Robust Hex-Dominant Mesh Generation using Field-Guided Polyhedral Agglomeration

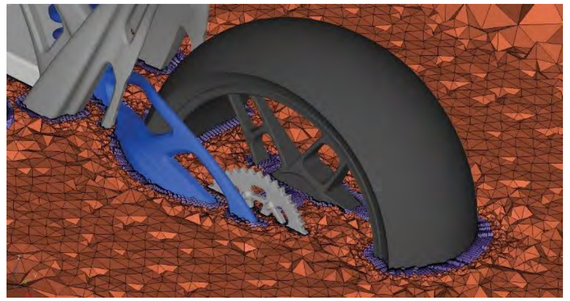
Xifeng Gao — New York University

Wenzel Jakob — École Polytechnique Fédérale de Lausanne (EPFL)

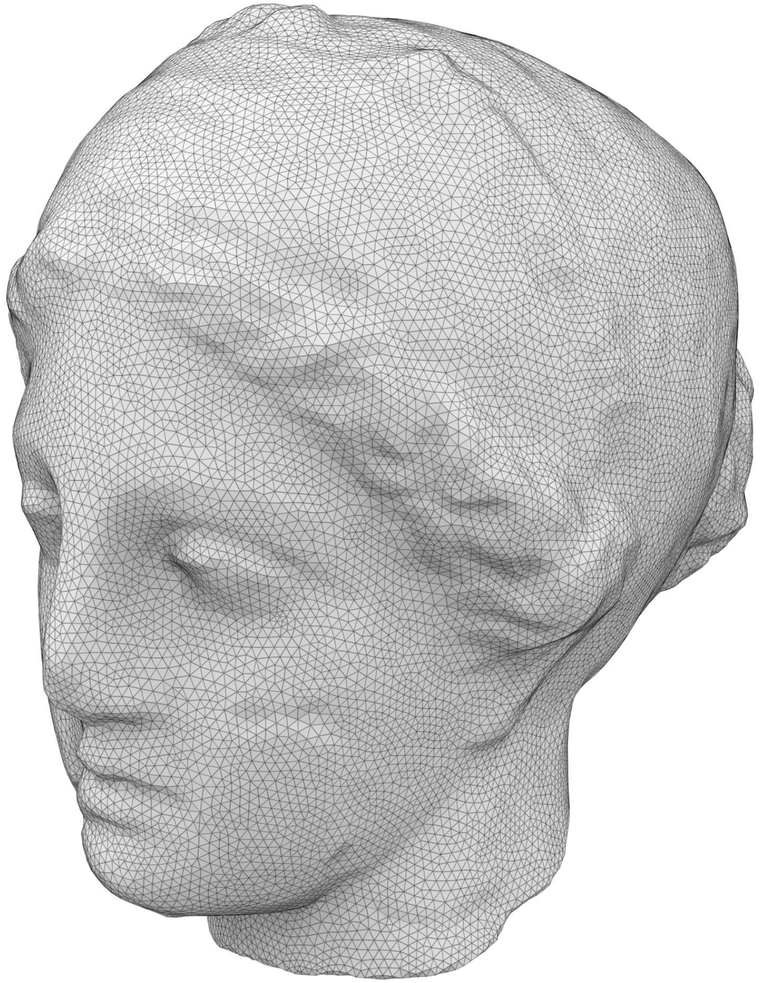
Marco Tarini — Università dell'Insubria and CNR-ISTI

Daniele Panozzo — New York University

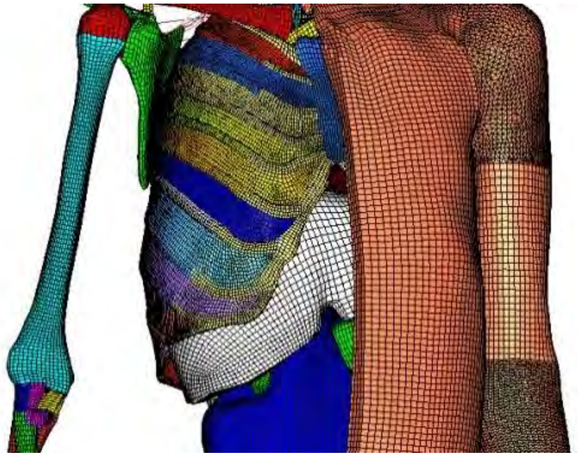
Volumetric meshes



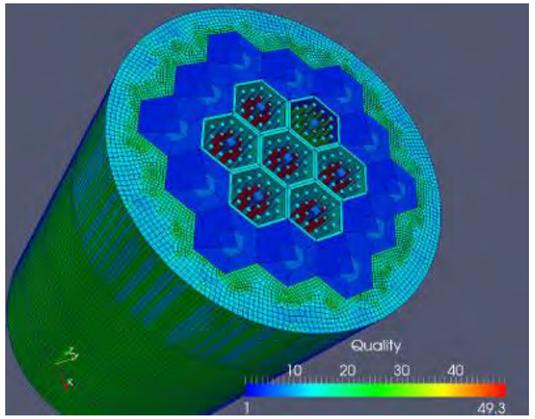
Aerodynamics



Material design



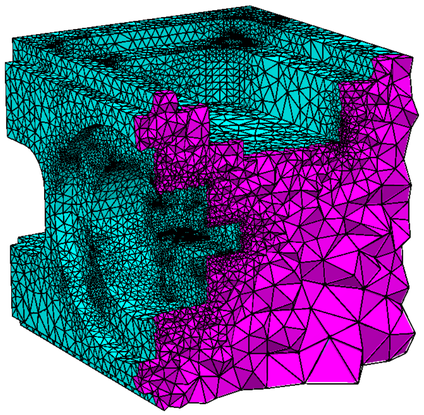
Biomechanics



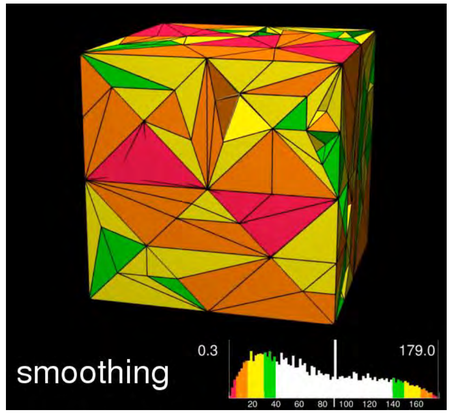
Engineering

Volumetric meshes

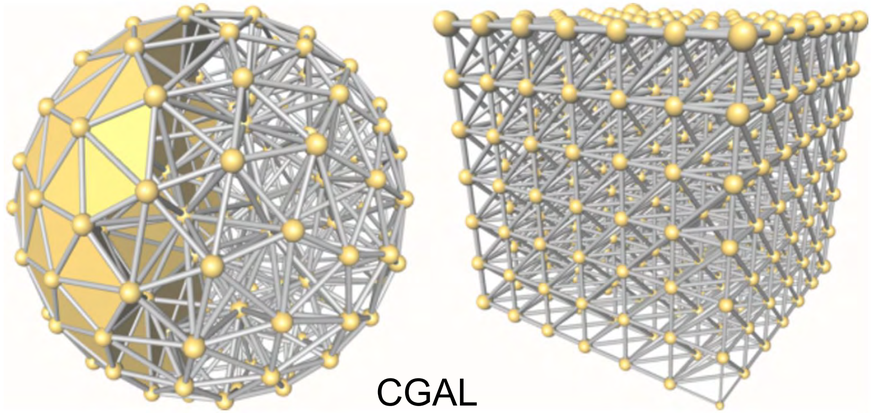
Tetrahedral-Mesh



TetGen

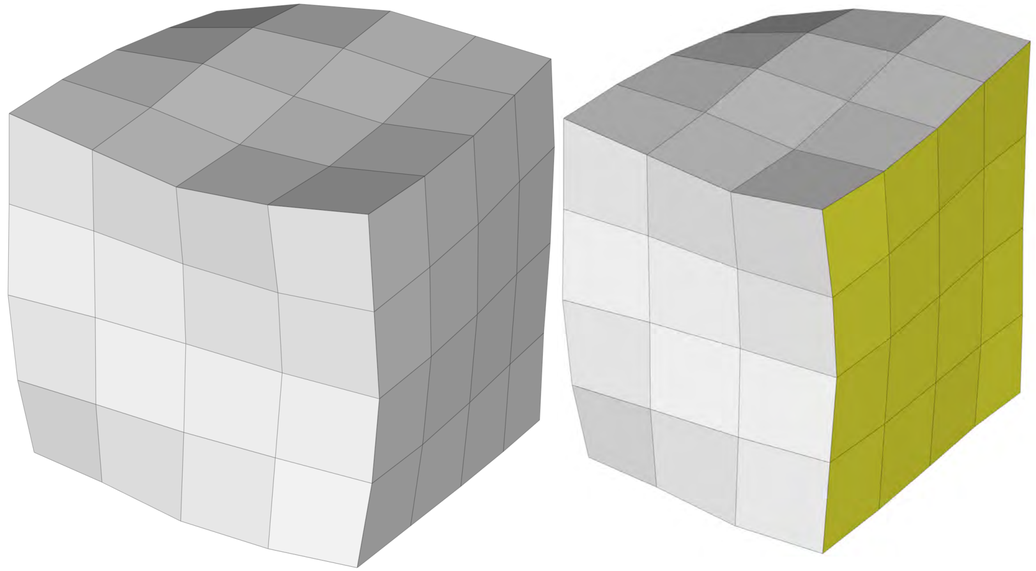


Stellar



CGAL

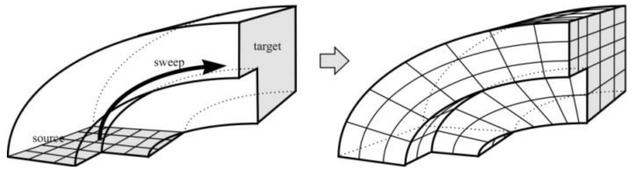
Hexahedral-Mesh



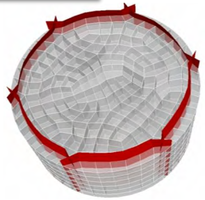
- Fewer elements
- Tensor product structure
- Superior numerical properties

Hexahedral-Meshing

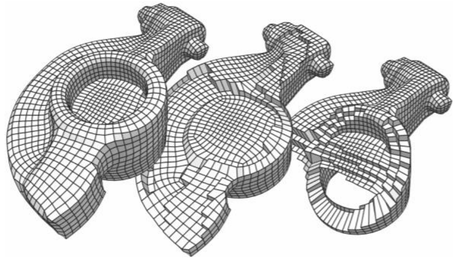
Sweeping, connectivity editing, and advancing



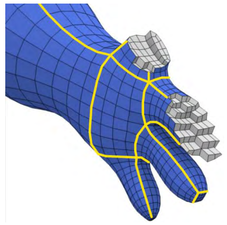
[Owen, et al. 1998]



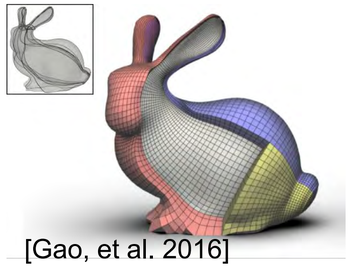
[Shepherd, et al. 2008]



[Kremer, et al. 2013]

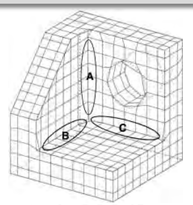


[Livesu, et al. 2016]

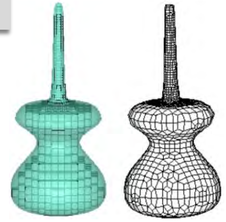


[Gao, et al. 2016]

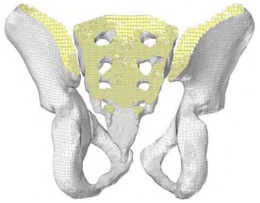
Space decomposition



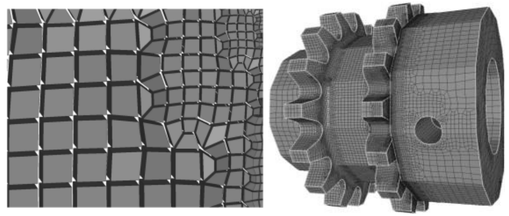
[Su, et al. 2004]



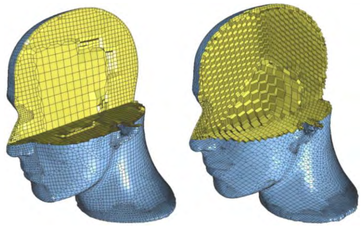
[Zhang, et al. 2007]



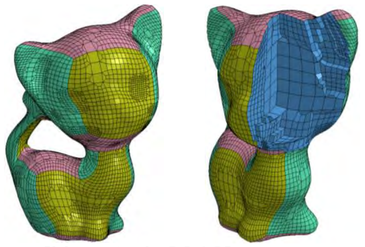
[Ito, et al. 2009]



[Maréchal, et al. 2009]

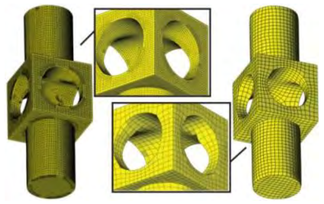


[Zhang, et al. 2013]

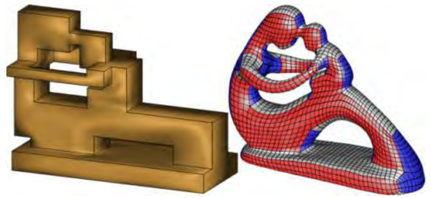


[Hu, et al. 2016]

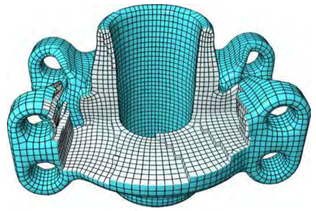
Polycube-based



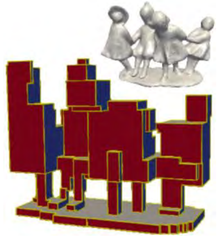
[Gregson, et al. 2011]



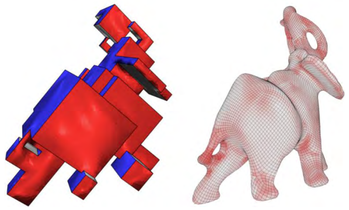
[Livesu, et al. 2013]



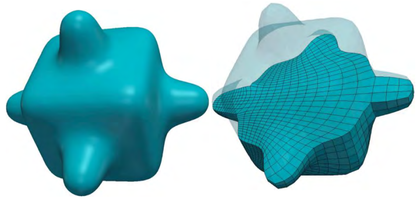
[Fang, et al. 2016]



[Huang, et al. 2014]



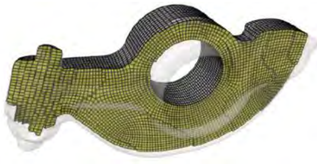
[Fu, et al. 2016]



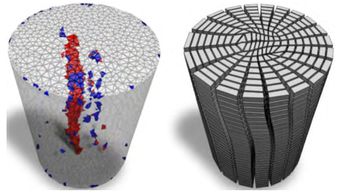
[Xu, et al. 2017]

Orientation field-guided

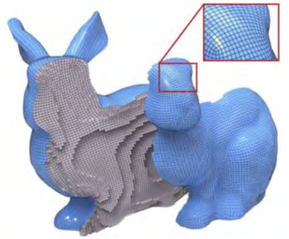
[Jiang, et al. 2013]



[Nieser, et al. 2011]

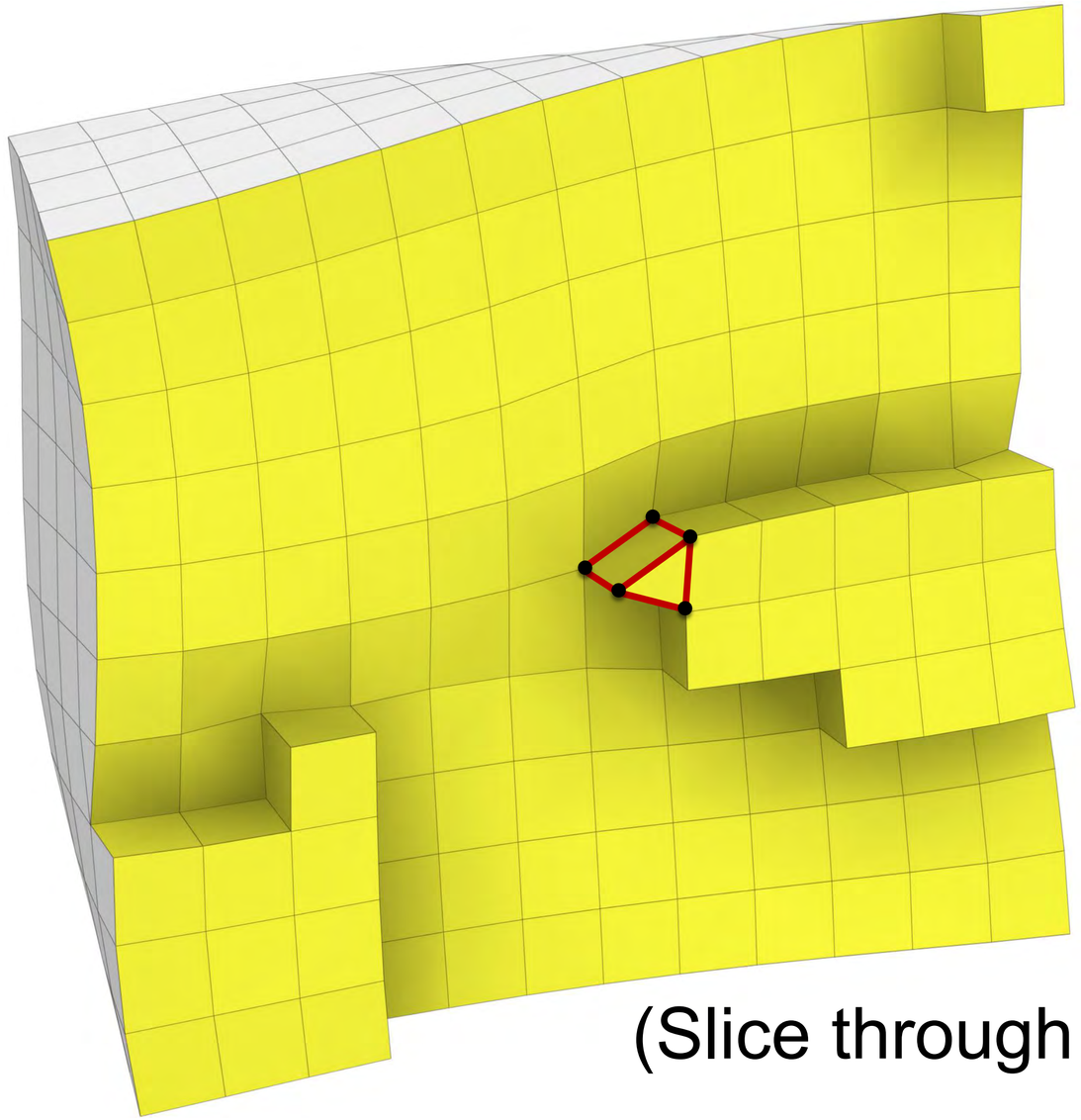


[Lyon, et al. 2016]



[Li, et al. 2012]

Hex-Dominant Mesh

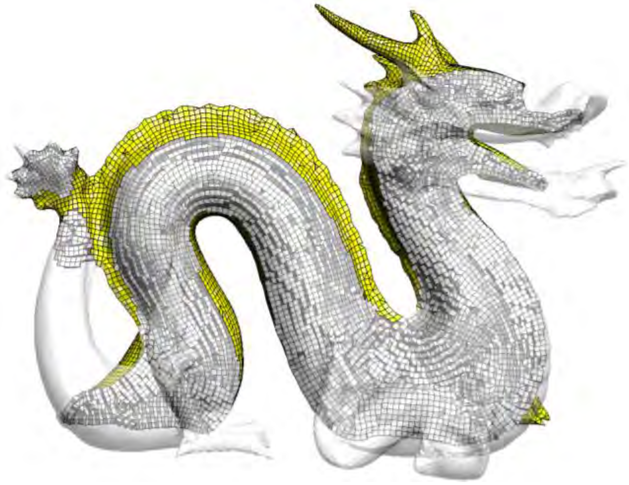


(Slice through the interior)

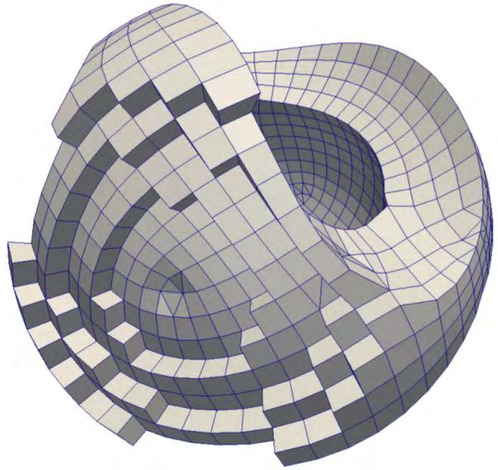
Hex-Dominant Meshing

[Meshkat and Talmor, 2000]

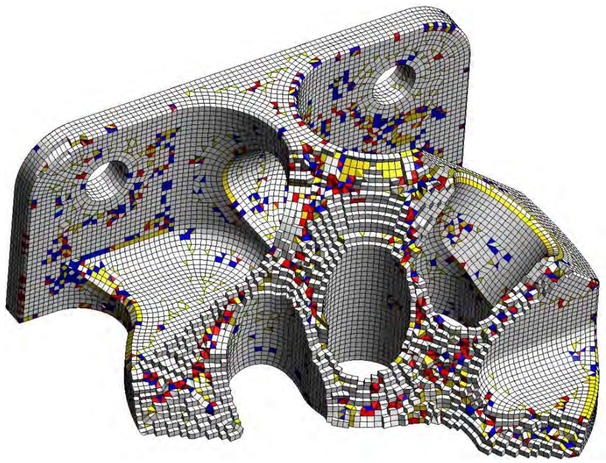
[Yamakawa, et al. 2003]



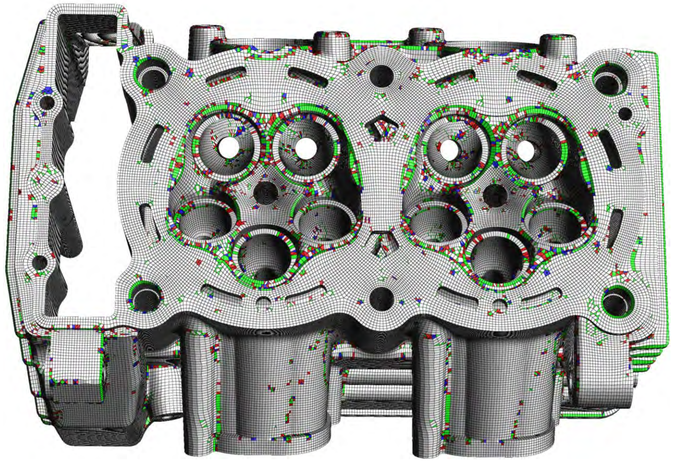
[Lévy, et al. 2010]



[Huang, et al. 2011]

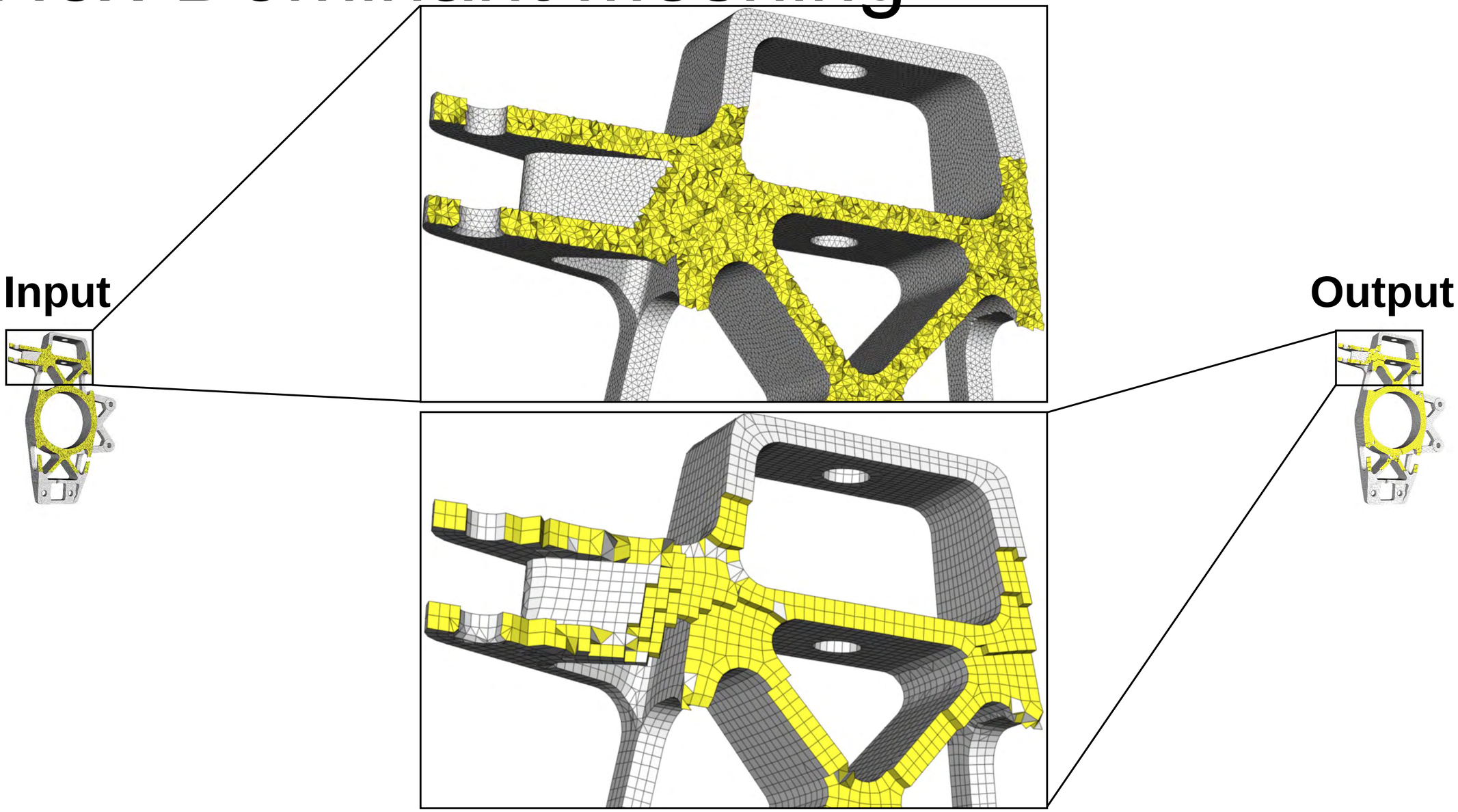


[Baudouin, et al. 2014]



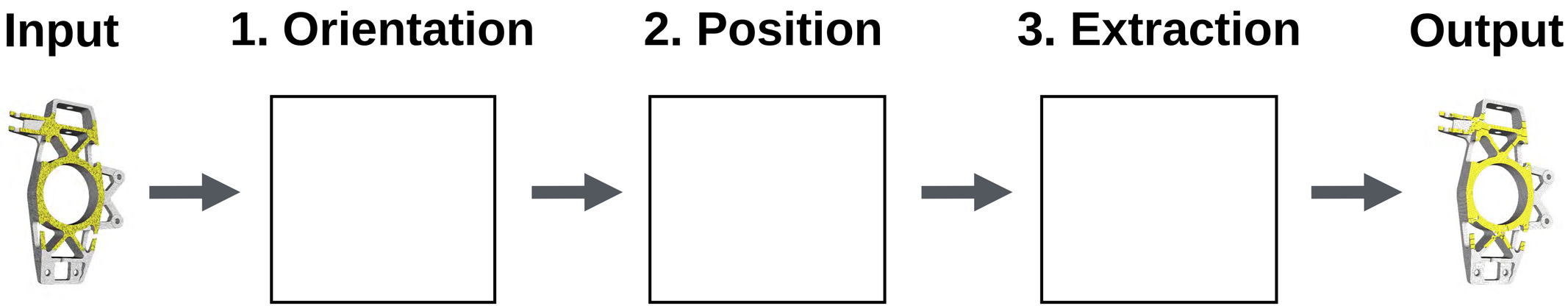
[Sokolov, et al. 2016]

Hex-Dominant Meshing



Hexahedral-dominant meshing, D. Sokolov, N. Ray, L. Untereiner, B. Lévy, 2016, ACM Trans. on Graphics.

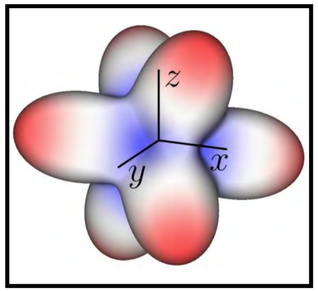
Field Guided Meshing Pipeline



Hex-Dominant Meshing

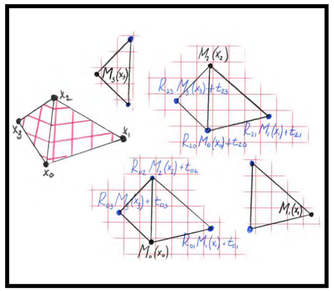
Input

1. Orientation



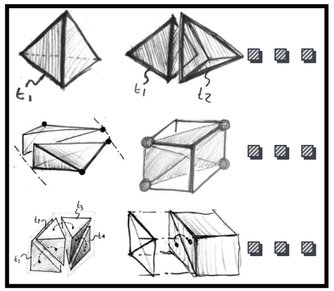
Spherical harmonic functions

2. Position



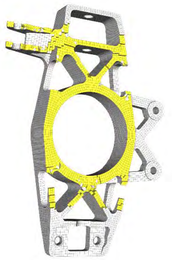
Periodic global parameterization

3. Extraction



Tetrahedral recombination

Output

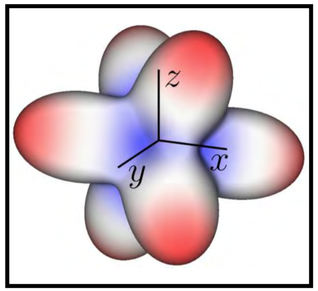


Hexahedral-dominant meshing, D. Sokolov, N. Ray, L. Untereiner, B. Lévy, 2016, ACM Trans. on Graphics.

Hex-Dominant Meshing

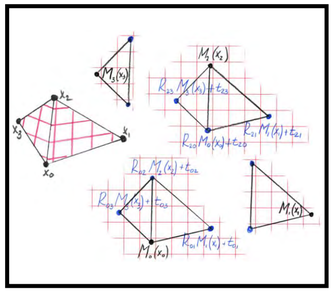
Input

1. Orientation



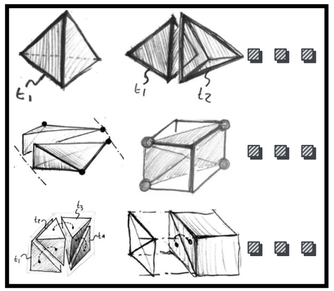
Spherical harmonic functions

2. Position



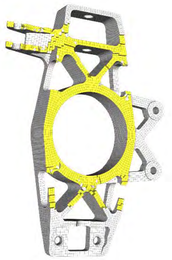
Periodic global parameterization

3. Extraction



Tetrahedral recombination

Output

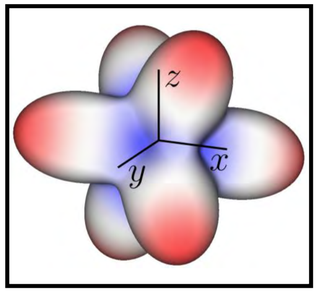


Hexahedral-dominant meshing, D. Sokolov, N. Ray, L. Untereiner, B. Lévy, 2016, ACM Trans. on Graphics.

Hex-Dominant Meshing

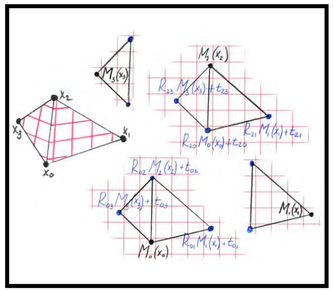
Input

1. Orientation



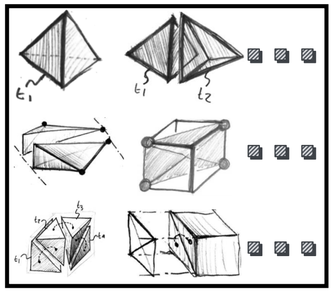
Spherical harmonic functions

2. Position



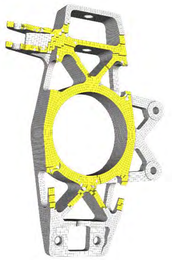
Periodic global parameterization

3. Extraction



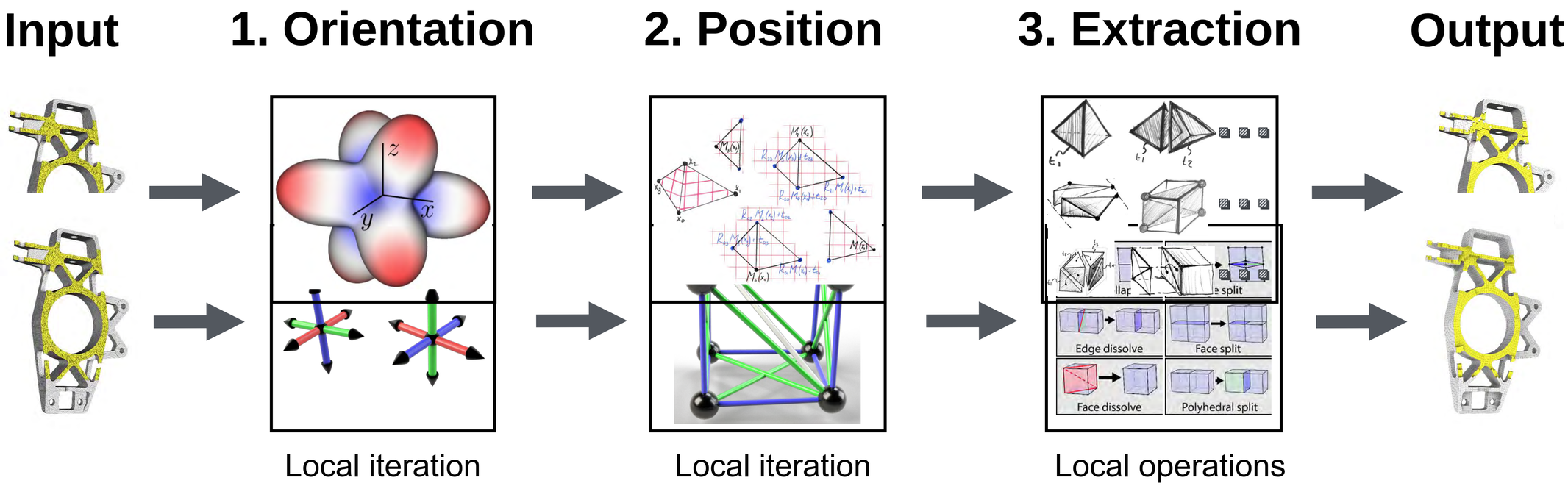
Tetrahedral recombination

Output



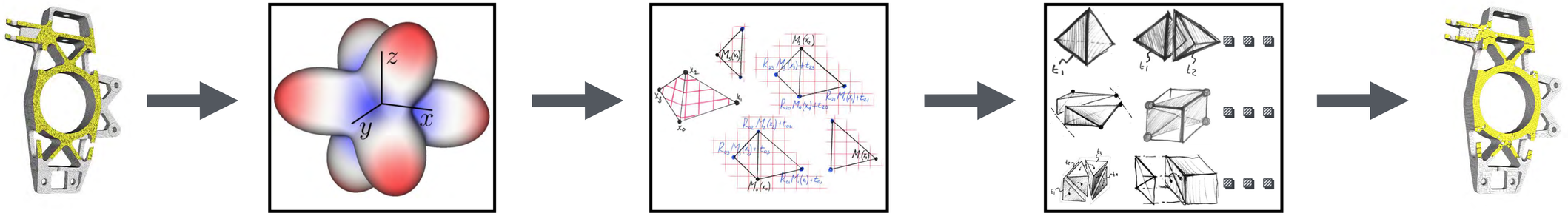
Hexahedral-dominant meshing, D. Sokolov, N. Ray, L. Untereiner, B. Lévy, 2016, ACM Trans. on Graphics.

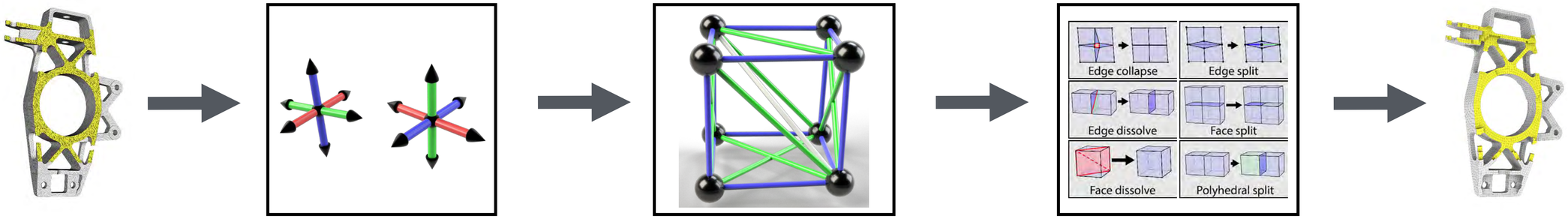
Hex-Dominant Meshing



Ours

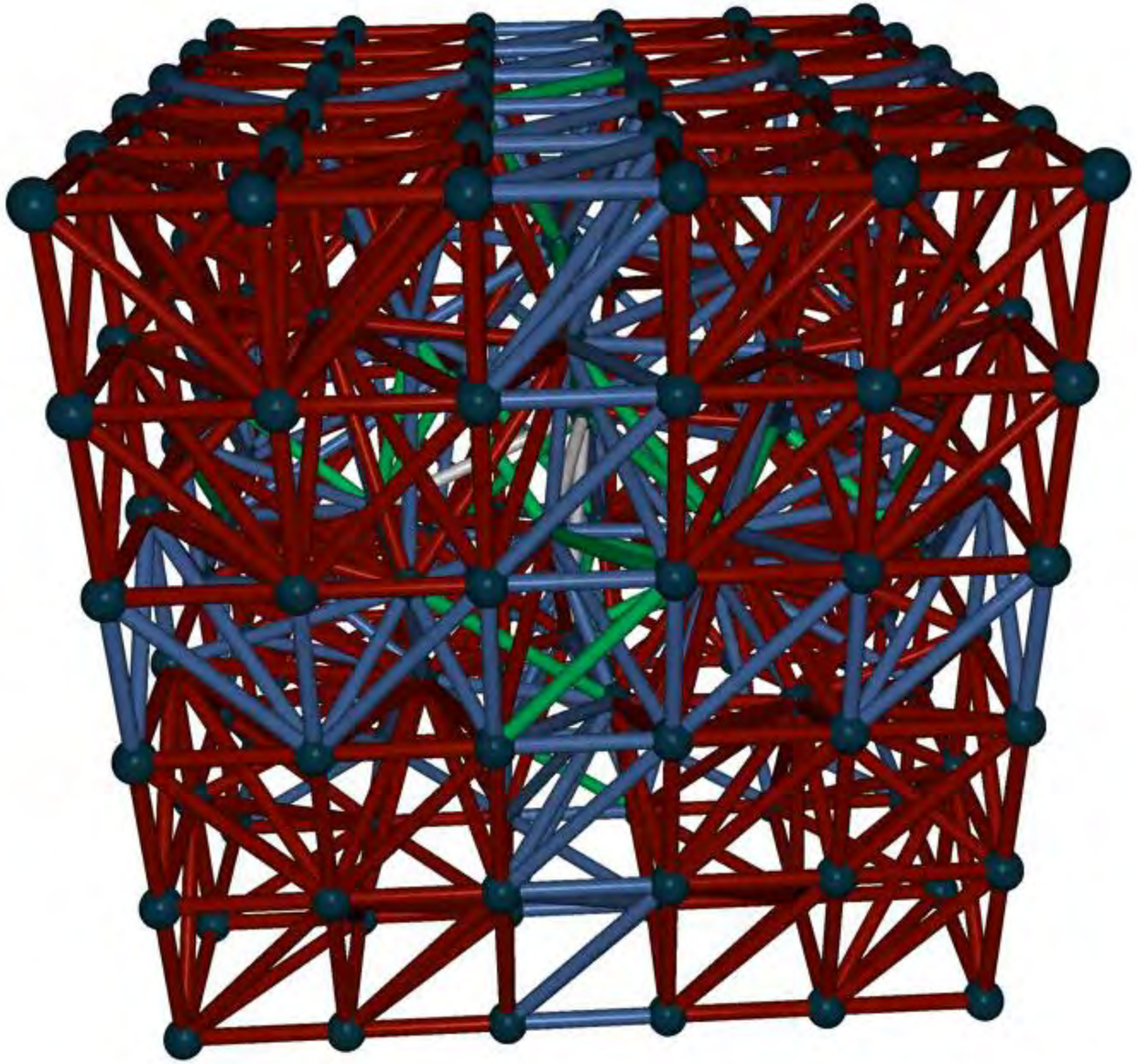
Input **1. Orientation** **2. Position** **3. Extraction** **Output**





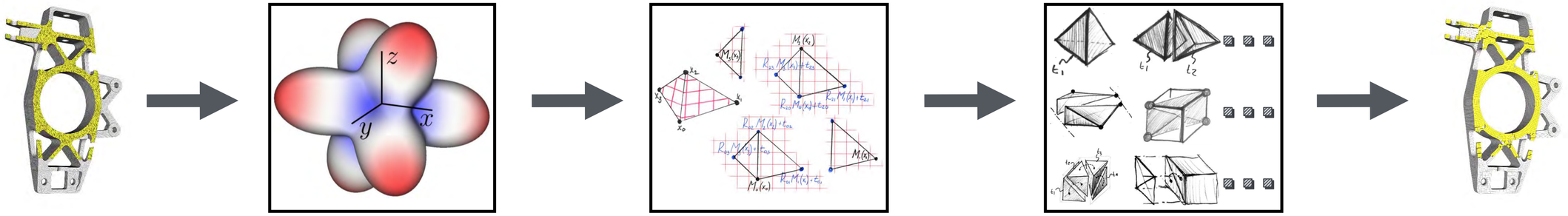
Local iteration Local iteration Local operations

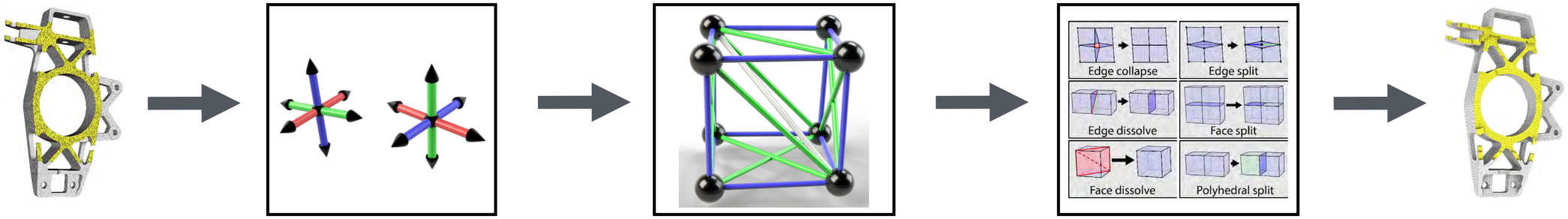
Key Idea



Ours

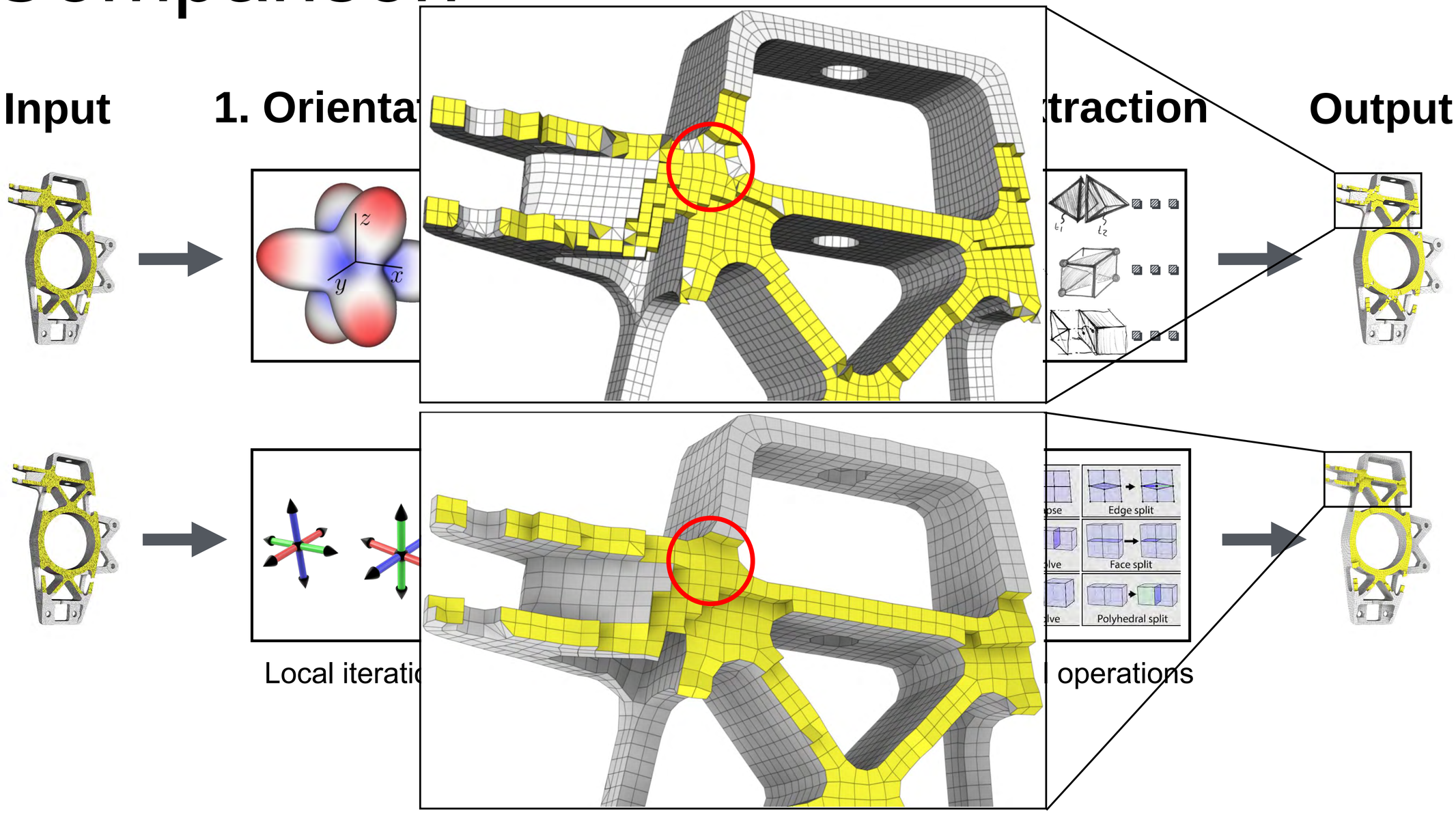
Input **1. Orientation** **2. Position** **3. Extraction** **Output**





Local iteration Local iteration Local operations

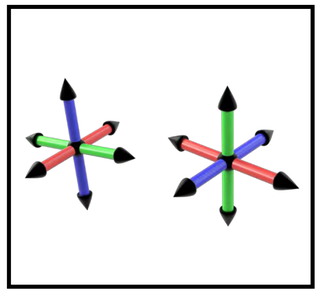
Comparison


Input
1. Orienta
traction
Output
z
y
x
Edge split
Face split
Polyhedral split
Local iterati
operations
Fewer irregular elements

Pipeline

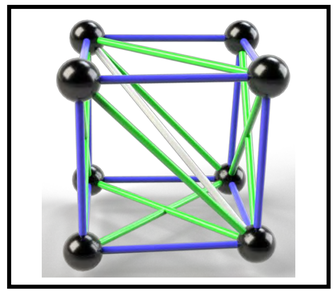
Input

1. Orientation



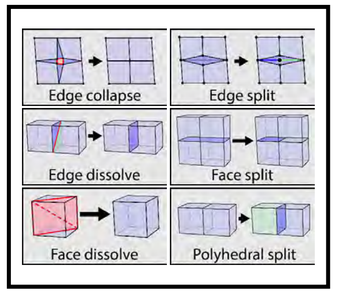
Local iteration

2. Position



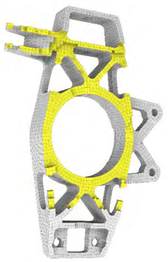
Local iteration

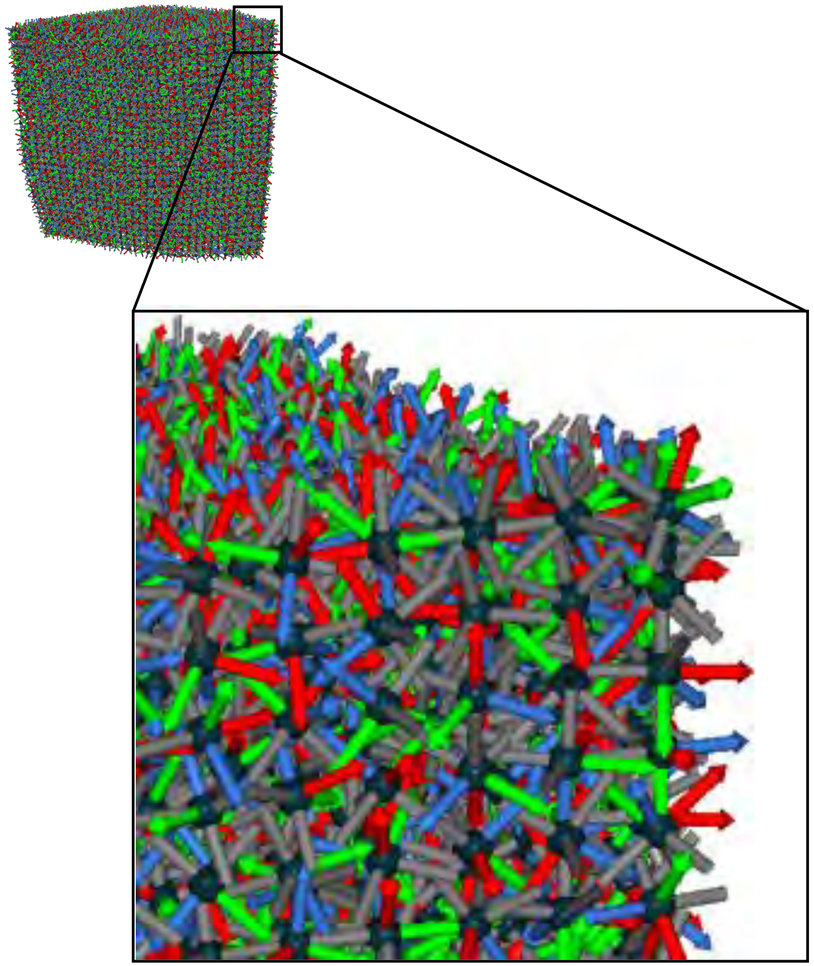
3. Extraction

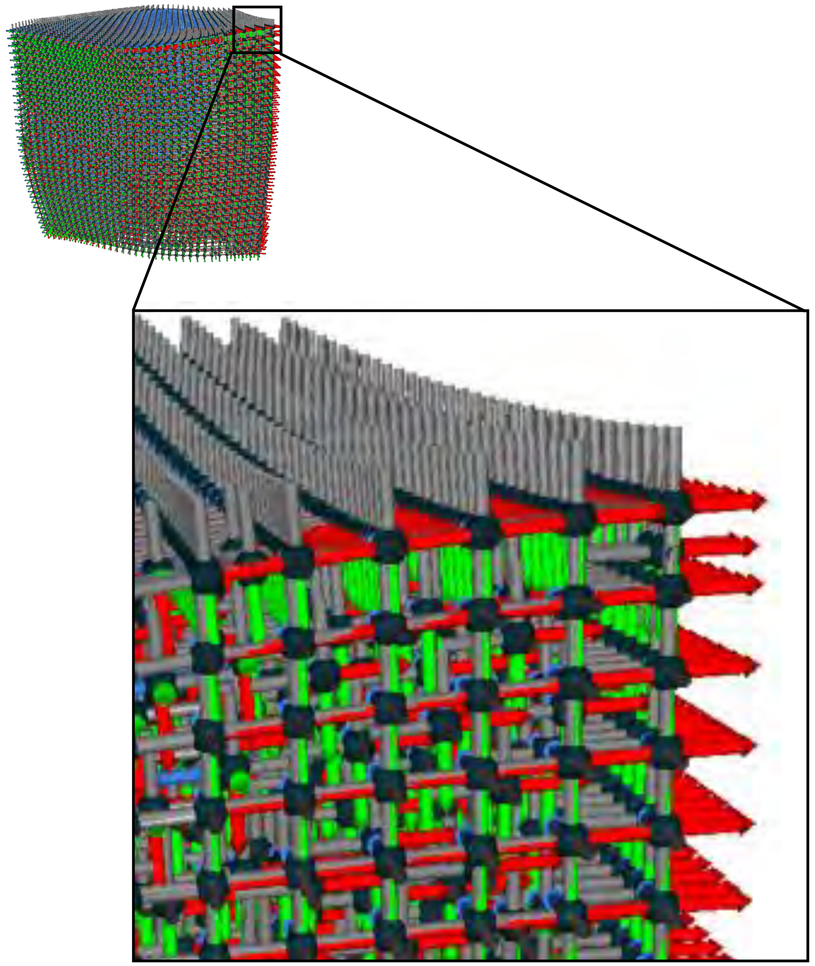


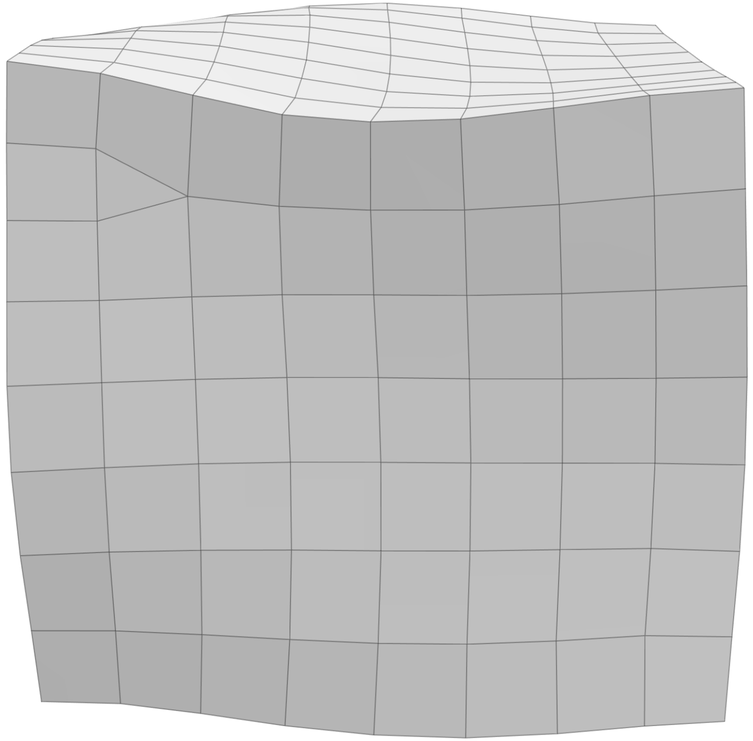
Local operations

Output



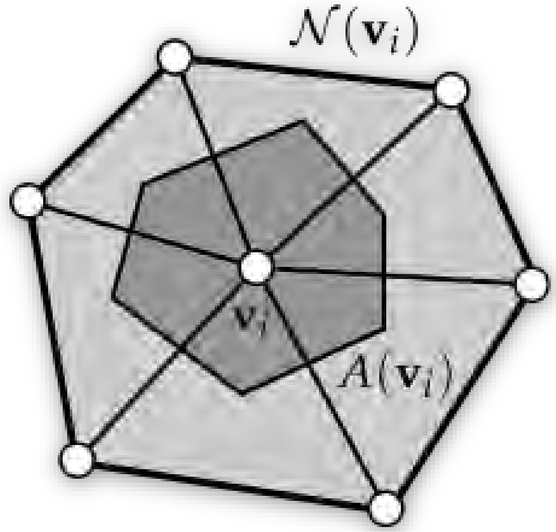






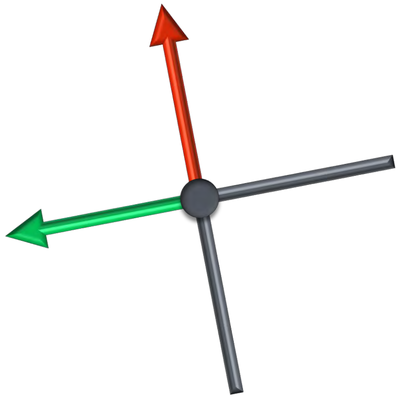
A simple smoothing algorithm

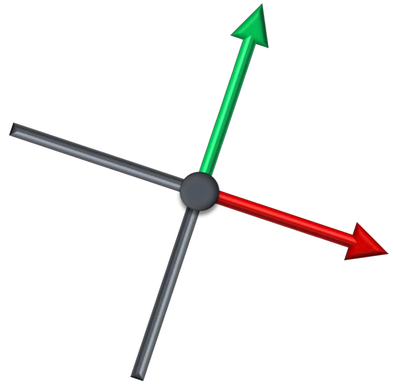
```
1. foreach v ∈ V:
2.     temp = 0
3.     foreach n ∈ N(v):
4.       temp += value[n]
5.     value[v] = temp / |N(v)|
```



A simple smoothing algorithm

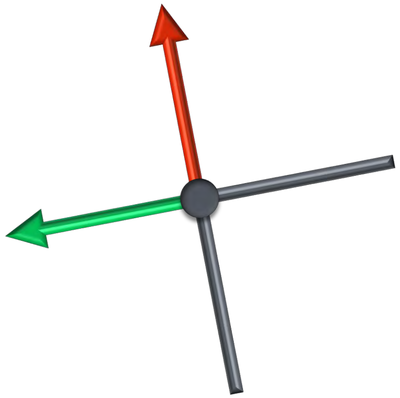
```
1. foreach v ∈ V:
2.     temp = 0
3.     foreach n ∈ N(v):
4.       temp += value[n]
5.     value[v] = temp / |N(v)|
```

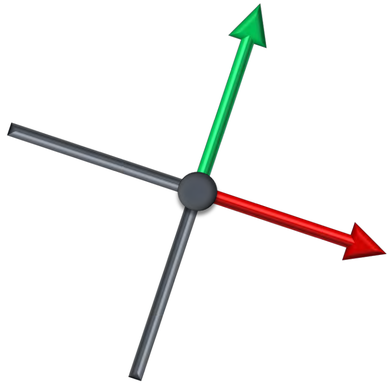




A simple smoothing algorithm

```
1. foreach v ∈ V:
2.     temp = 0
3.     foreach n ∈ N(v):
           matching of crosses
4.         temp += value[n]
5.     value[v] = temp / |N(v)|
```

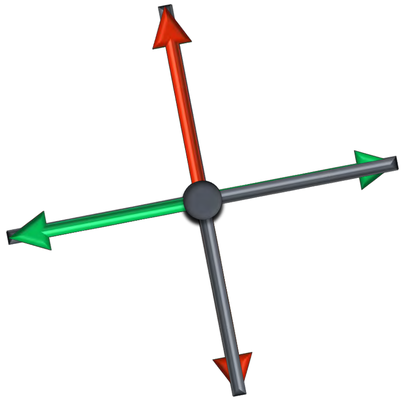


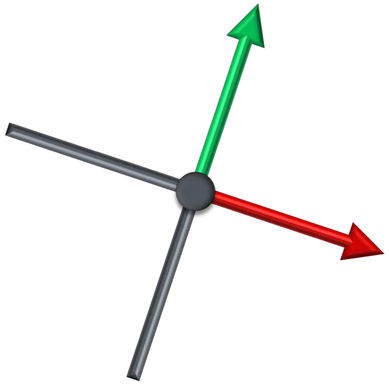


Instant Field-Aligned Meshes, Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung, 2015, ACM Trans. on Graphics.

A simple smoothing algorithm

```
1. foreach v ∈ V:
2.     temp = 0
3.     foreach n ∈ N(v):
           matching of crosses
4.         temp += value[n]
5.     value[v] = temp / |N(v)|
```

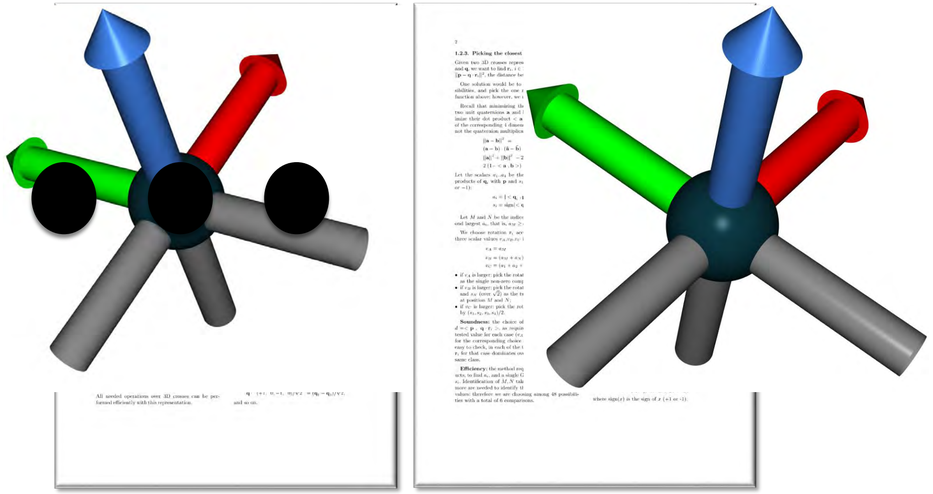




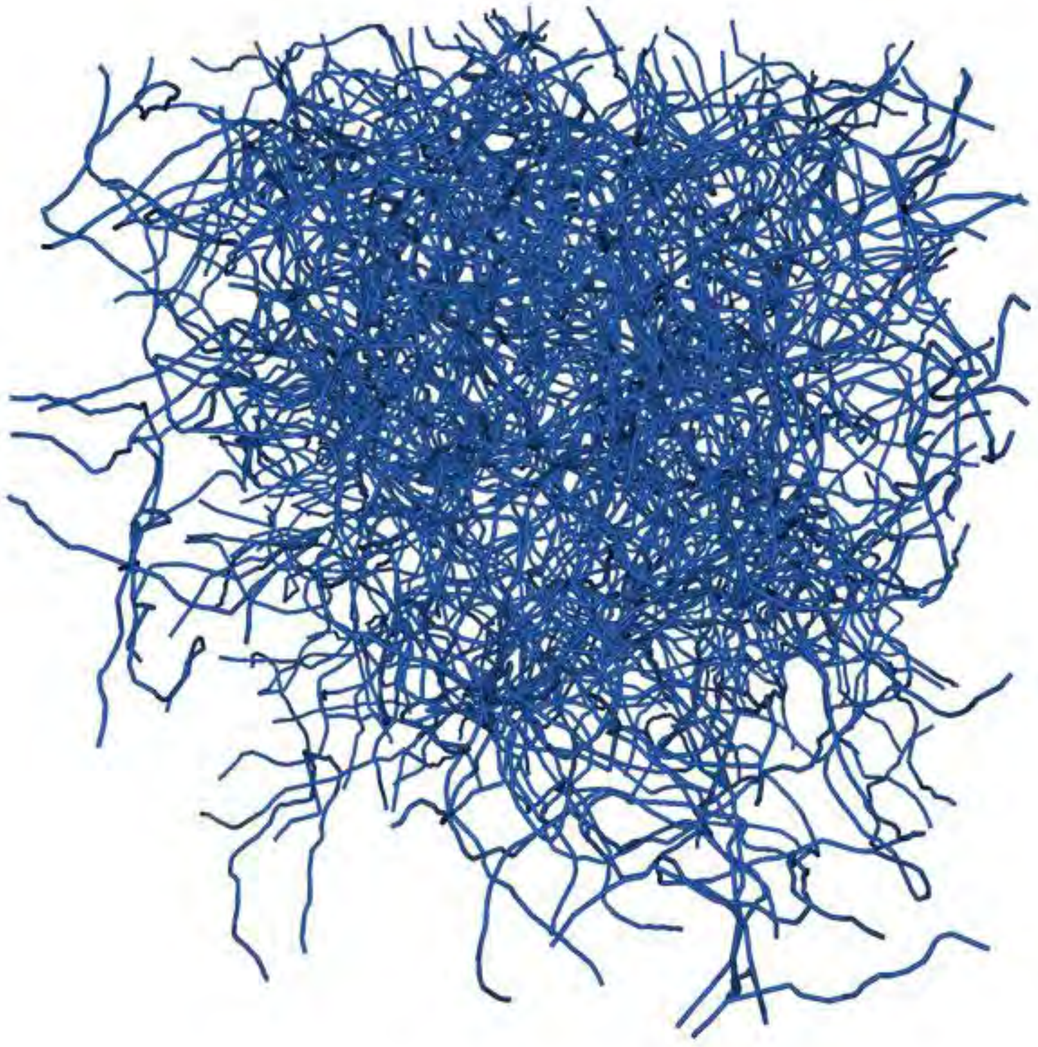
Instant Field-Aligned Meshes, Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung, 2015, ACM Trans. on Graphics.

The simplest smoothing algorithm

```
1. foreach v ∈ V:
2.     temp = 0
3.     foreach n ∈ N(v):
           matching of crosses
4.         temp += value[n]
5.     value[v] = temp / |N(v)|
```



24 equivalents

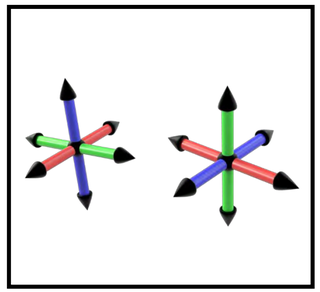


Level [redacted]

Pipeline

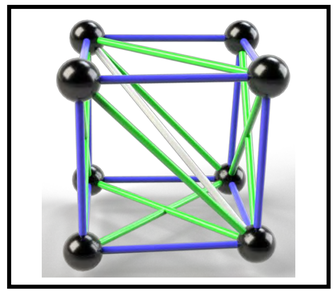
Input

1. Orientation



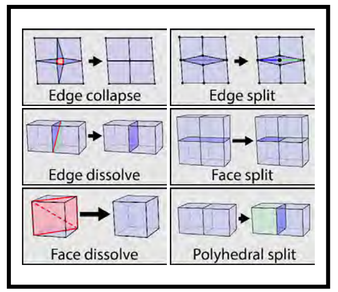
Local iteration

2. Position



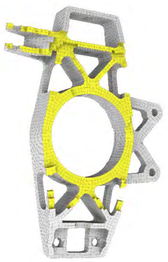
Local iteration

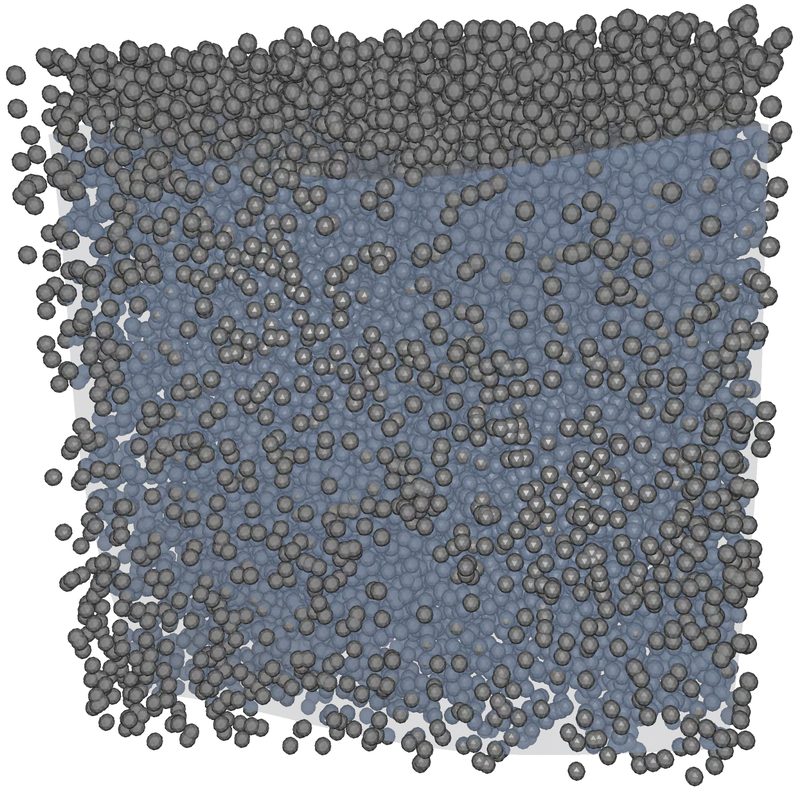
3. Extraction

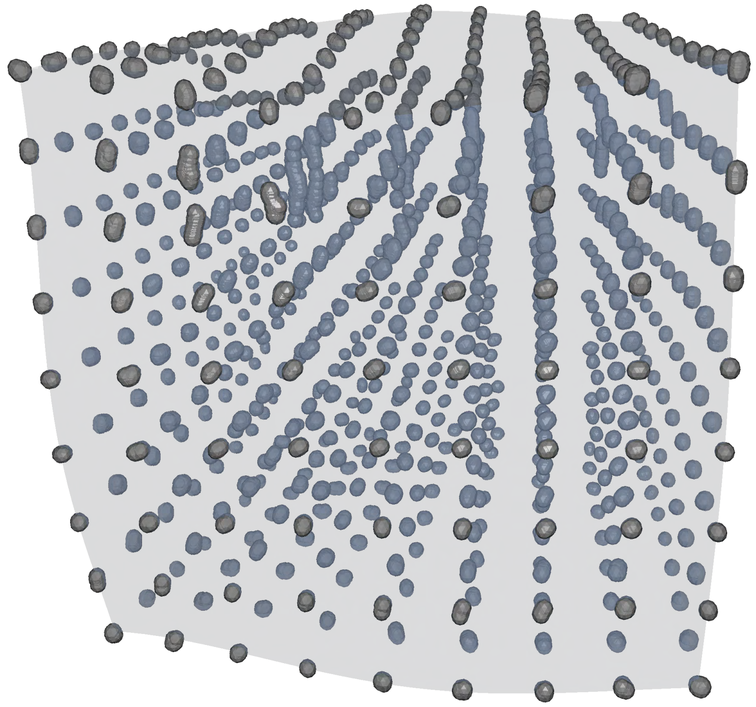


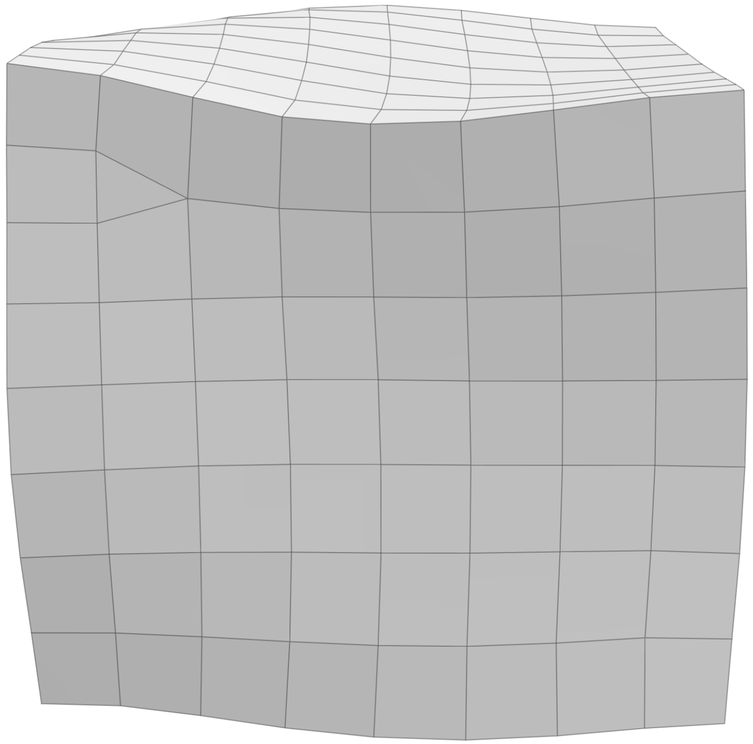
Local operations

Output





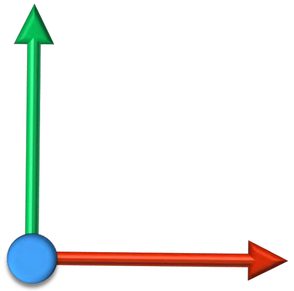




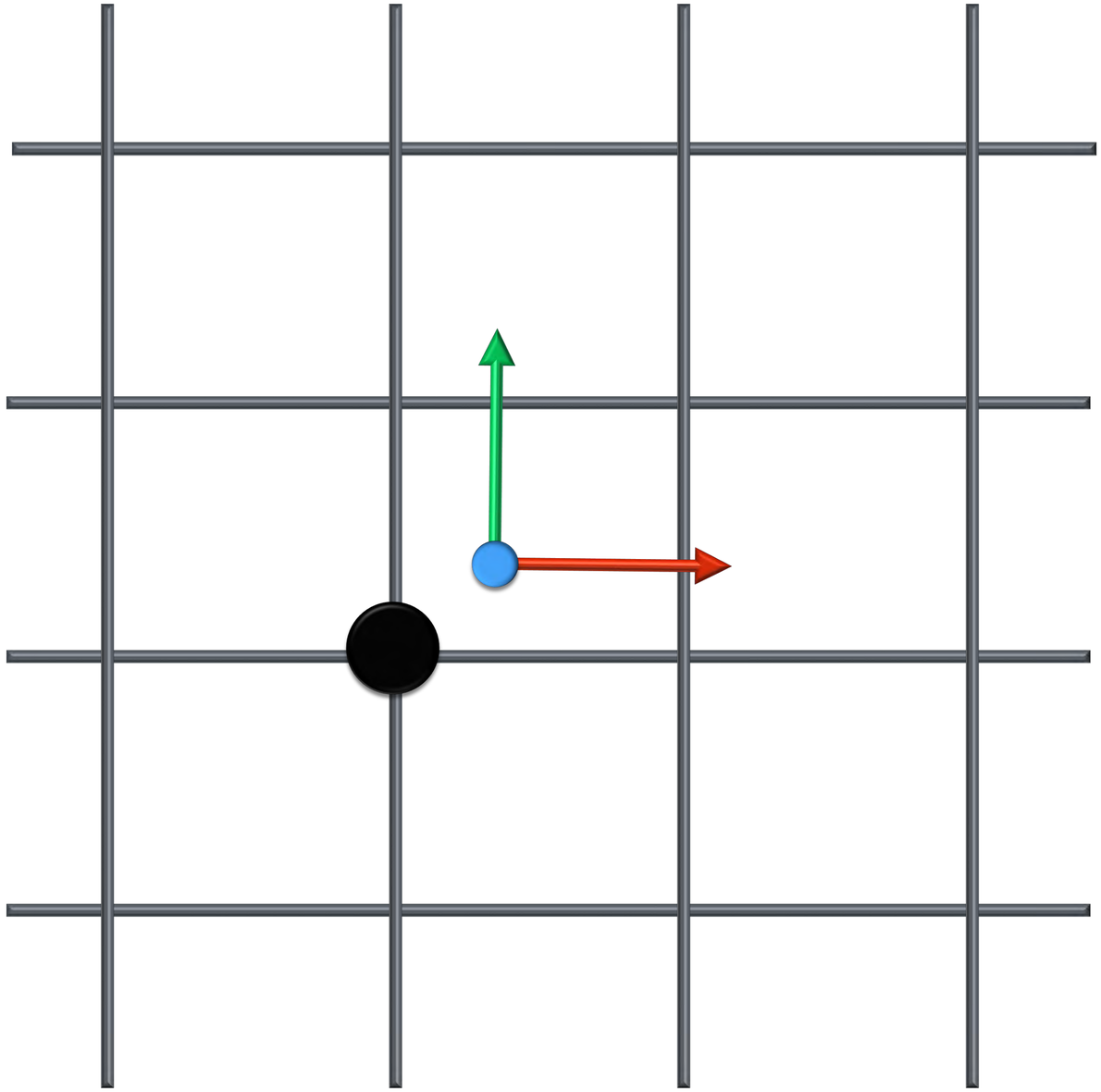
A simple smoothing algorithm

```
1. foreach v ∈ V:
2.    temp = 0
3.    foreach n ∈ N(v):
        matching of local parameterization
4.      temp += value[n]
5.    value[v] = temp / |N(v)|
```

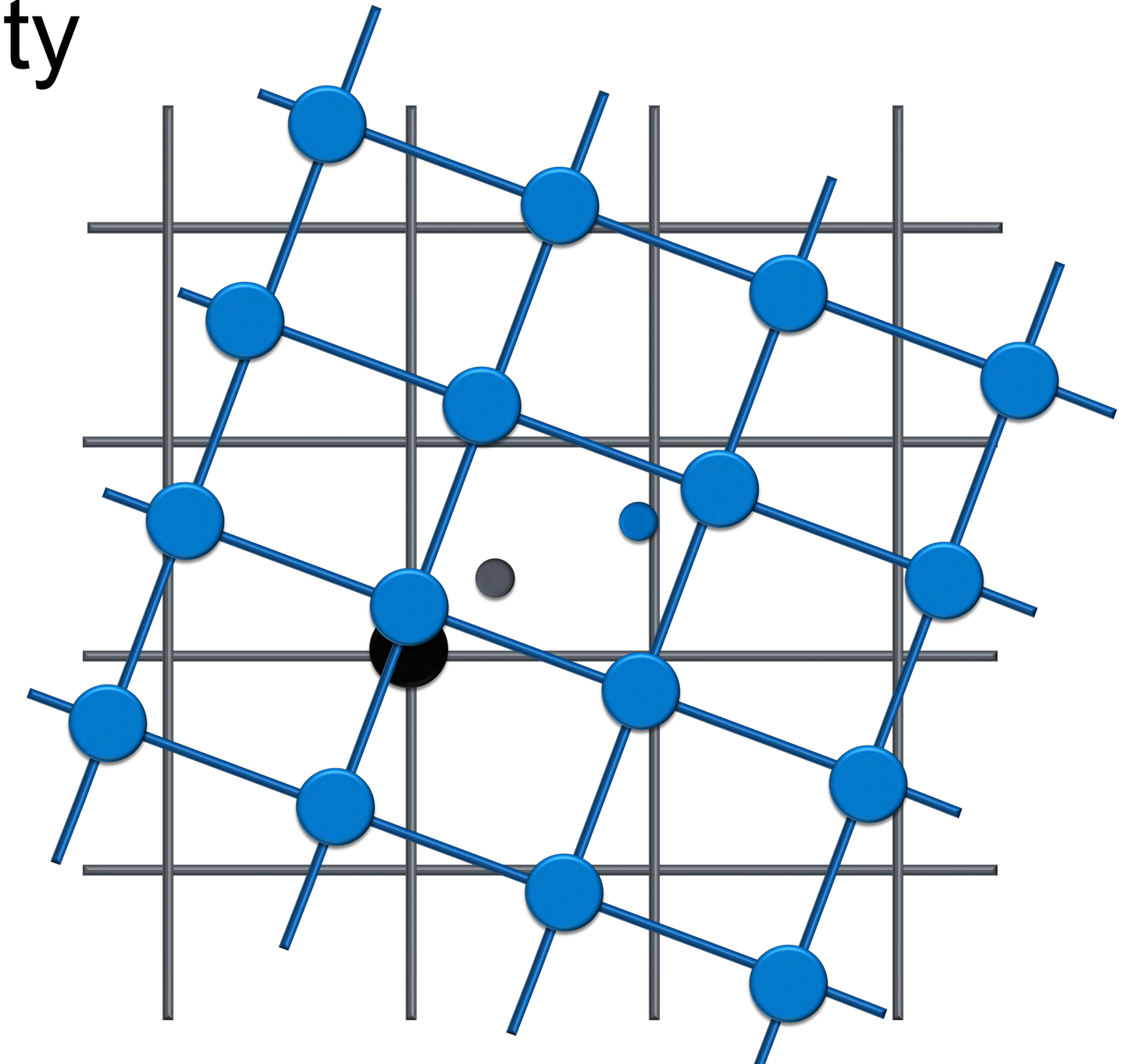
Local Parameterization



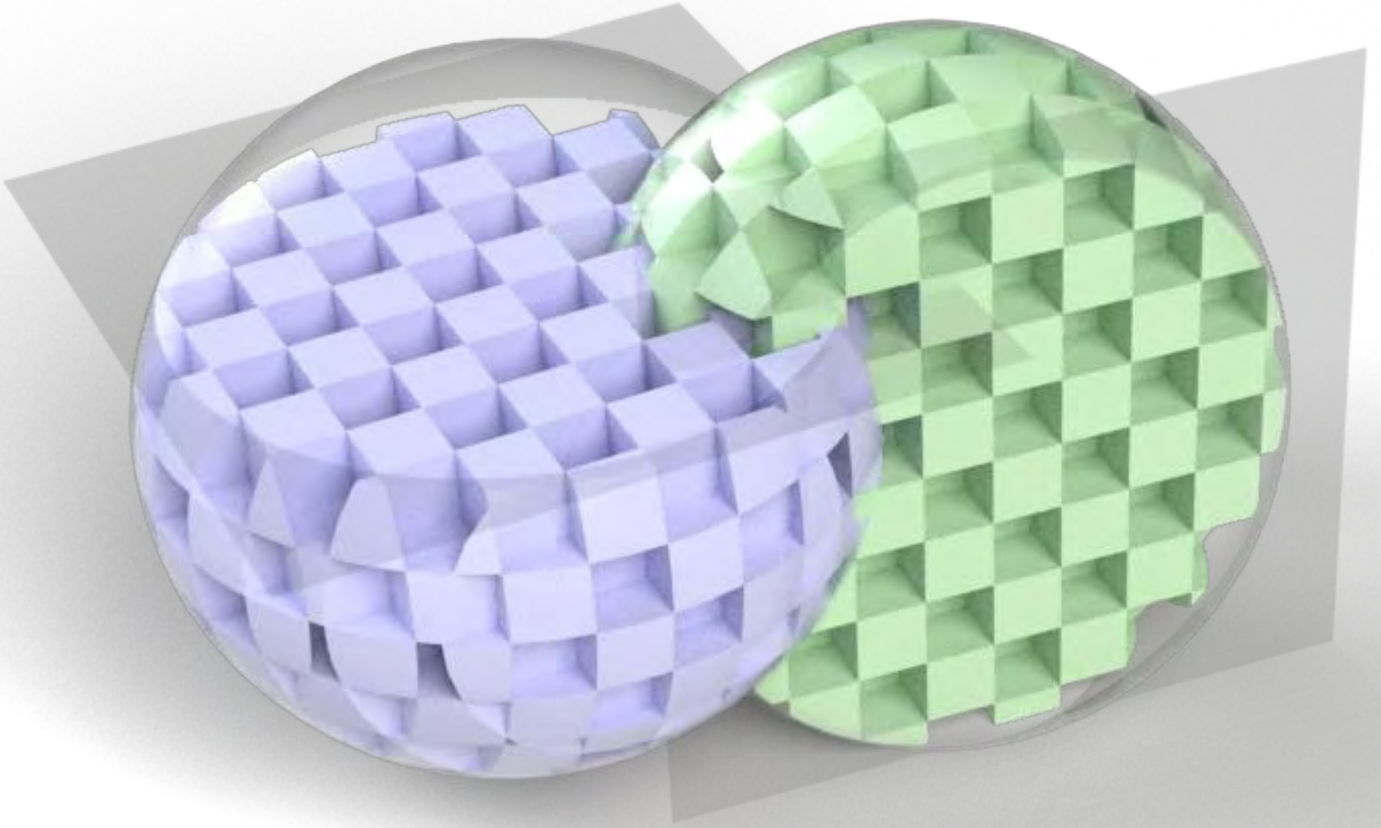
Local Parameterization

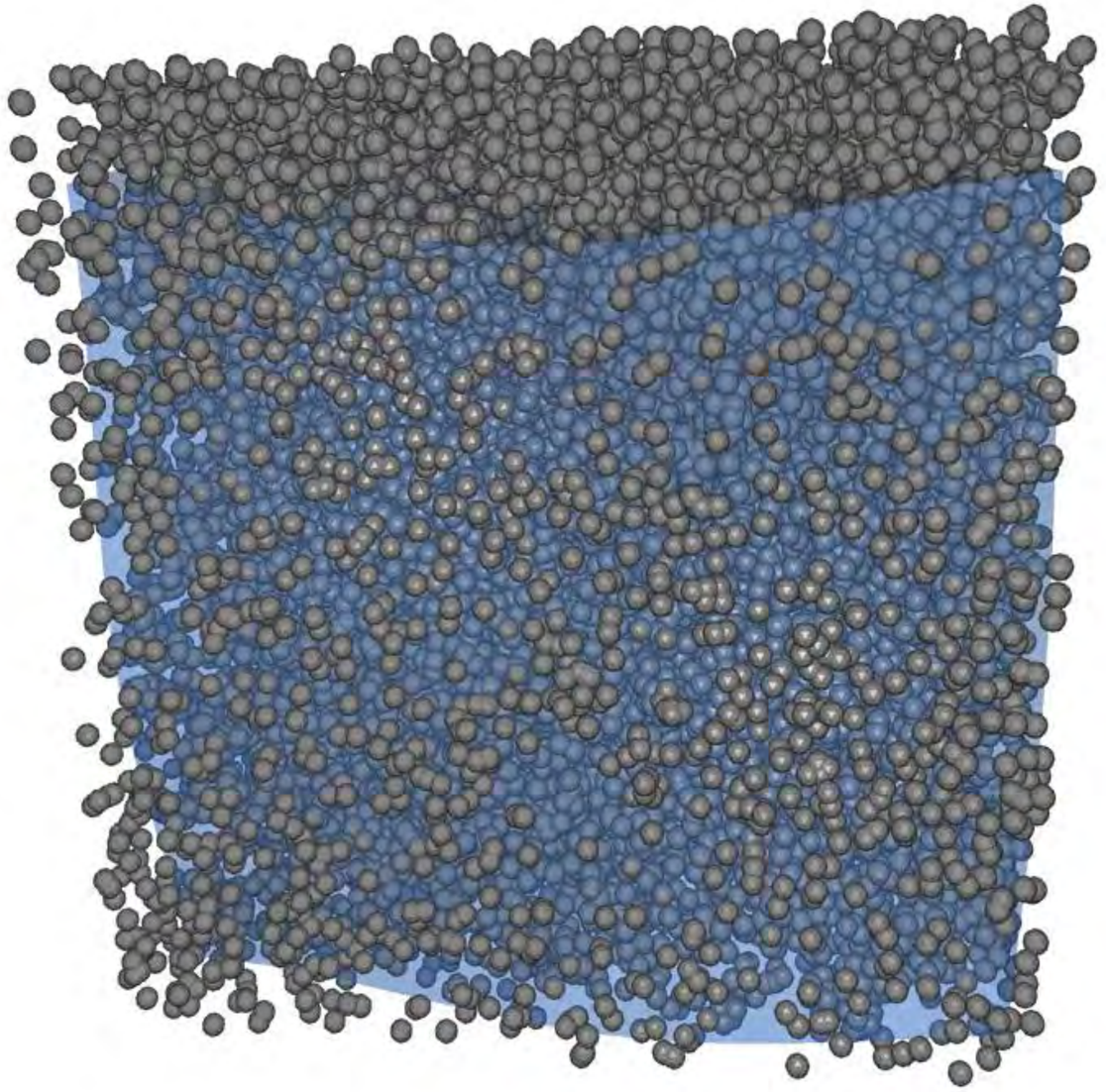


Similarity



3D Position Field



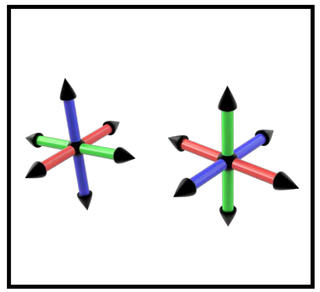


Level [redacted]

Pipeline

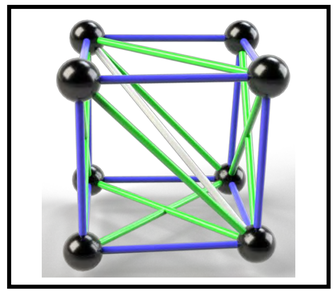
Input

1. Orientation



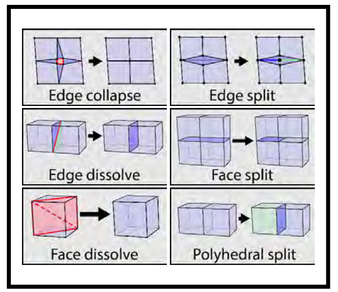
Local iteration

2. Position



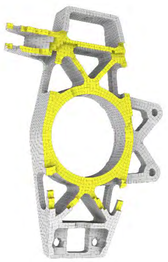
Local iteration

3. Extraction

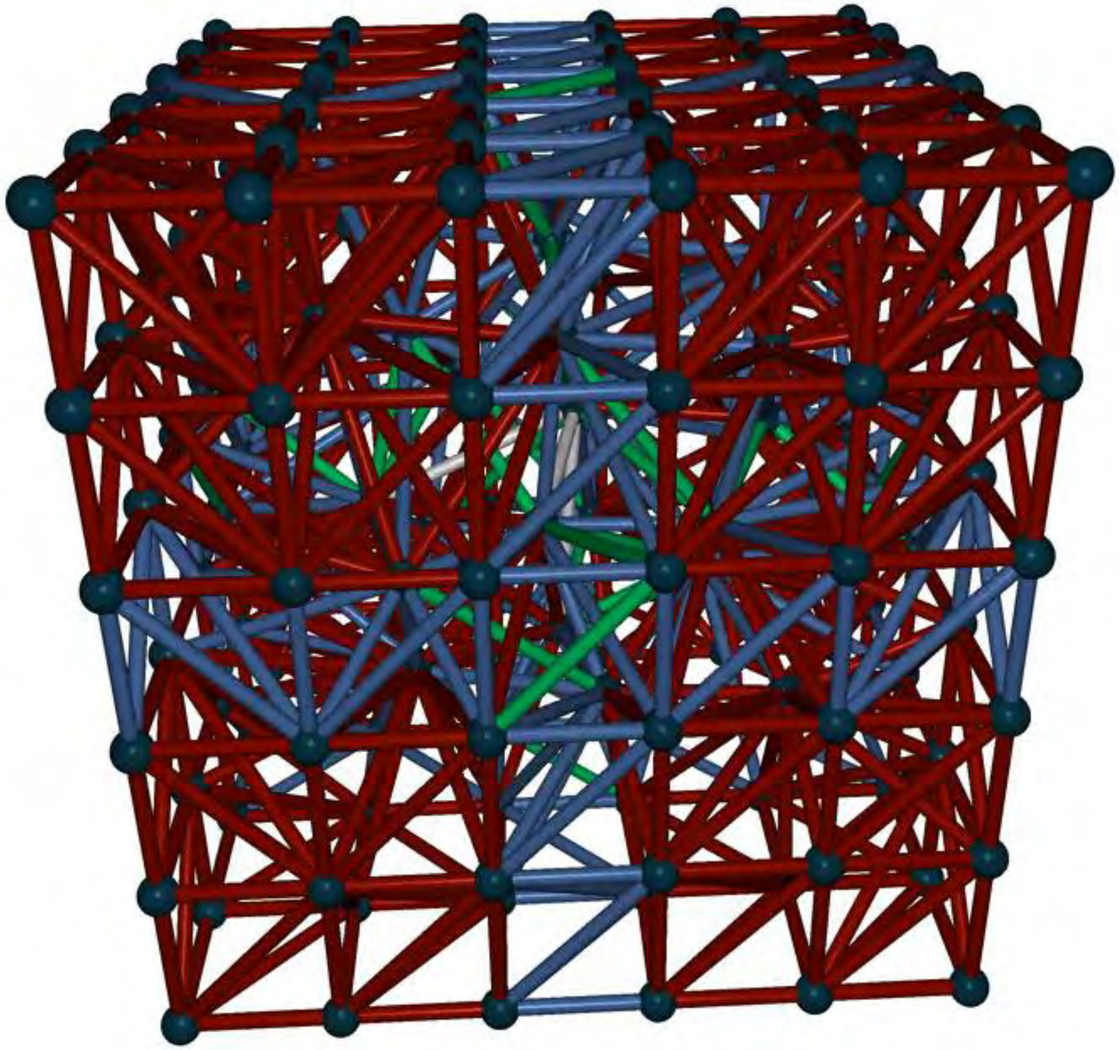


Local operations

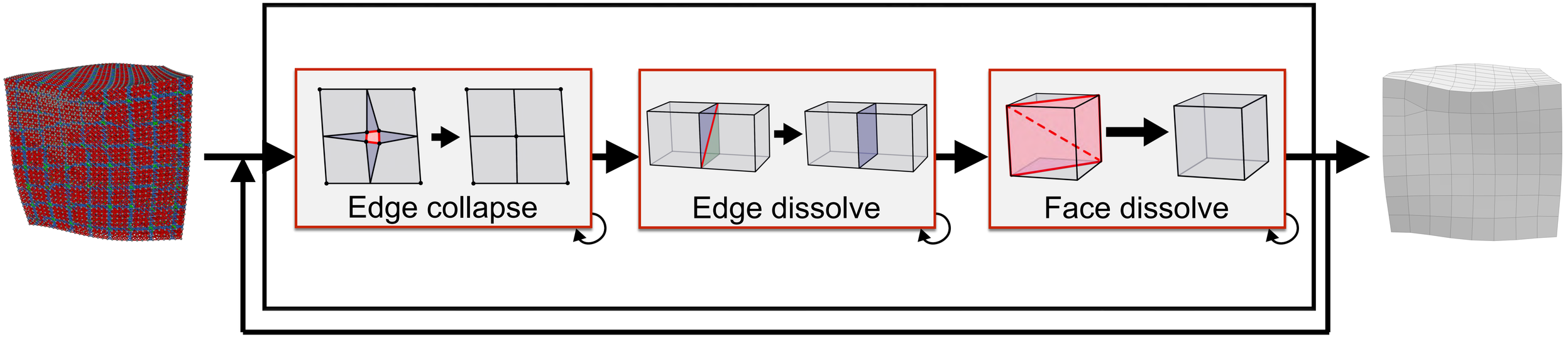
Output



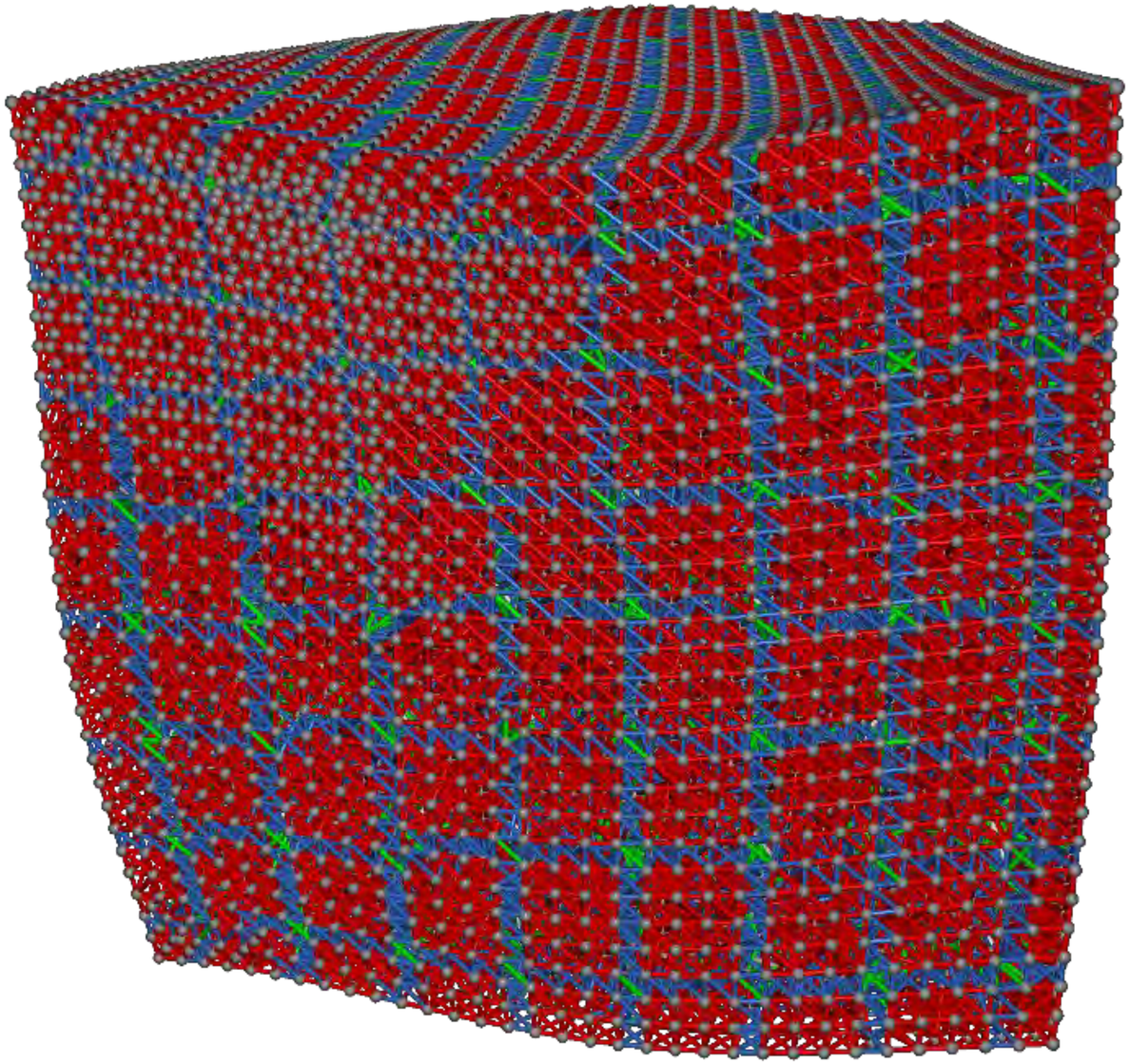
Edge Classification

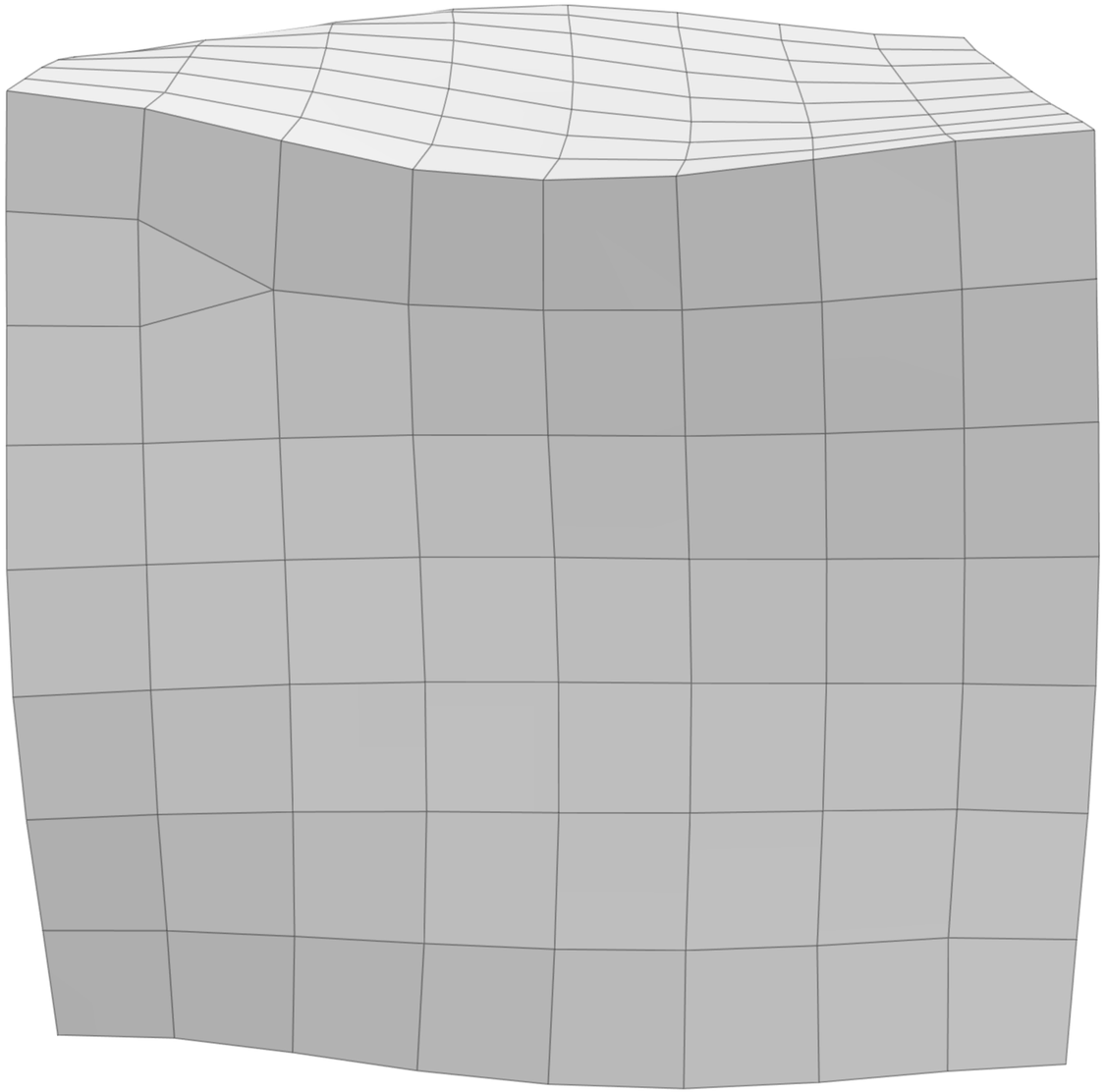


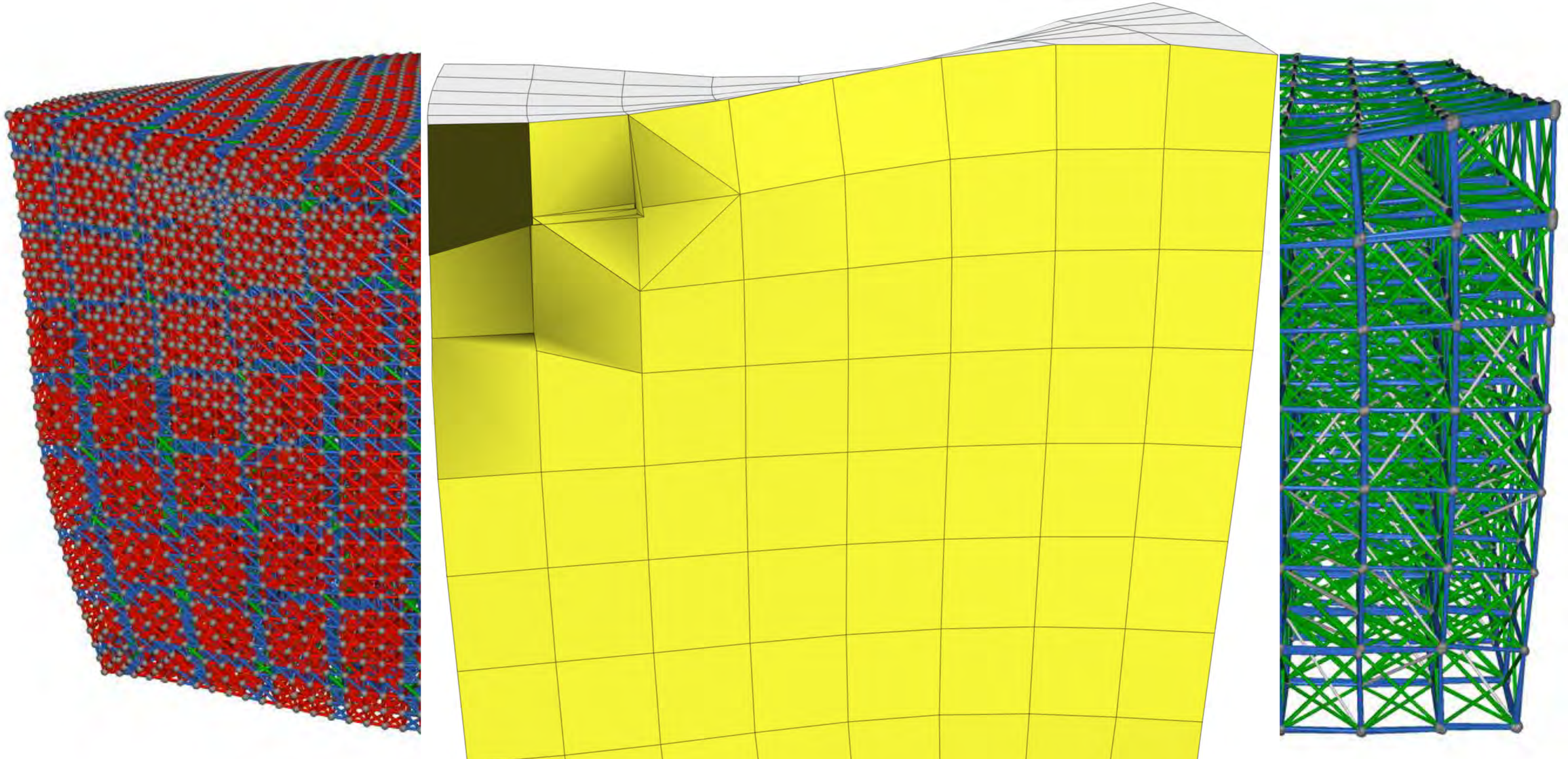
Local Coarsening



Local Coarsening





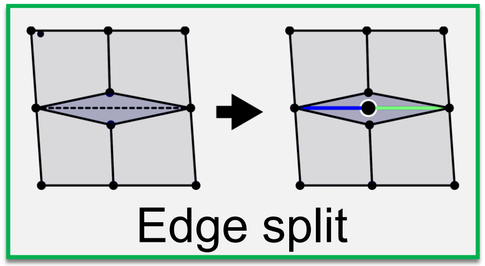


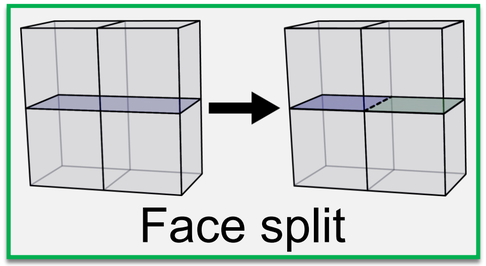
Original coordinates
(demonstration purpose only)

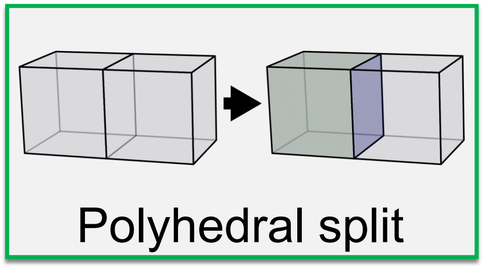
Position fields (output)

Unflippable Tetrahedral Complexes, Dougherty, R., Faber, V. and Murphy, M. , 2004, Discrete & Computational Geometry.

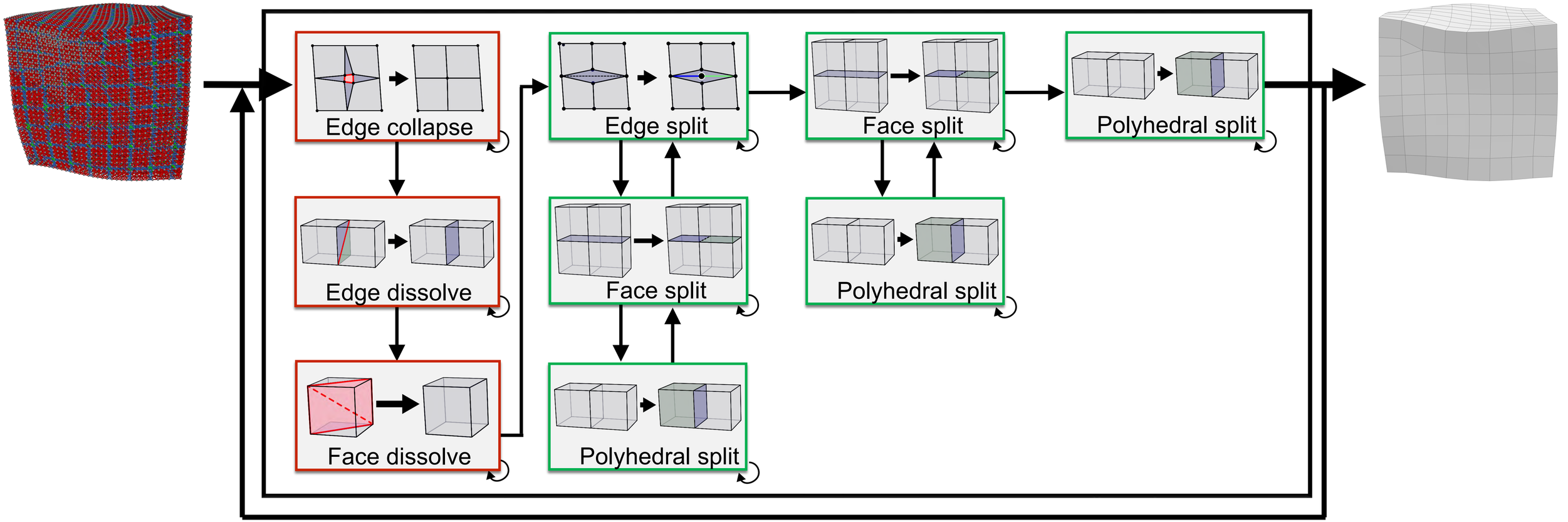
Local Refinement

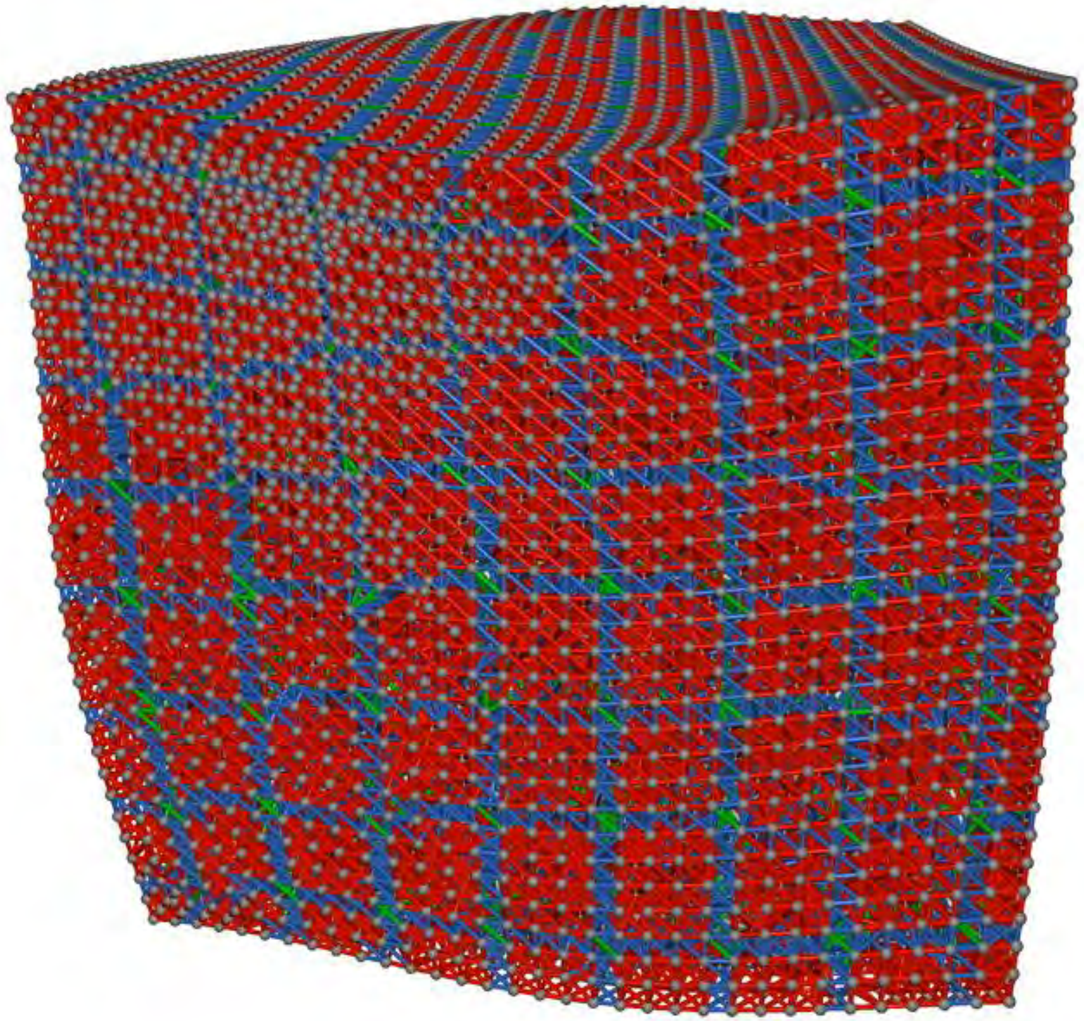


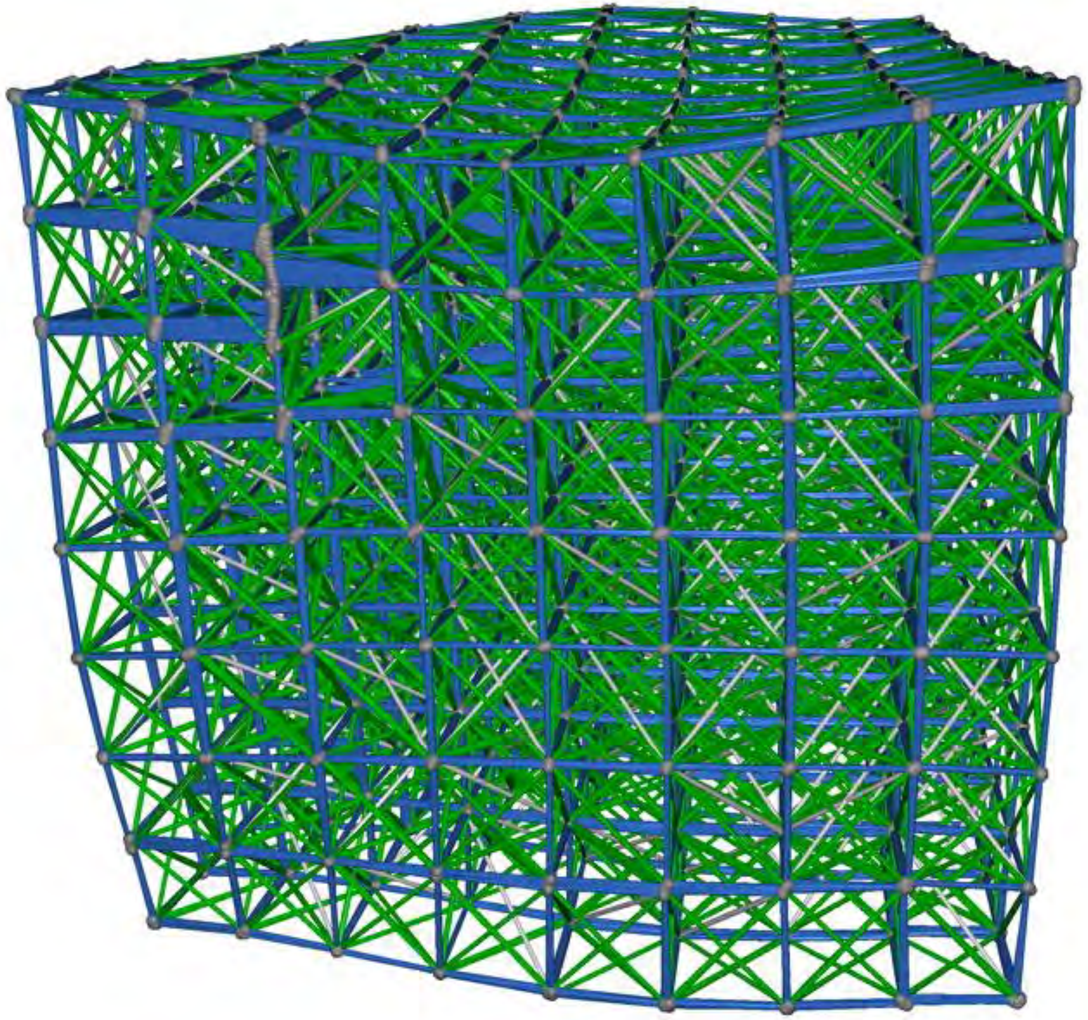




Local Coarsening + Local Refinement







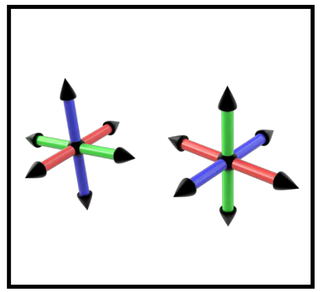
Original coordinates
(demonstration purpose only)

Position fields (output)

Pipeline

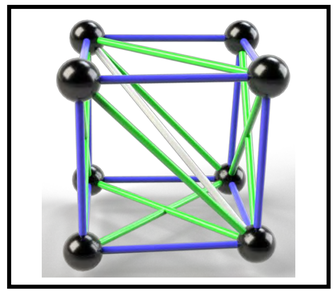
Input

1. Orientation



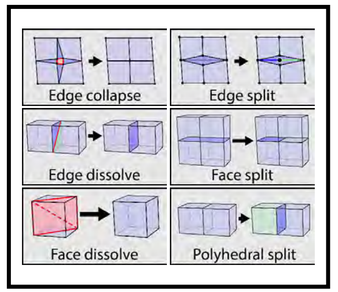
Local iteration

2. Position



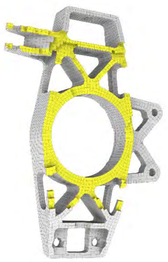
Local iteration

3. Extraction



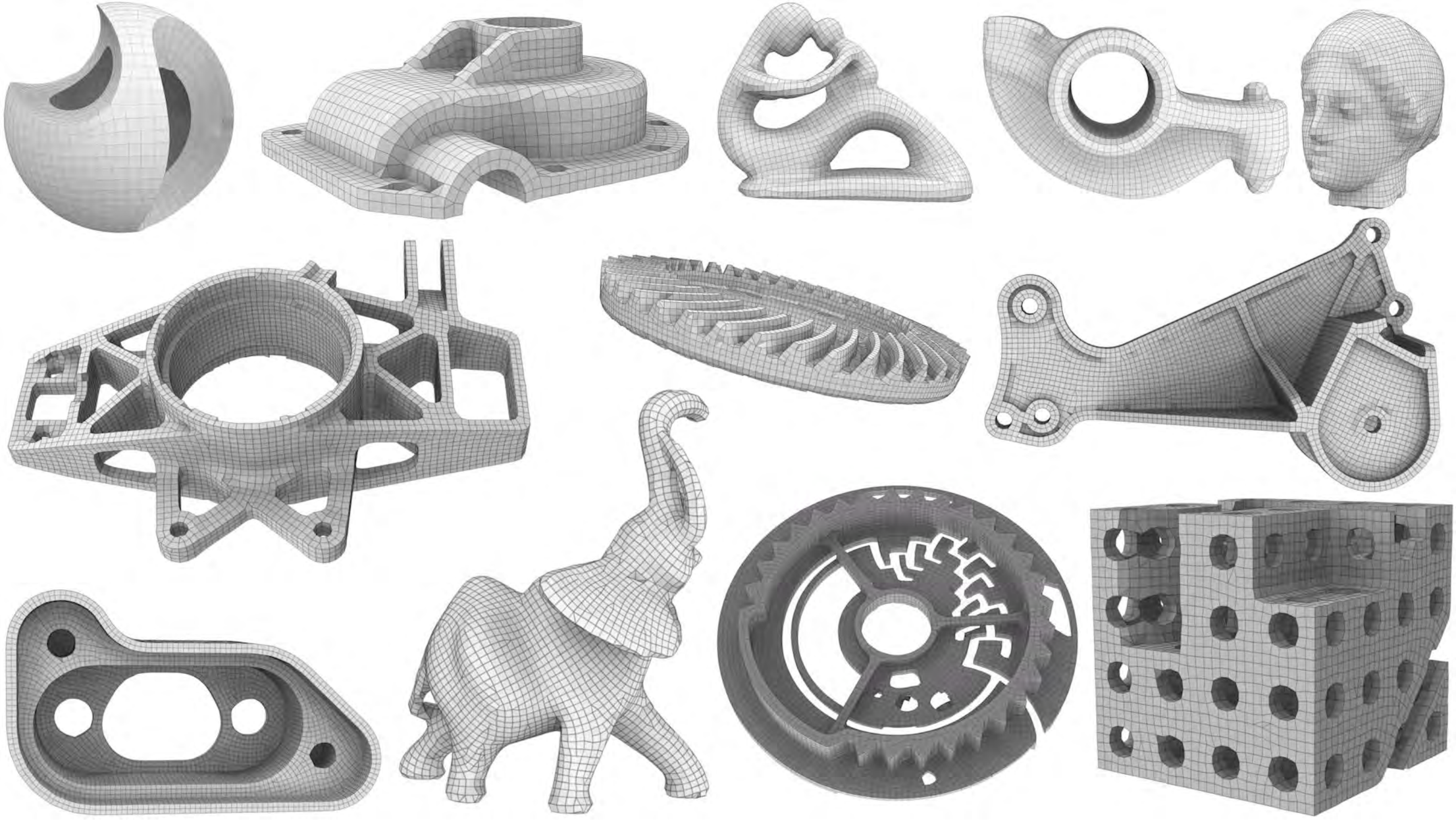
Local operations

Output

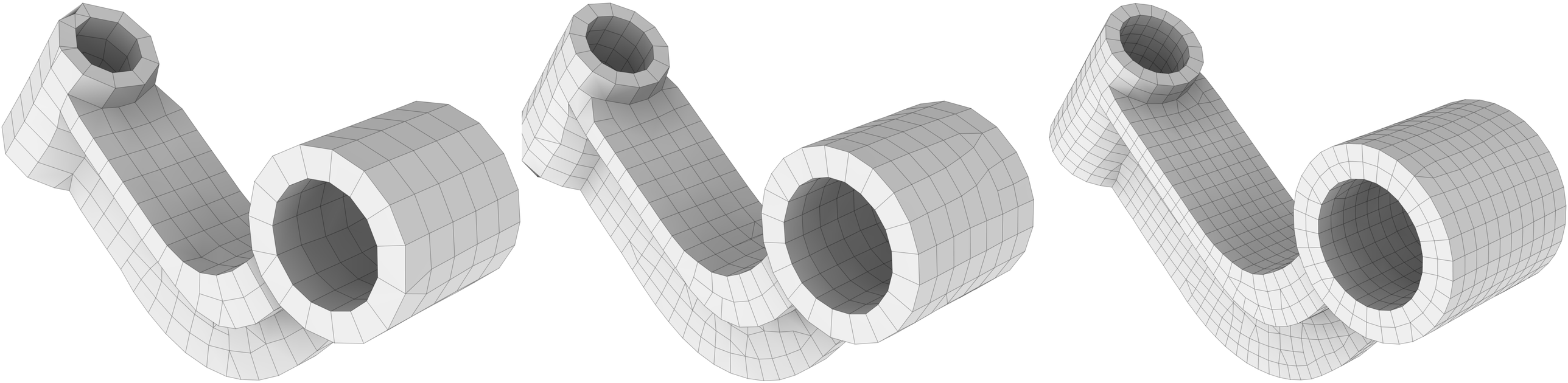


Robustness

110 models

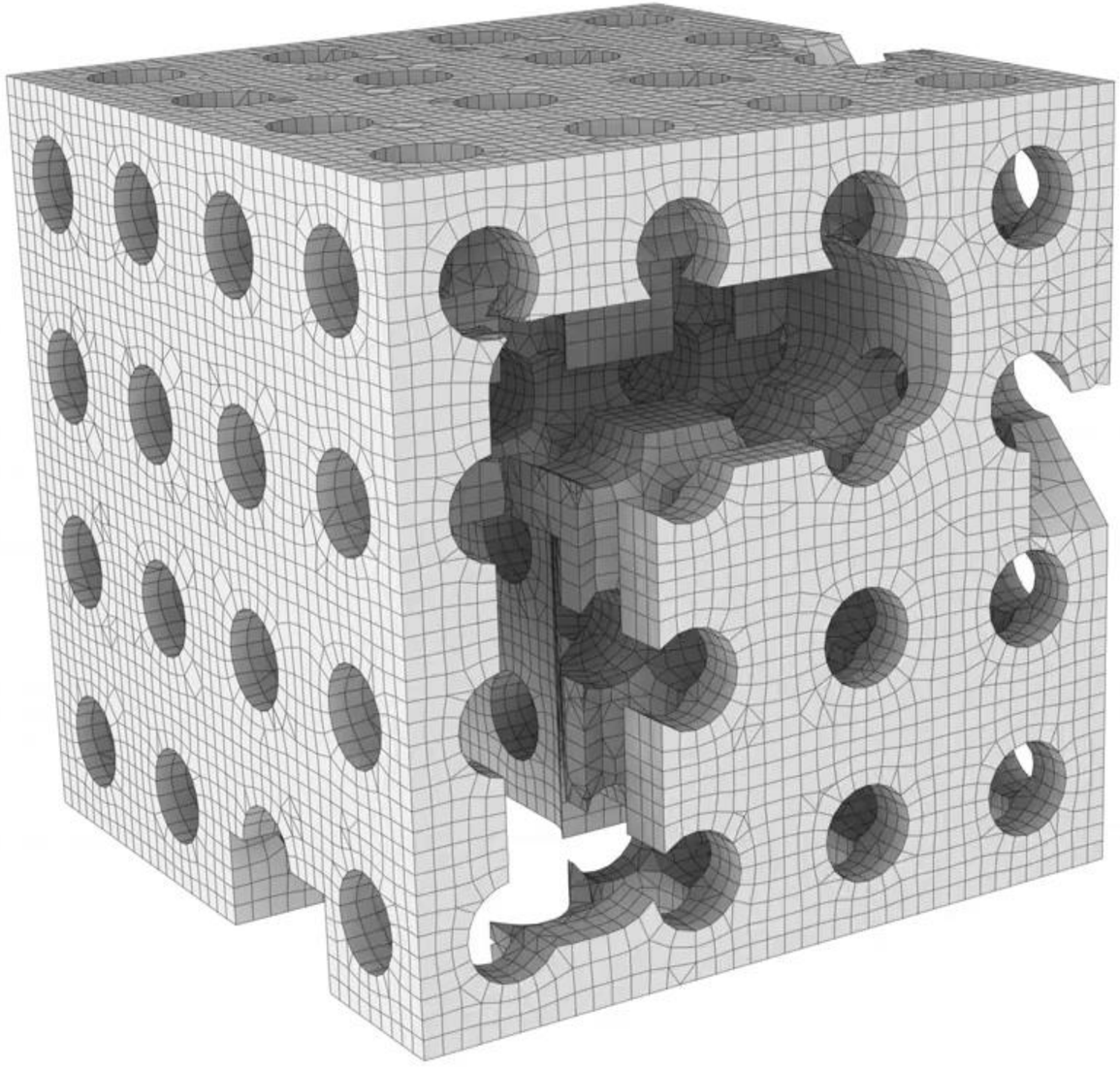


Single Parameter

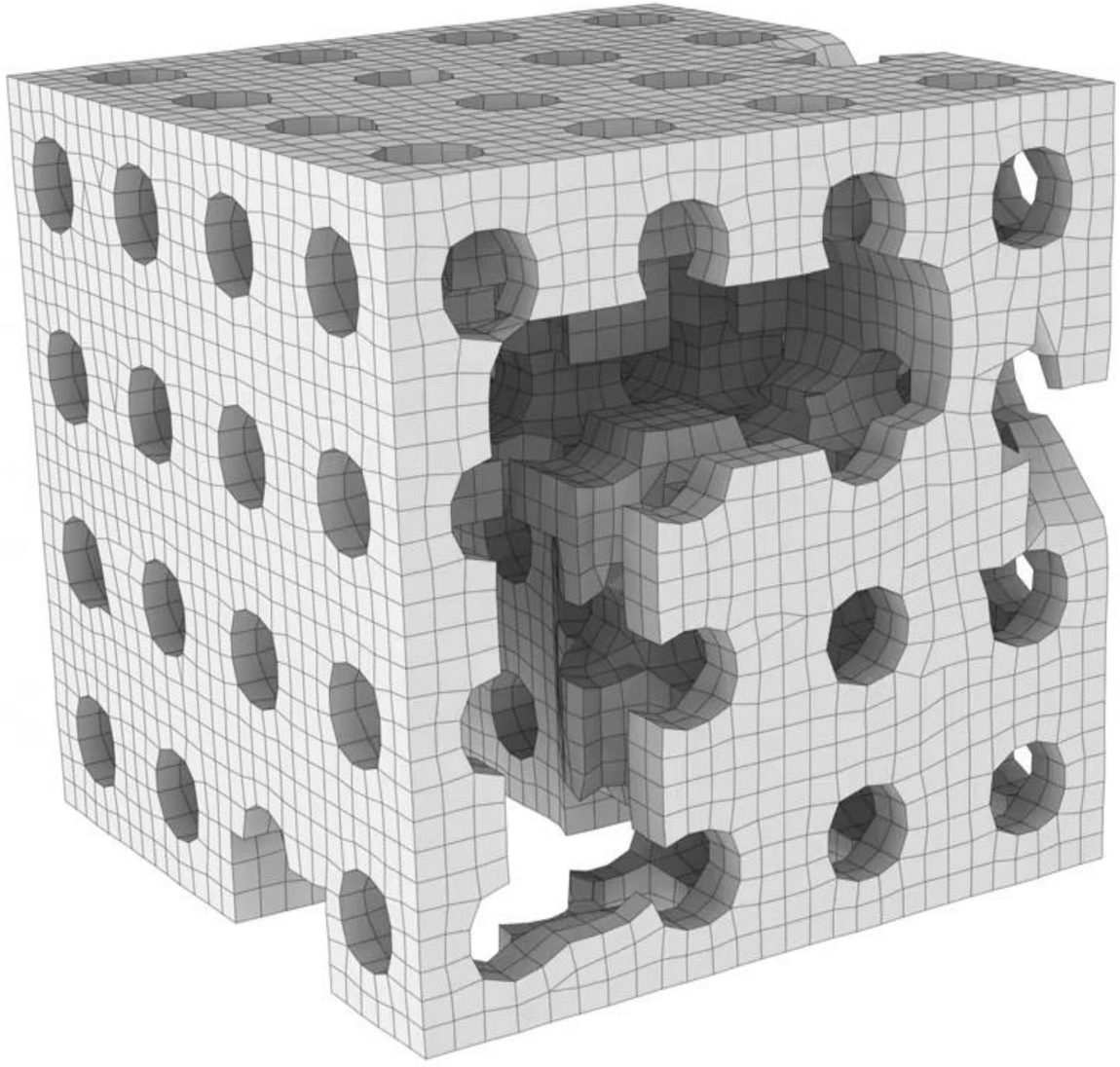


Different target element sizes

Comparisons

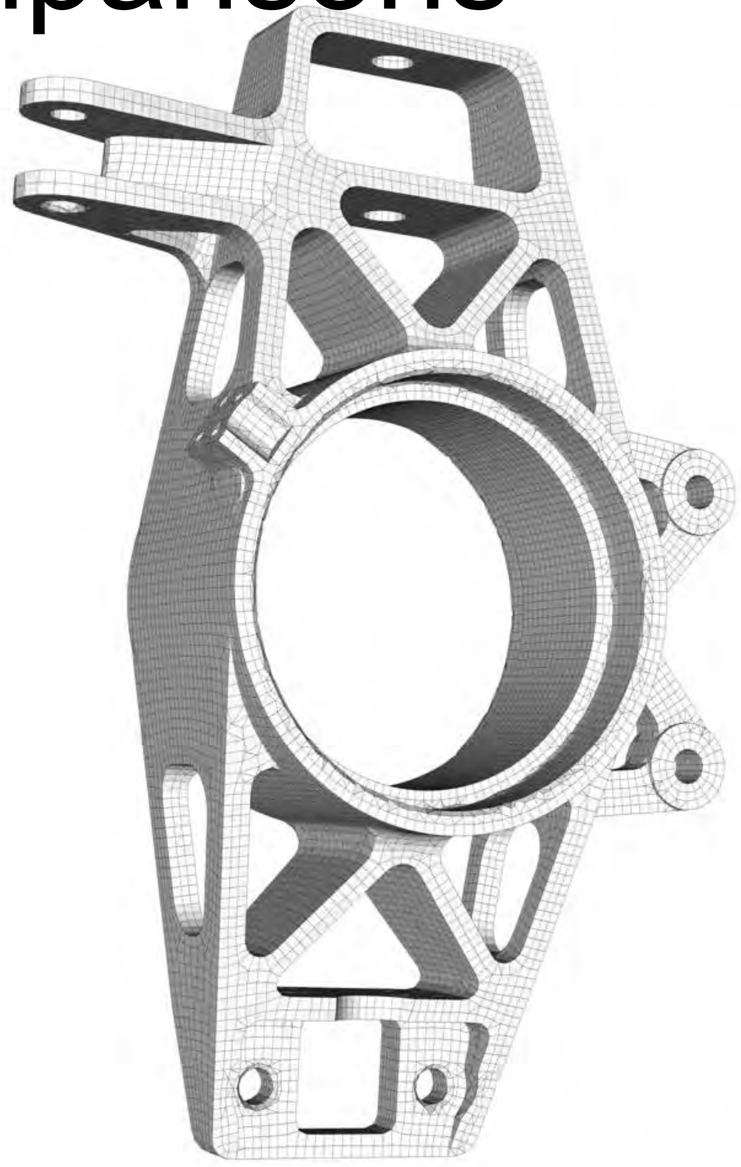


[Sokolov et al., 2016]

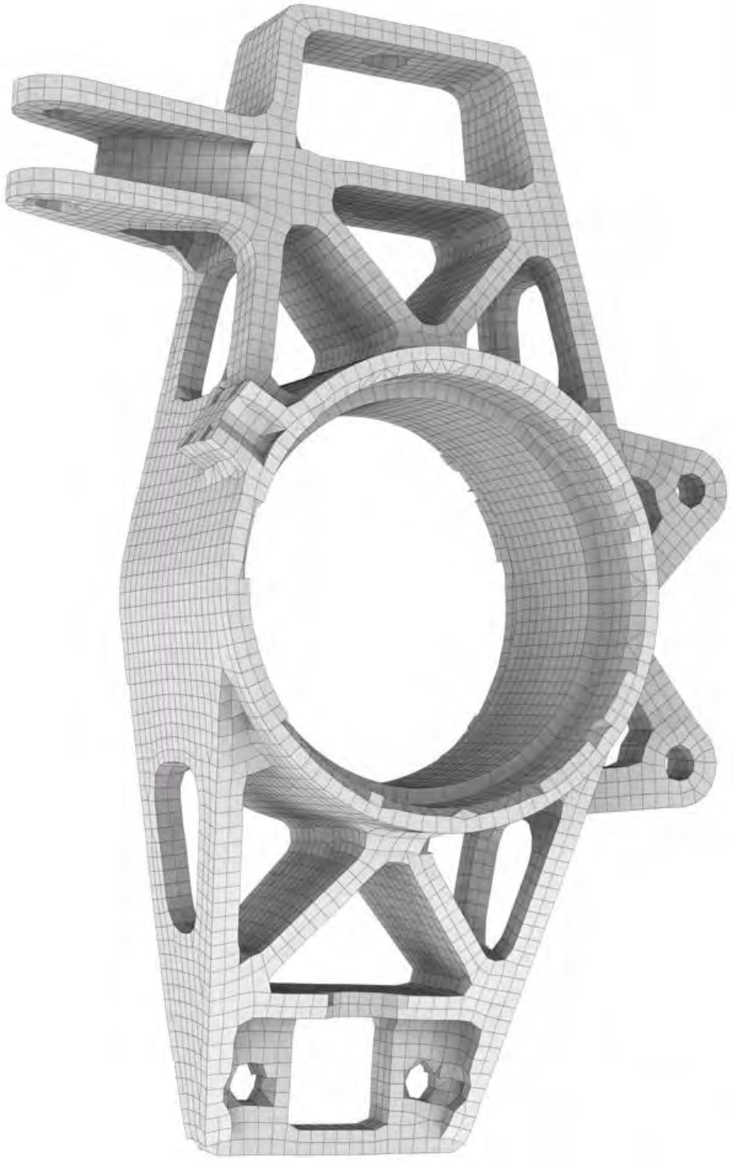


Ours

Comparisons

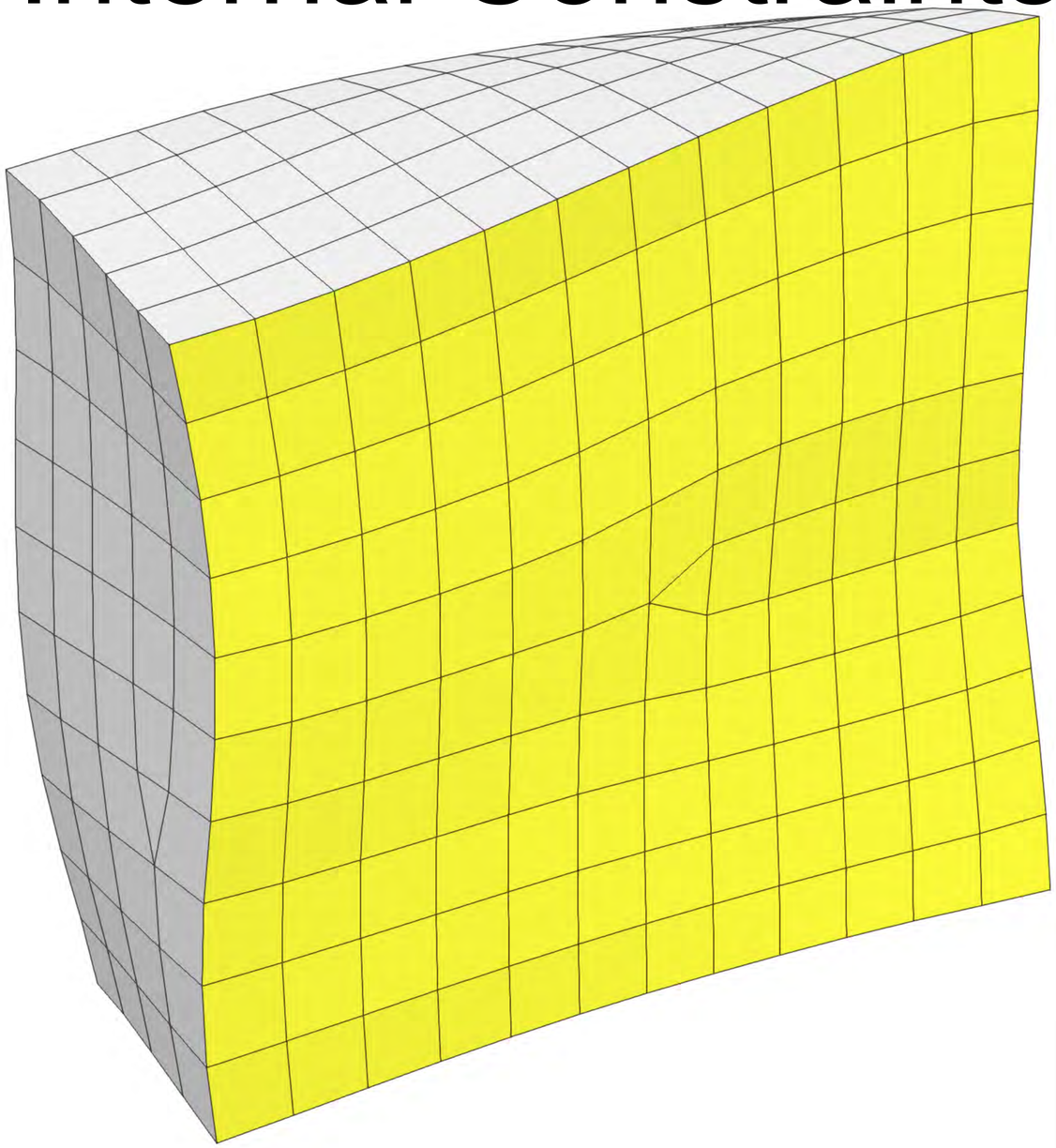


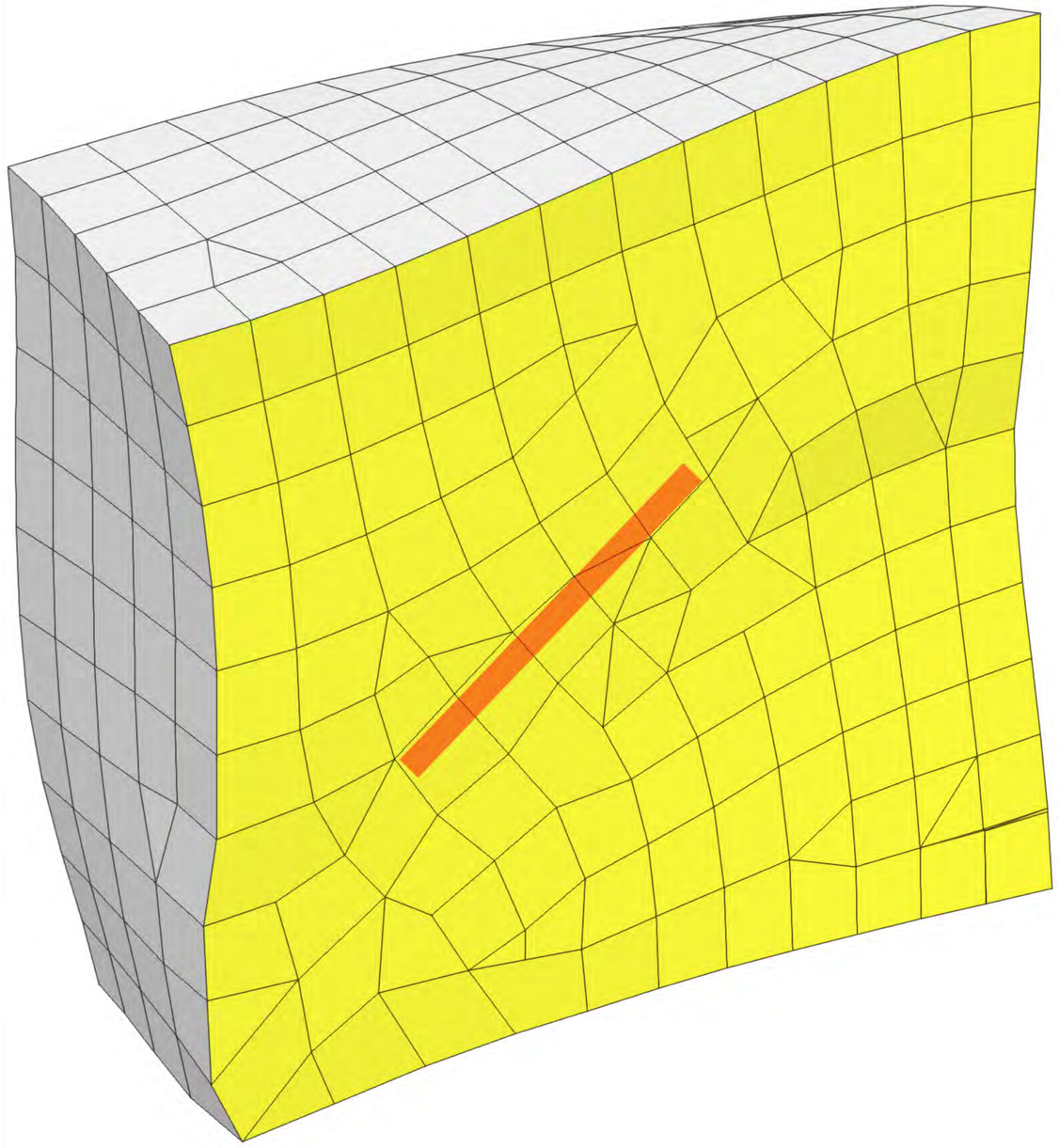
[Sokolov et al., 2016]



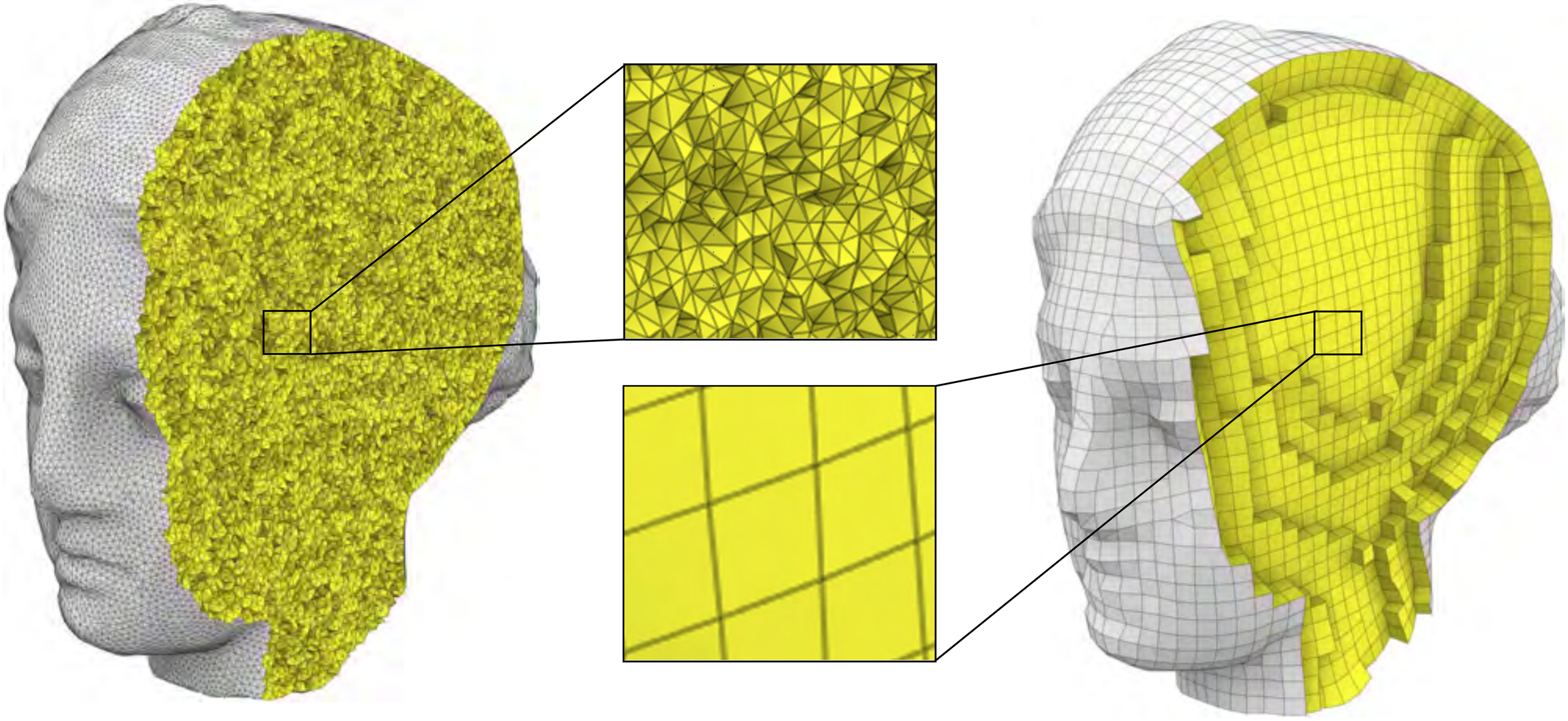
Ours

Internal Constraints



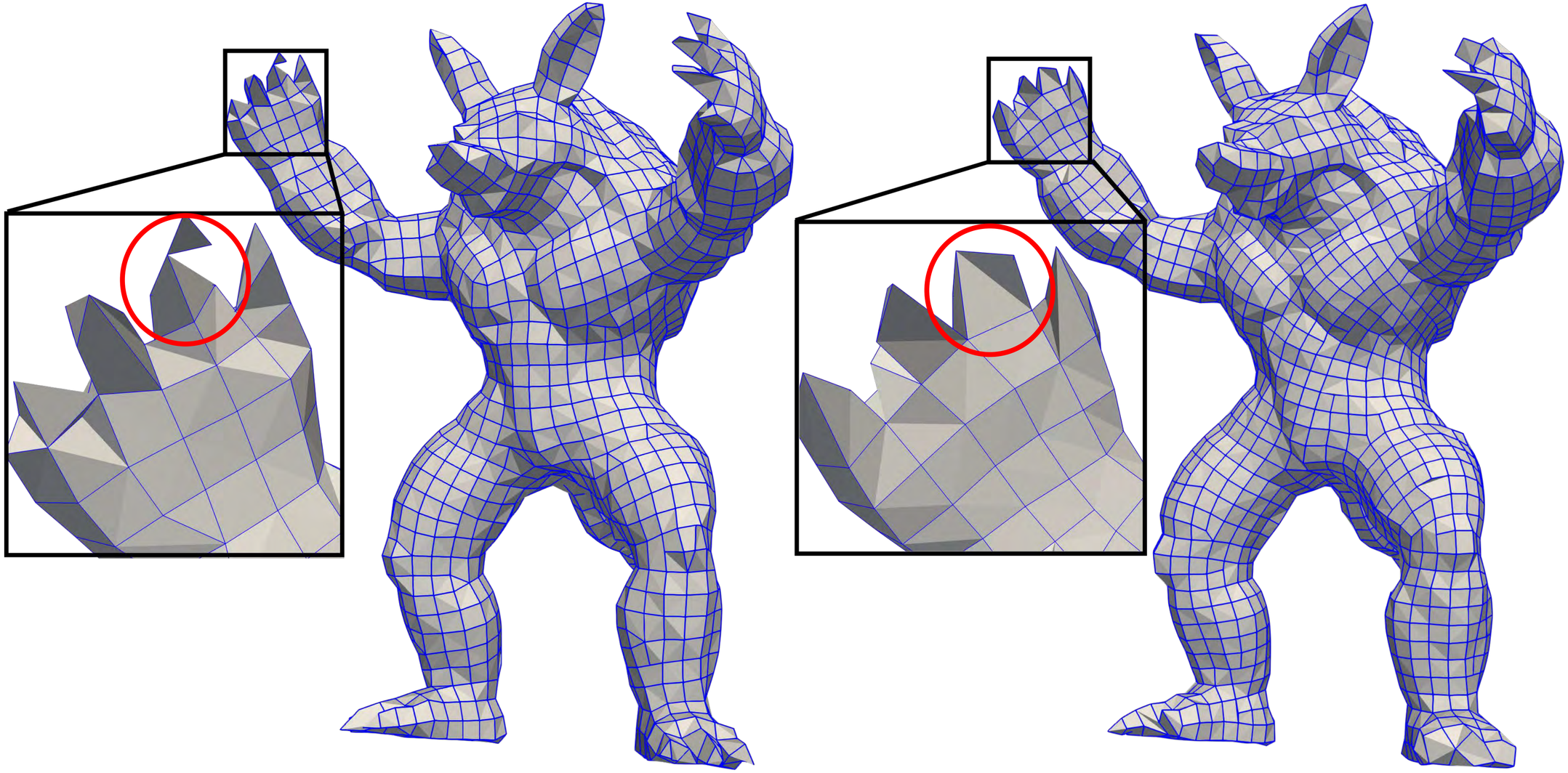


Scalability



Takes 84 minutes for 1.2 millions tets

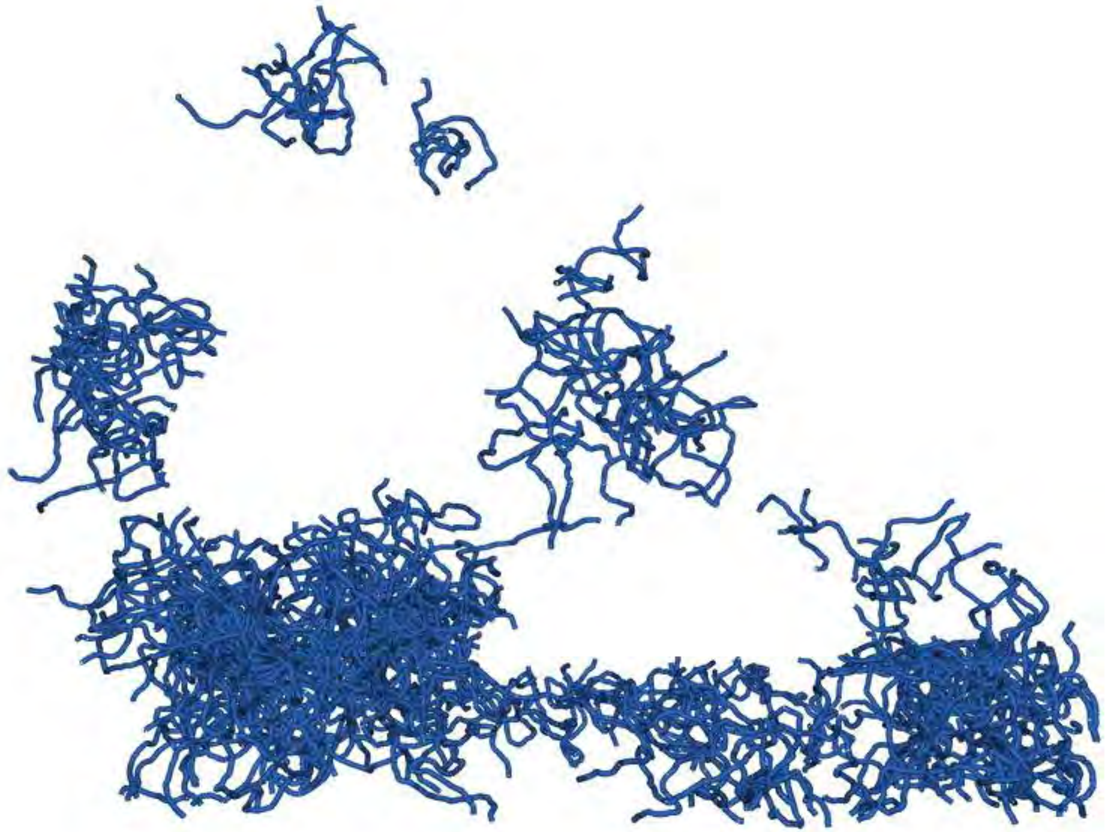
Surface Mesh Extraction



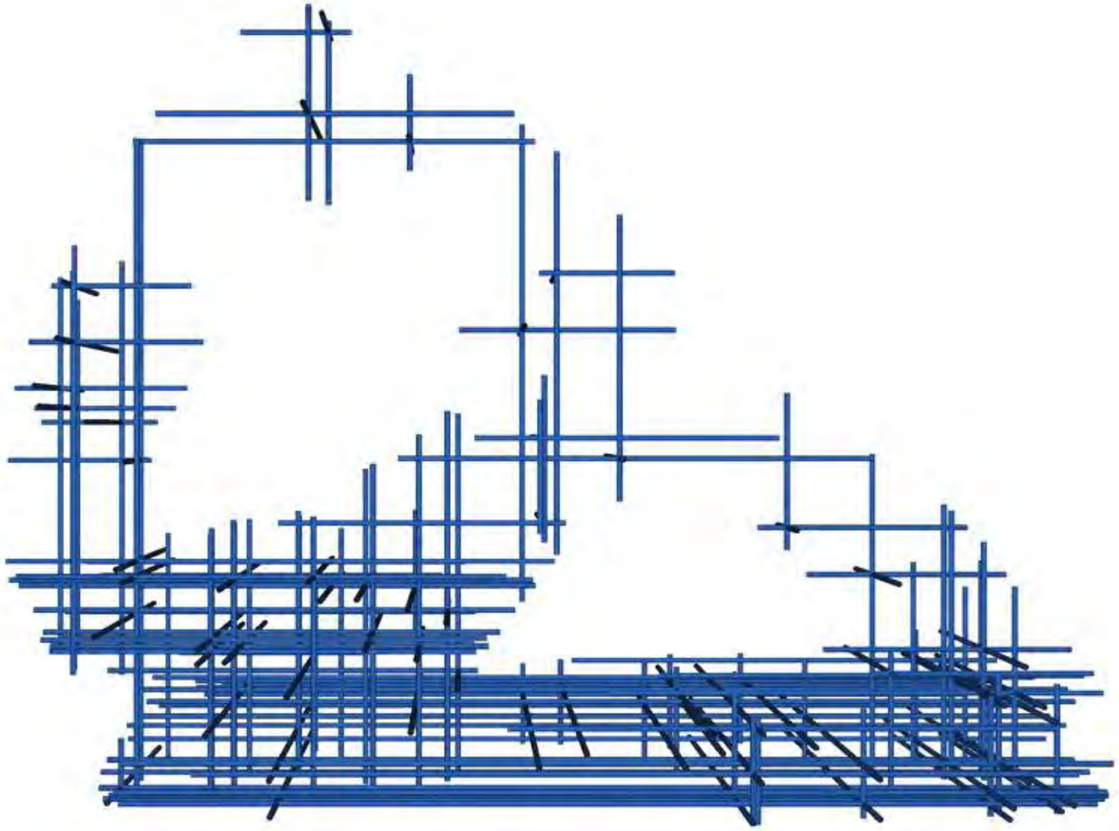
[Jakob et al., 2015]

Ours

Arbitrary Initialization

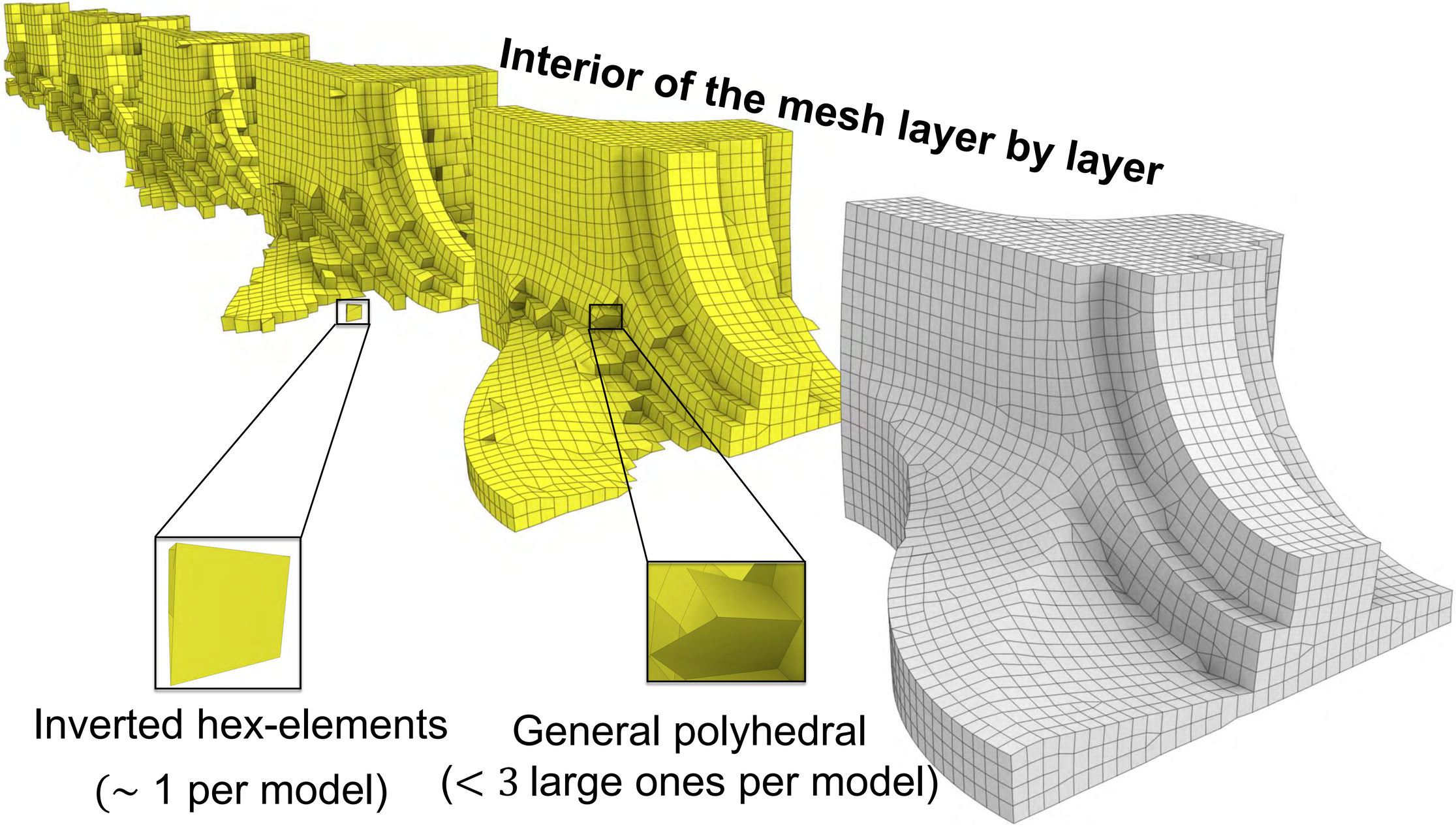


Random initialization



Constant initialization

Limitations



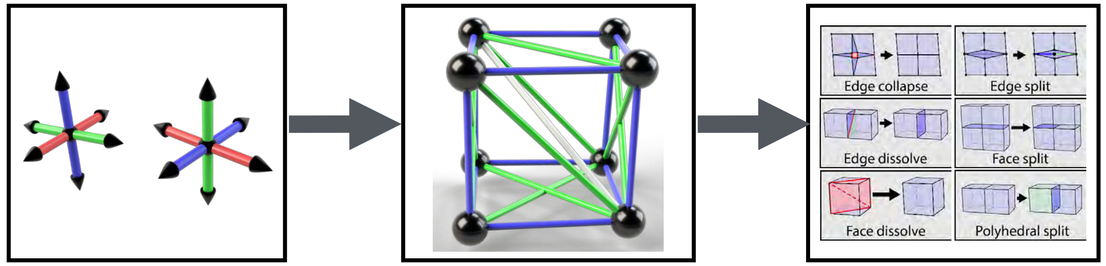
Inverted hex-elements

(~ 1 per model)

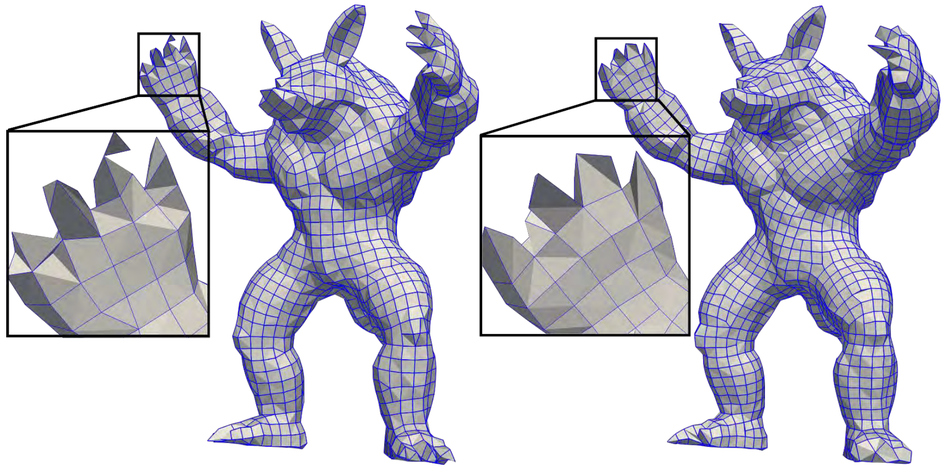
General polyhedral

(< 3 large ones per model)

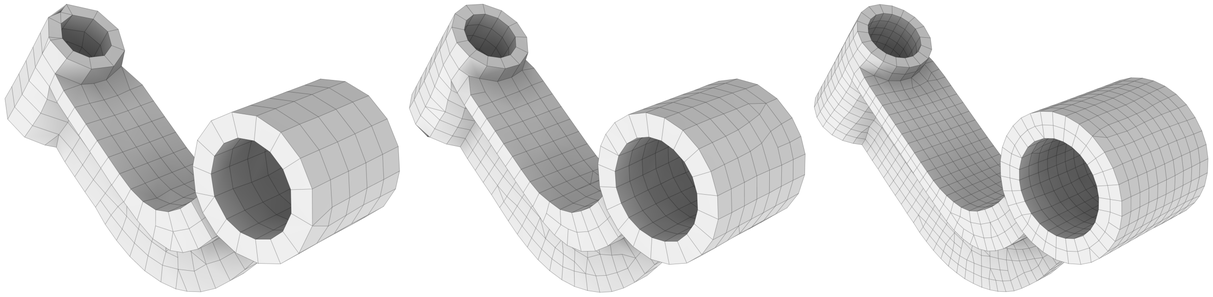
Summary



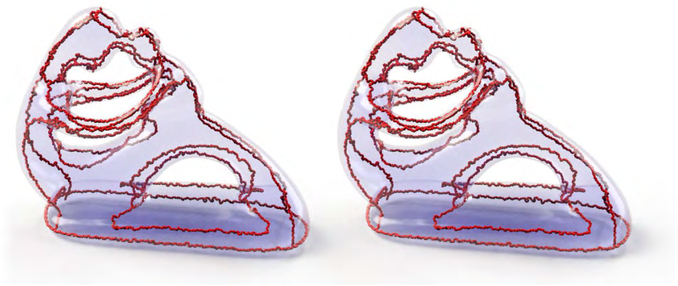
Simplicity



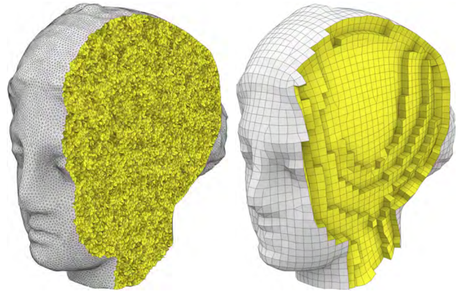
Topological guarantees



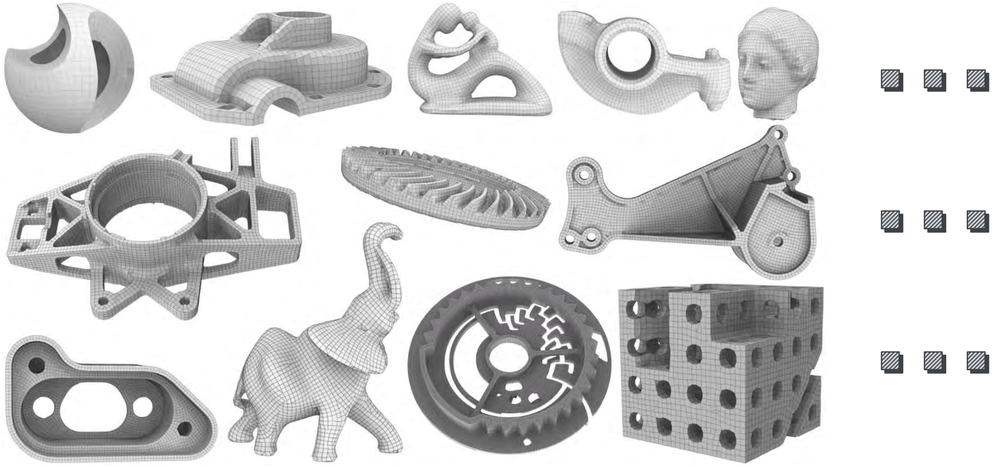
Single parameter



Arbitrary initializations



Scalability

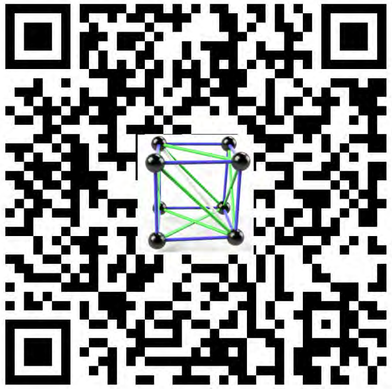


Robustness: 110 models

Thank you!

Acknowledgement

We thank Bruno Levy for providing the mechanical models for our experiments and comparisons, Yixin Hu for preparing video clips, Zhongshi Jiang for helping with the statistics, Alexandra Trif for coding a preliminary prototype, and Olga Sorkine-Hornung for the insightful discussions. The elephant, fertility, and rocker-arm models are part of the AIM@SHAPE Shape Repository. This work was supported in part by the NSF CAREER award 1652515 and by the MIUR project Dsurf.



Code: https://github.com/gaoxifeng/robust_hex_dominant_meshing