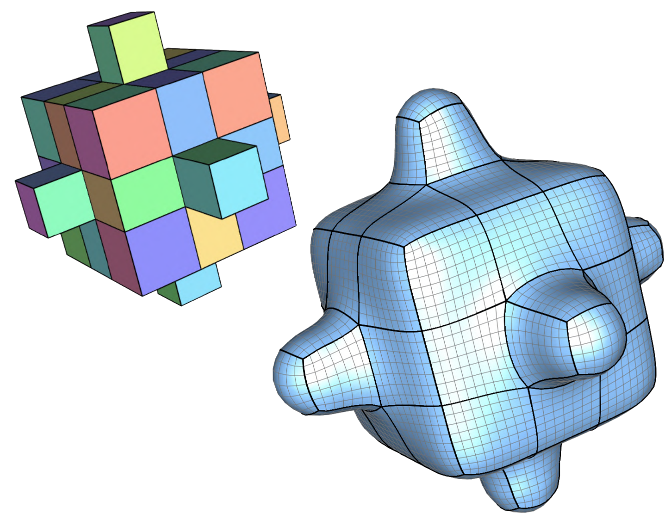
Eurographics Symposium on Geometry Processing 2016

Polycube Simplification for Coarse Layouts of Surfaces and Volumes

G. Cherchi[1], M. Livesu[2], R. Scateni[1]

[1] University of Cagliari, Italy
Department of Mathematics and Computer Science

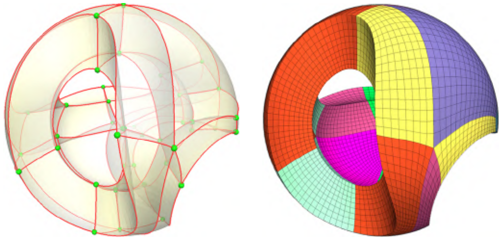
[2] CNR IMATI, Genoa, Italy
Institute for Applied Mathematics and Information Technologies



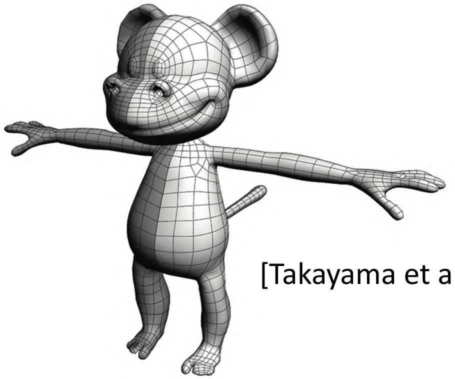
Berlin, June 20-24th

Singularities alignment

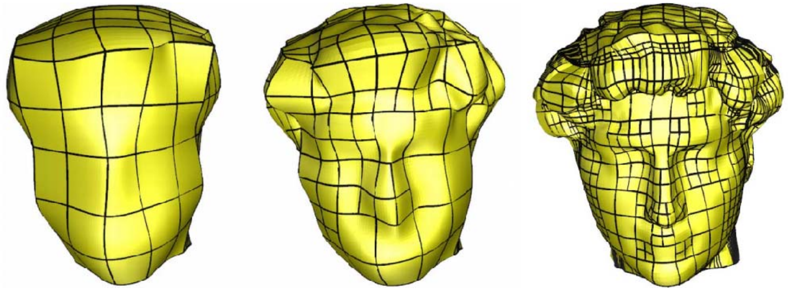
- Having a good singularities alignment is important in a number of applications:
  - High quality hex-meshes for simulation
  - High quality quad-meshes for animation
  - Higher order-meshing
  - Benefits for memory requirements
  - Benefits for performance speedup



[Gao et al. 2015]



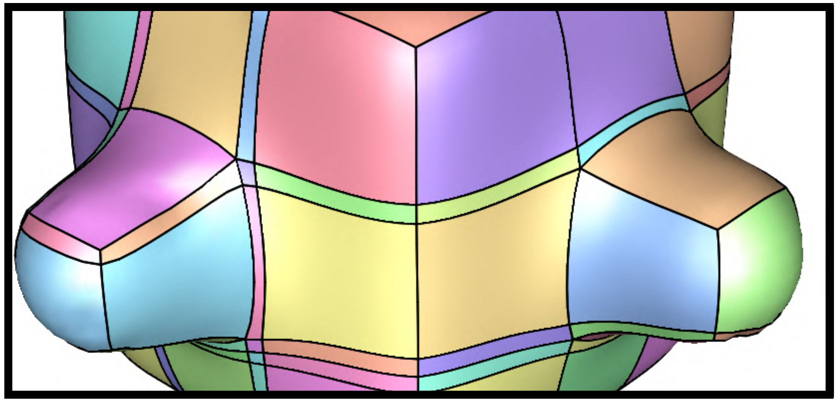
[Takayama et al. 2013]



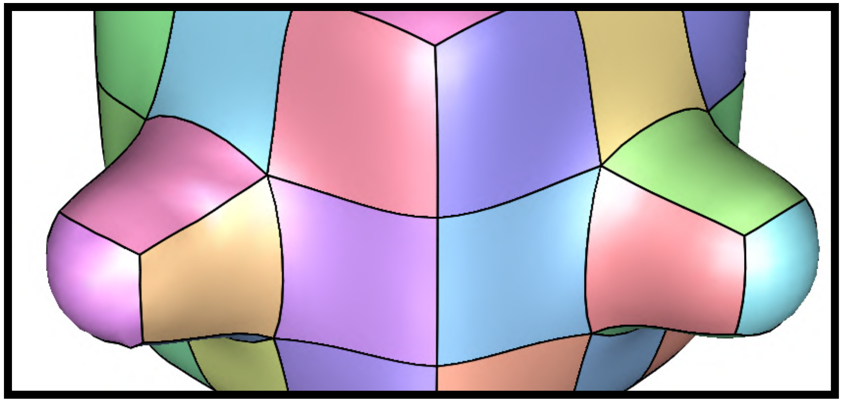
[Li et al. 2013]

The singularity misalignment problem

- Meshes with singularity misalignments → poor structure
- Meshes **without** singularity misalignments → good structure



The “**nearly miss**” problem



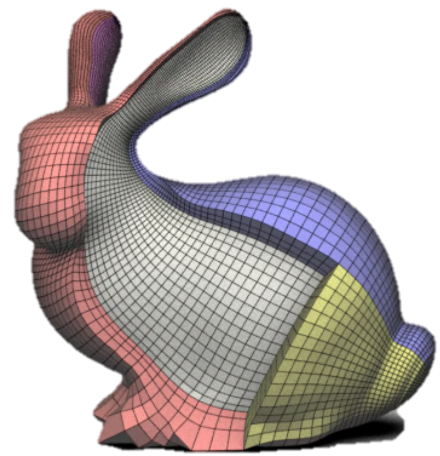
Aligned singularities

How can we solve the
singularity misalignment problem?

State of the art

- On volumes:
  - Iterative collapse of hexahedral sheets in the base-complex (Gao et al. 2015)[1]
    - Slow in some cases
  - User-designed harmonic function al. 2015)[2]
    - Often too coarse, user assisted

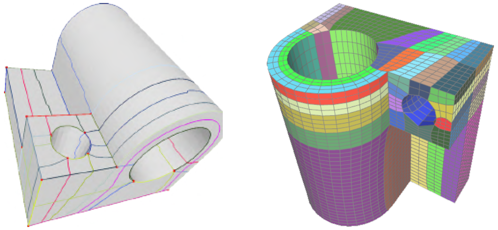
[1][Gao et al. 2015]



[2][Gao et al. 2015]

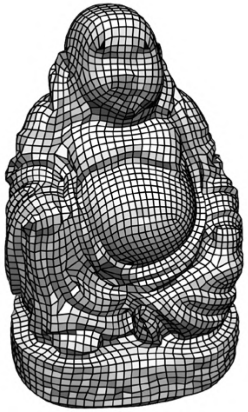
State of the art

- On surfaces:
  - Greedy approach using a graph of the separatrices of the mesh (Tarini et al. 2011)



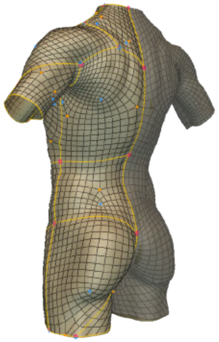
[Tarini et al. 2011]

  - Integer-grid maps (Bommes et al. 2013)



[Bommes et al. 2013]

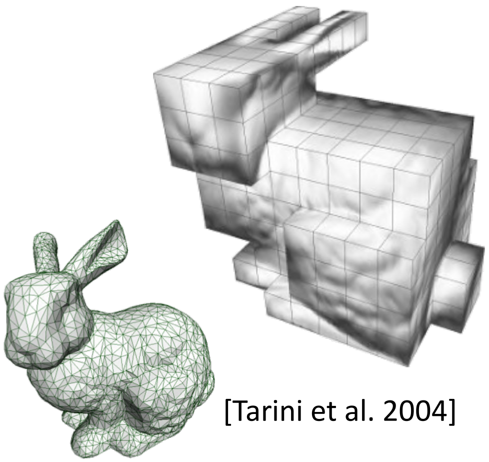
  - User assisted approaches



[Marcias et al. 2015]

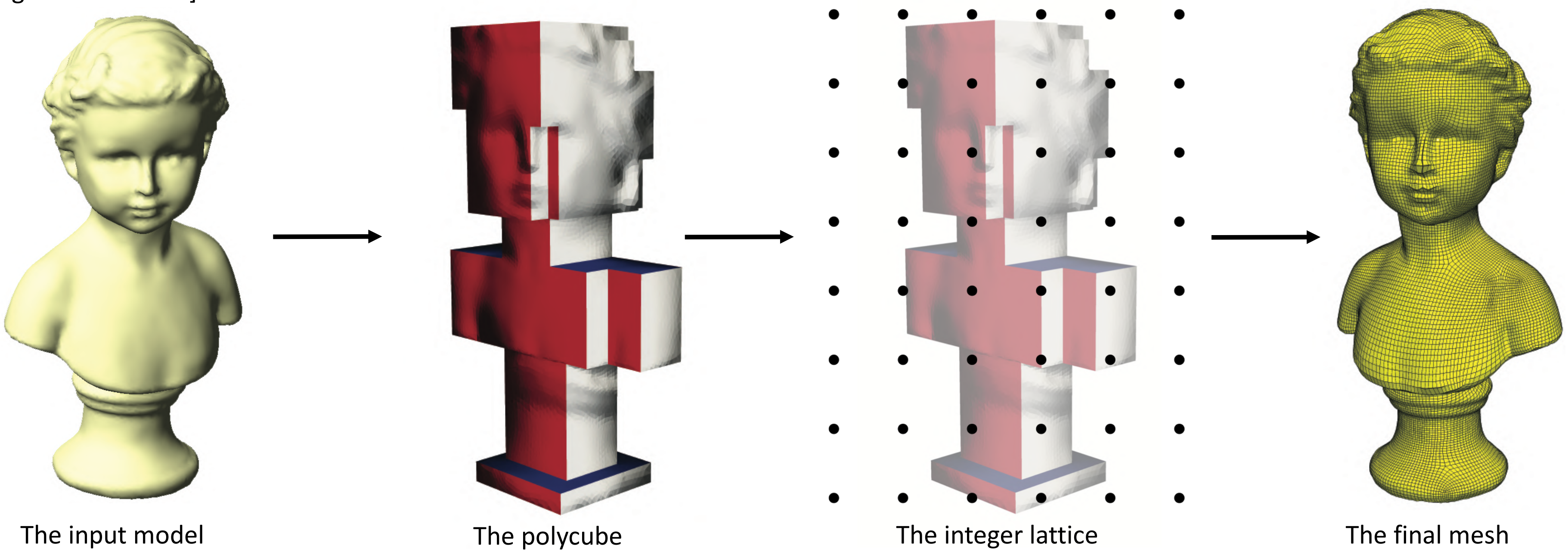
Polycubes

- Shapes transformed into a collection of connected cuboids
- Cuboid turned into mesh via gridding



[Tarini et al. 2004]

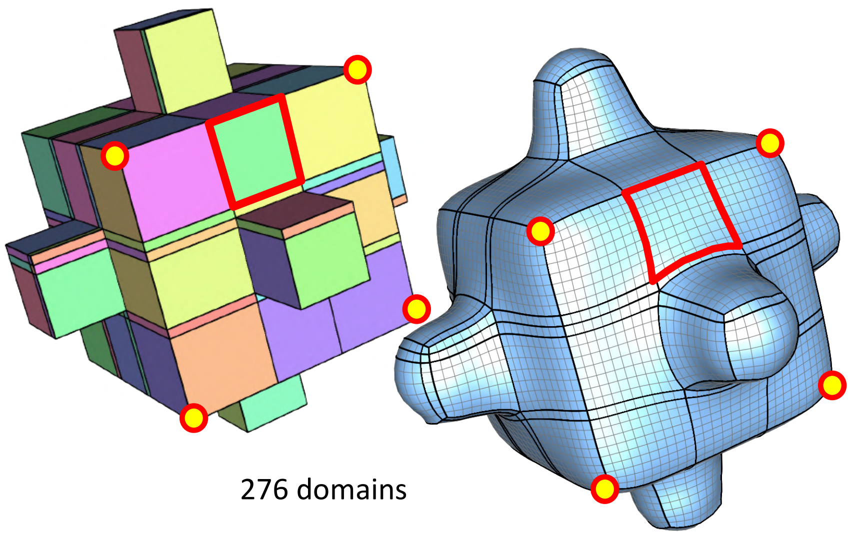
[Gregson et al. 2011]



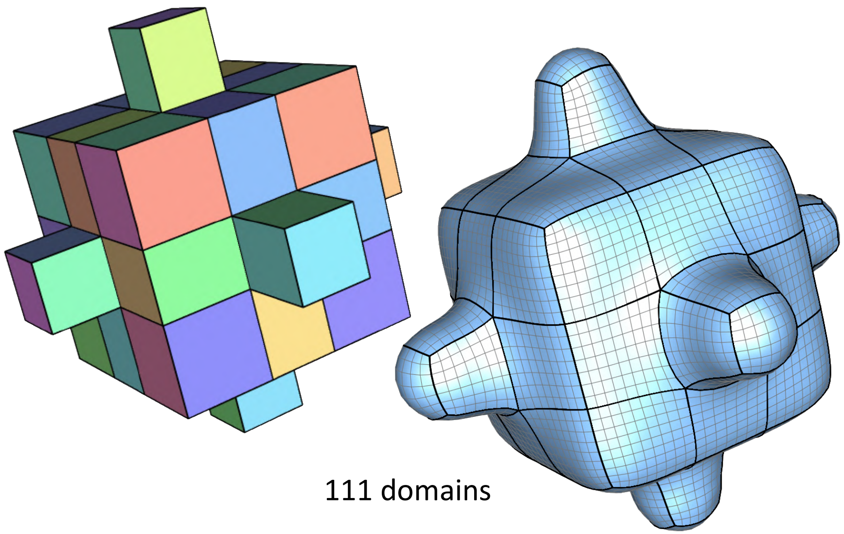
Polycube to mesh

- The polycube shape defines the structure of the final mesh
- Polycube corners are mesh singularities
- Polycube edges define the final base-complex

- Corners alignment influences the base-complex of the final mesh



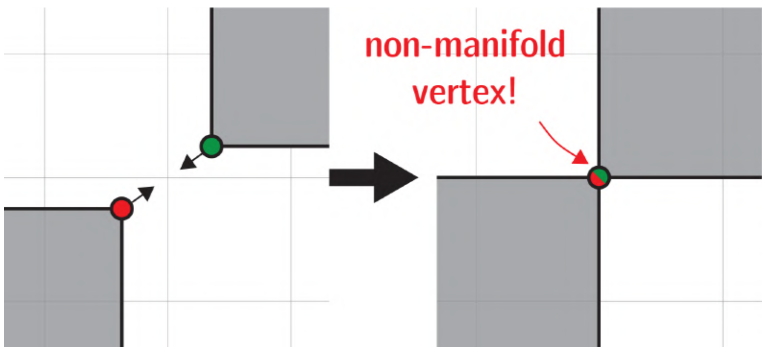
276 domains



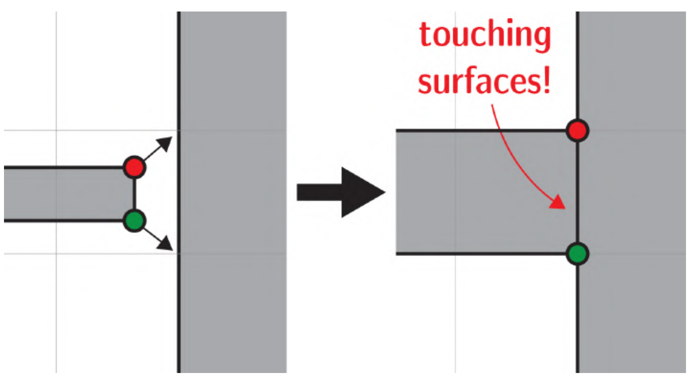
111 domains

First problem

- Polycube in an integer lattice
- Topological inconsistencies



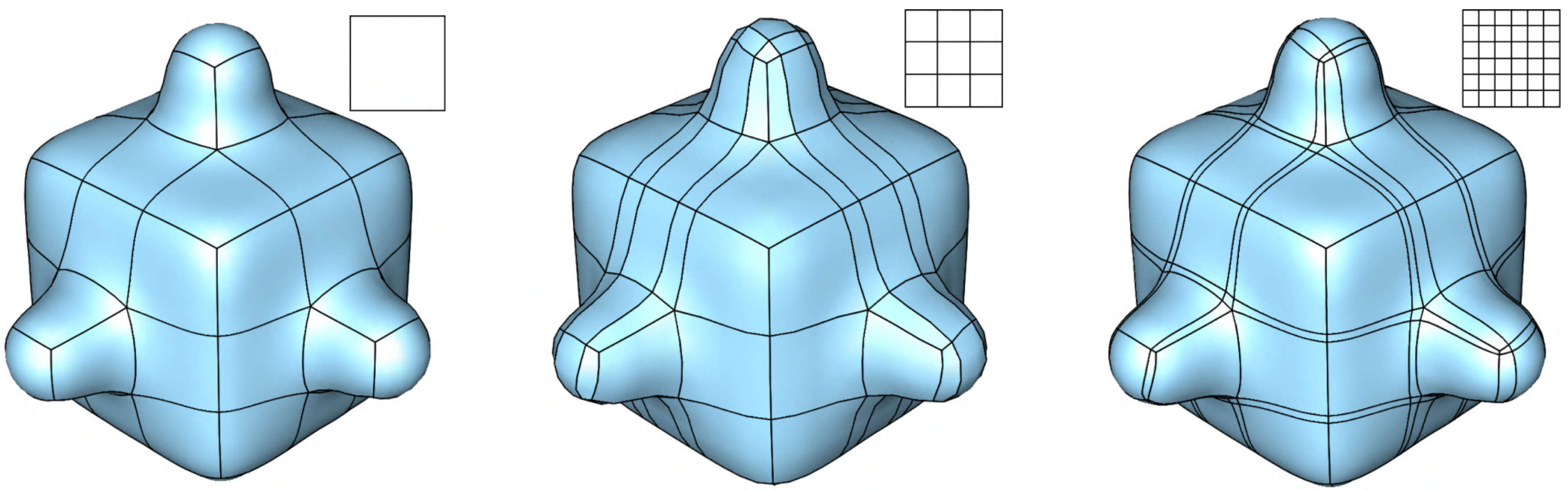
two corners map the same integer location



overlap between portions of the polycube

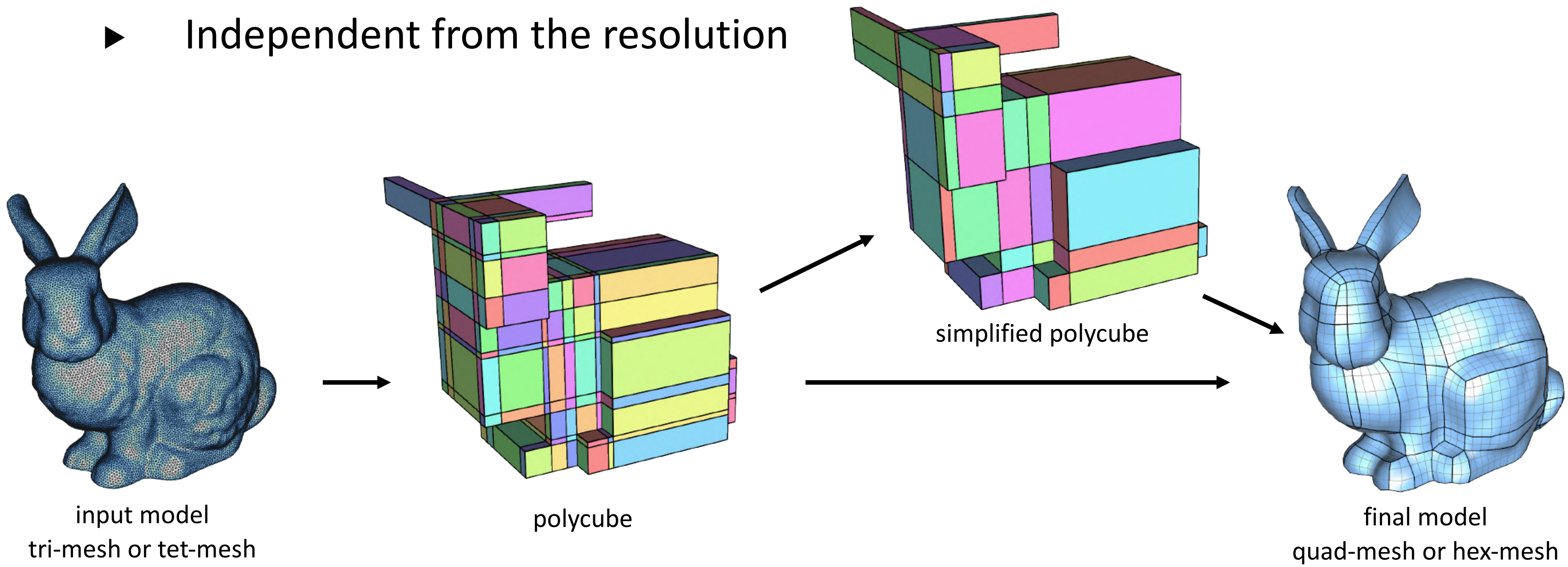
Second problem

- Polycube in an integer lattice
- Different lattice densities generate different mesh structures



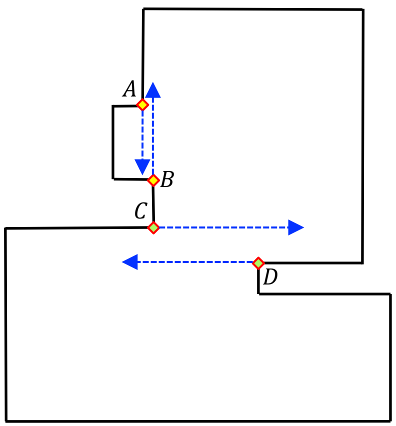
A new meshing pipeline

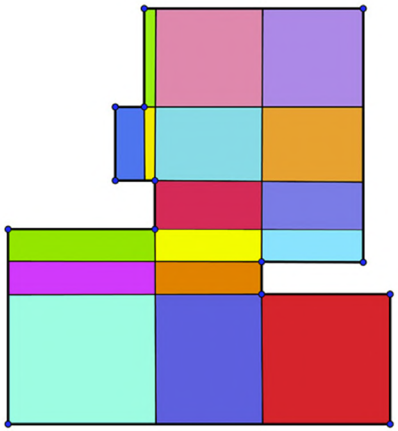
- A new step in the meshing pipeline:
  - Optimize the corners' position in the integer lattice
  - Optimize the base-complex decomposition
  - Avoid topological inconsistencies
  - Independent from the resolution



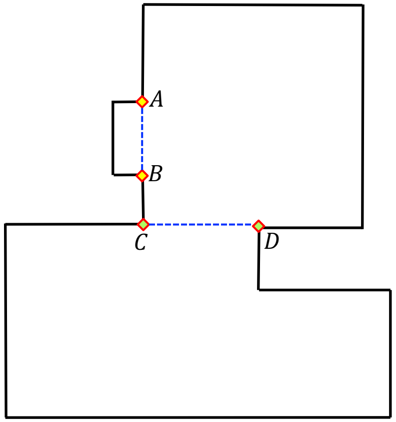
Our idea

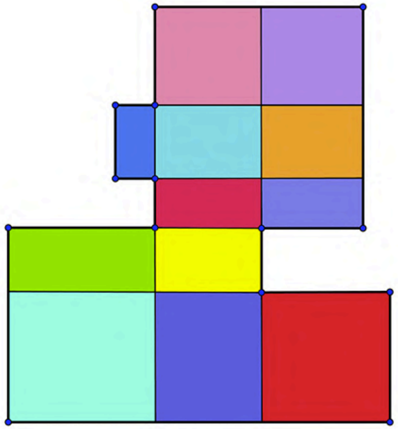
- Corner alignment





17 domains

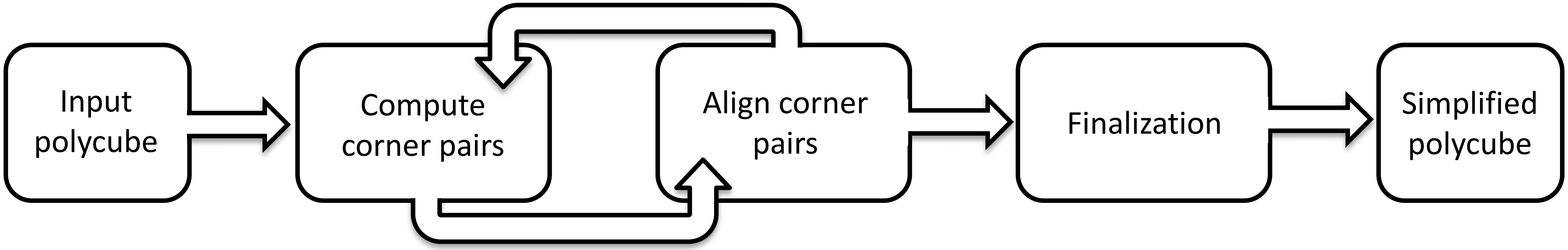


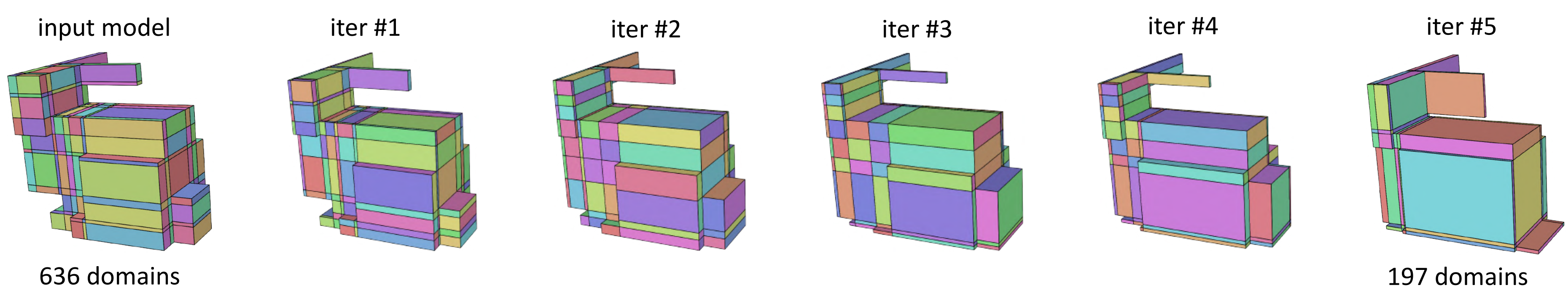


12 domains

The proposed method

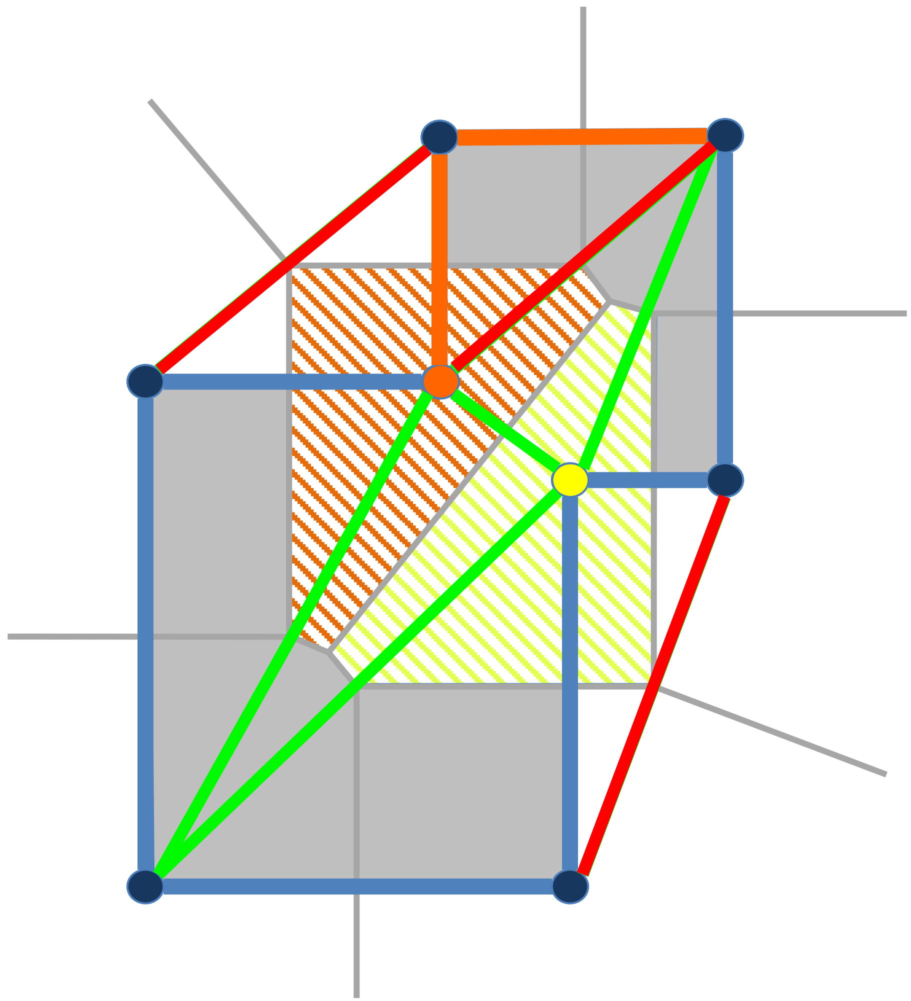
- Concise and iterative method:
  - Compute corner pairs
  - Align corner pairs





Corner pairing

- Voronoi based heuristic to find corner pairs (set A)
- Pruning of the graph of adjacencies
  - end-points of the same edge
  - end-points of adjacent edges
  - external adjacencies
- Splitting of A in A_x, A_y and A_z



Corner alignment

- A mathematical model
- Integer variables
- The objective function

$$\min E = \boxed{E_{align}(A)} + \boxed{\lambda} \cdot \boxed{E_{shape}}$$

$s.t.$

$\boxed{\text{structural constraints}}$

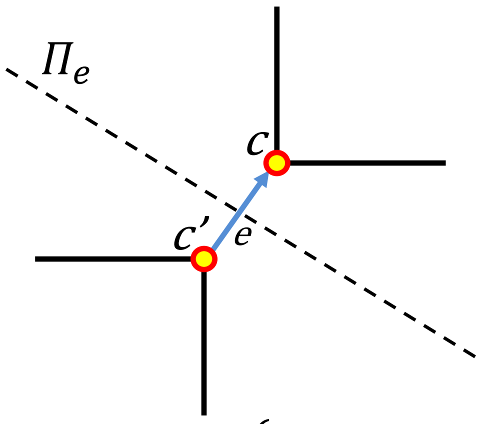
$$E_{align} = \sum_{(c,c')\in A_x} (c_x - c'_x)^2 + \sum_{(c,c')\in A_y} (c_y - c'_y)^2 + \sum_{(c,c')\in A_z} (c_z - c'_z)^2$$

$$E_{shape} = \sum_{c} \|c - \tilde{c}\|^2$$

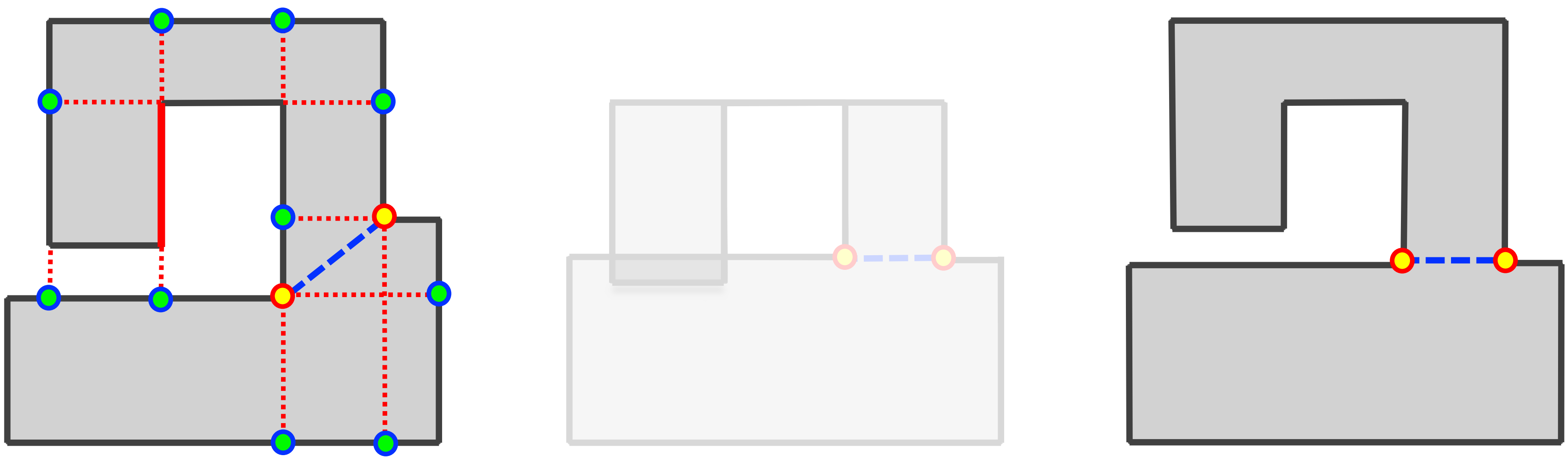
λ=*scaling factor*

Structural constraints

- A set of **linear** constraints:
  - Collinearity of end-points
  - Avoid edge collapse (length >= 1)
  - Avoid corner collapse
  - Dummy vertices and edges



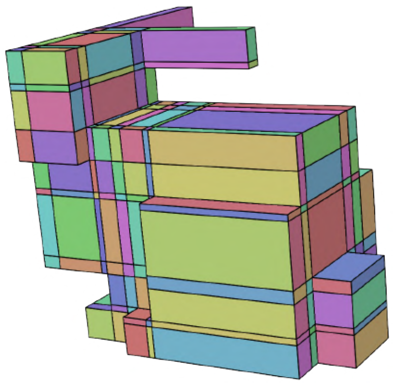
$$\begin{cases} \Pi_e(c) > 0 \\ \Pi_e(c') < 0 \end{cases}$$

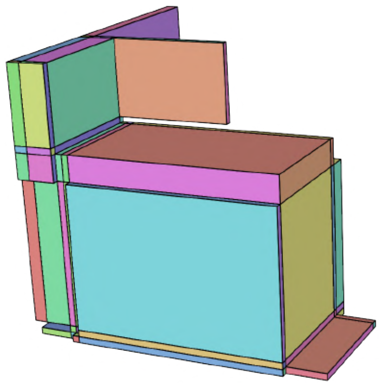


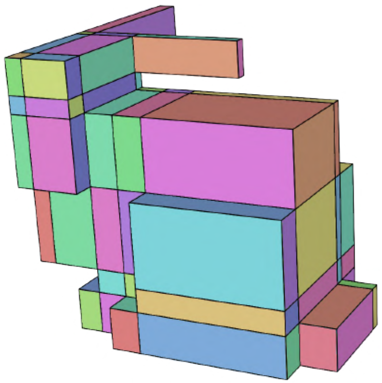
Finalization

- Final simplification step ($min\ E = \ E_{shape}$)

$$E_{shape} = \sum_{c} \|c - \tilde{c}\|^2$$

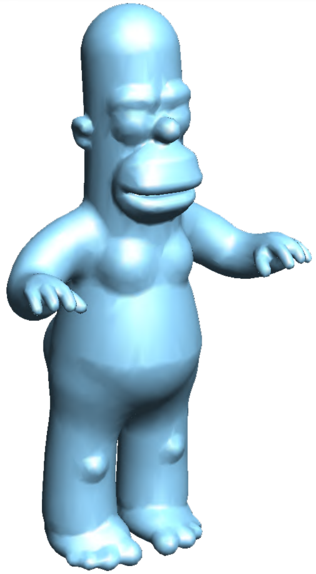

original model


model at convergence

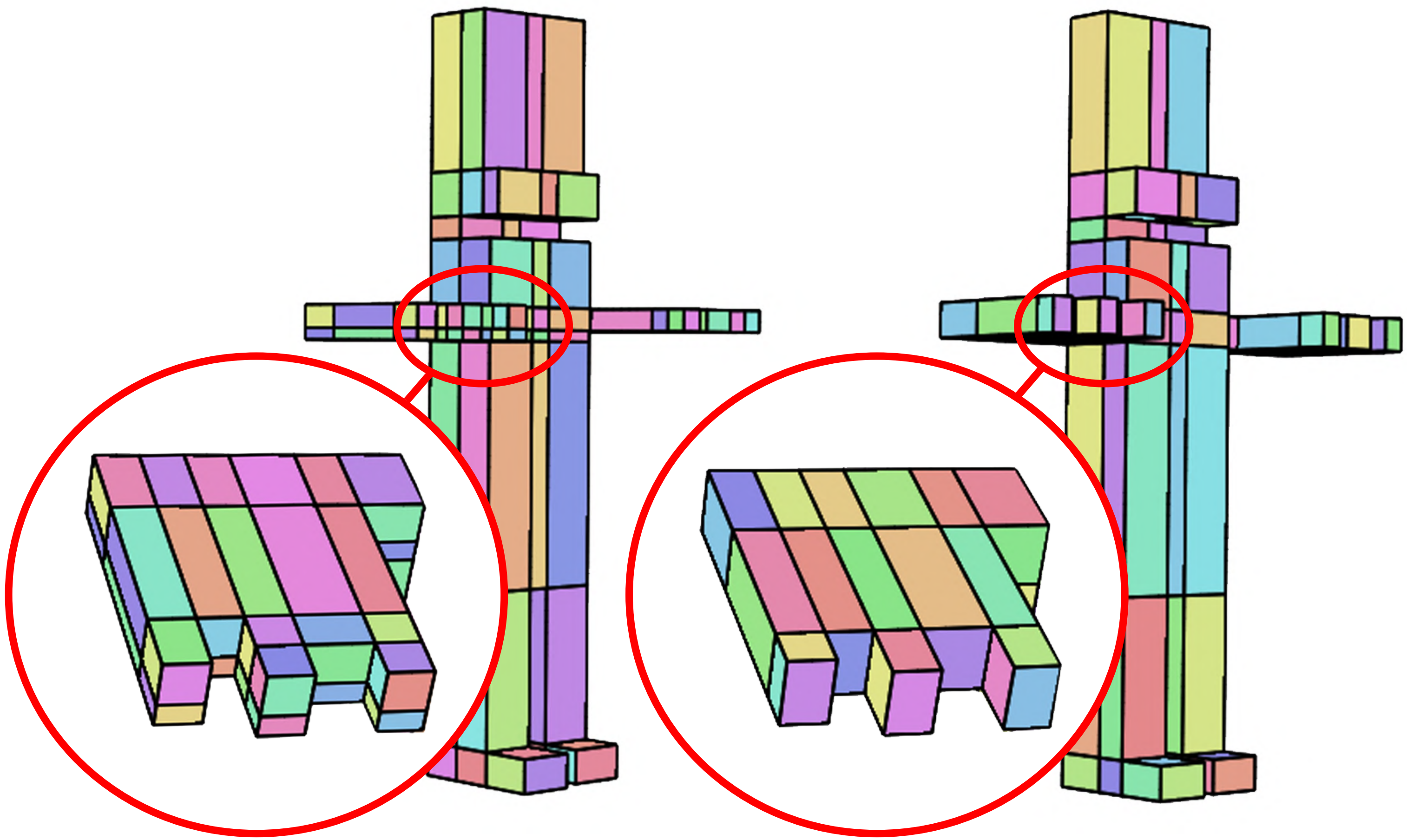

model after the final step

- We fit the input polycube into our simplified structure solving a simple Laplacian problem $\nabla P = 0$
  - For surfaces we use the cotangent weights
  - For volumes we use the 3D mean value coordinates (Floater et al. 2005)

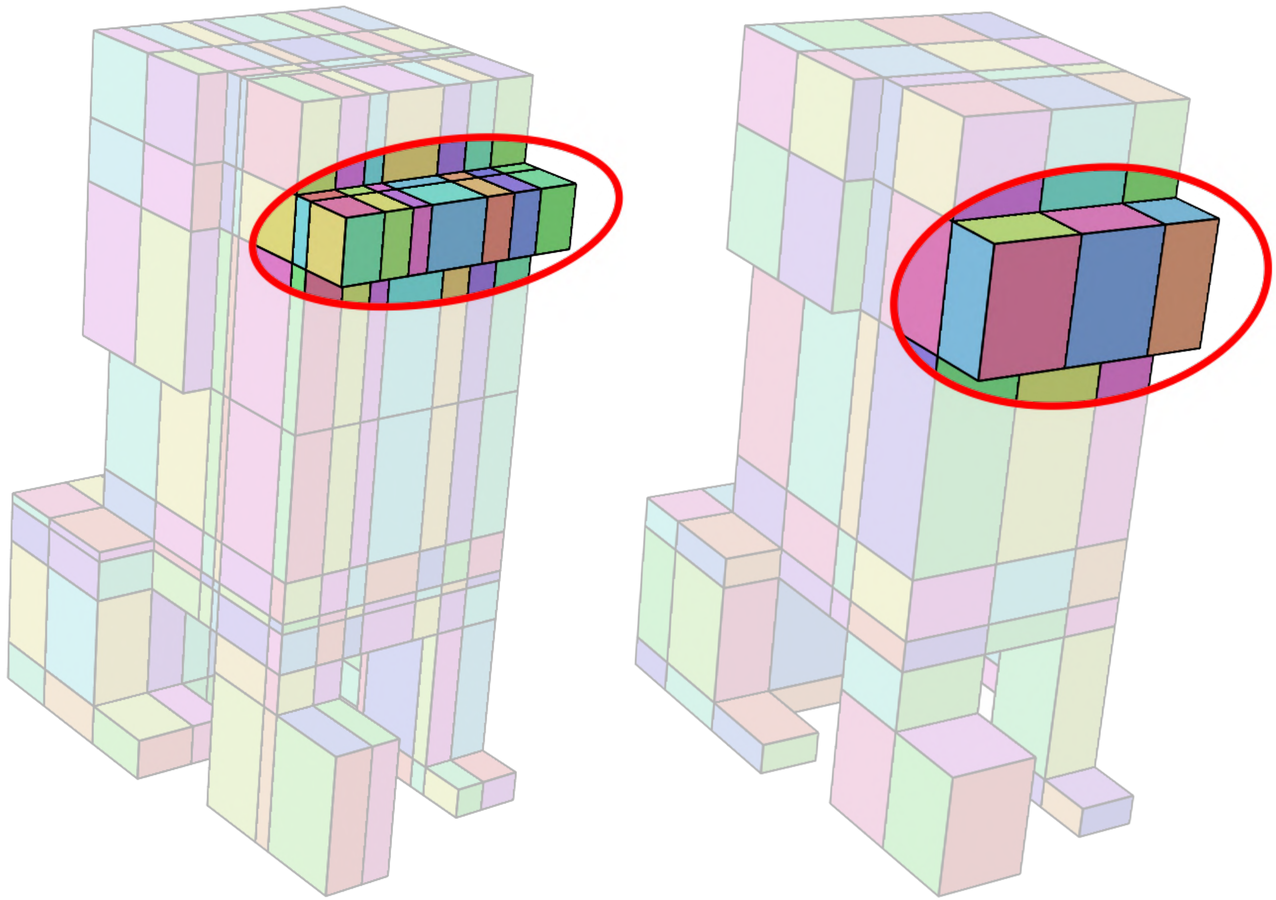
Results



INPUT:	260 dom
OUTPUT:	202 dom
GAIN:	22%
TIME:	14.7 sec



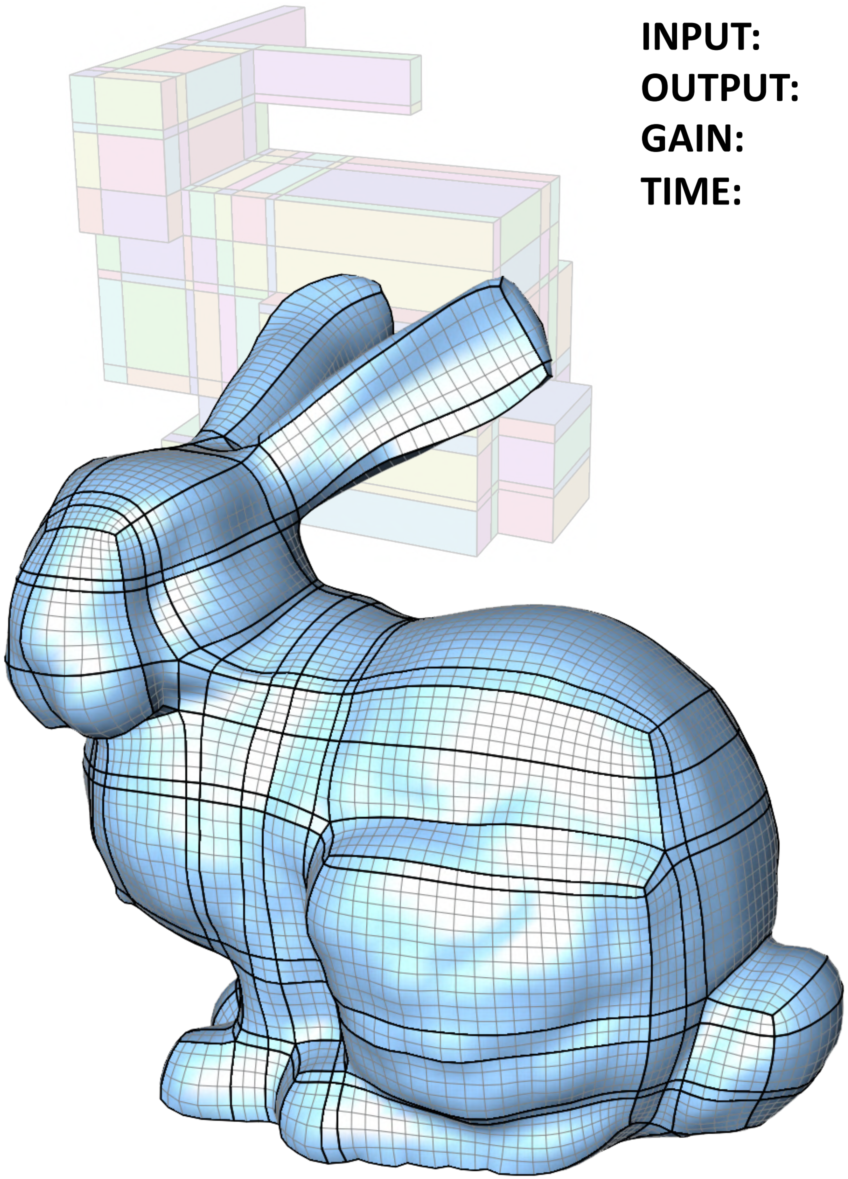
Results

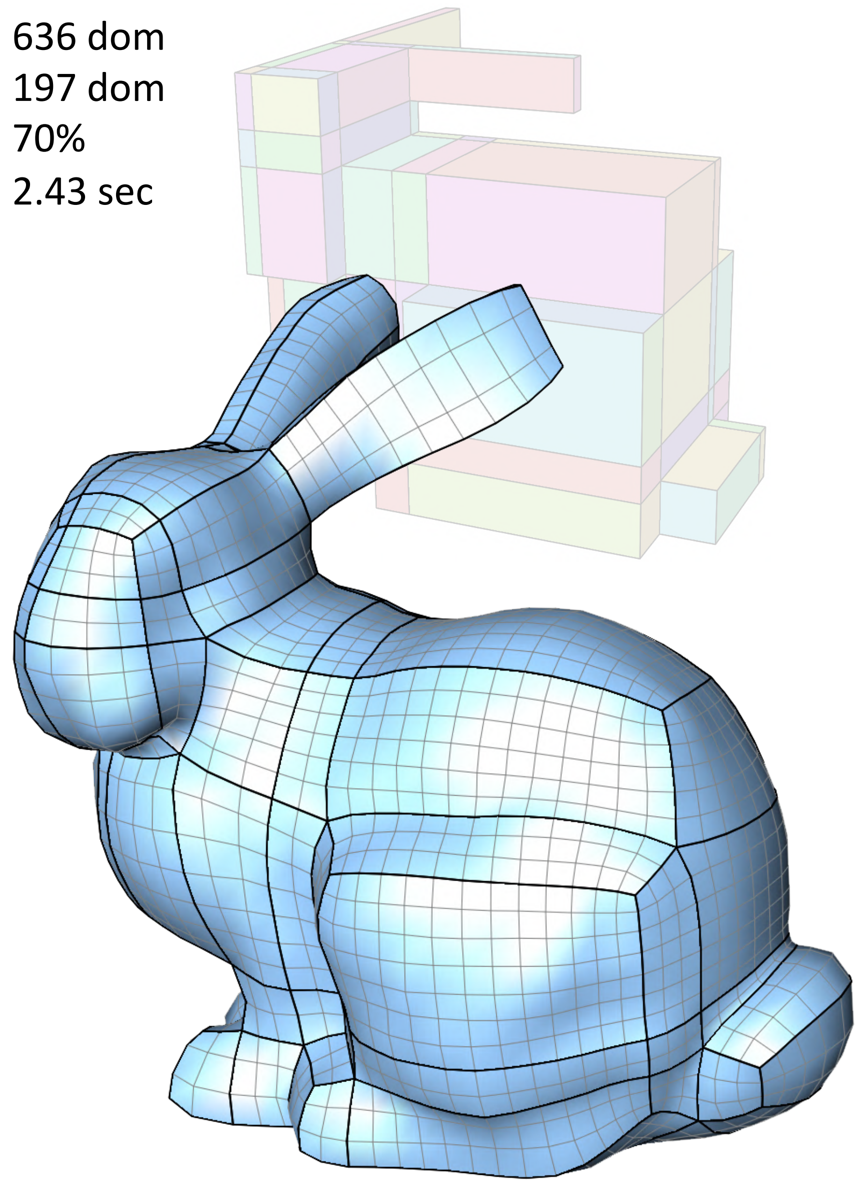


INPUT:	248 dom
OUTPUT:	192 dom
GAIN:	23%
TIME:	8.13 sec

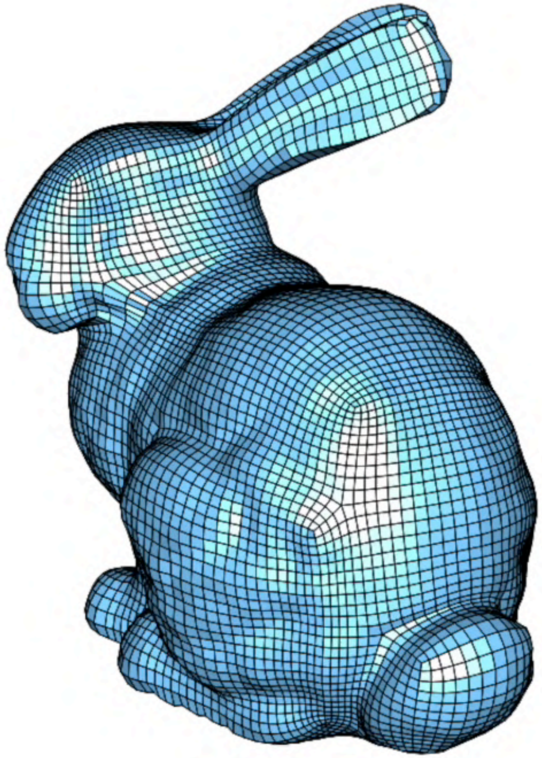
Results

INPUT:	636 dom
OUTPUT:	197 dom
GAIN:	70%
TIME:	2.43 sec



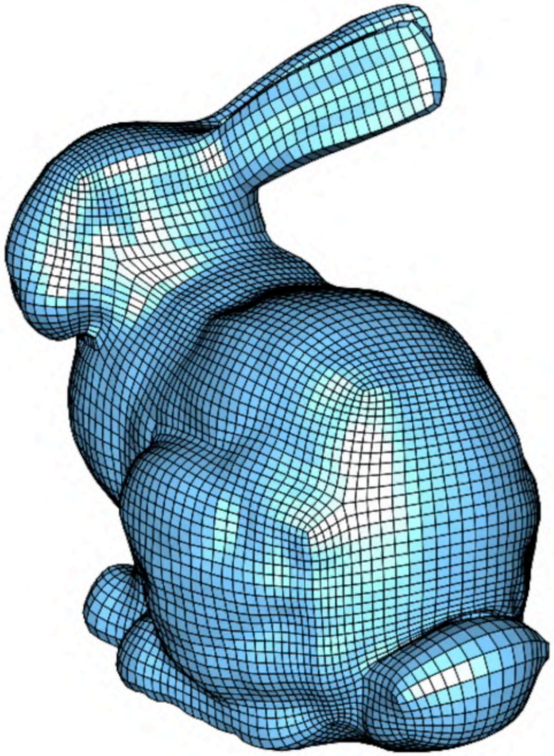


Results



Without simplification

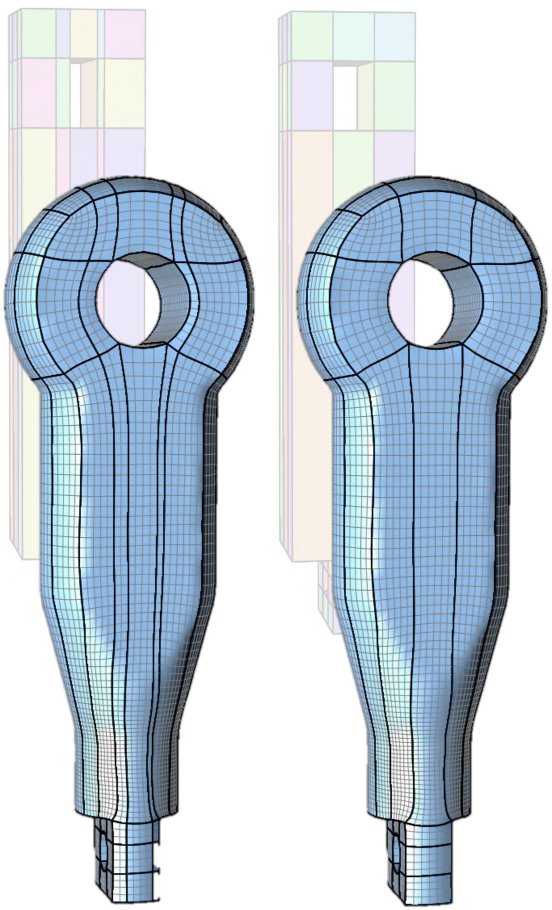
min / avg SJ
.180 / .966



With simplification

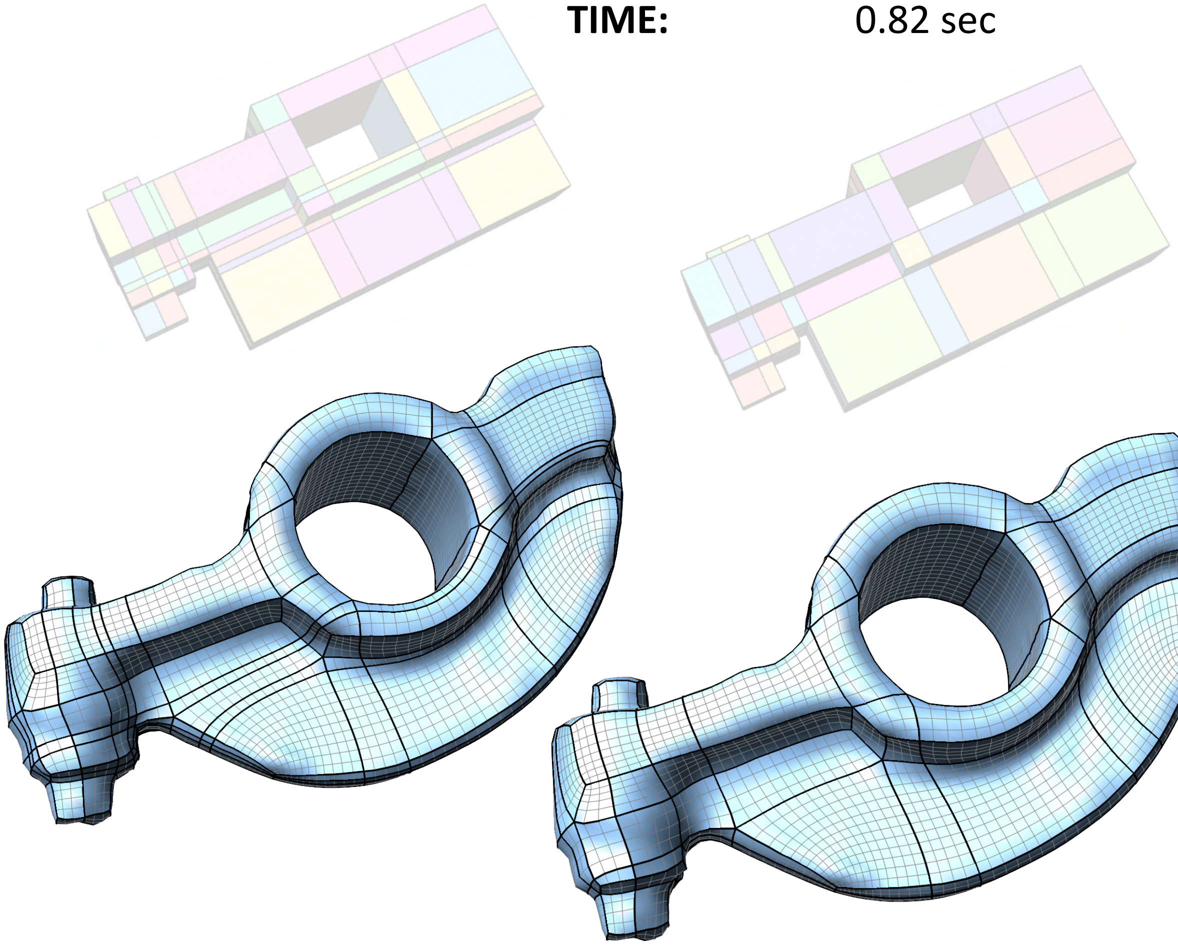
min / avg SJ
.205 / .935

Results

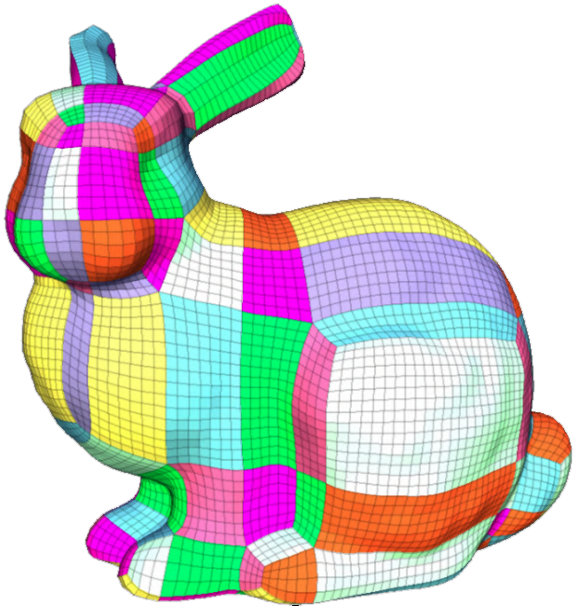


INPUT: 184 dom
OUTPUT: 114 dom
GAIN: 38%
TIME: 1.33 sec

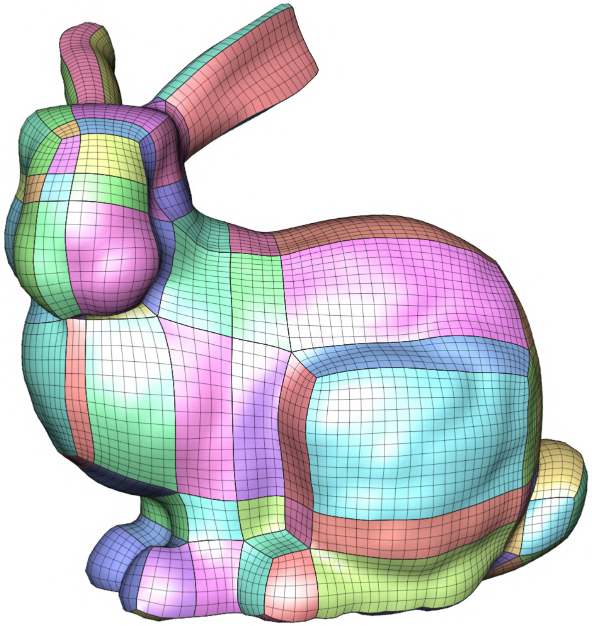
INPUT: 684 dom
OUTPUT: 332 dom
GAIN: 51%
TIME: 0.82 sec



Comparison



[Gao et al. 2015]



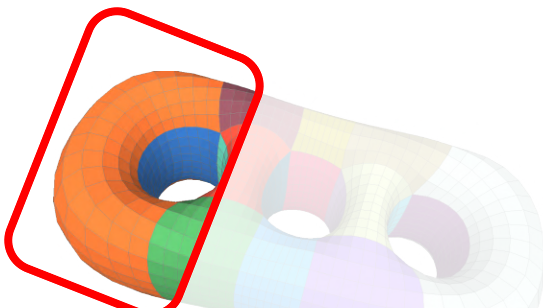
[Ours]

INPUT:	580 dom		**INPUT:**	636 dom
OUTPUT:	194 dom		**OUTPUT:**	197 dom
GAIN:	67%		**GAIN:**	70%
TIME:	*from 1 m to ½ h*		**TIME:**	2.43 sec

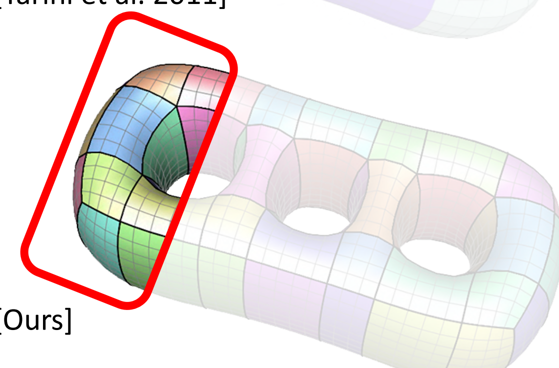
- Comparable results
- Time → two orders of magnitude lower

Limitations

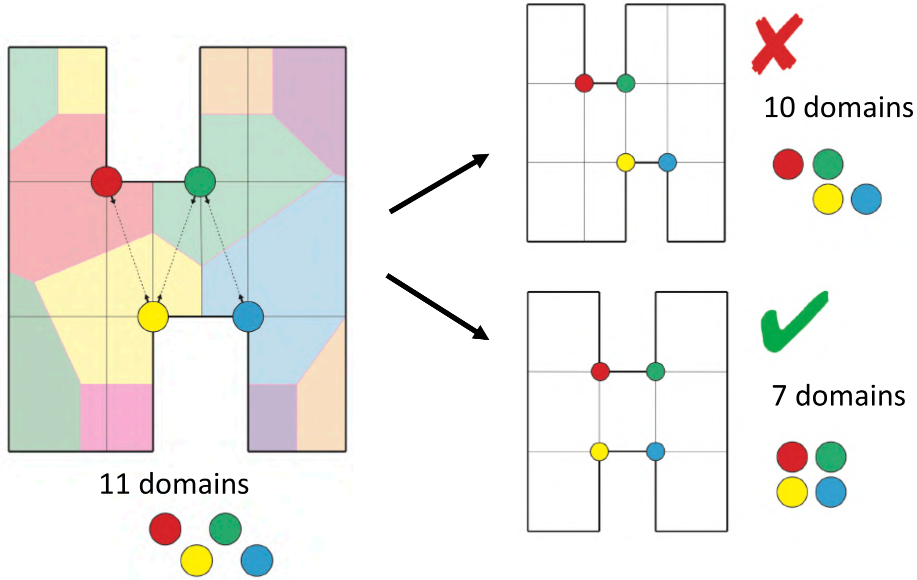
- Corner pairing
- Map distortion
- Domains

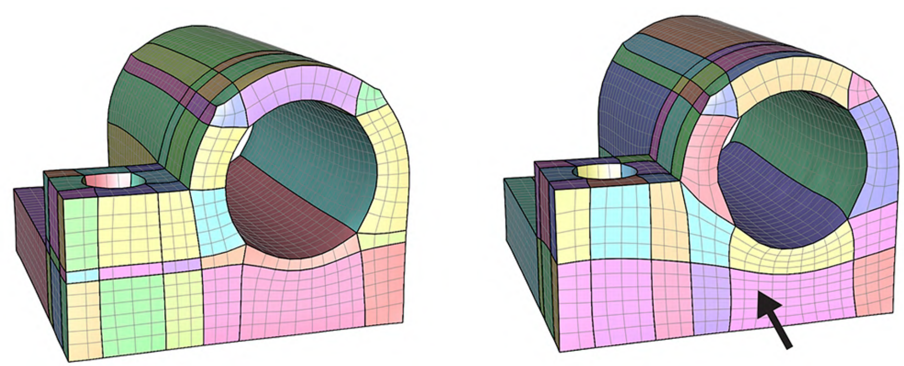


[Tarini et al. 2011]



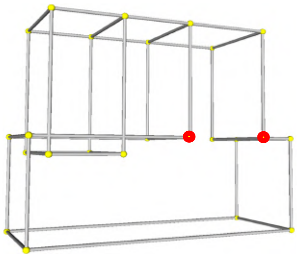
[Ours]

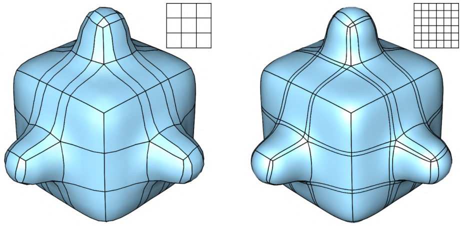


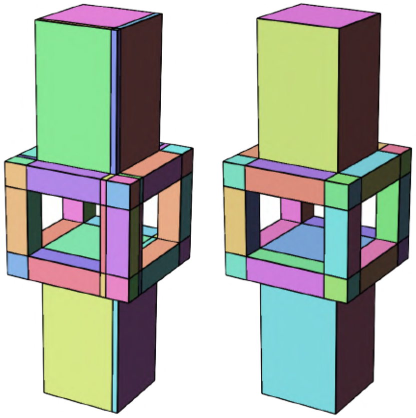


Conclusions

- We have proposed a novel method for the polycube simplification that:
  - Avoids topological inconsistencies
  - Is independent from the sampling resolution
  - Reduces the number of domains
  - Is simple to insert in the meshing pipeline
  - Is general (surface and volumetric meshes)
  - Is very fast







Thanks!