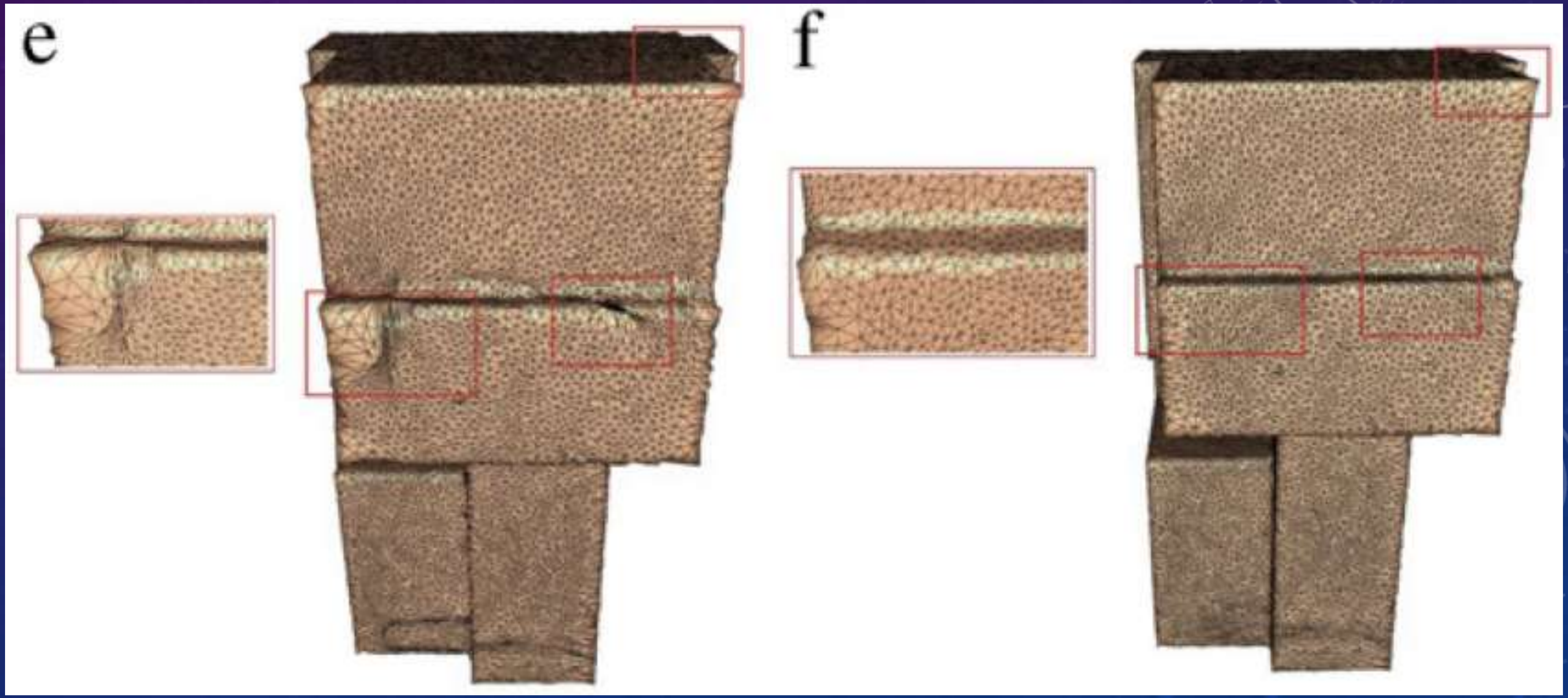
Optimization

- Corner number → Domain simplicity E_c
- Geometric deviation E_g



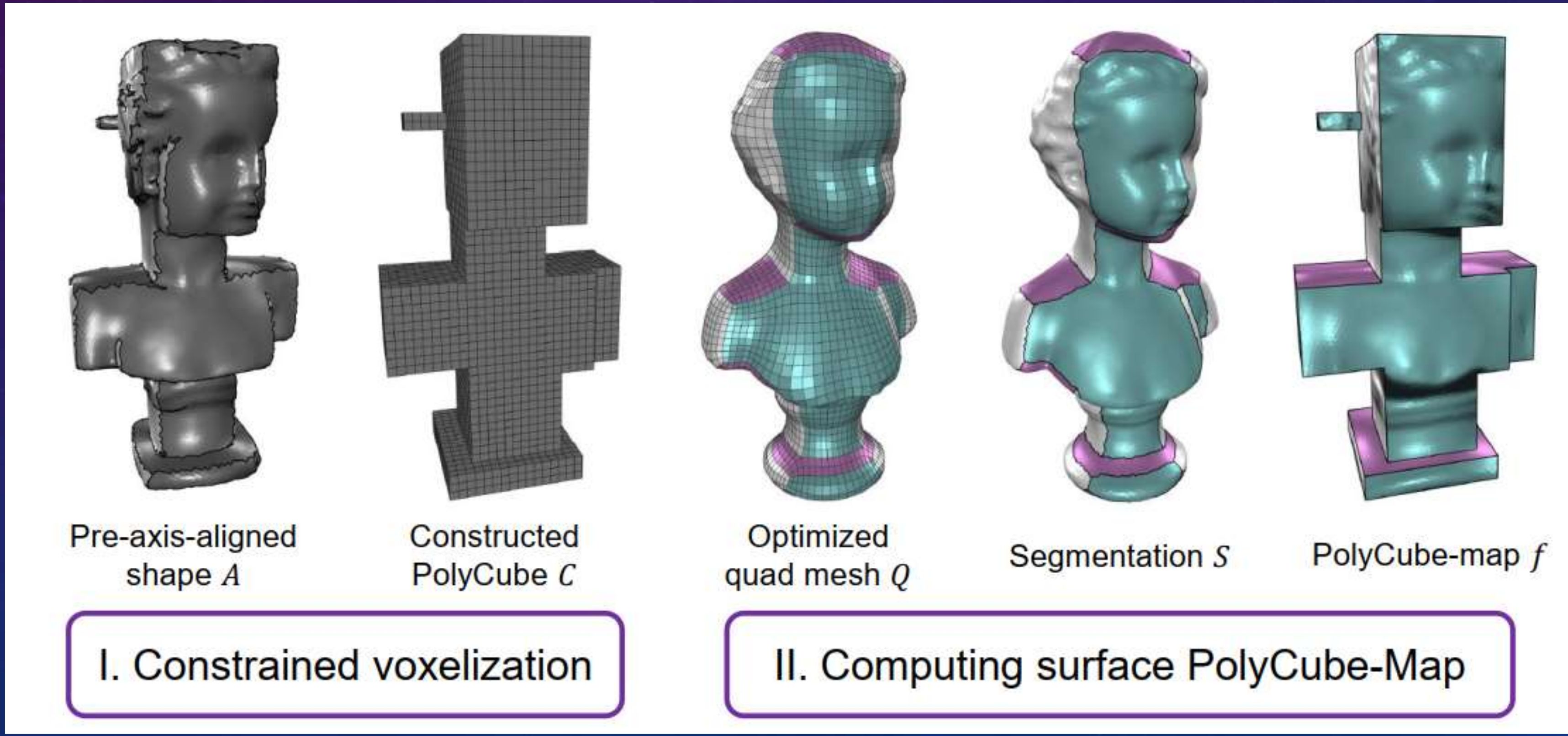
Mapping computation

- Projection
- Fixed boundary mapping



More papers

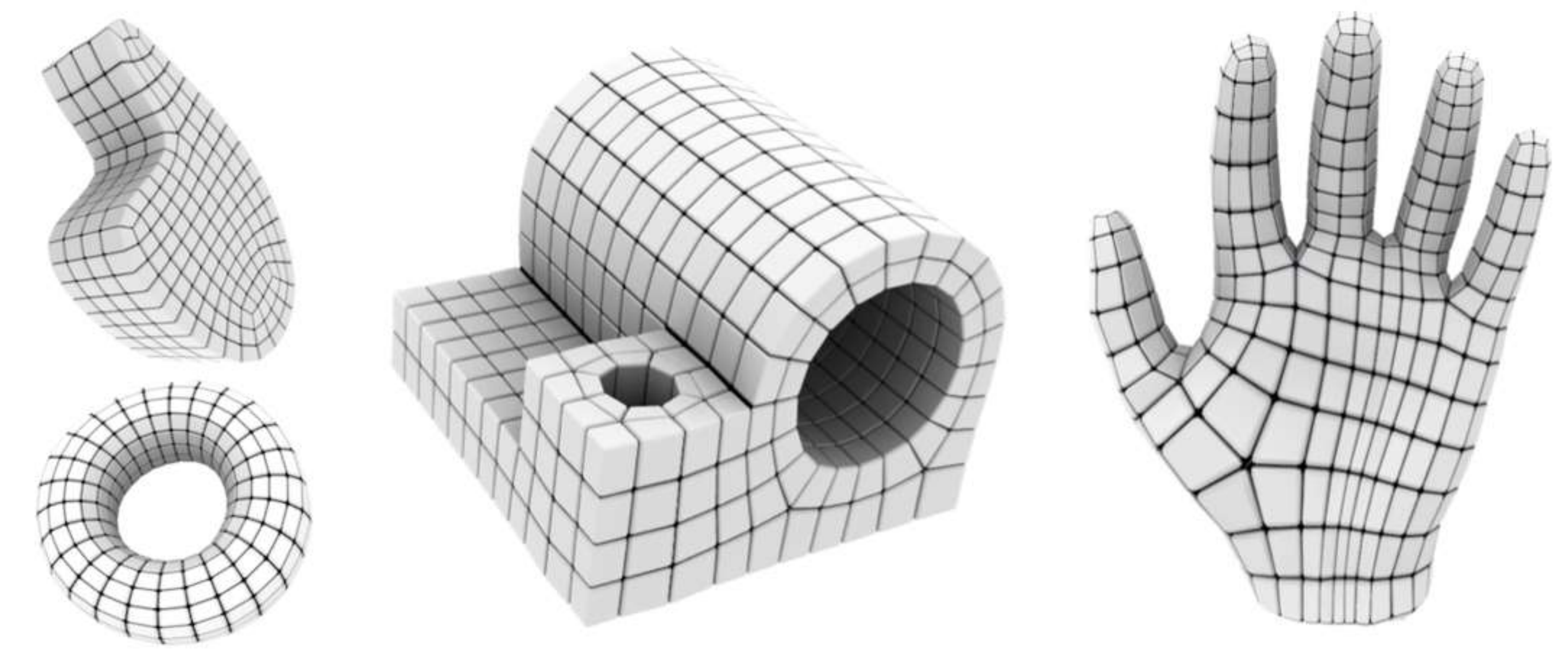
- Computing Surface PolyCube-Maps by Constrained Voxelization (PG2019)



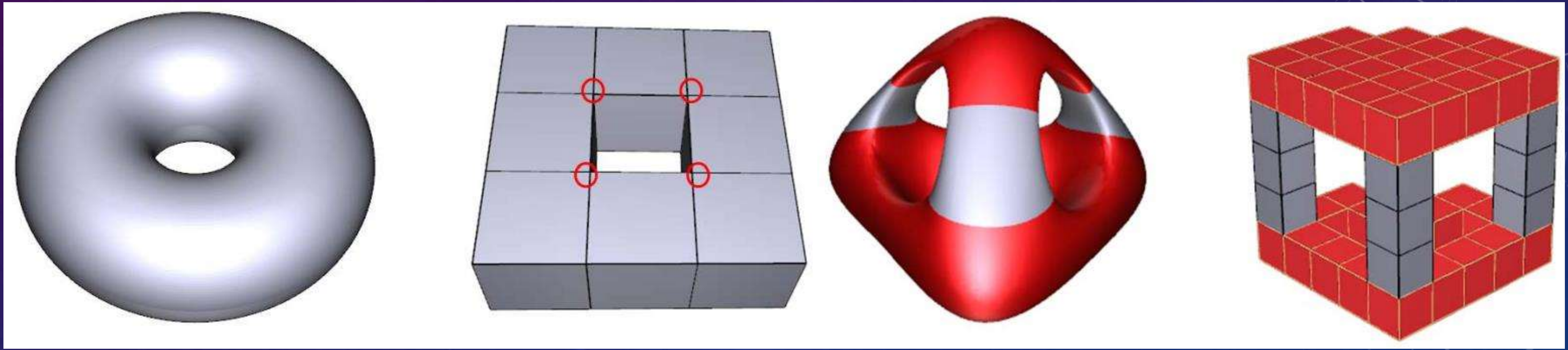
Generalized PolyCube

- Conventional PolyCube:
  - A shape composes of axis-aligned unit cubes that abut with each other.
  - Unit cubes as the building block.
  - All cubes are glued together and embedded in the 3D space.
- Generalized PolyCube:
  - A shape composes of a set of cuboids glued together topologically.
  - Topological simplicity and elegance

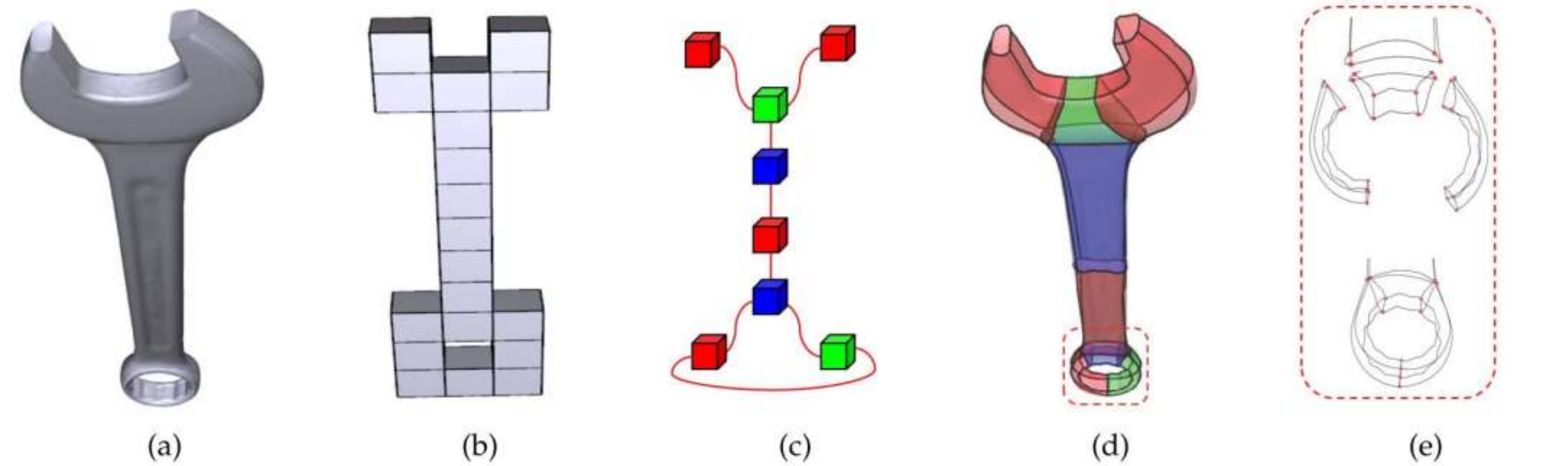
Generalized PolyCube



Comparisons



Thinking from topology



Generalized poly-cubes: (a) The wrench model; (b) The conventional poly-cube (CPC); (c) The generalized poly-cube (GPC) as a topological graph; (d-e) The cuboid edges are overlaid onto the model to visualize the GPC global structure.

Frame field

- All-Hex Meshing using Closed-Form Induced Polycube

Figure 3: *Pipeline of our algorithm. From the left to the right are input mesh, cut faces, frame field, deformed cut mesh, polycube parametrization and final hexahedral mesh.*