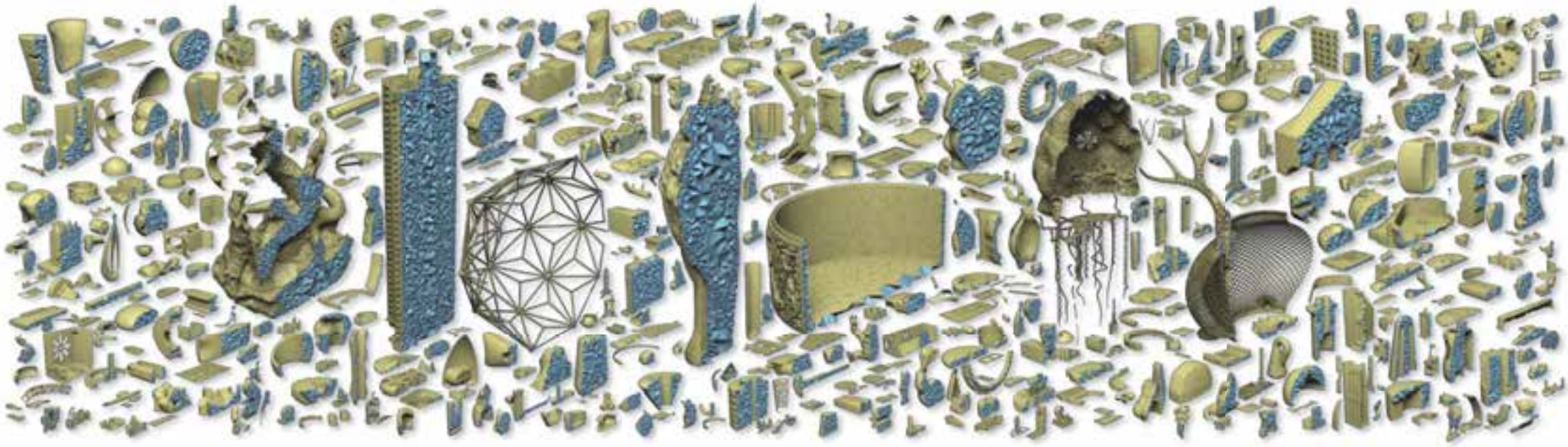
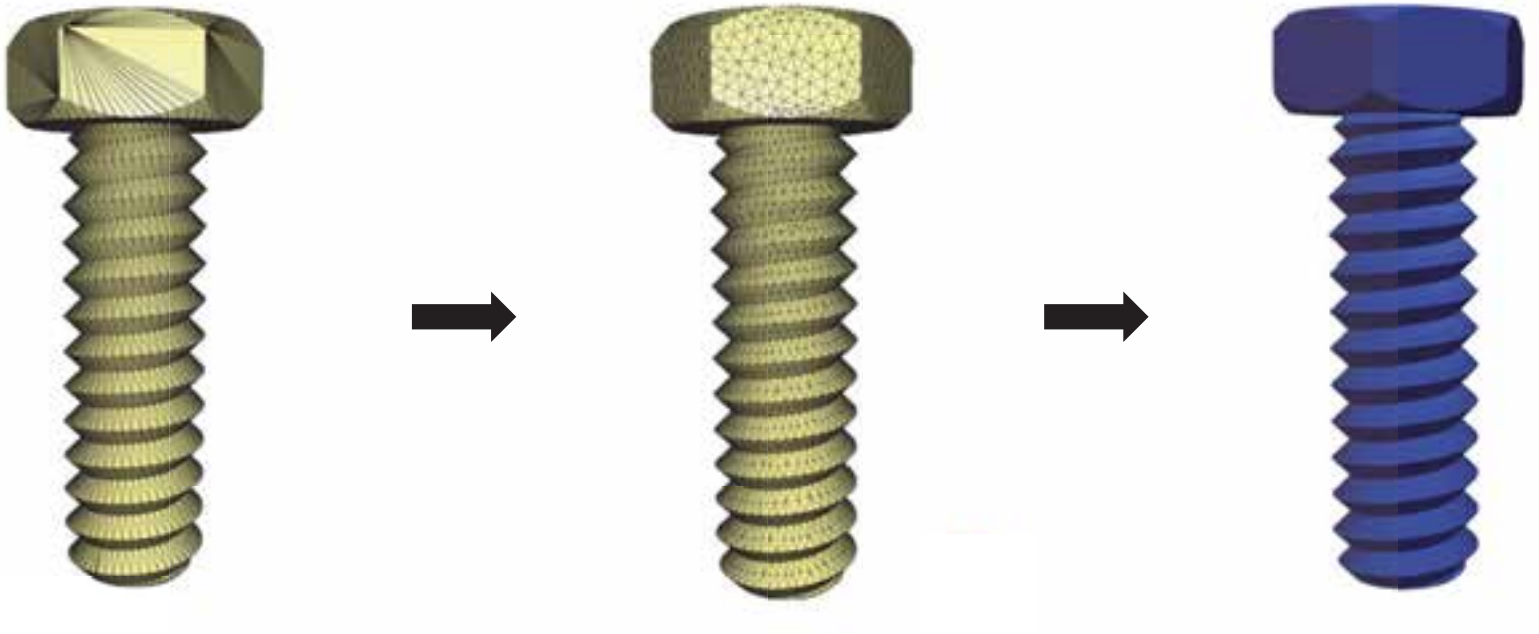


Tetrahedral Meshing in the Wild

Yixin Hu[1], Qingnan Zhou[2], Xifeng Gao[1],
Alec Jacobson[3], Denis Zorin[1], Daniele Panozzo[1]

1 NYU COURANT INSTITUTE OF MATHEMATICAL SCIENCES
2 Adobe
3 Computer Science UNIVERSITY OF TORONTO

Tetrahedral Meshing



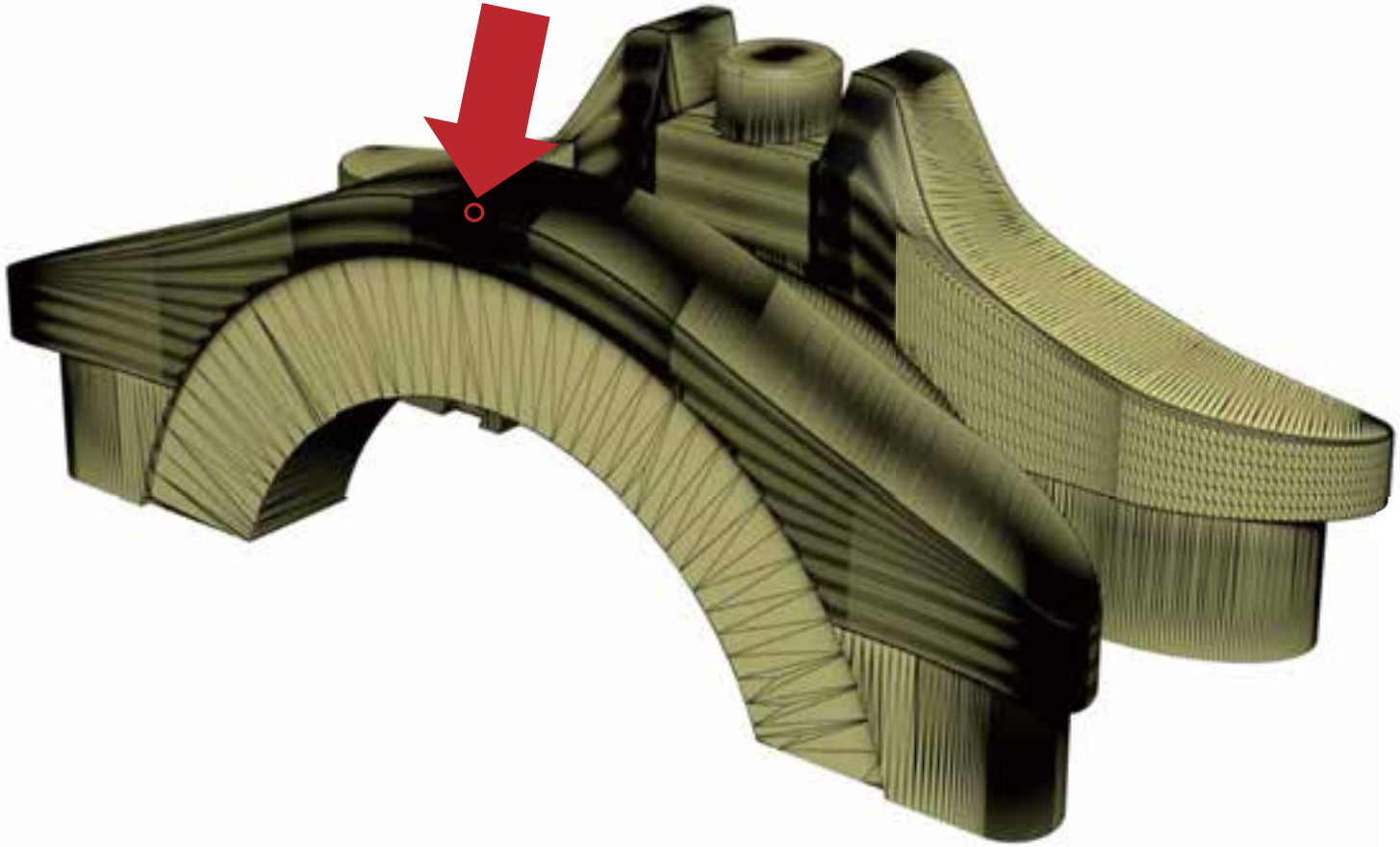
Surface Representation
(From Thingi10k)

Volumetric Representation
(Generated by TetWild)

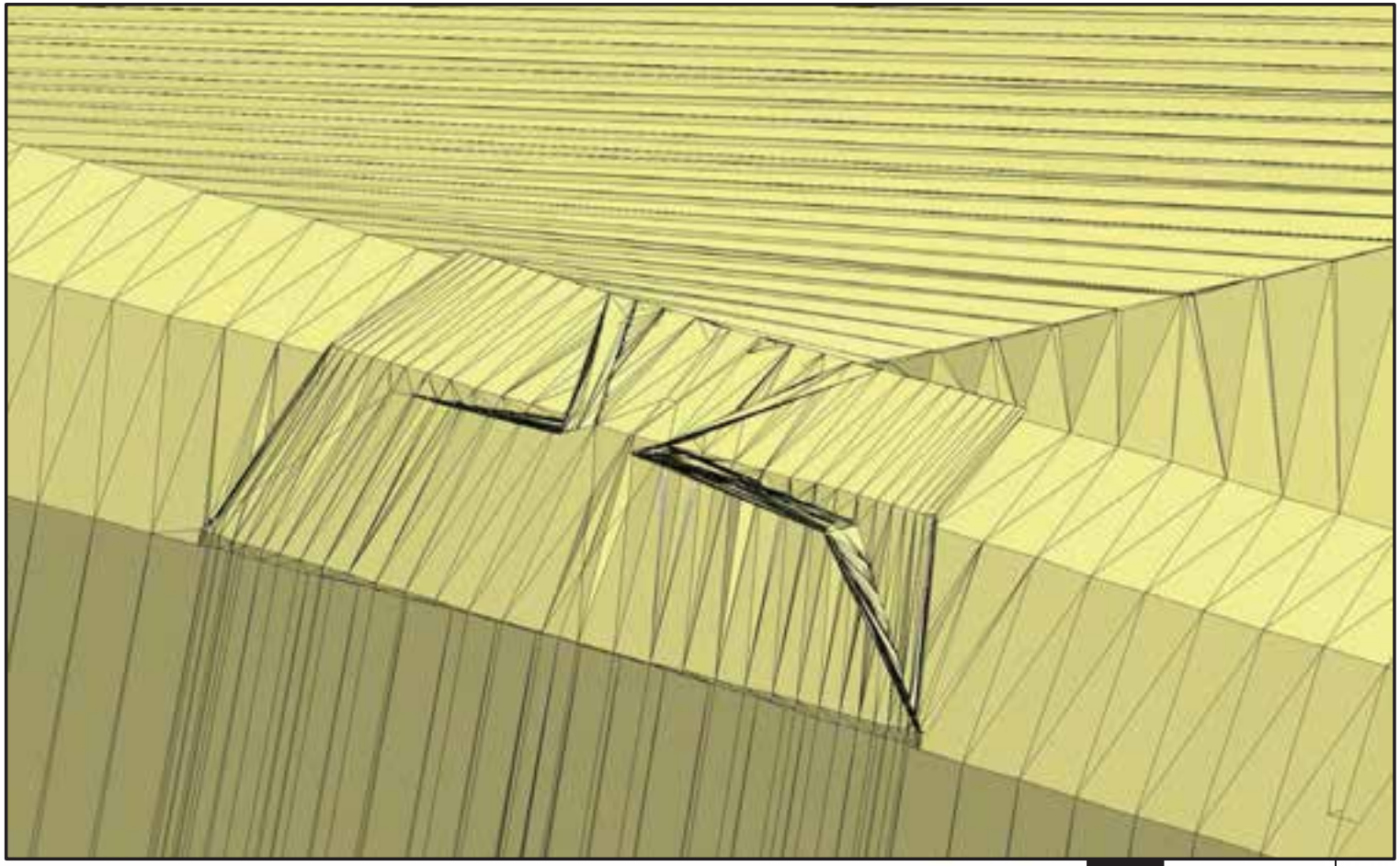
Physical Simulation

NYU | COURANT INSTITUTE OF MATHEMATICAL SCIENCES

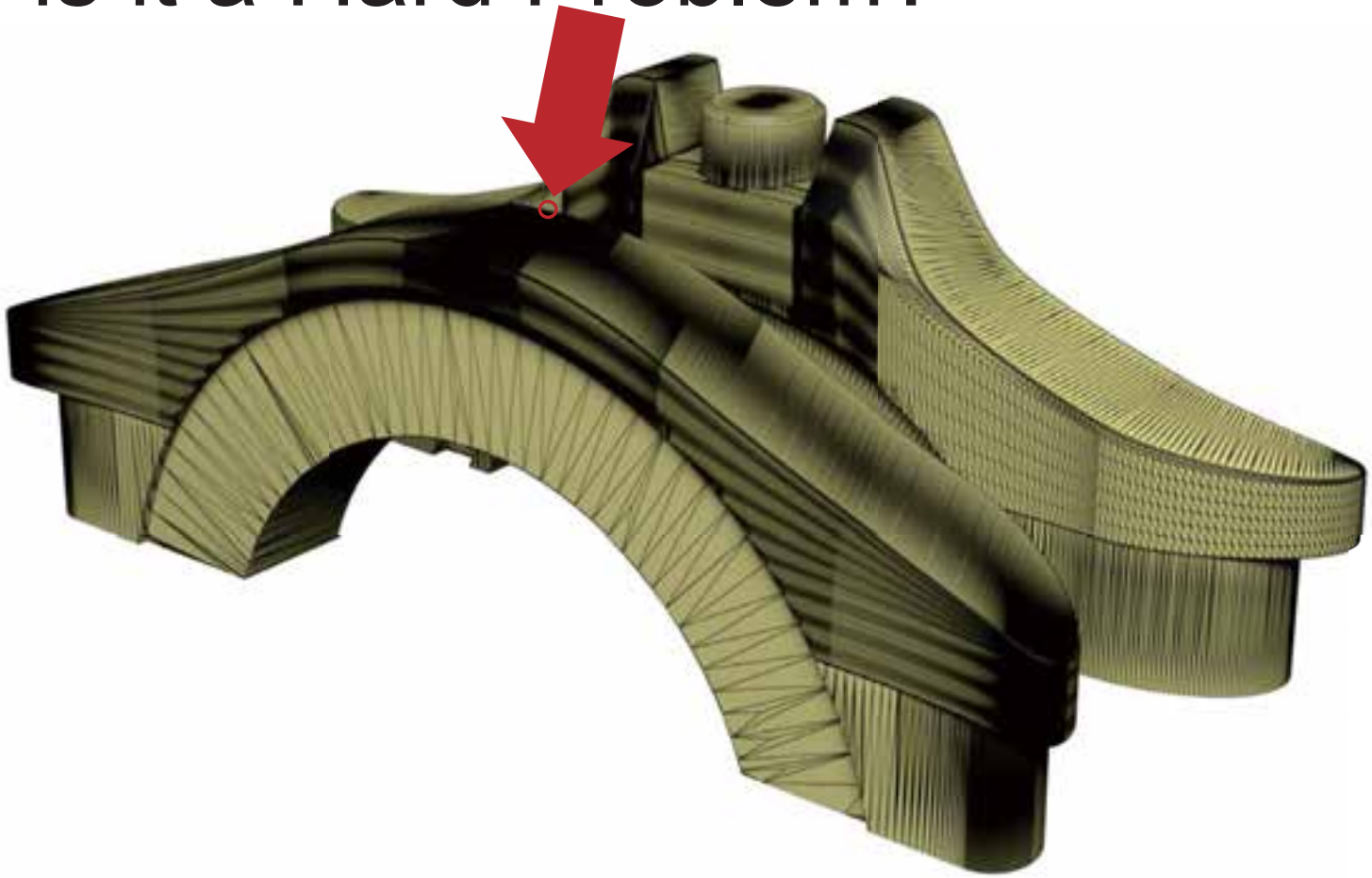
Why is it a Hard Problem?



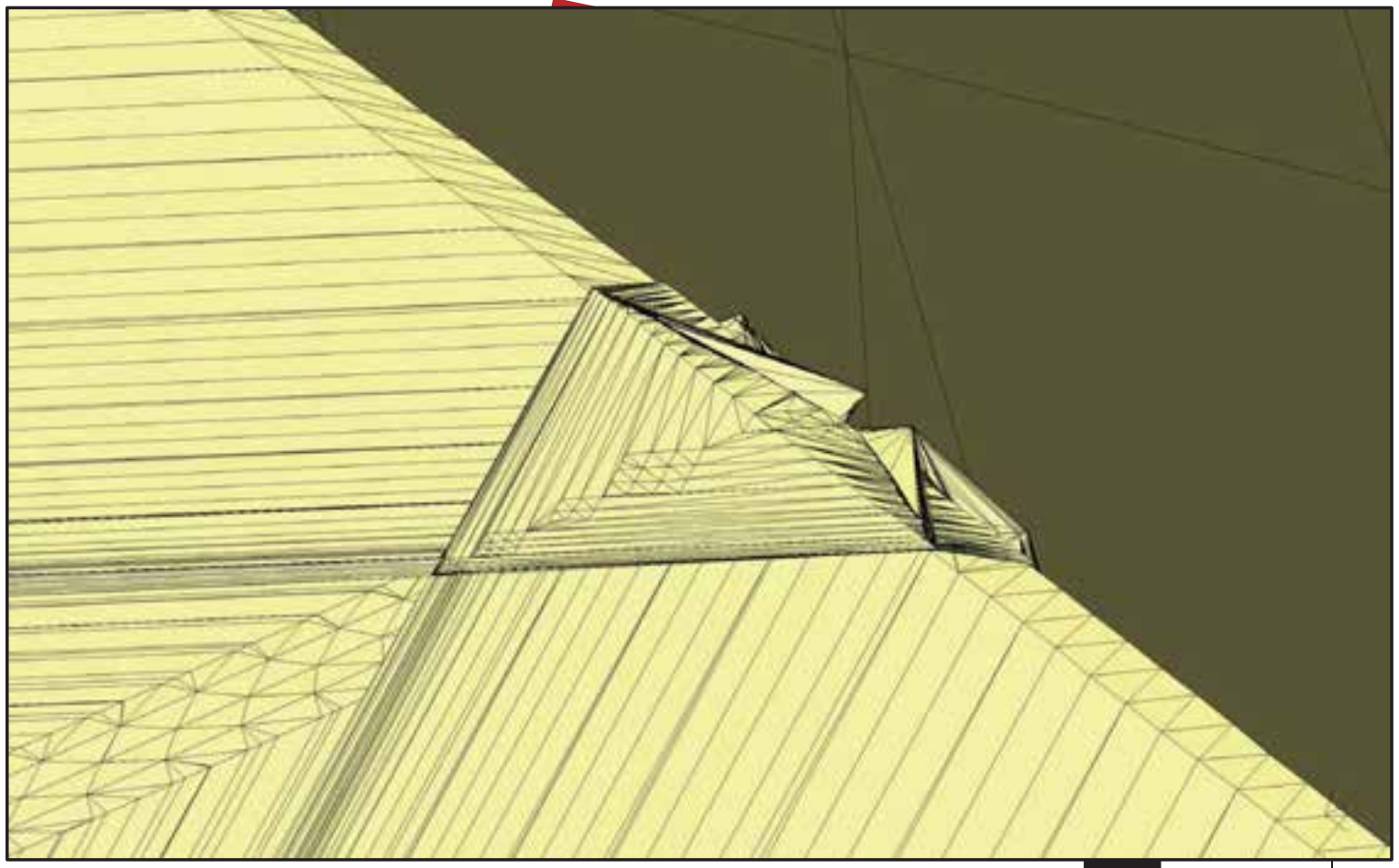
Why is it a Hard Problem?



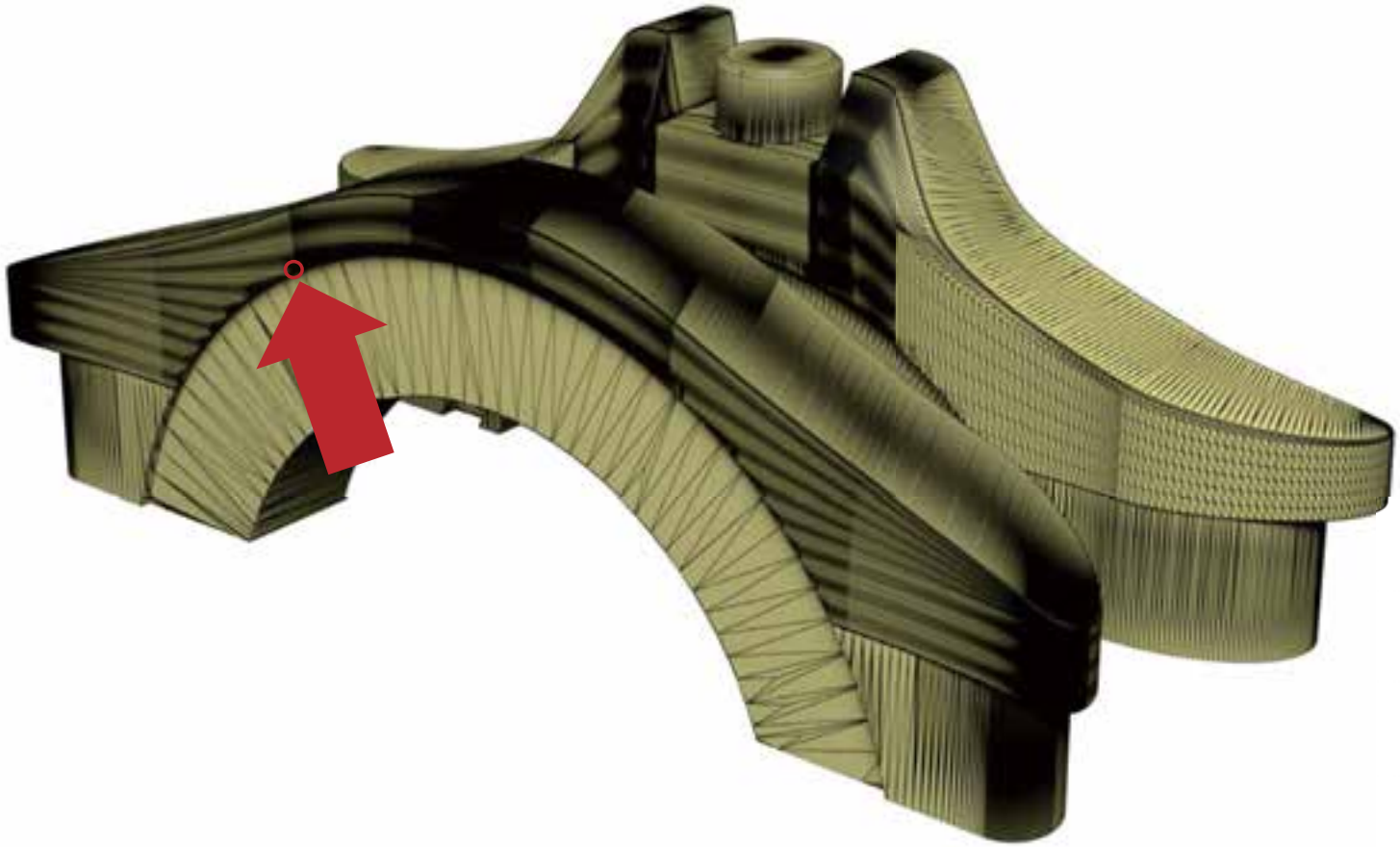
Why is it a Hard Problem?



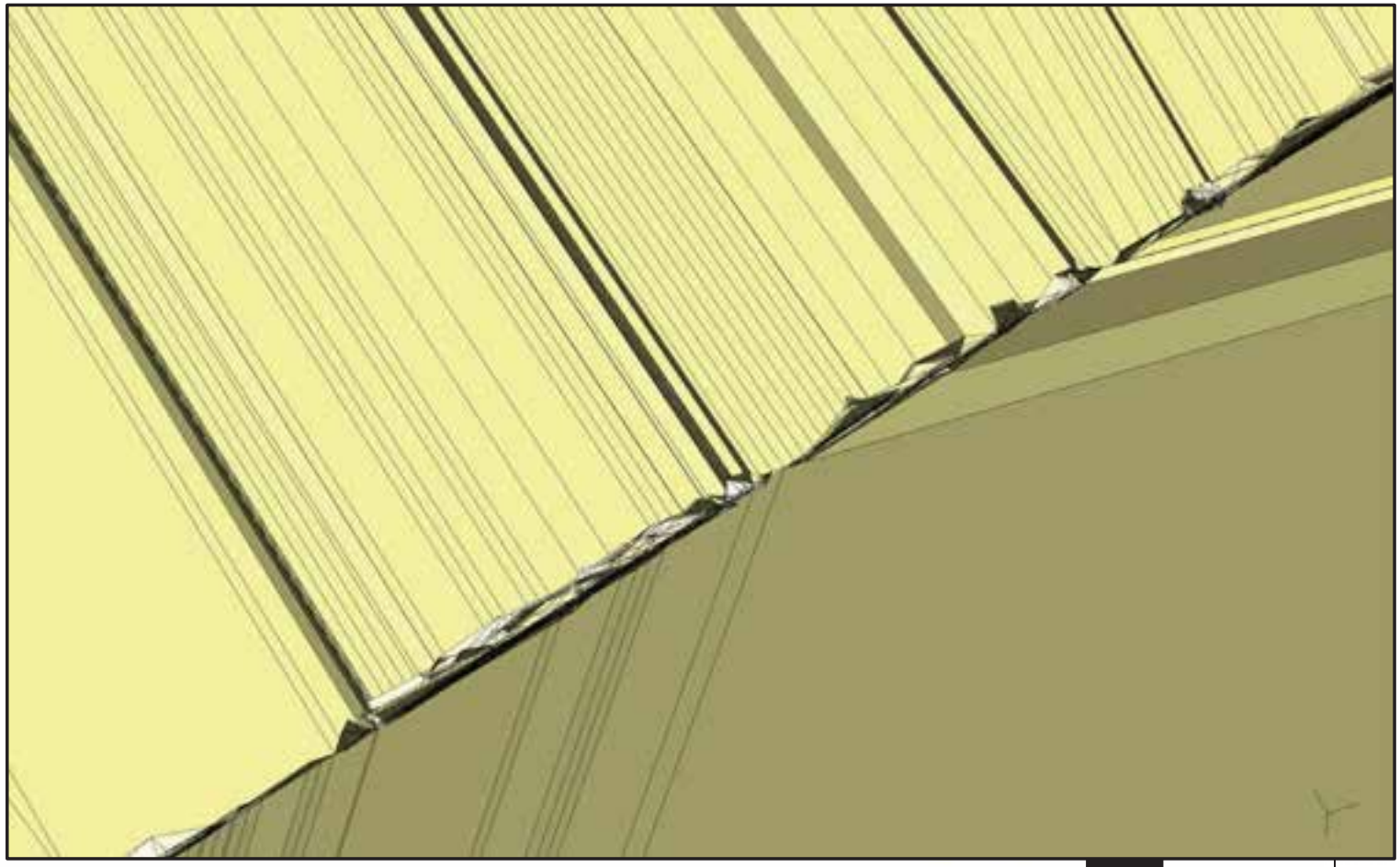
Why is it a Hard Problem?



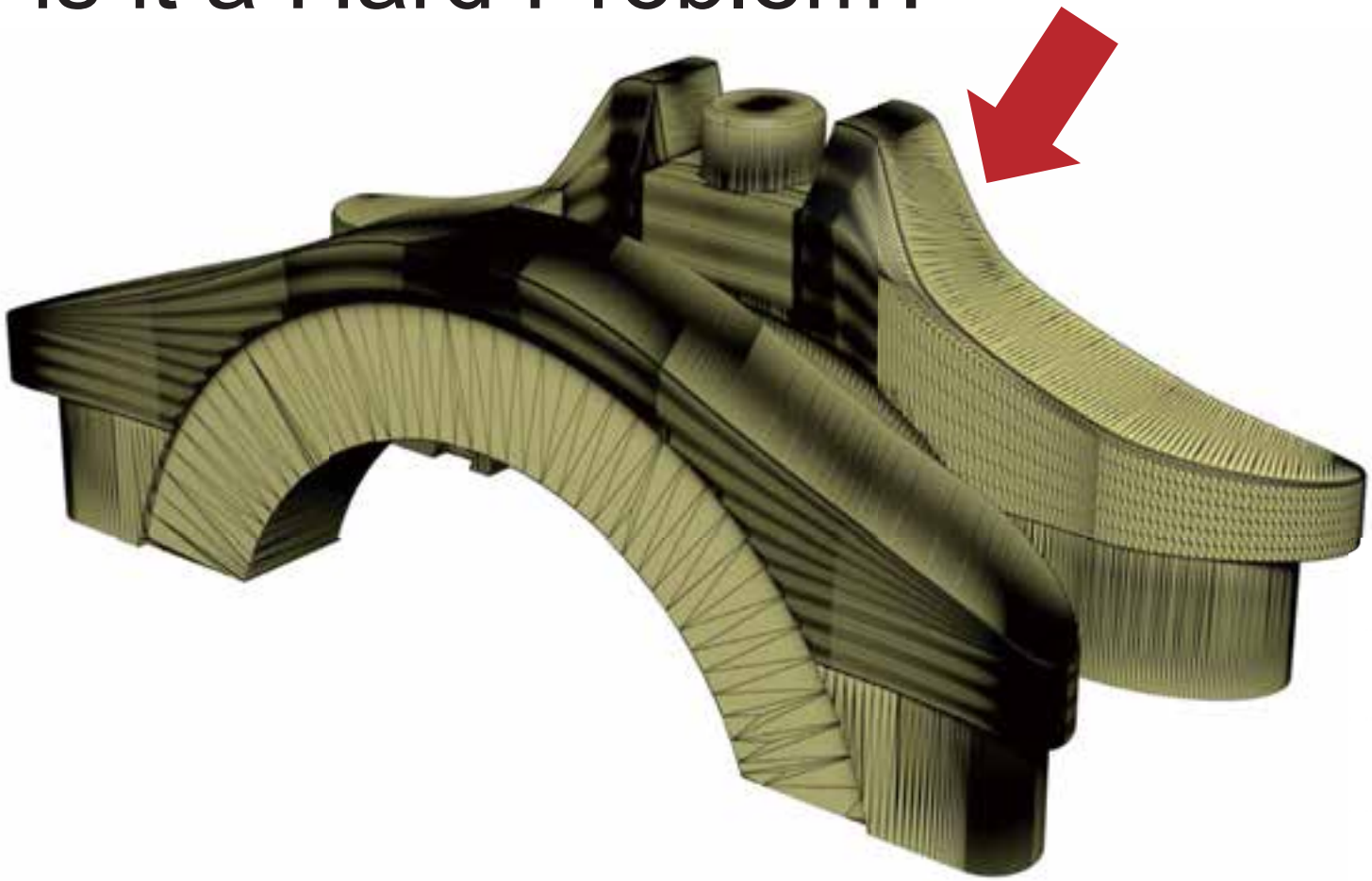
Why is it a Hard Problem?



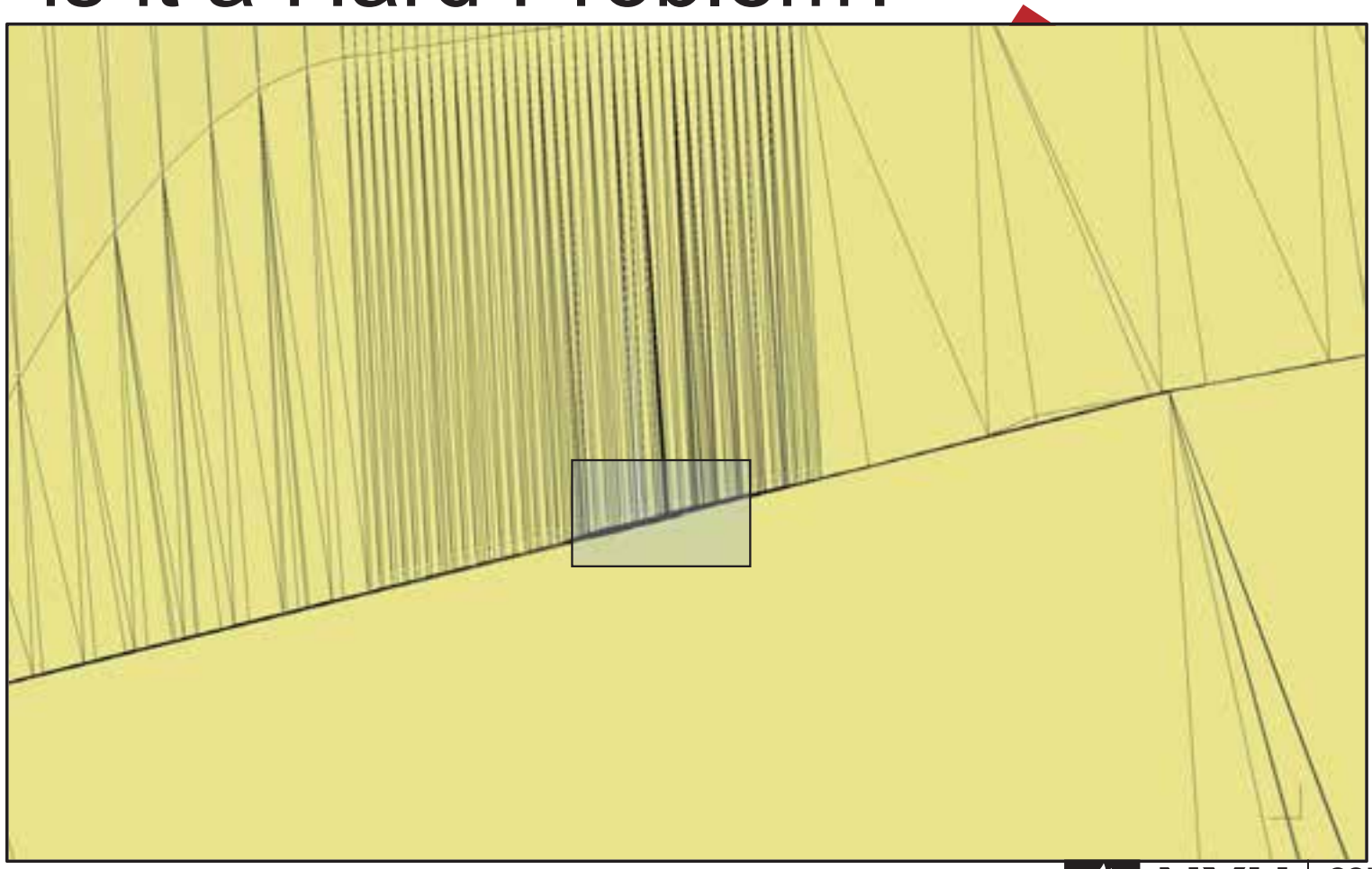
Why is it a Hard Problem?



Why is it a Hard Problem?

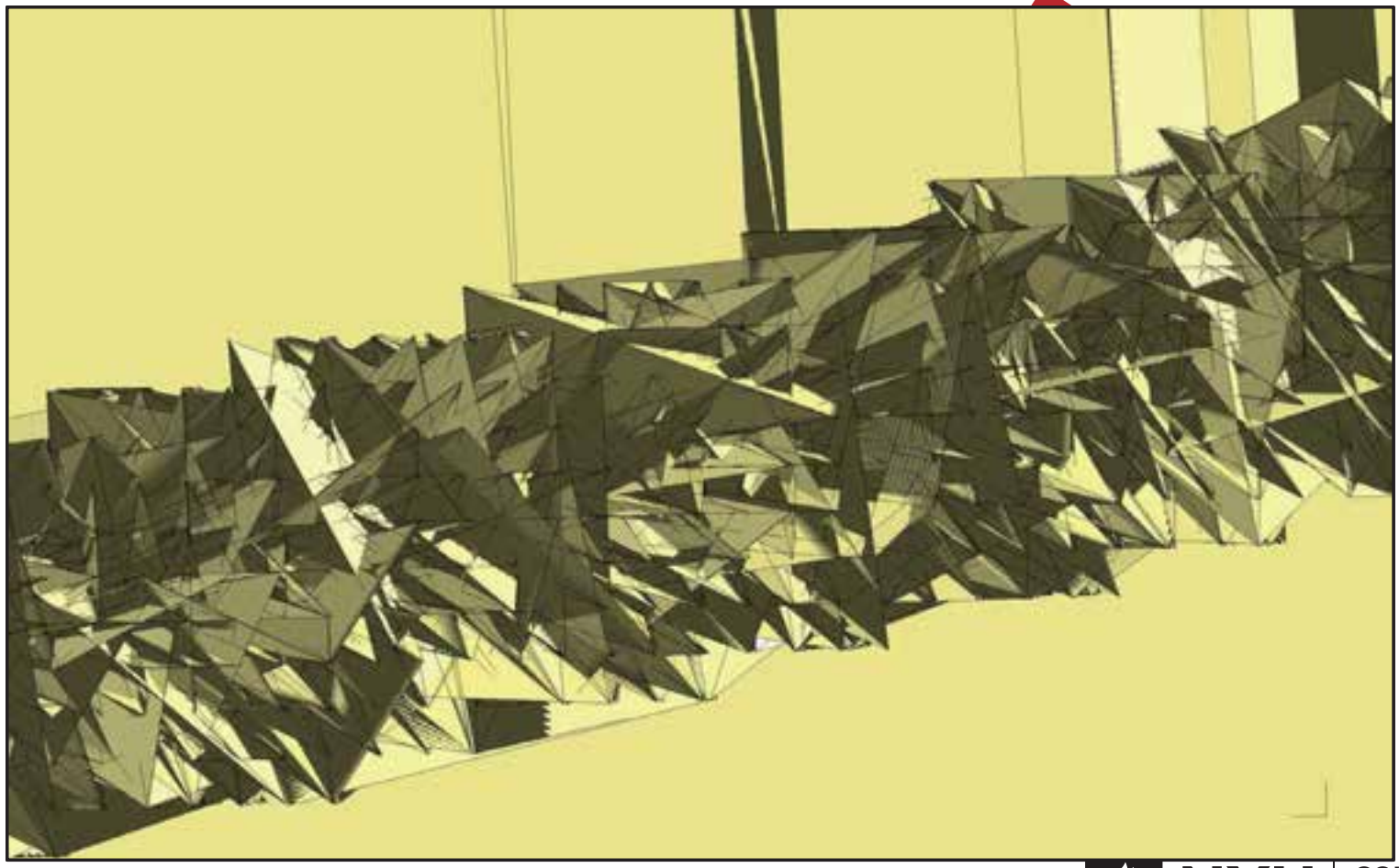


Why is it a Hard Problem?

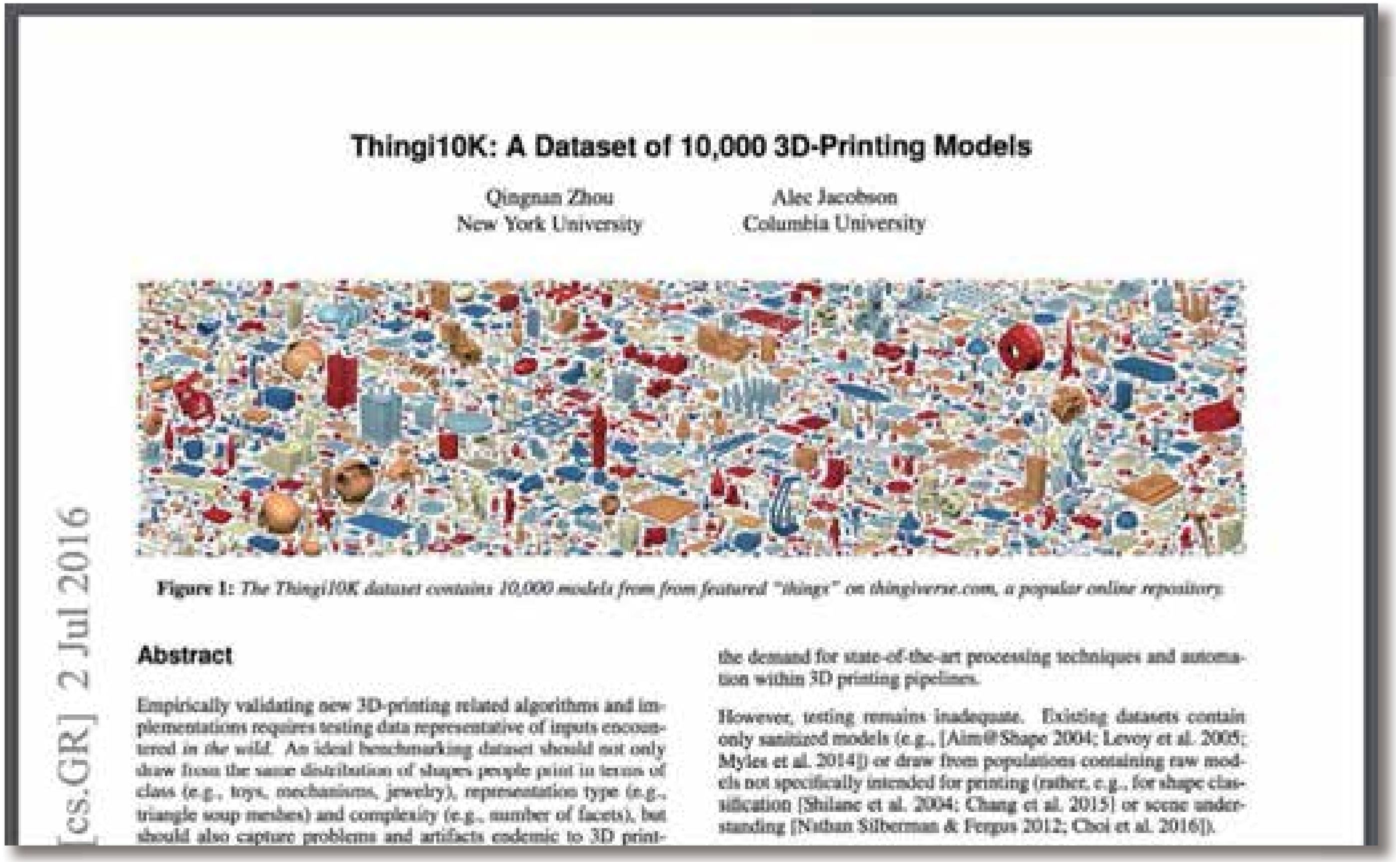


Why is it a Hard Problem?

Why is it a Hard Problem?



New Problem



Thingi10K: A Dataset of 10,000 3D-Printing Models

Qingnan Zhou
New York University

Alec Jacobson
Columbia University

Figure 1: *The Thingi10K dataset contains 10,000 models from from featured "things" on thingiverse.com, a popular online repository.*

Abstract

Empirically validating new 3D-printing related algorithms and implementations requires testing data representative of inputs encountered *in the wild*. An ideal benchmarking dataset should not only draw from the same distribution of shapes people print in terms of class (e.g., toys, mechanisms, jewelry), representation type (e.g., triangle soup meshes) and complexity (e.g., number of facets), but should also capture problems and artifacts endemic to 3D print-

the demand for state-of-the-art processing techniques and automation within 3D printing pipelines.

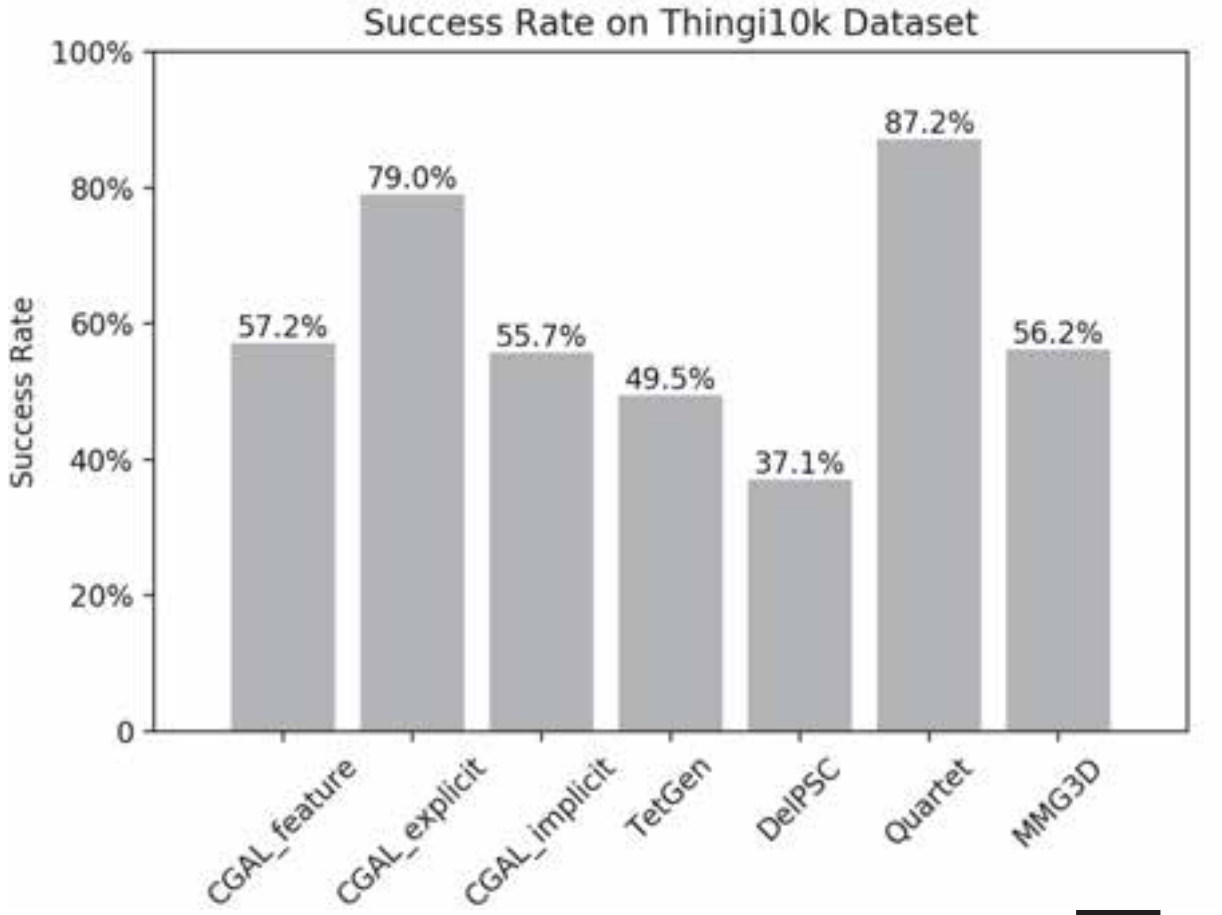
However, testing remains inadequate. Existing datasets contain only sanitized models (e.g., [Aim@Shape 2004; Levoy et al. 2005; Myles et al. 2014]) or draw from populations containing raw models not specifically intended for printing (rather, e.g., for shape classification [Shilane et al. 2004; Chang et al. 2015] or scene understanding [Nathan Silberman & Fergus 2012; Choi et al. 2016]).

NYU | COURANT INSTITUTE OF MATHEMATICAL SCIENCES

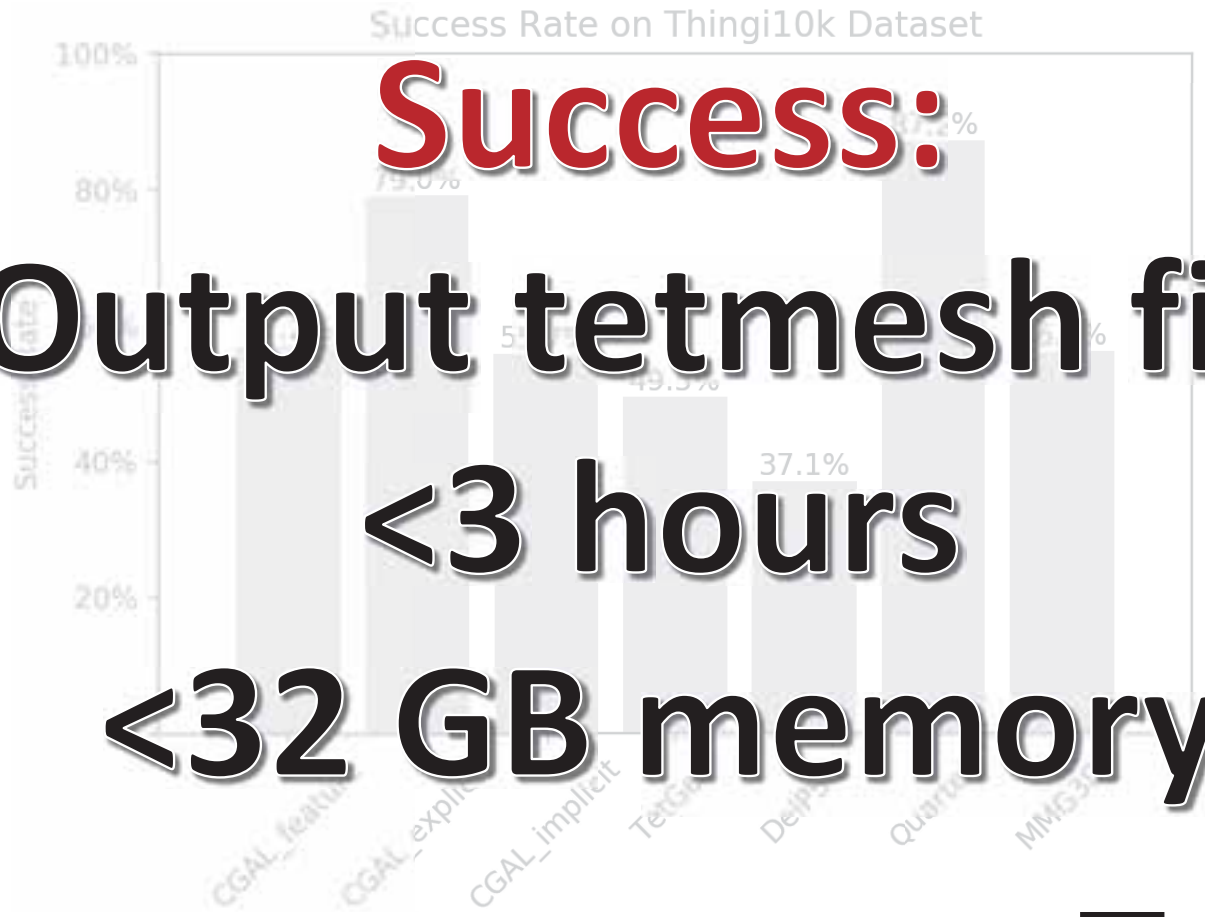
Tetrahedral Meshing on Messy Surface?

NYU | COURANT INSTITUTE OF MATHEMATICAL SCIENCES

Existing Methods



Existing Methods



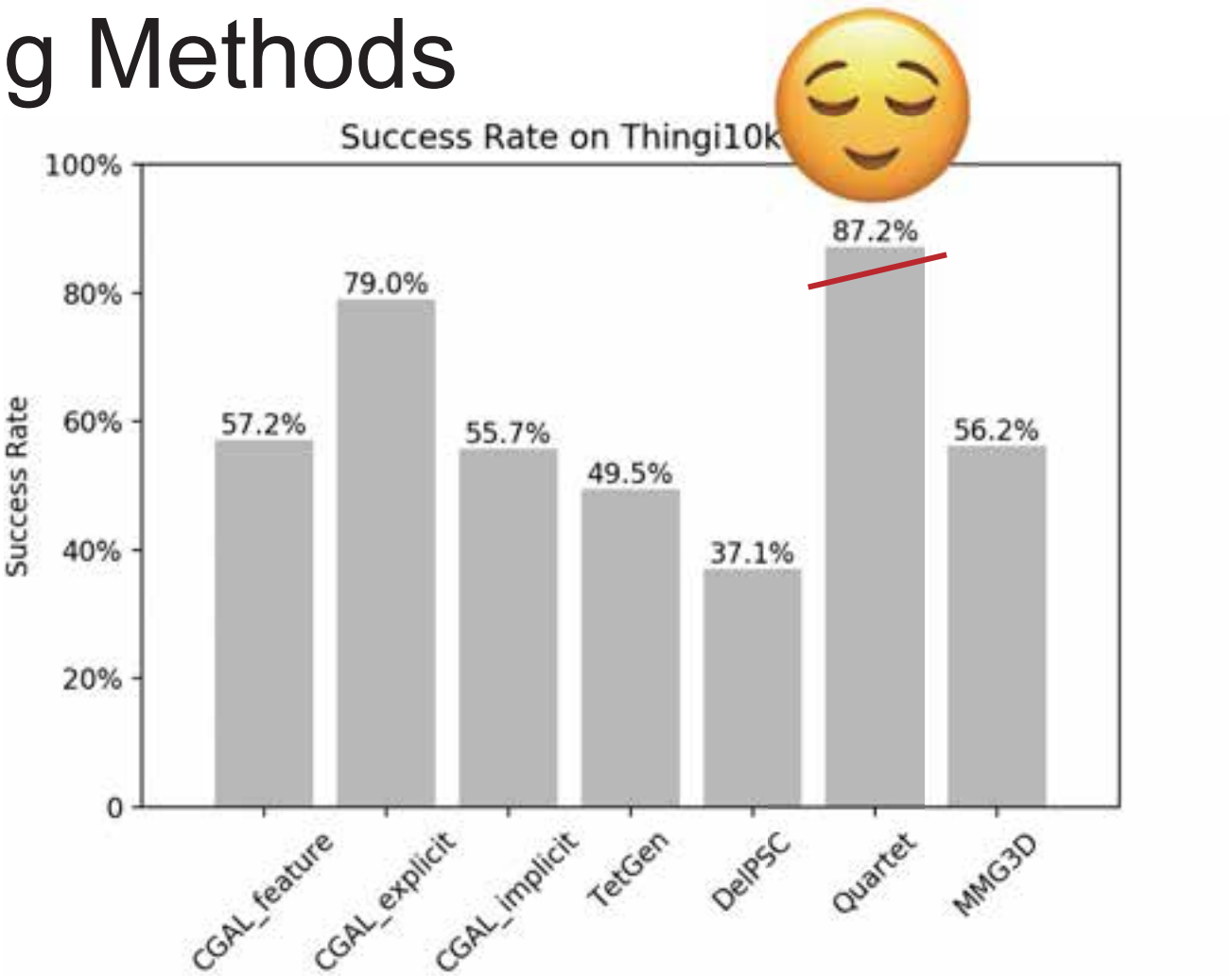
Success:

Output tetmesh file

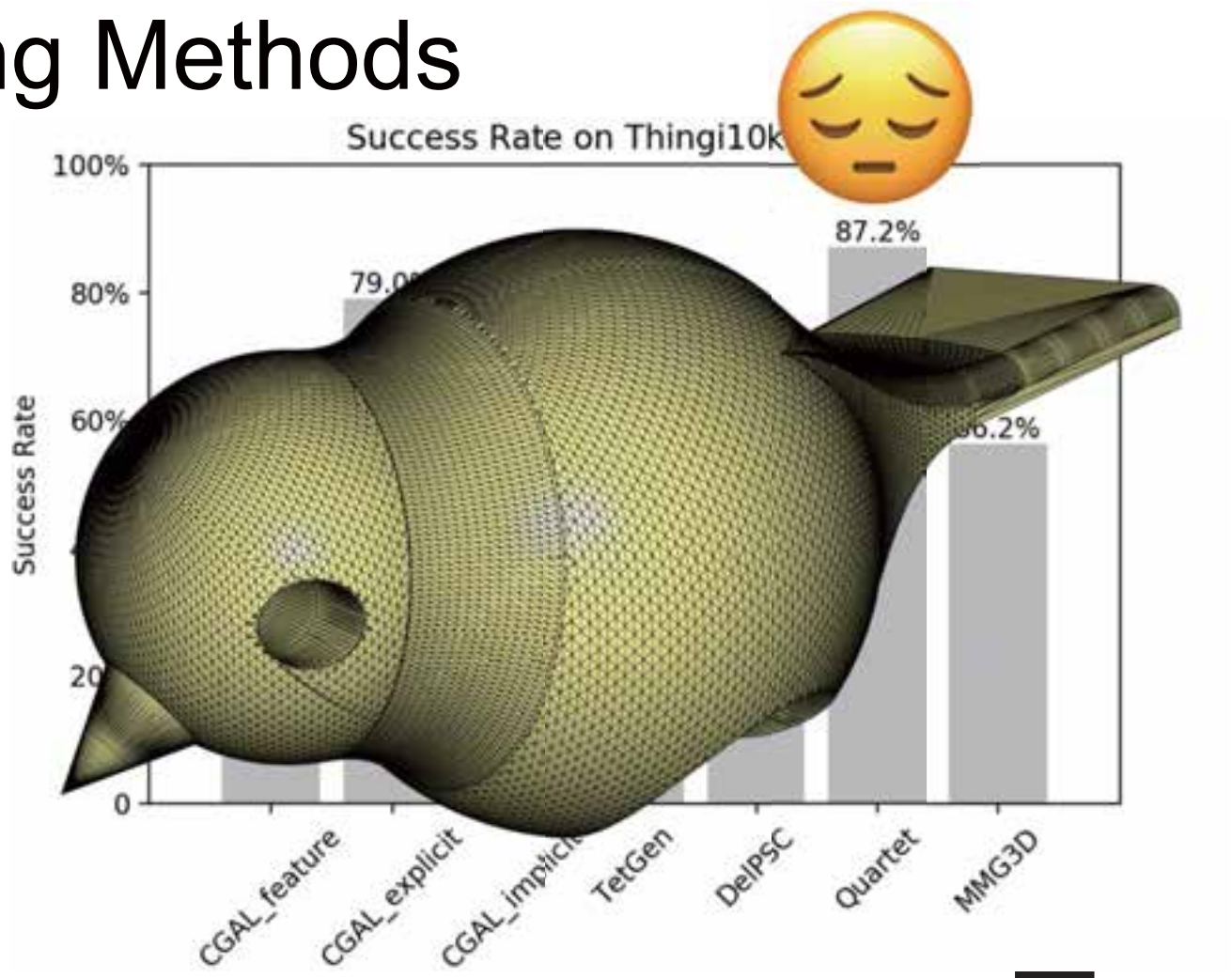
<3 hours

<32 GB memory

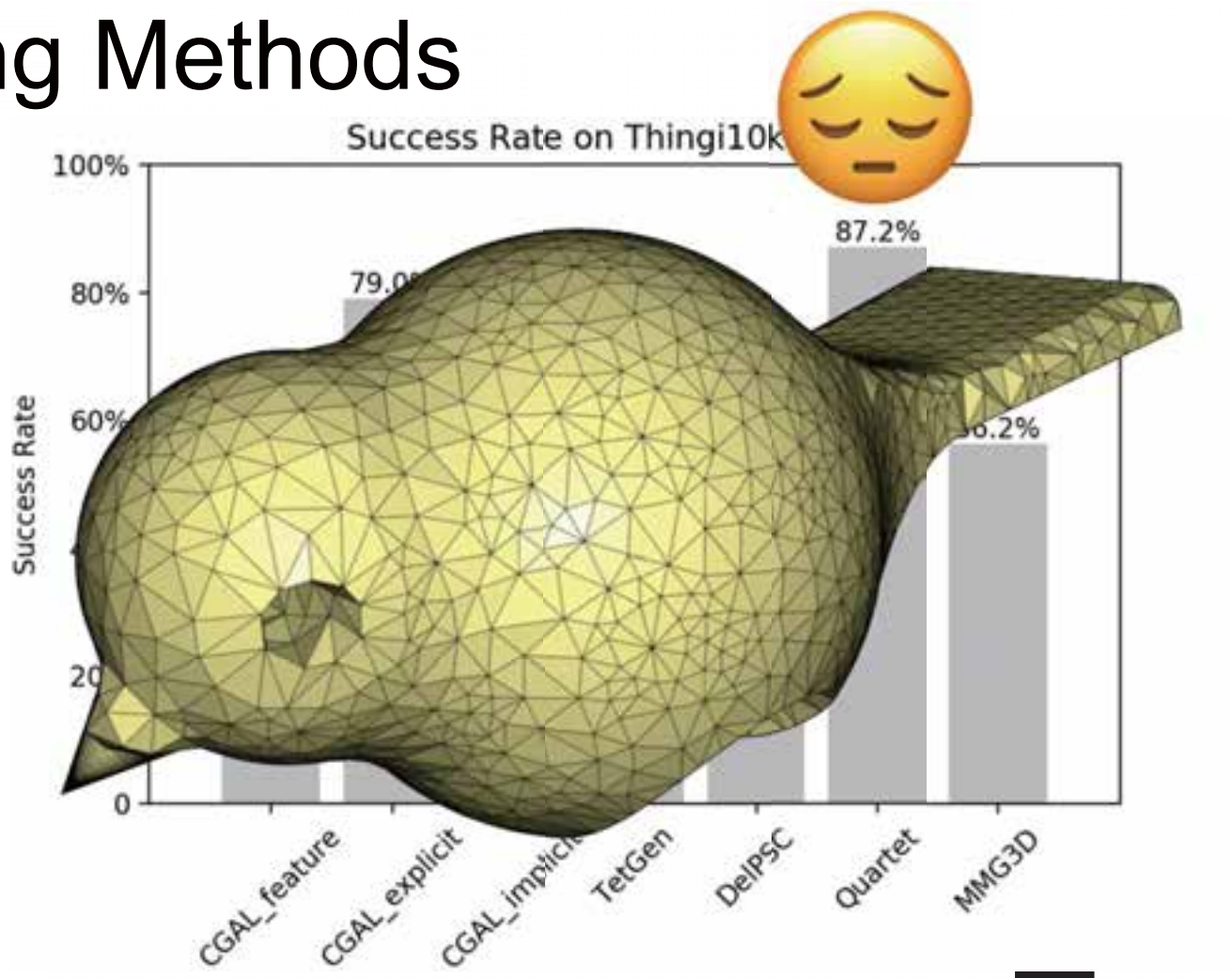
Existing Methods



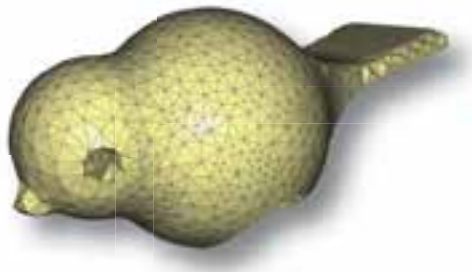
Existing Methods



Existing Methods

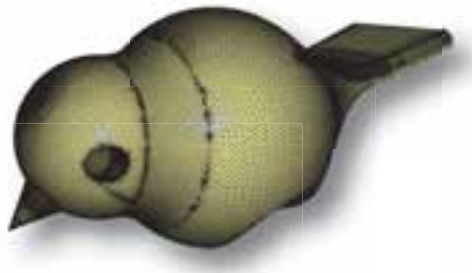


New Method



(Generated by Quartet)

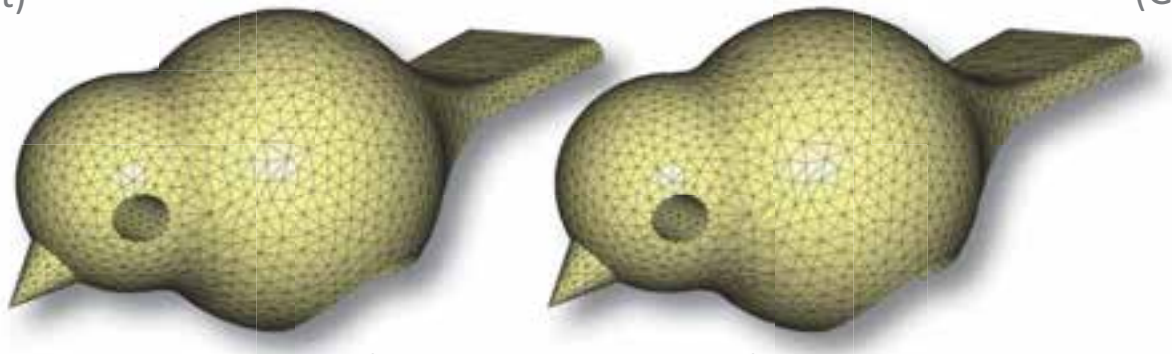
Exactly conform to the input surface



(Generated by TetGen)

Want robustness?

Want faithfulness?

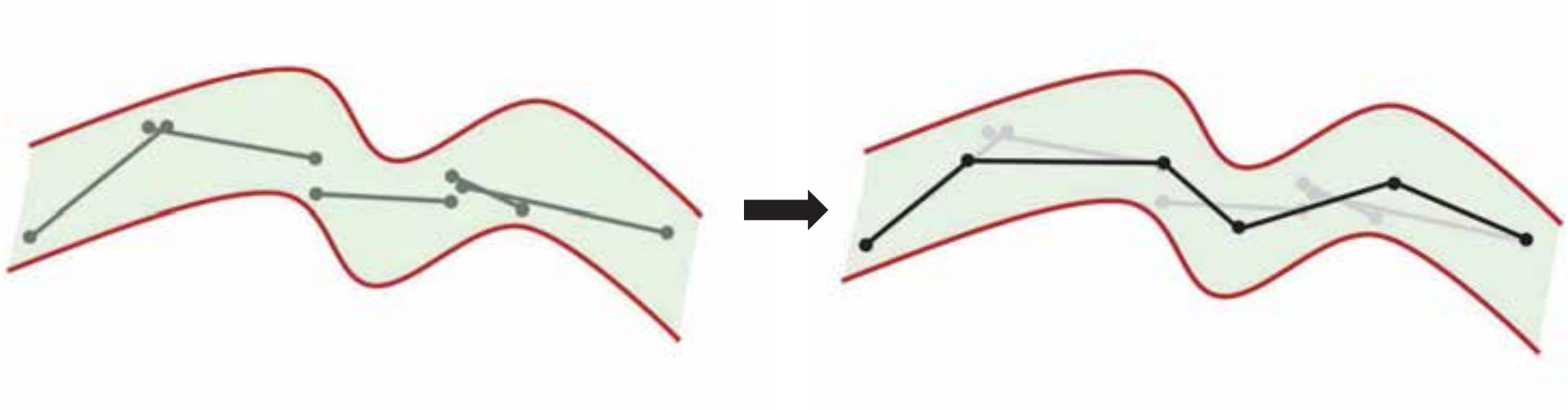


(Generated by TetWild)

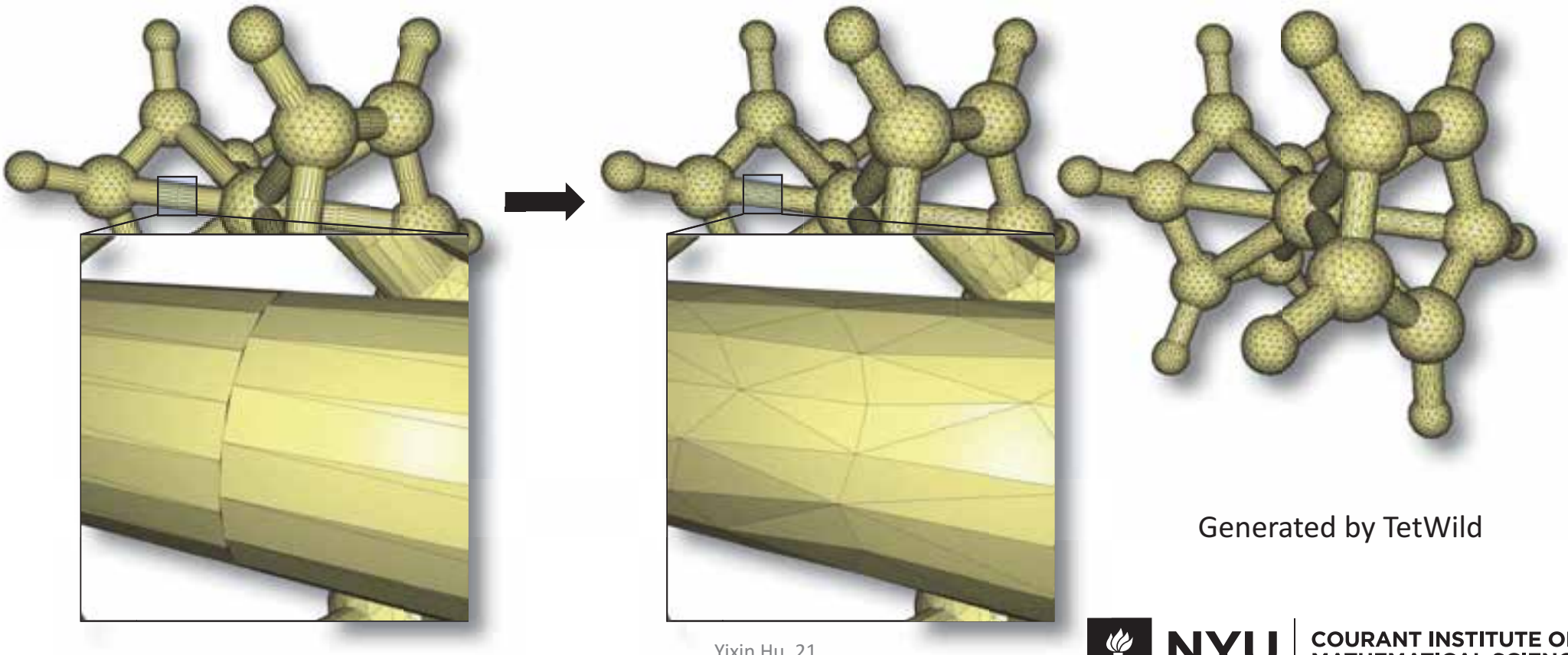
Our method: 1. Closed under rationals. 2. Approximate input surface within an envelope.

Why Envelope?

- Optimize boundary inside an envelope



Input with Gaps



Generated by TetWild

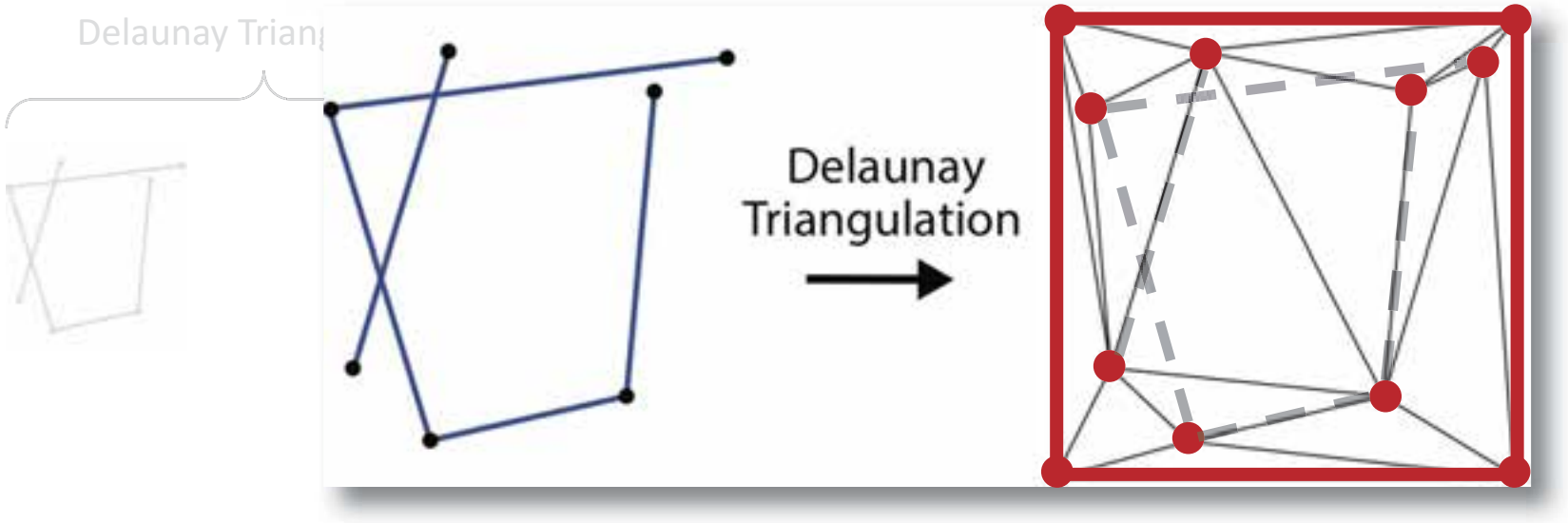
TetWild

- **Input**: Triangle soup that forms the surface of a 3D shape.
  - No assumption about the input.

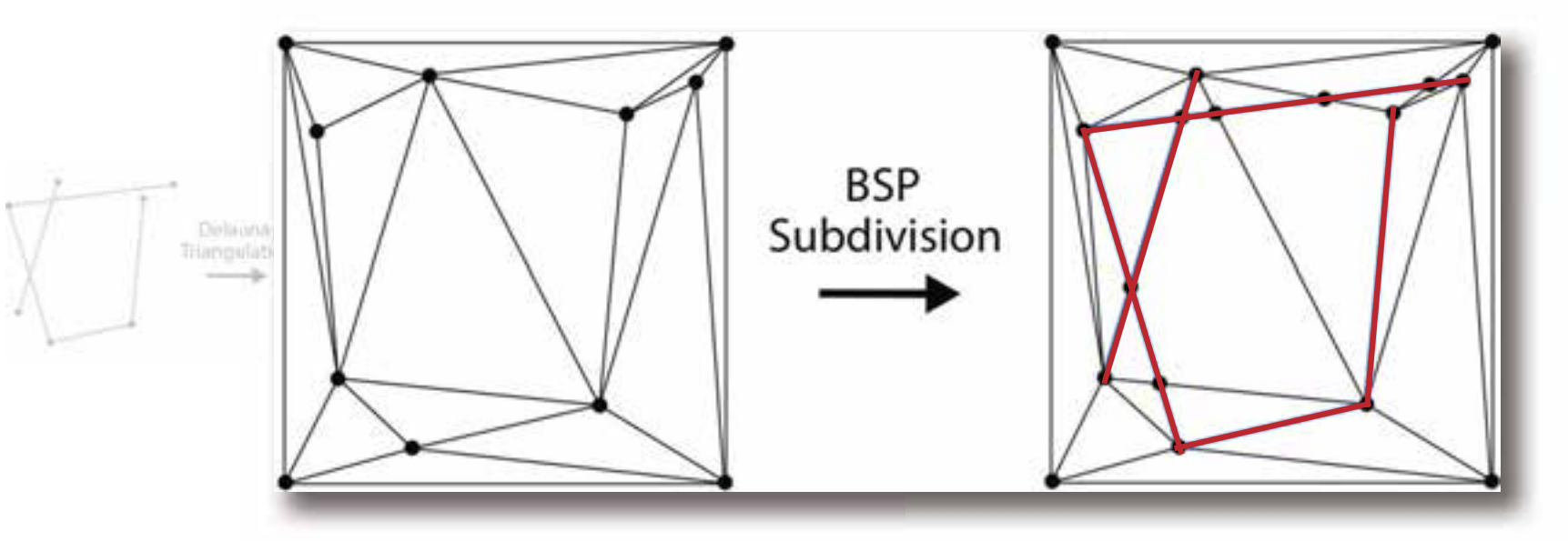
- **Output**: Approximated constrained tetrahedral mesh. (Envelope)
  - Able to clean up imperfect regions on the input and produce high-quality output.

Pipeline

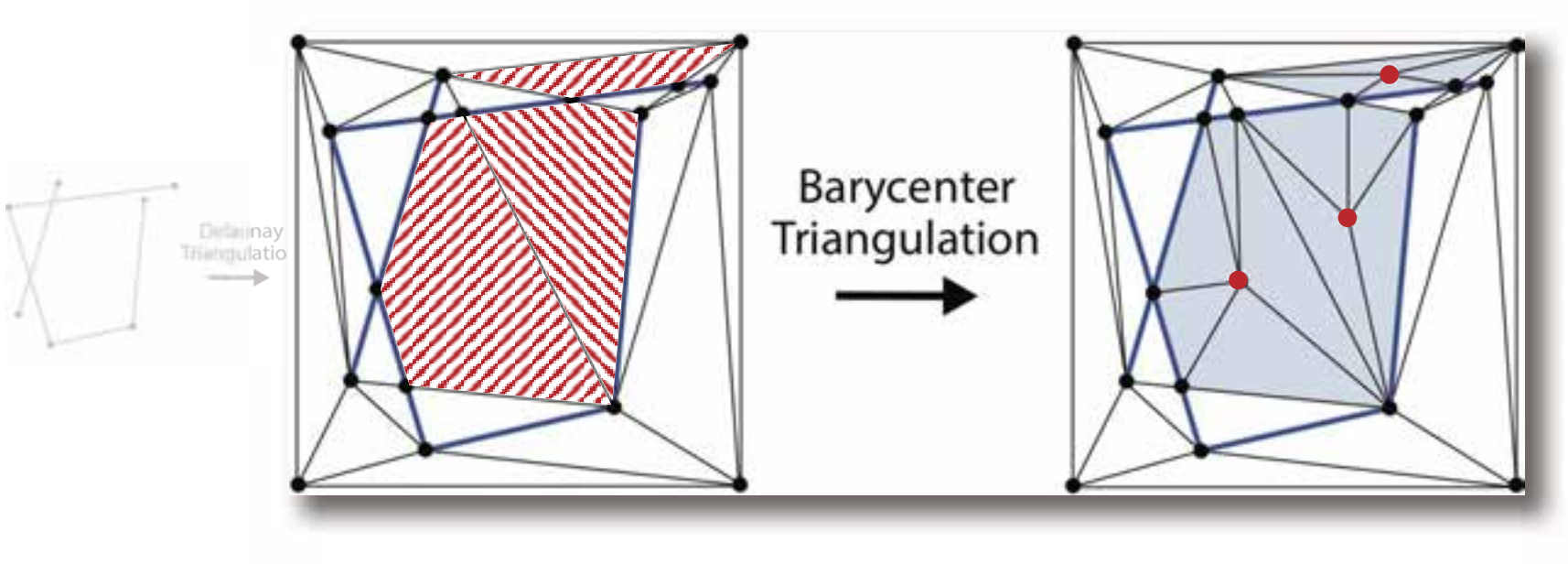
Delaunay Triang



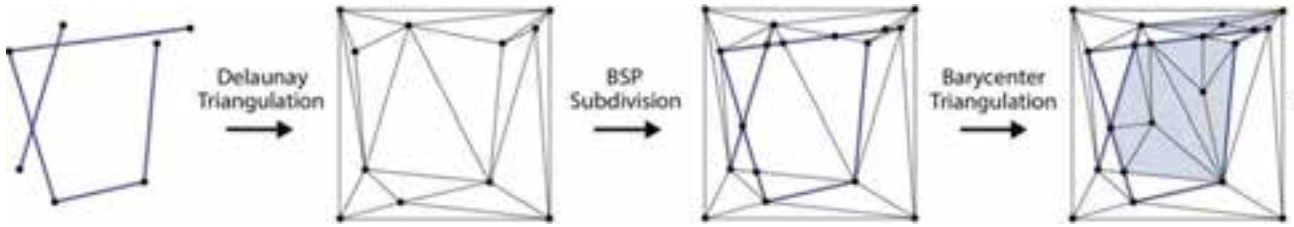
Pipeline



Pipeline



Pipeline



Stage I: Valid Mesh Generation
(Rational)

Pipeline

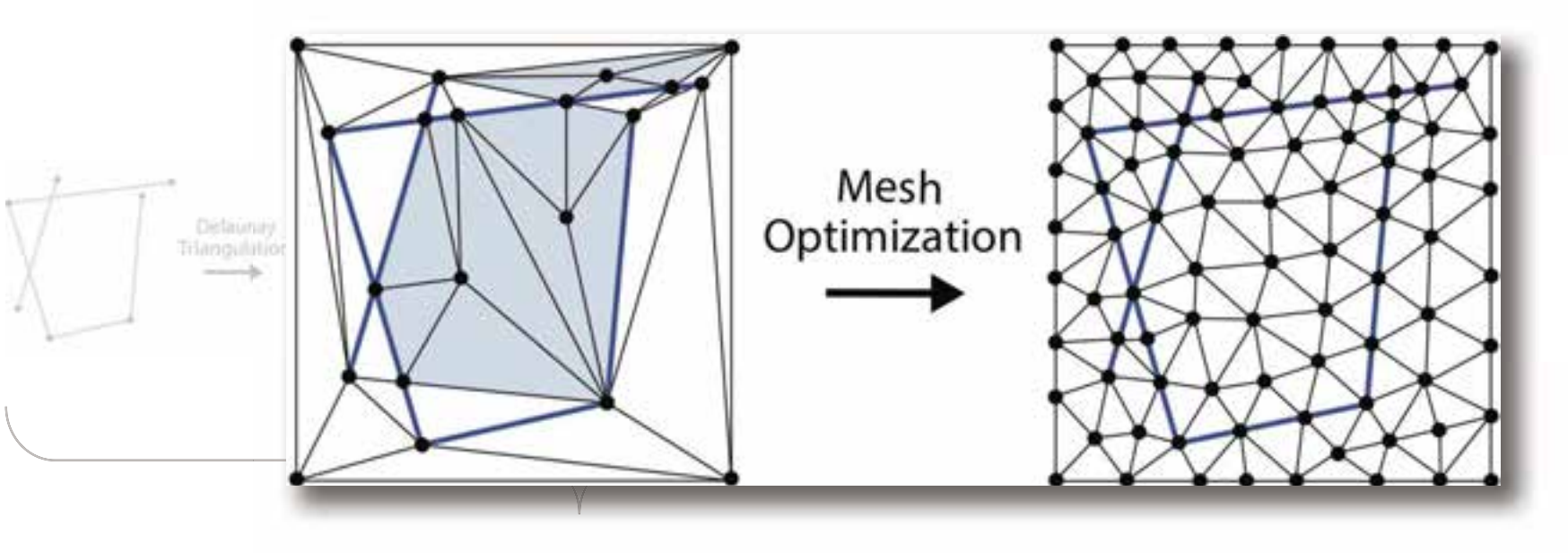
Valid:

No inversion or self-intersection

Input boundary preserved

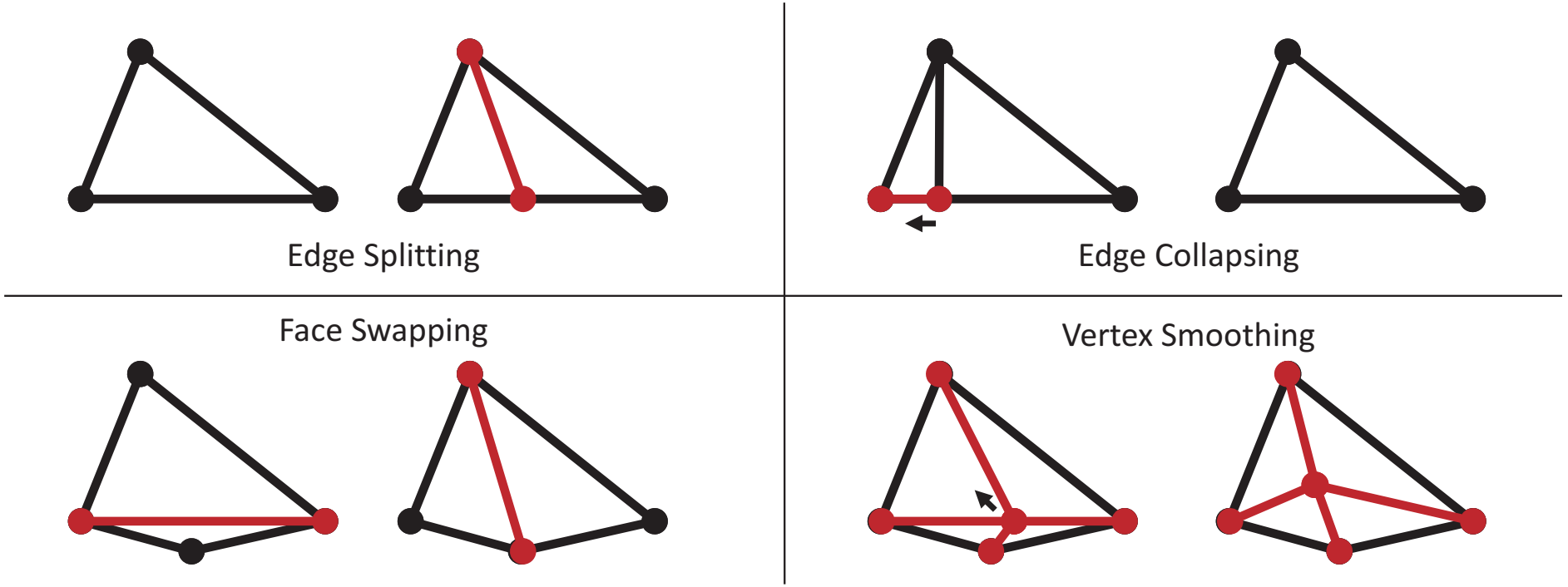
Input: Volumetric Mesh
(Rational)

Pipeline



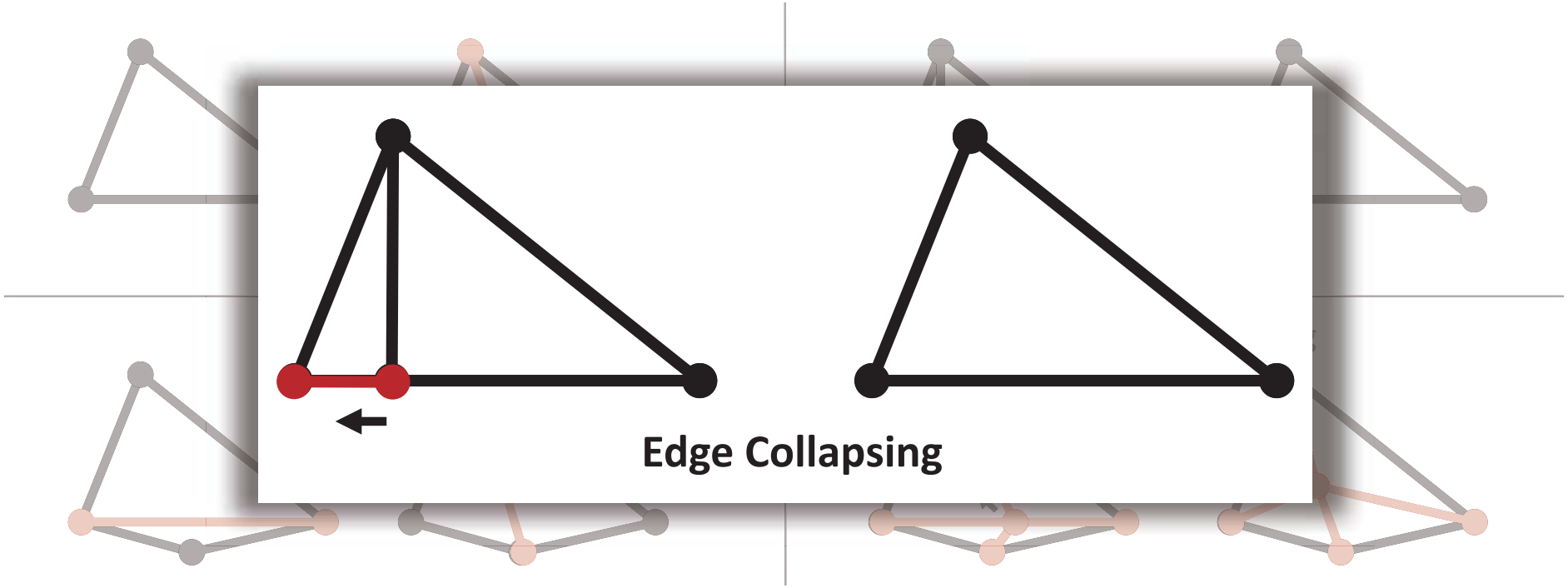
Stage I: Valid Mesh Generation
(Rational)

Mesh Optimization

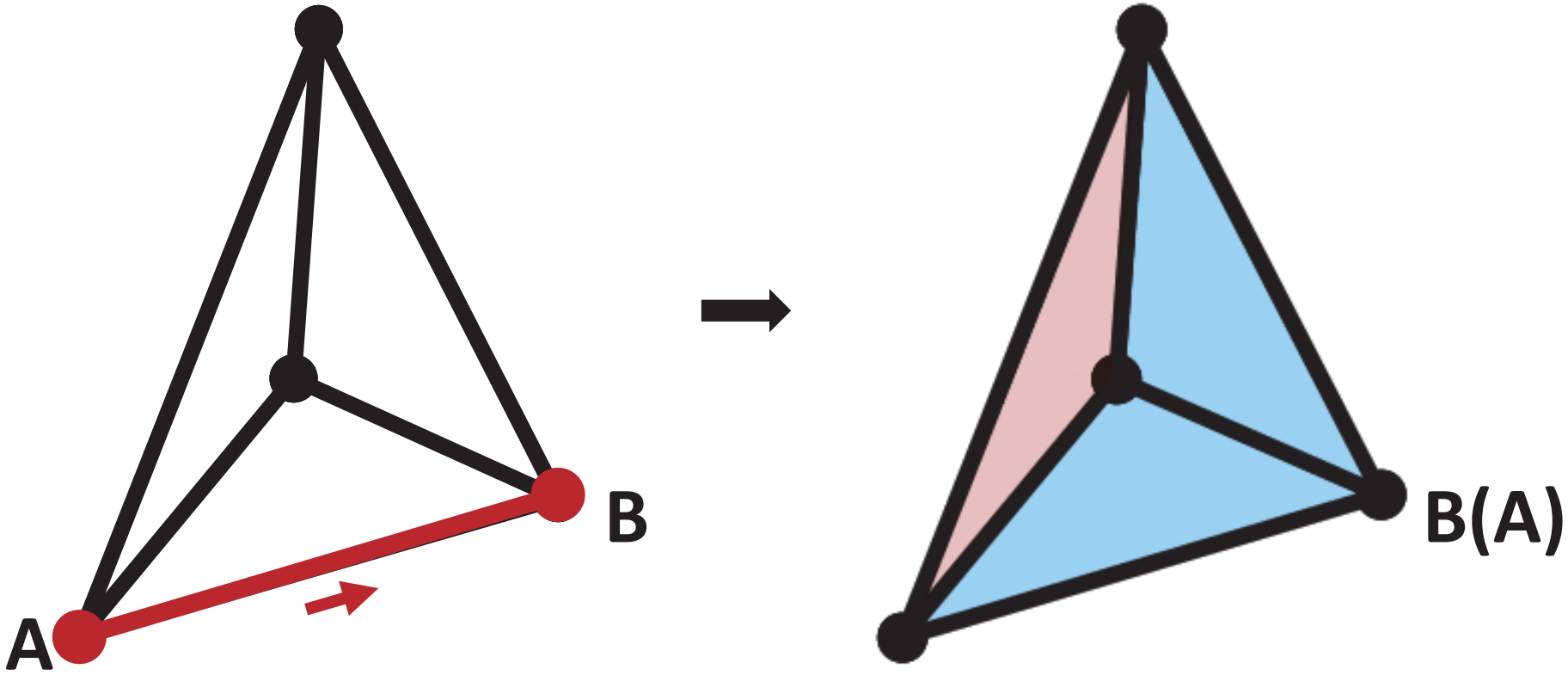


Mesh Optimization

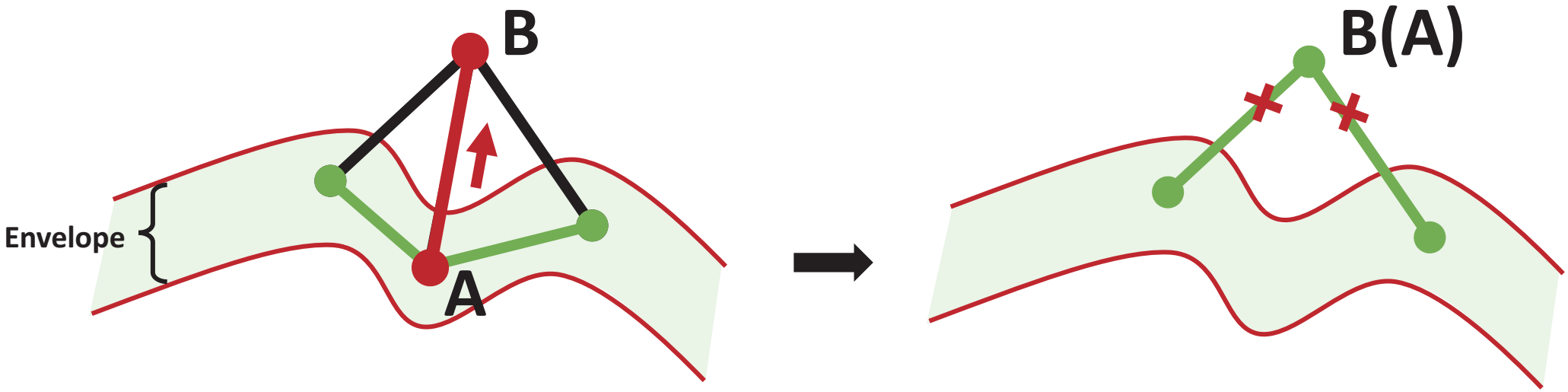
Conformal AMIPS 3D Energy



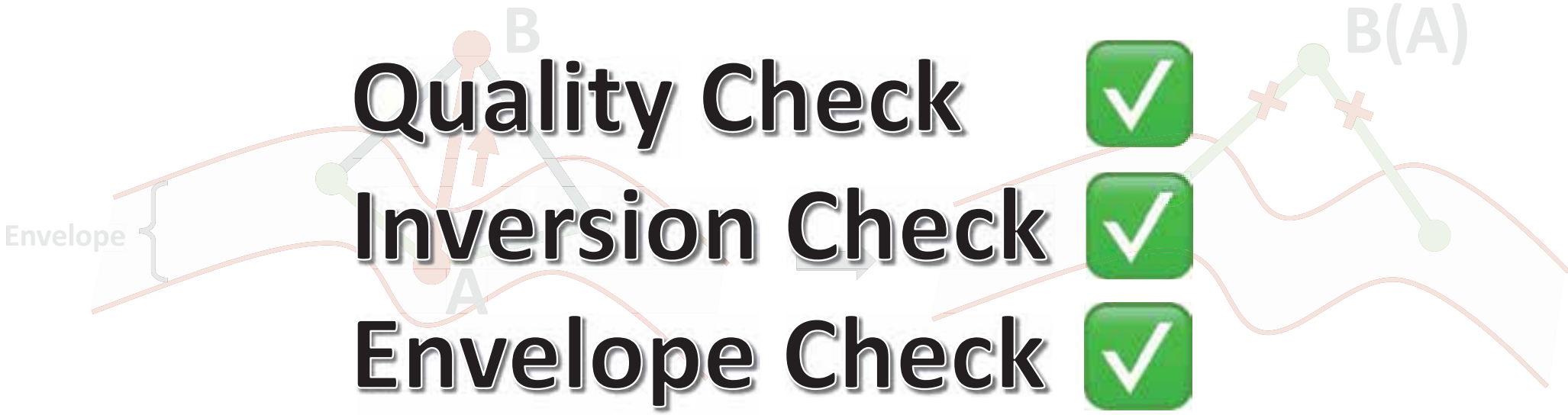
Inversion Check



Envelope Check



Envelope Check



Quality Check ✅

Inversion Check ✅

Envelope Check ✅

Pipeline

Delaun
Triangula

Filtering

Robust Inside-Outside Segmentation using Generalized Winding Numbers

Alec Jacobson[1] Ladislav Kavan[2,1] Olga Sorkine-Hornung[1]
[1]ETH Zurich [2]University of Pennsylvania

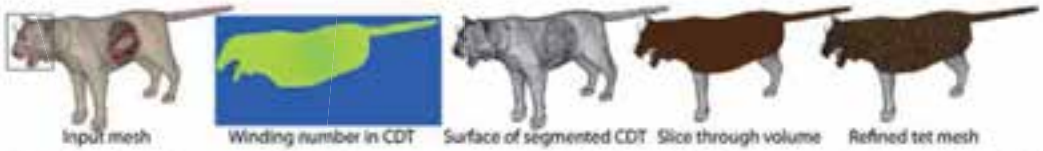


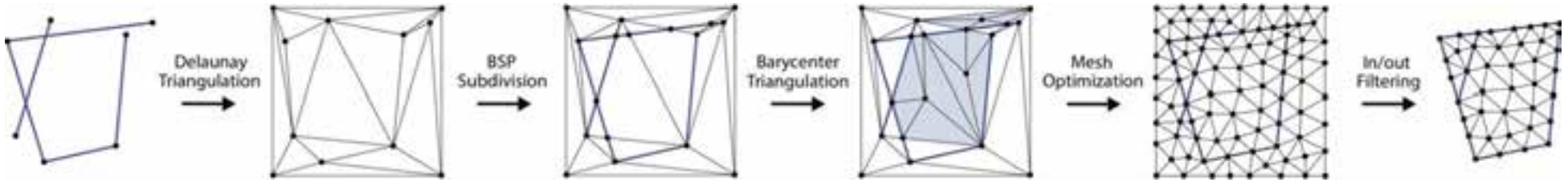
Figure 1: *The* Big SigCat *input mesh has 3442 pairs of intersecting triangles (bright red), 1020 edges on open boundaries (dark red), 344 non-manifold edges (purple) and 67 connected components (randomly colored). On top of those problems, a SIGGRAPH logo shaped hole is carved from her side.*

Abstract

Solid shapes in computer graphics are often represented with boundary descriptions, *e.g.* triangle meshes, but animation, physically-based simulation, and geometry processing are more realistic and accurate when explicit volume representations are available. Tetrahedral meshes which exactly contain (interpolate) the input boundary description are desirable but difficult to construct for a large class of input meshes. Character meshes and CAD models are

physically-based simulation of a hippopotamus would look quite different (and unrealistic) if handled as a thin shell, rather than a solid. Since many operations in animation, simulation and geometry processing require an explicit representation of an object's volume, for example for finite element analysis and solving PDEs, a conforming[1] tetrahedral meshing of the surface is highly desired, as it enables volumetric computation with direct access to and assignment of boundary surface values. However, a wide range of "real-life" models, although they *appear* to describe the boundary of a solid ob-

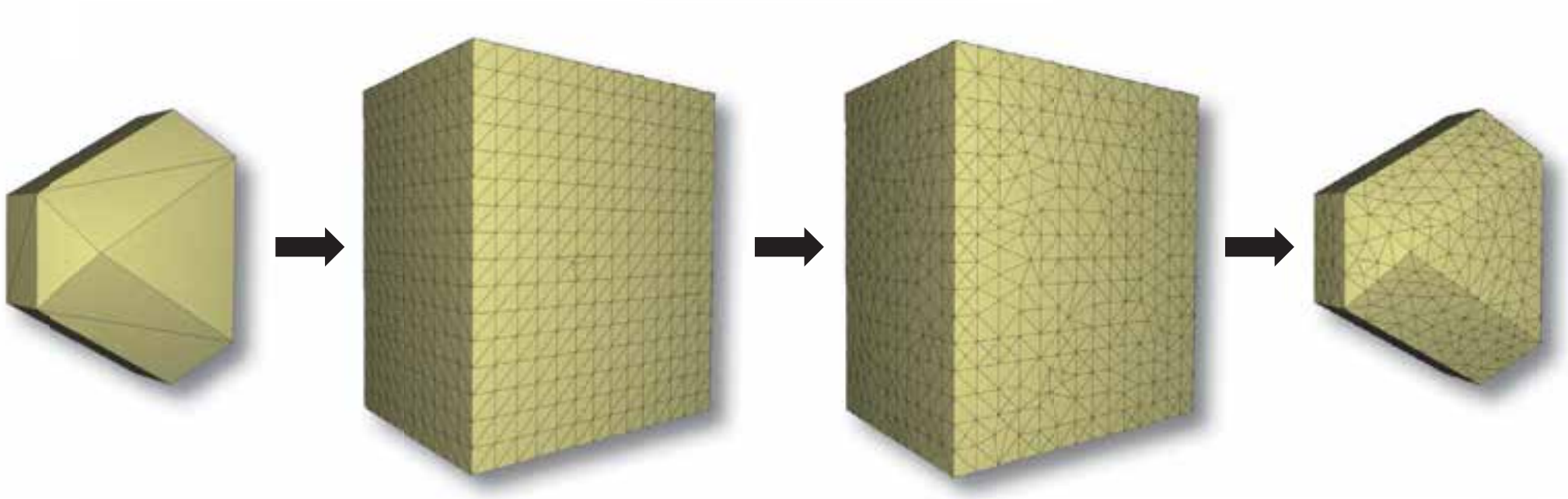
Pipeline



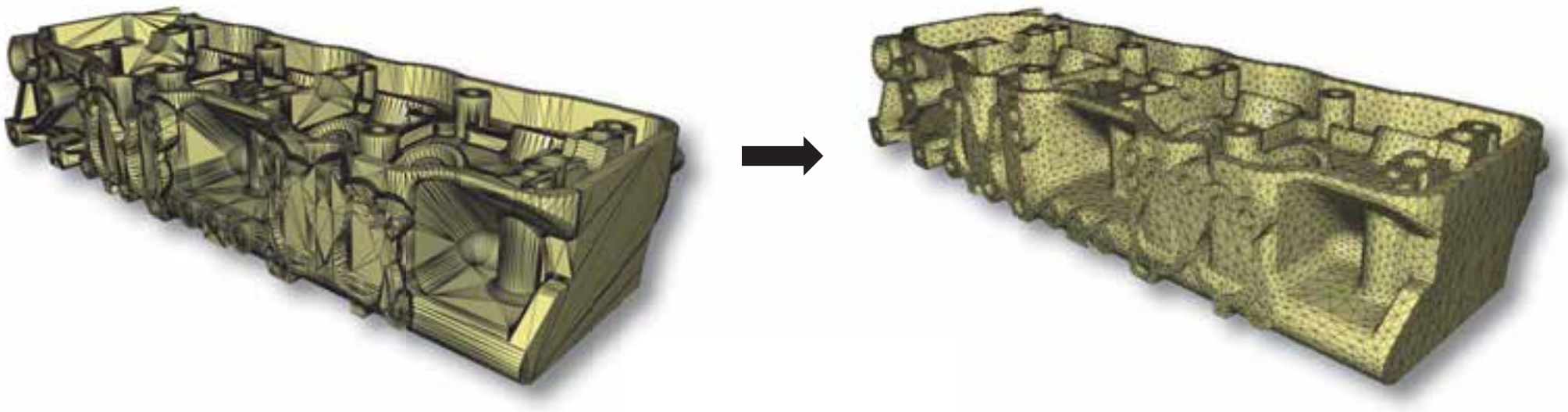
Stage I: Valid Mesh Generation (Rational)

Stage II: Mesh Improvement (Mixed → Double)

Pipeline



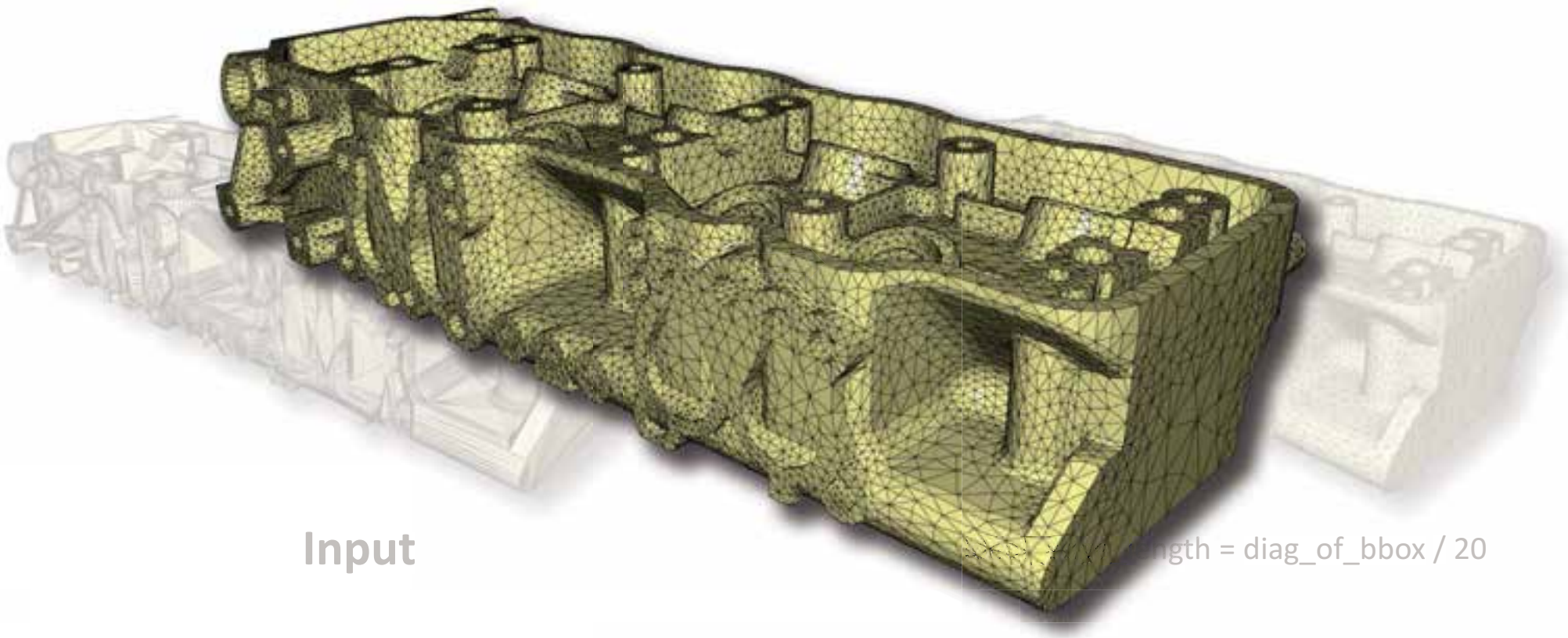
Parameters: Ideal Edge Length



Input

Ideal_edge_length = diag_of_bbox / 20

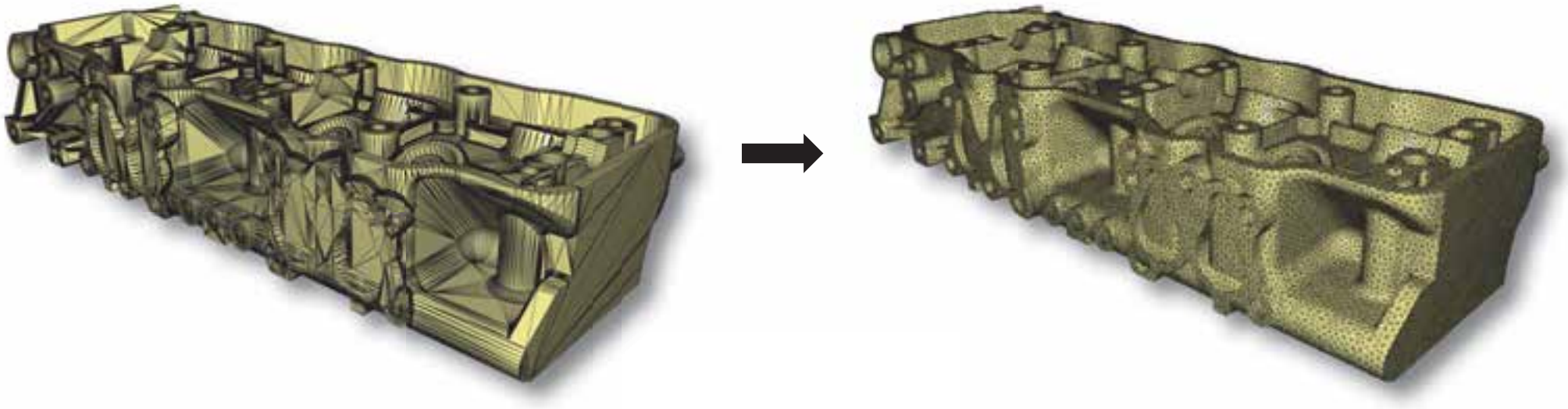
Parameters: Ideal Edge Length



Input

ngth = diag_of_bbox / 20

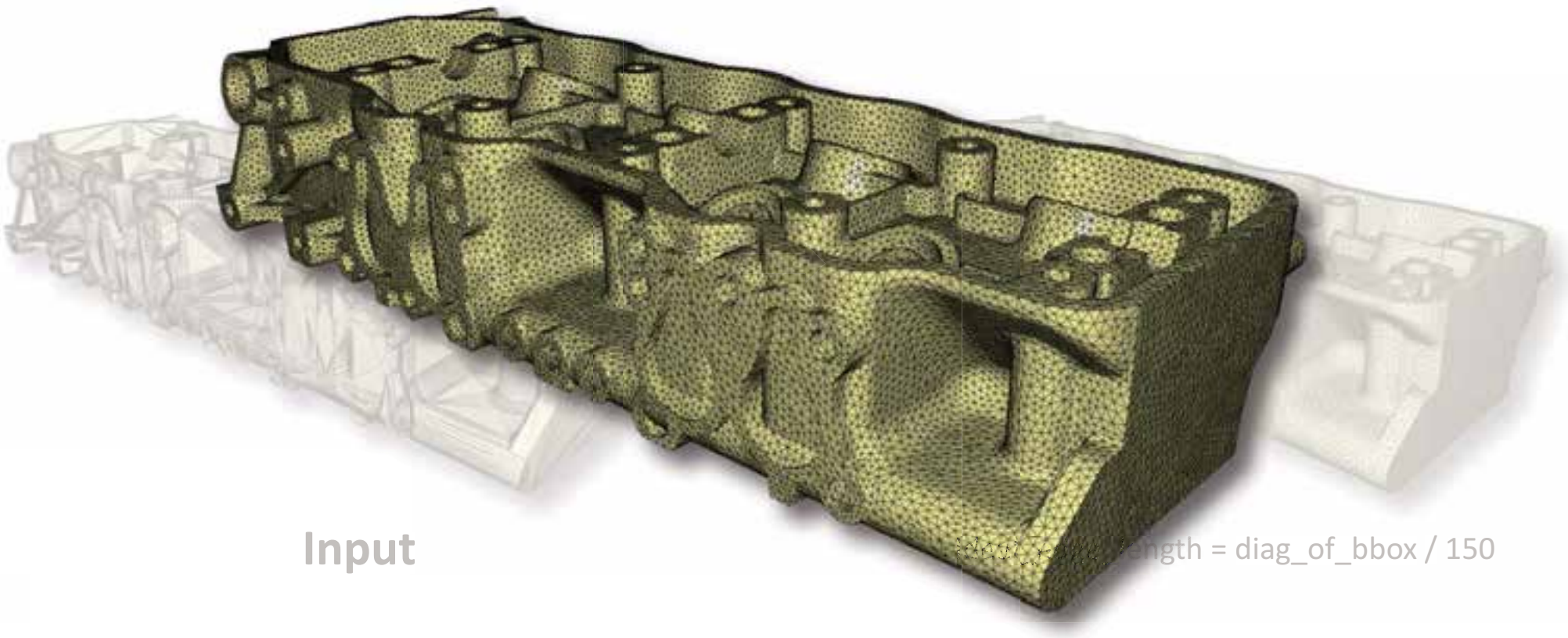
Parameters: Ideal Edge Length



Input

Ideal_edge_length = diag_of_bbox / 150

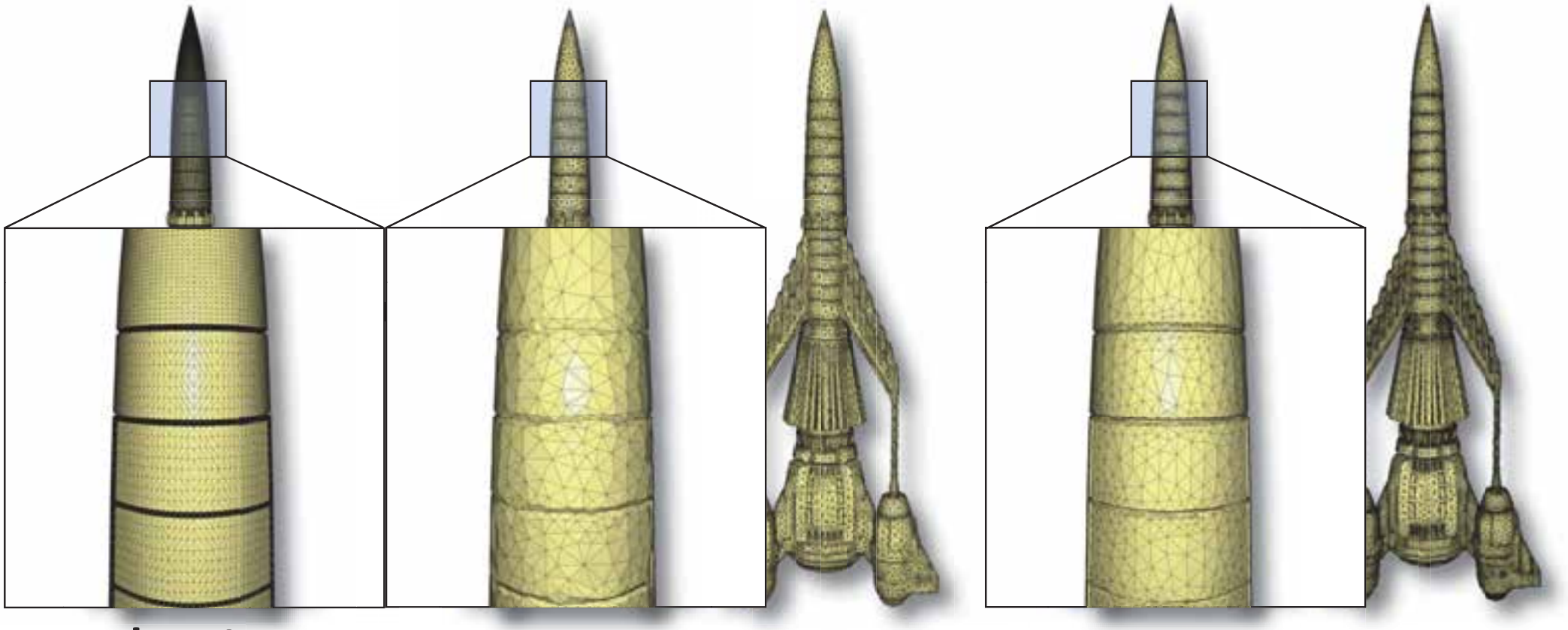
Parameters: Ideal Edge Length



Input

ngth = diag_of_bbox / 150

Parameters: Envelope Size



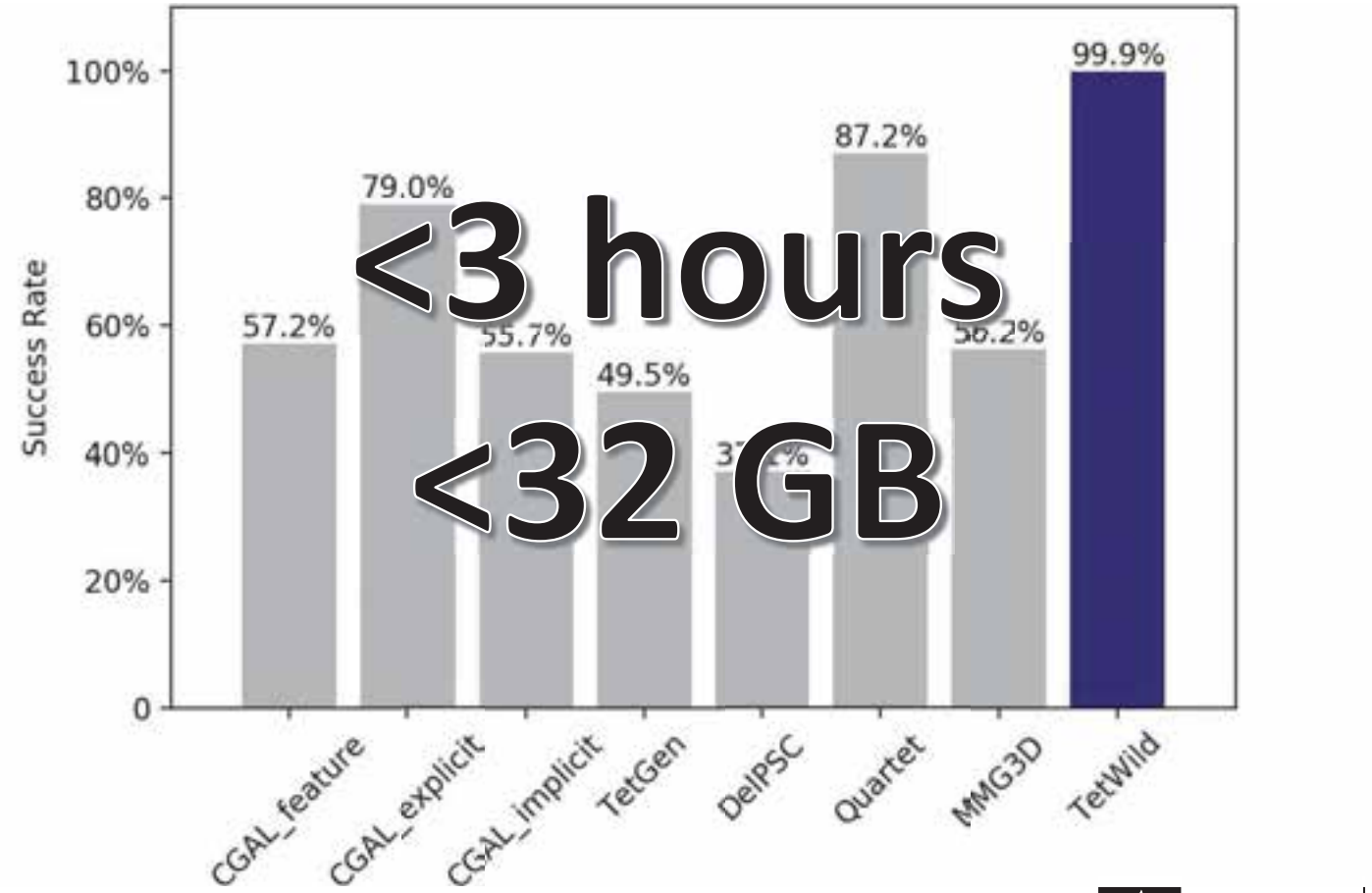
Input

Envelope size = diag_of_bbox / 1000

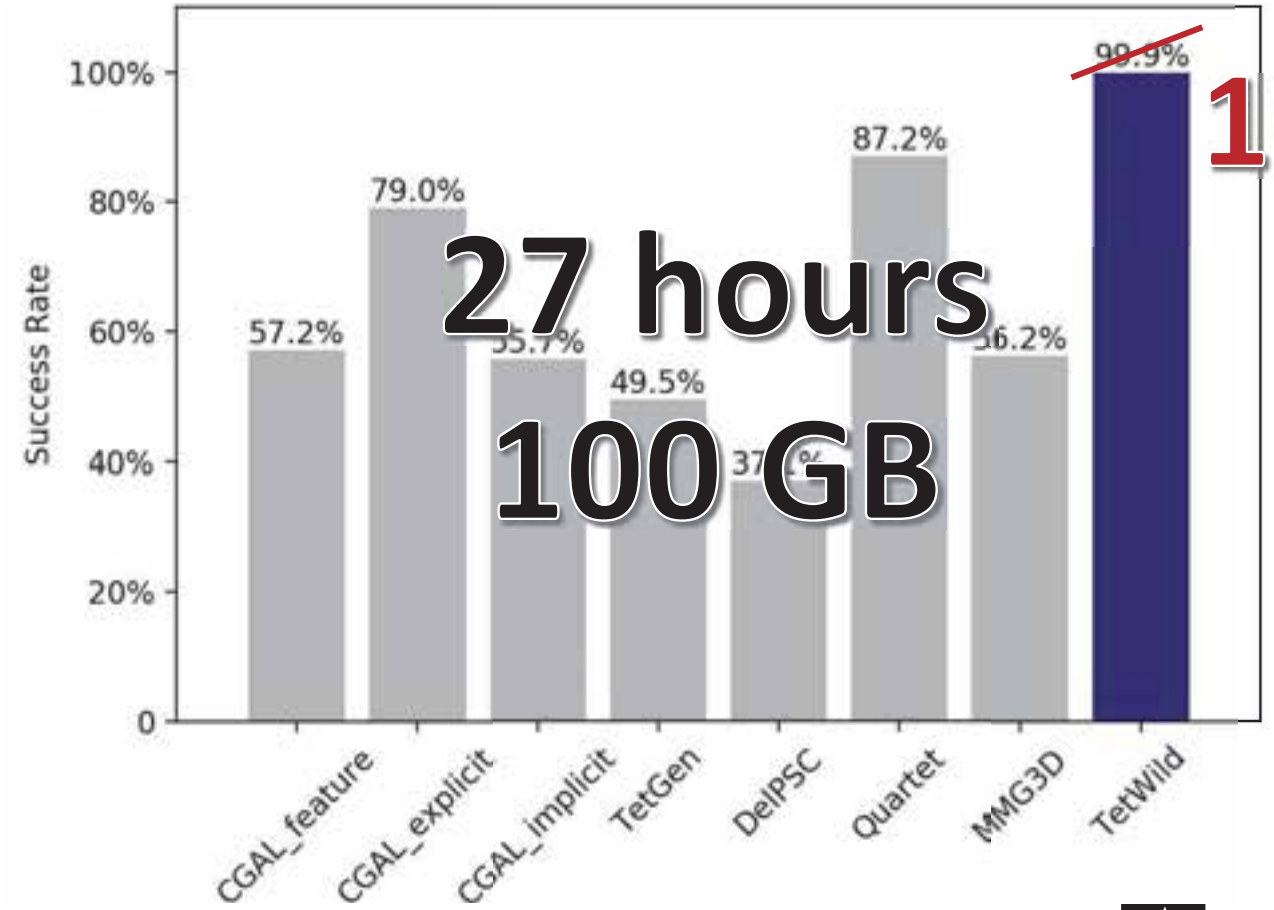
Envelope size = diag_of_bbox / 3000

Parameters: Envelope Size

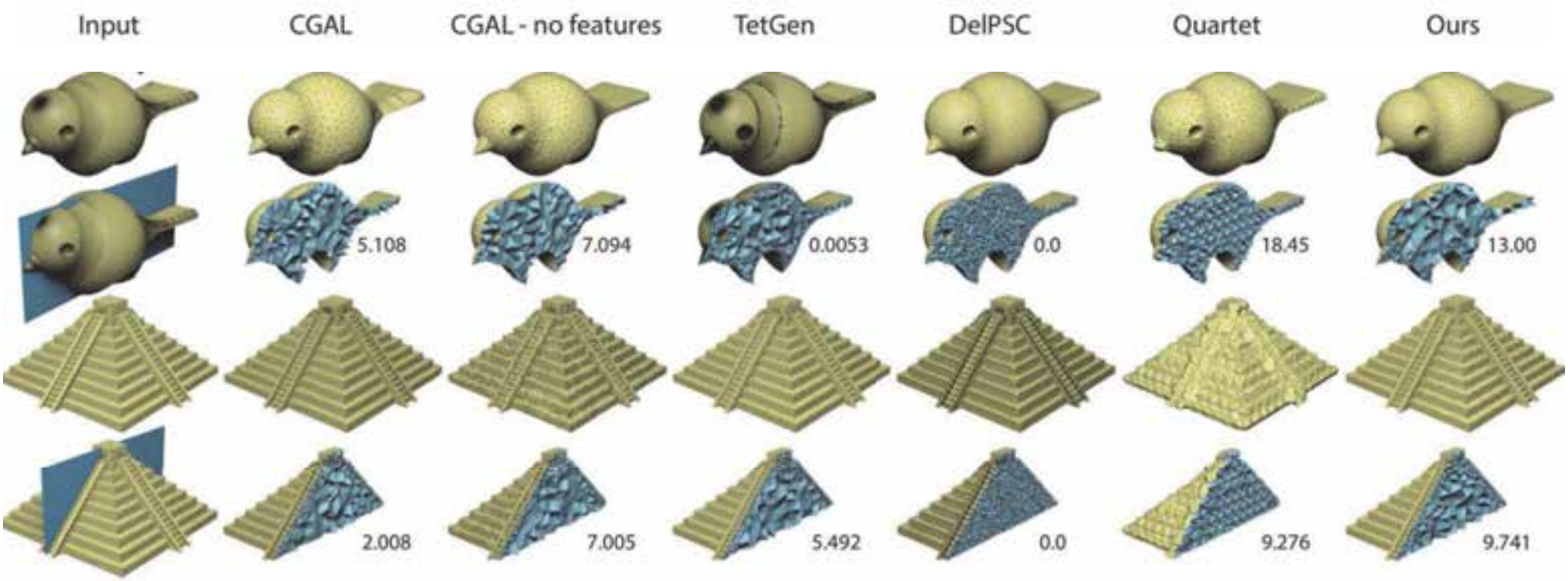
Comparison: Success Rate on 10k Models



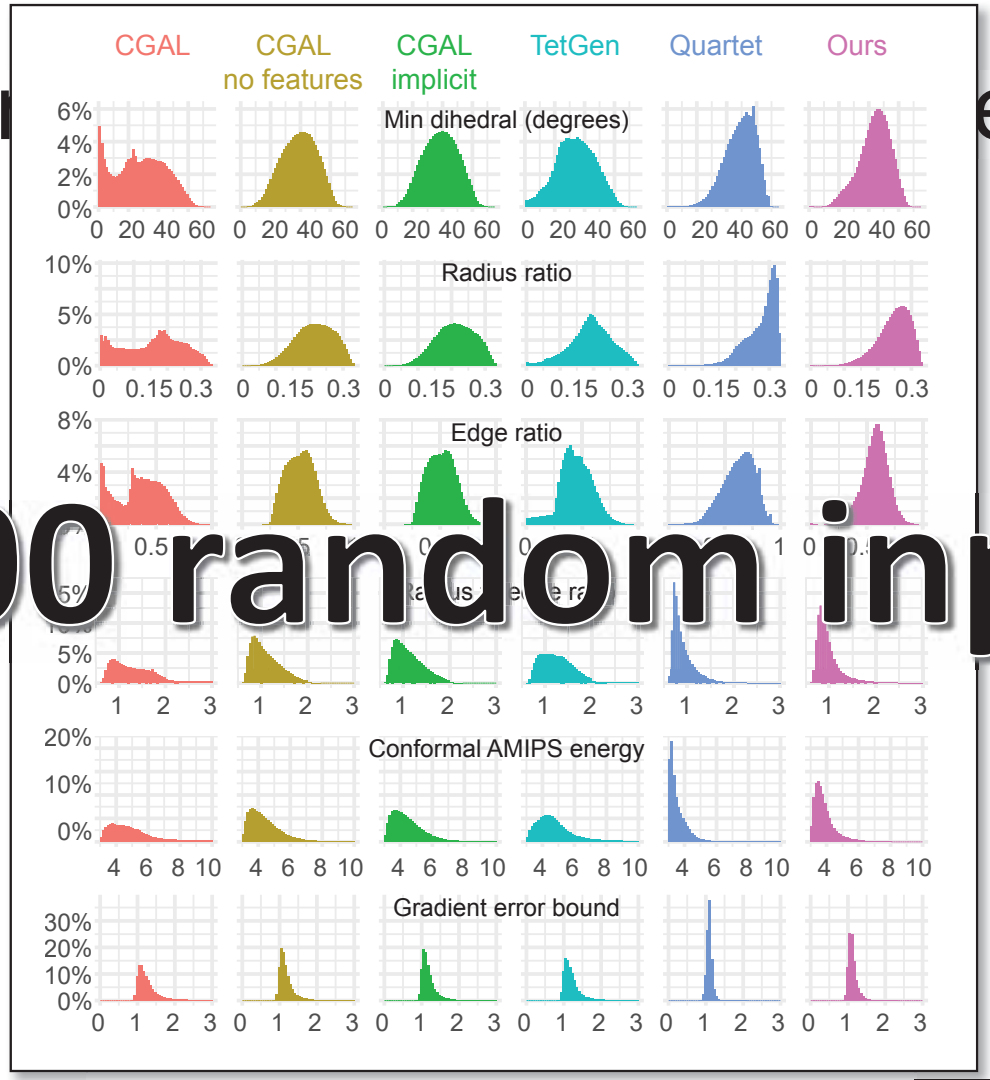
Comparison: Success Rate on 10k Models



Comparison

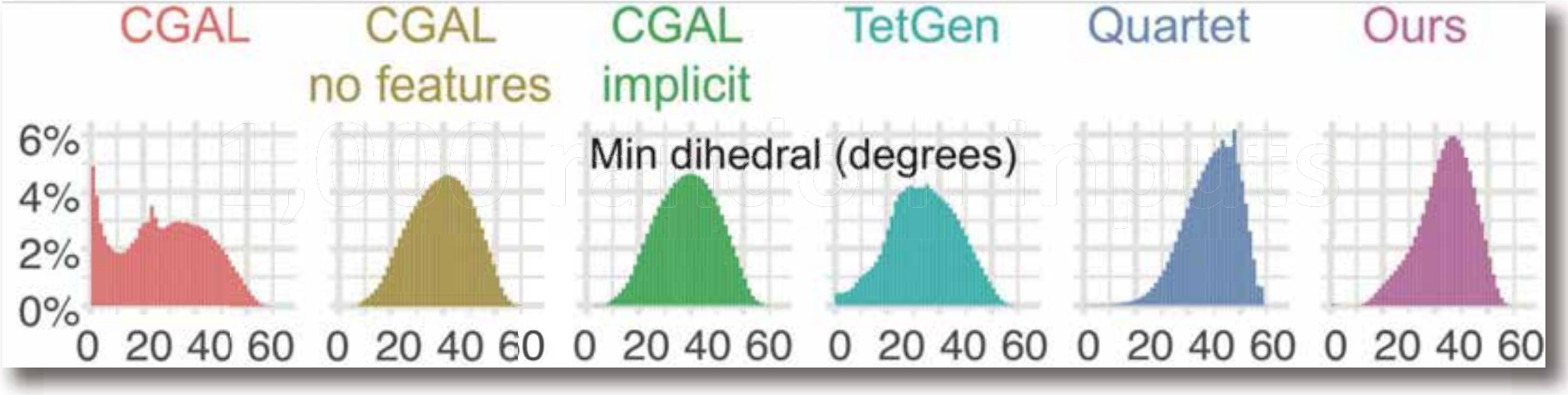

Input
CGAL
CGAL - no features
TetGen
DelPSC
Quartet
Ours
5.108
7.094
0.0053
0.0
18.45
13.00
2.008
7.005
5.492
0.0
9.276
9.741

Comparison easures



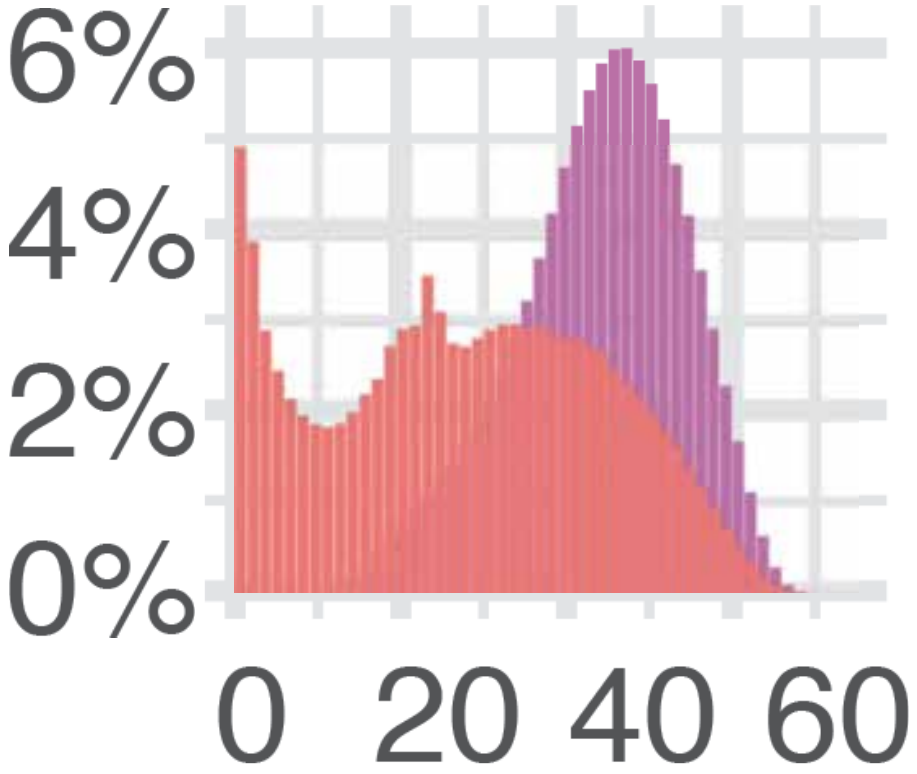
1,000 random inputs

Comparison: Quality in Different Measures



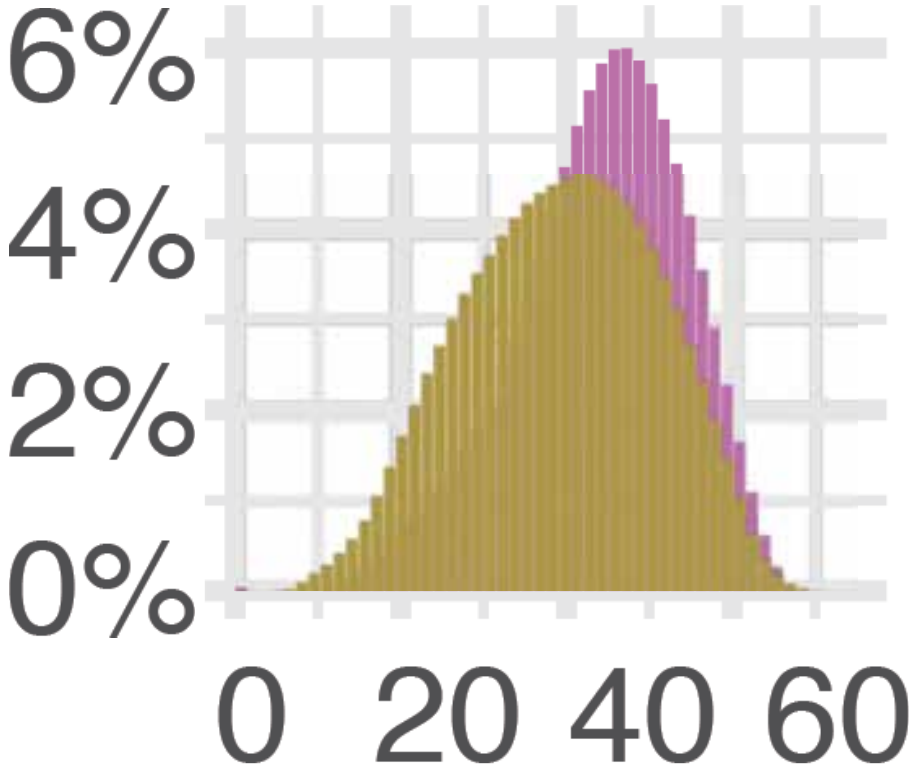
Comparison: Quality in Different Measures

Min dihedral angle:
TetWild vs. CGAL



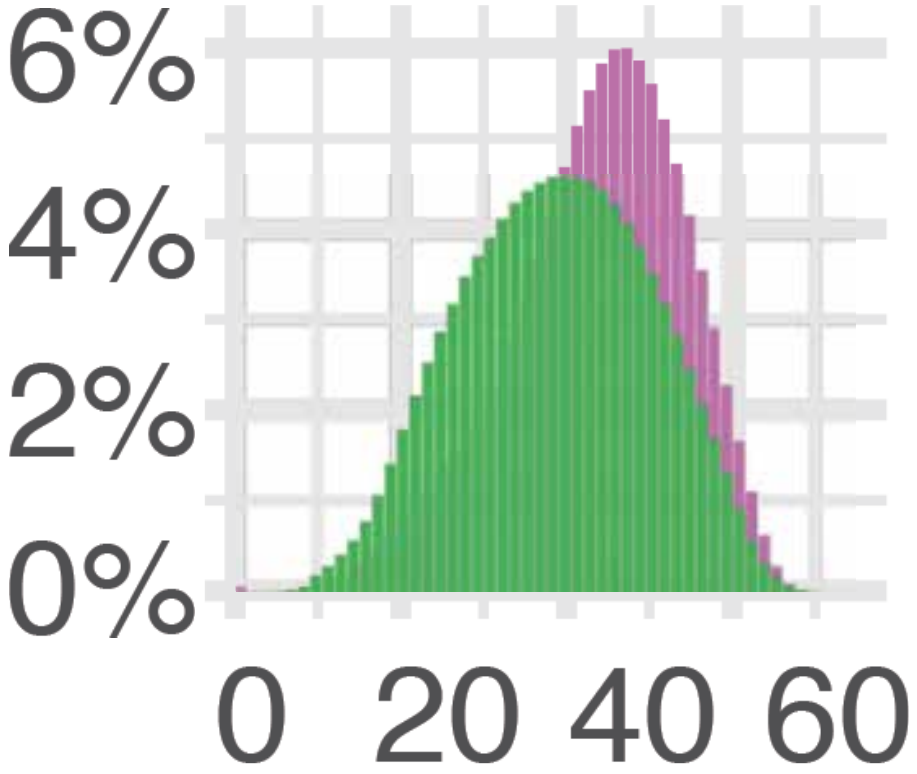
Comparison: Quality in Different Measures

Min dihedral angle:
TetWild vs.
CGAL no feature



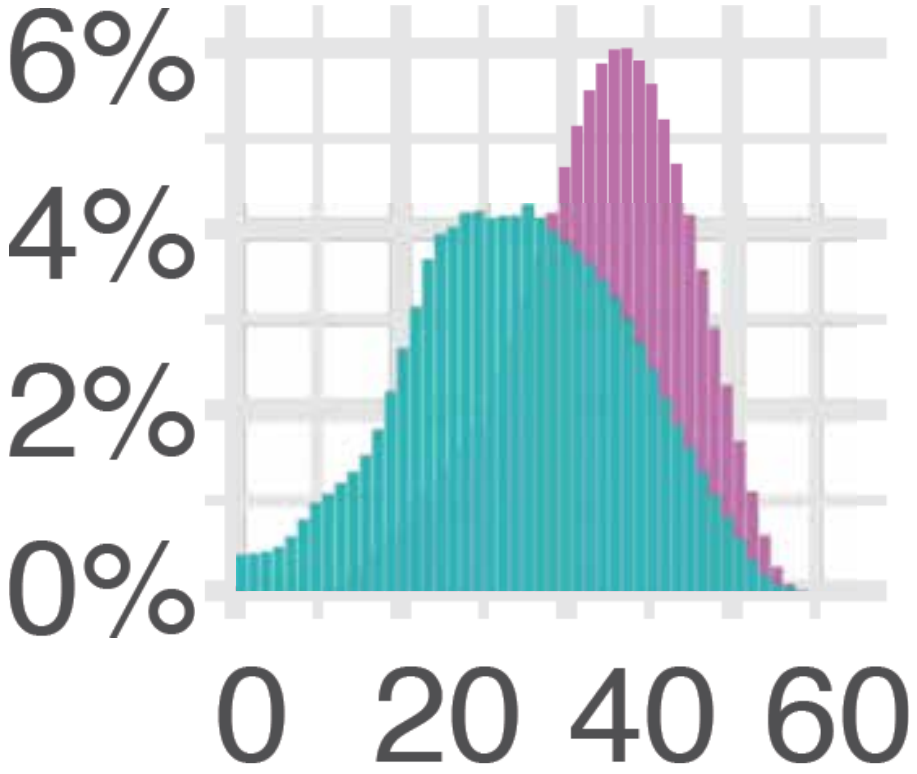
Comparison: Quality in Different Measures

Min dihedral angle:
TetWild vs.
CGAL implicit



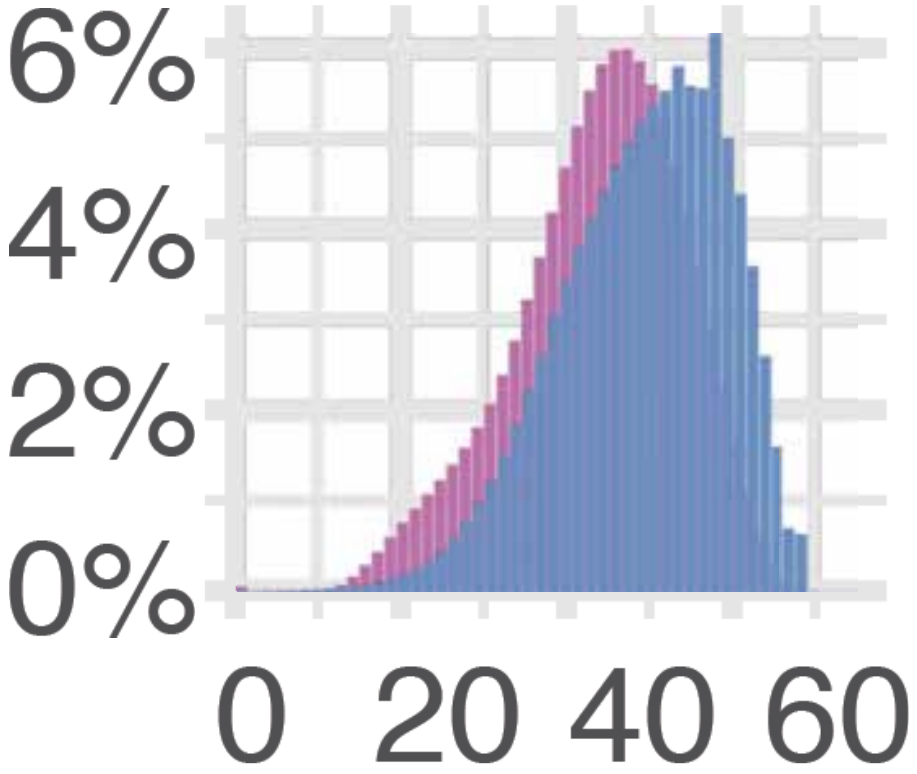
Comparison: Quality in Different Measures

Min dihedral angle:
TetWild vs. TetGen



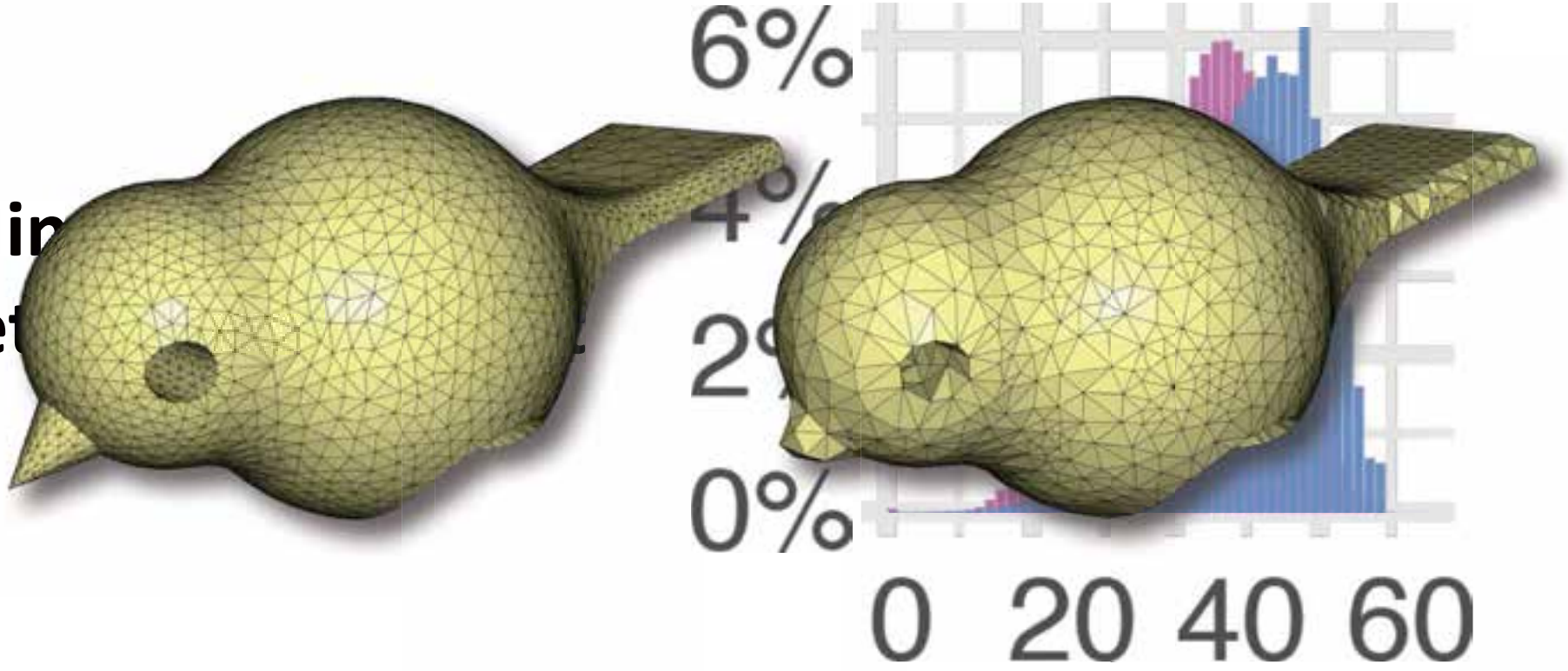
Comparison: Quality in Different Measures

Min dihedral angle:
TetWild vs. Quartet

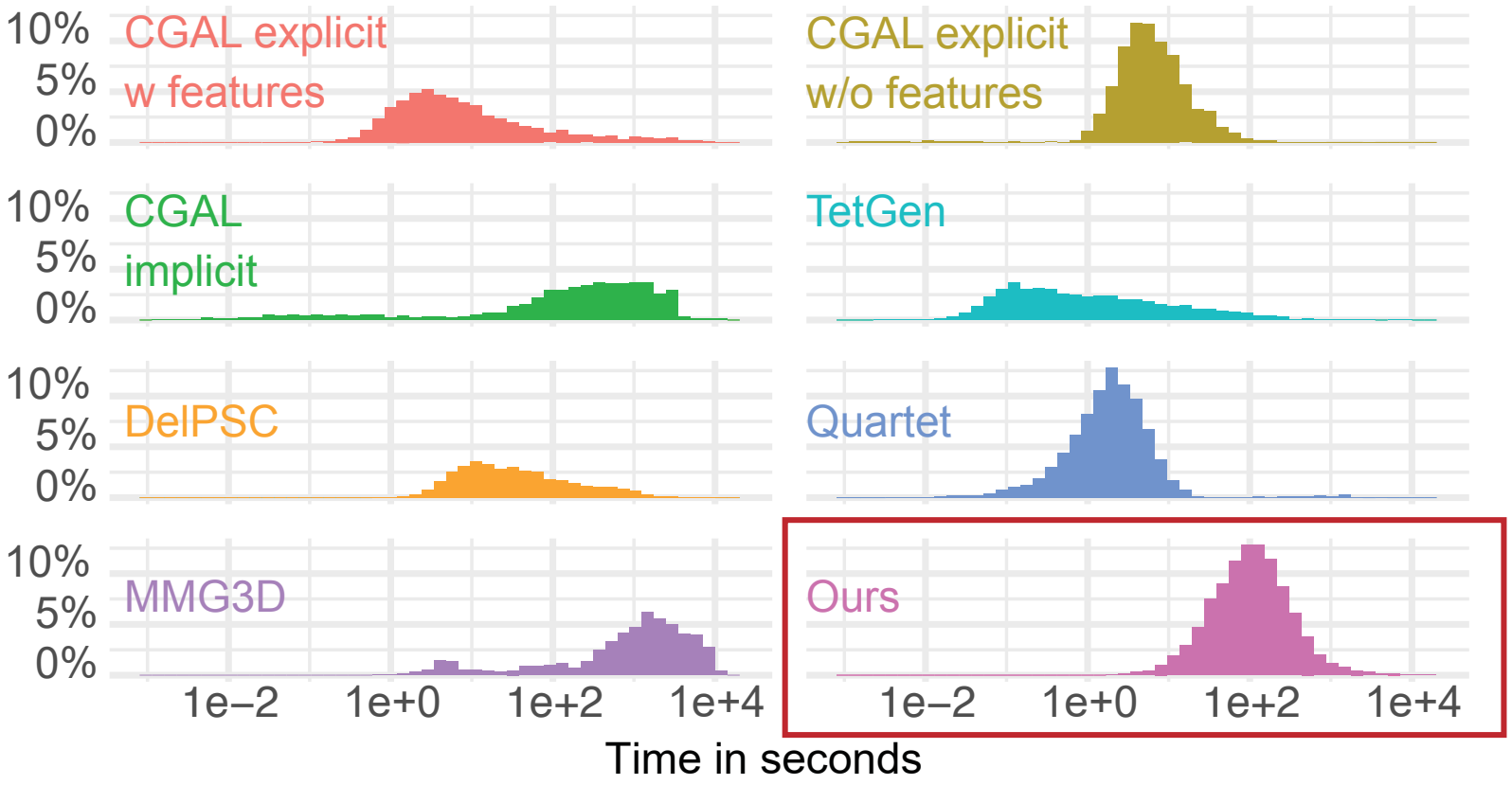


Comparison: Quality in Different Measures

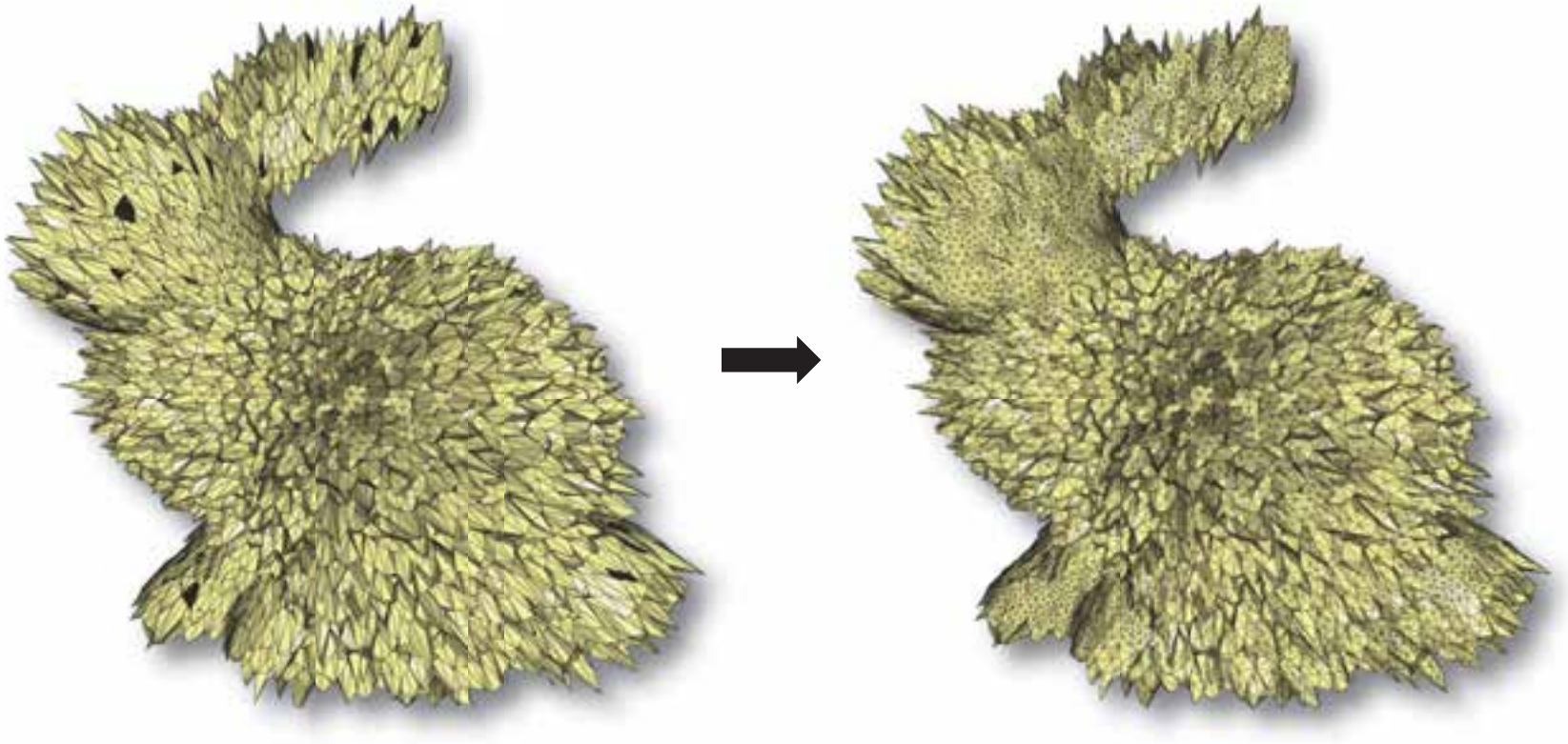
Min
Tet



Comparison: Running Time on 10k Models

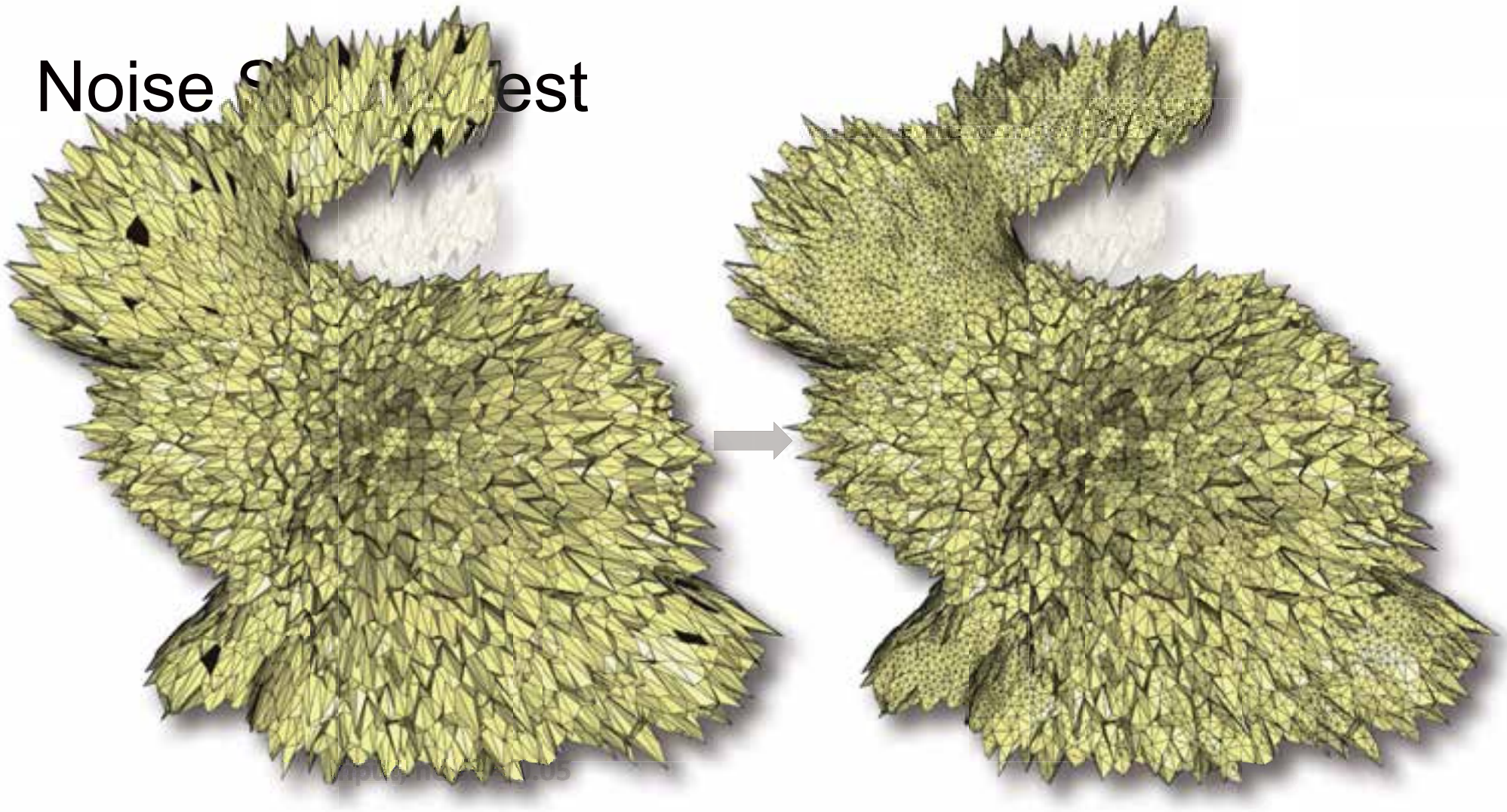


Noise Stress-Test

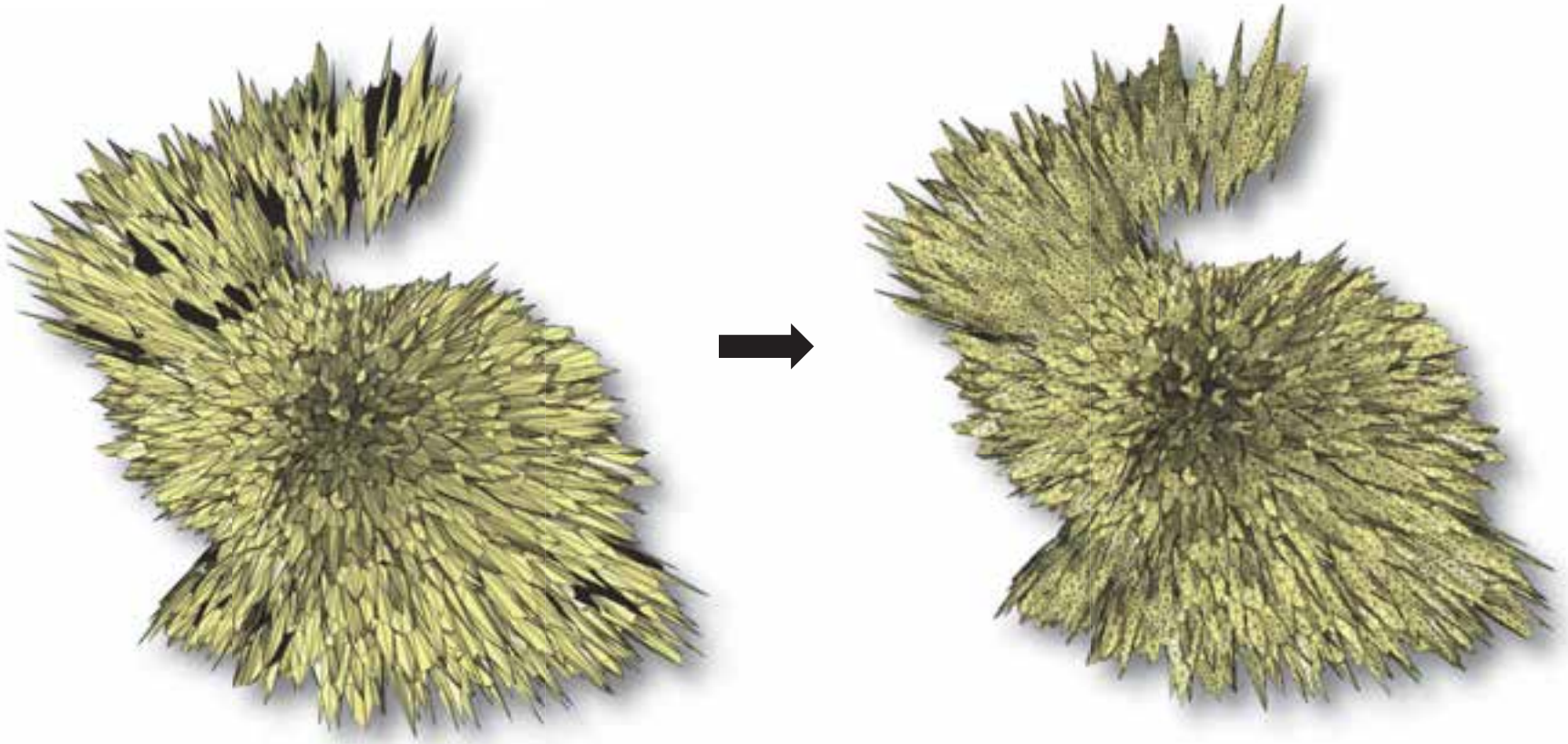


Input, noise = 0.05

Noise S[illegible]est

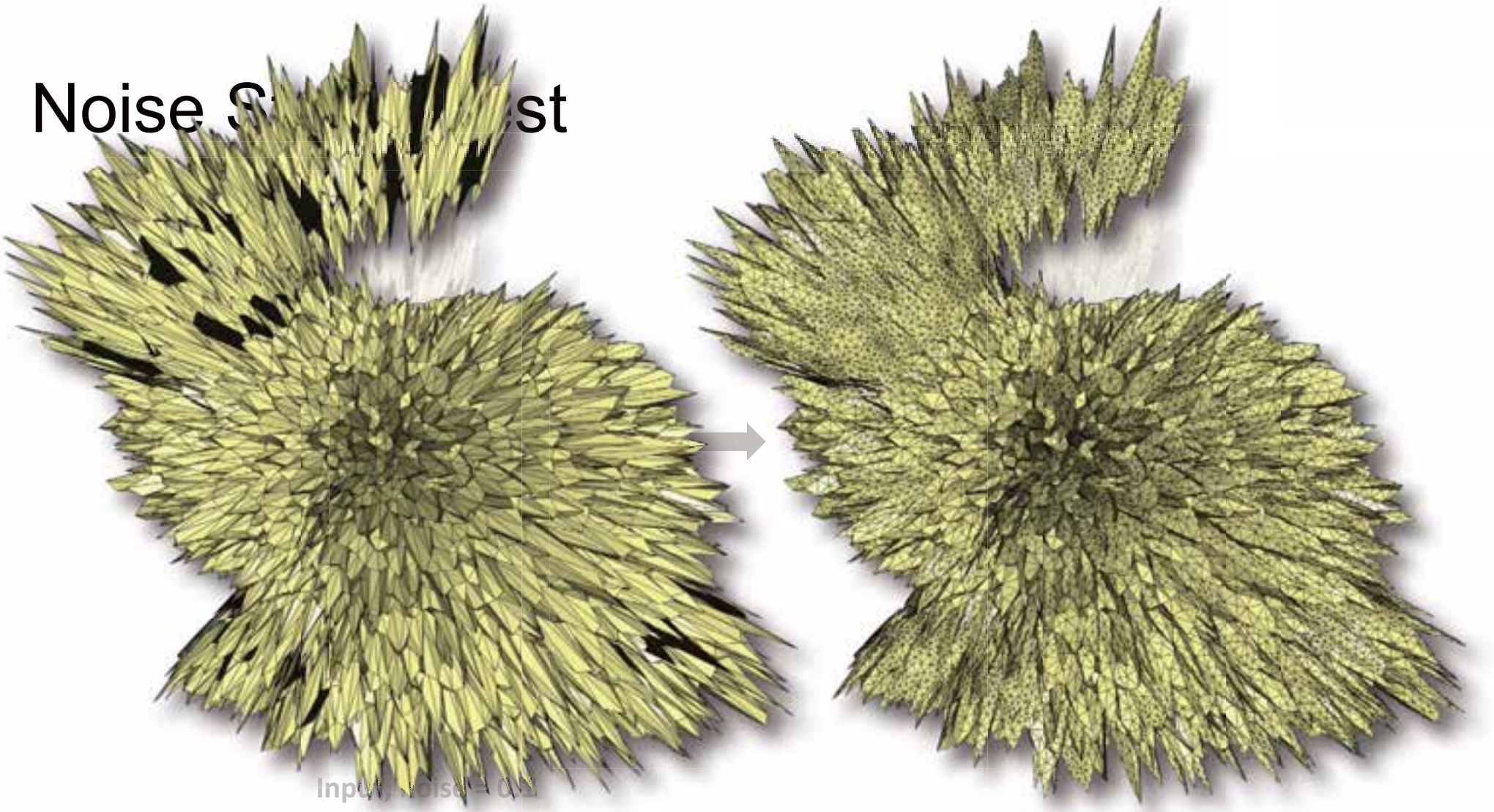


Noise Stress-Test

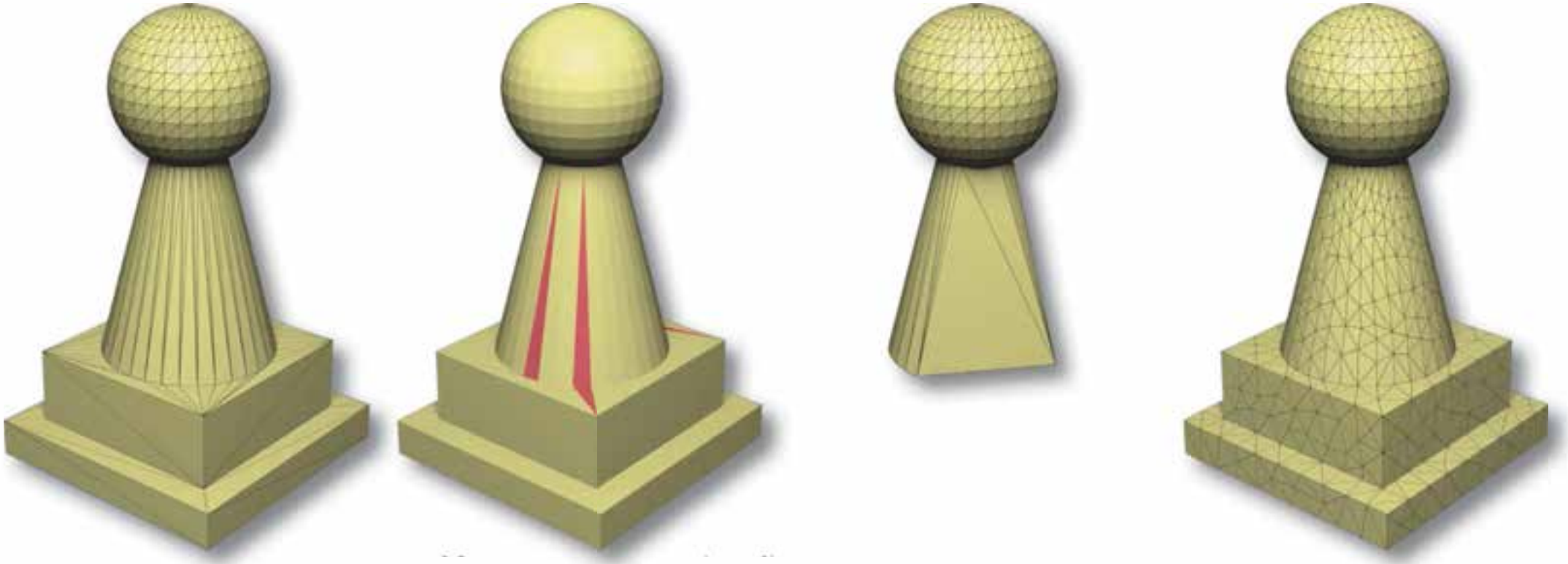


Input, noise = 0.1

Noise S[illegible]st

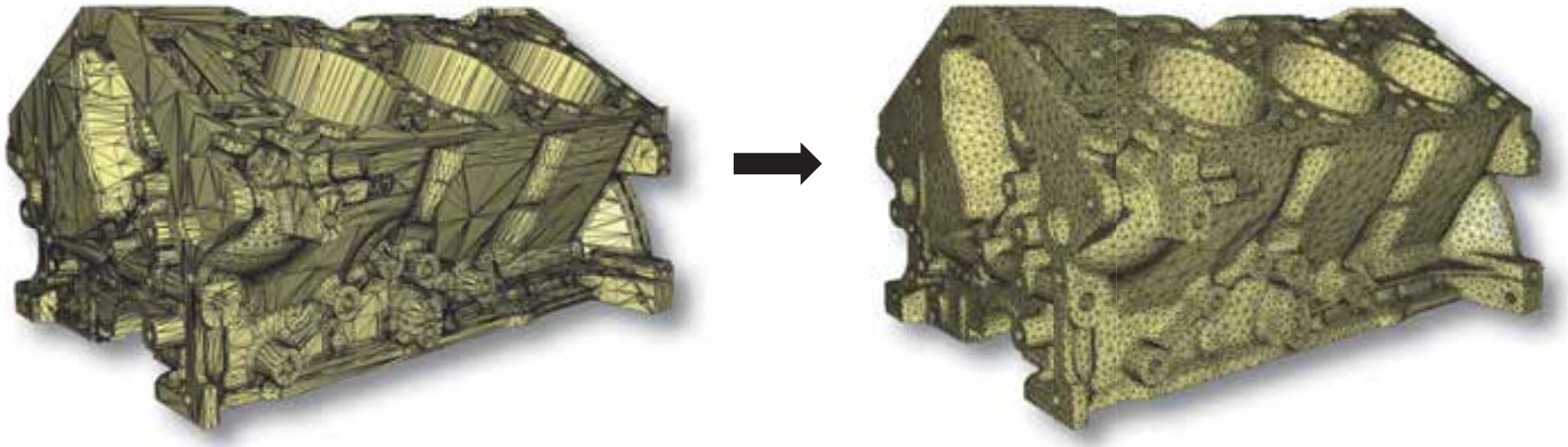


Surface Repair



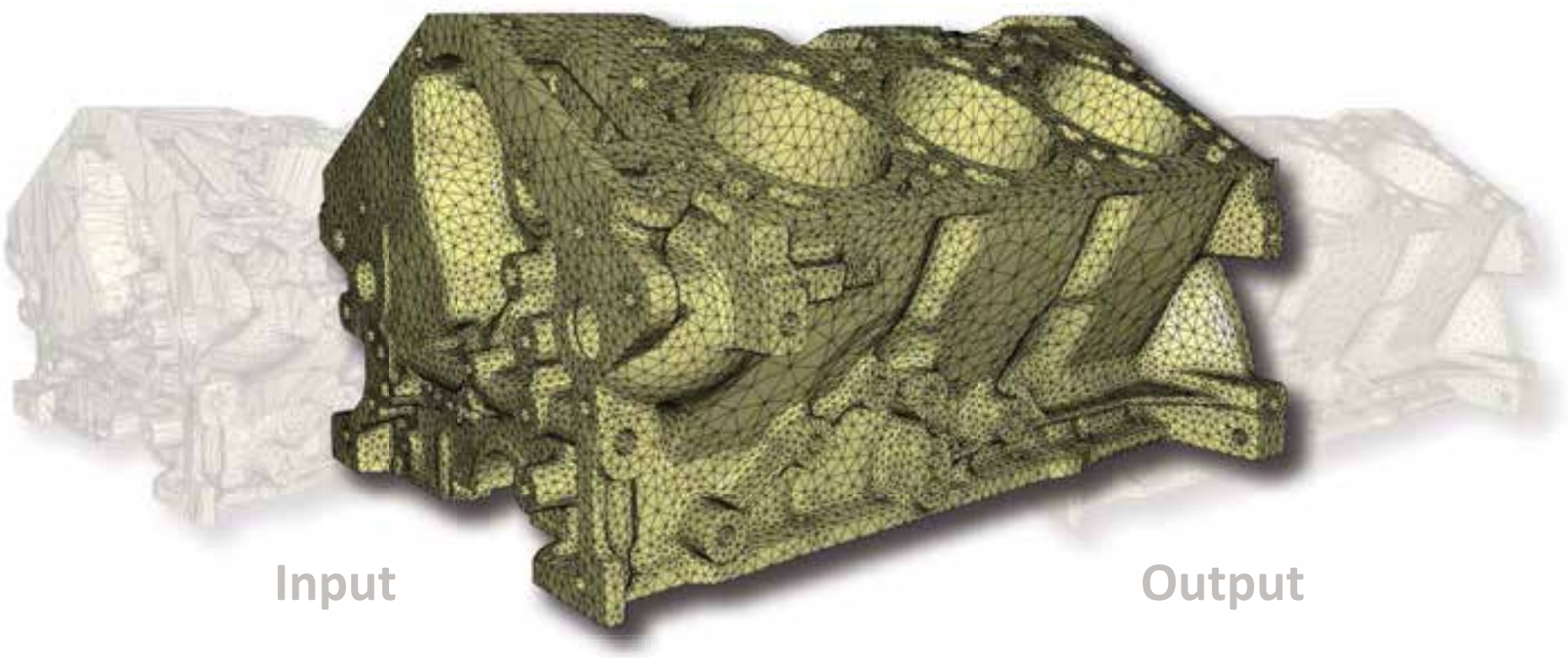
Input **Self-intersection(red)** **MeshFix** **TetWild**

CAD models

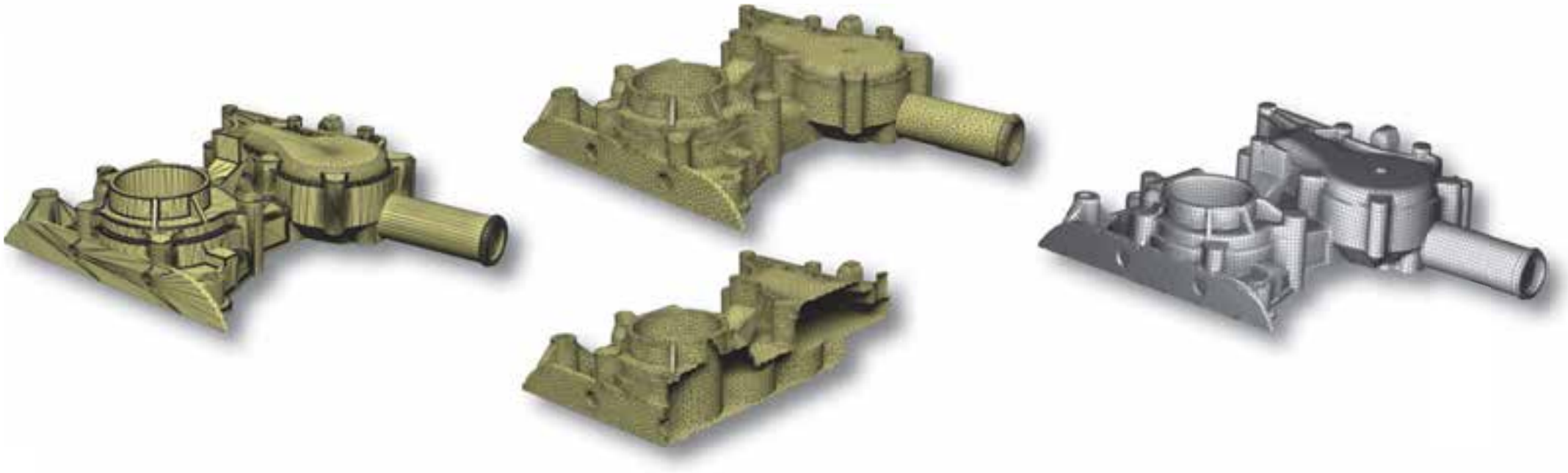


Input **Output**

CAD models

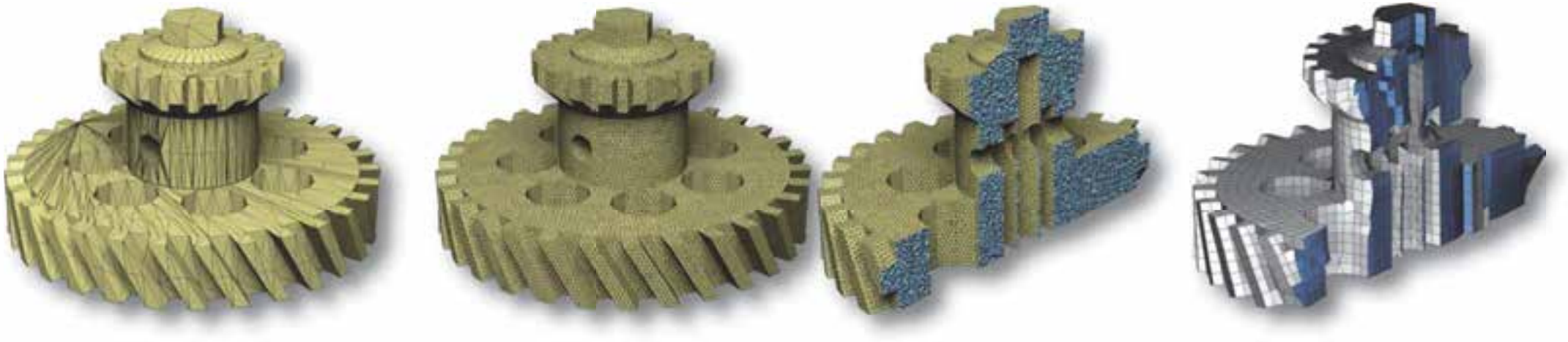


Structured Meshing: Quadmesh



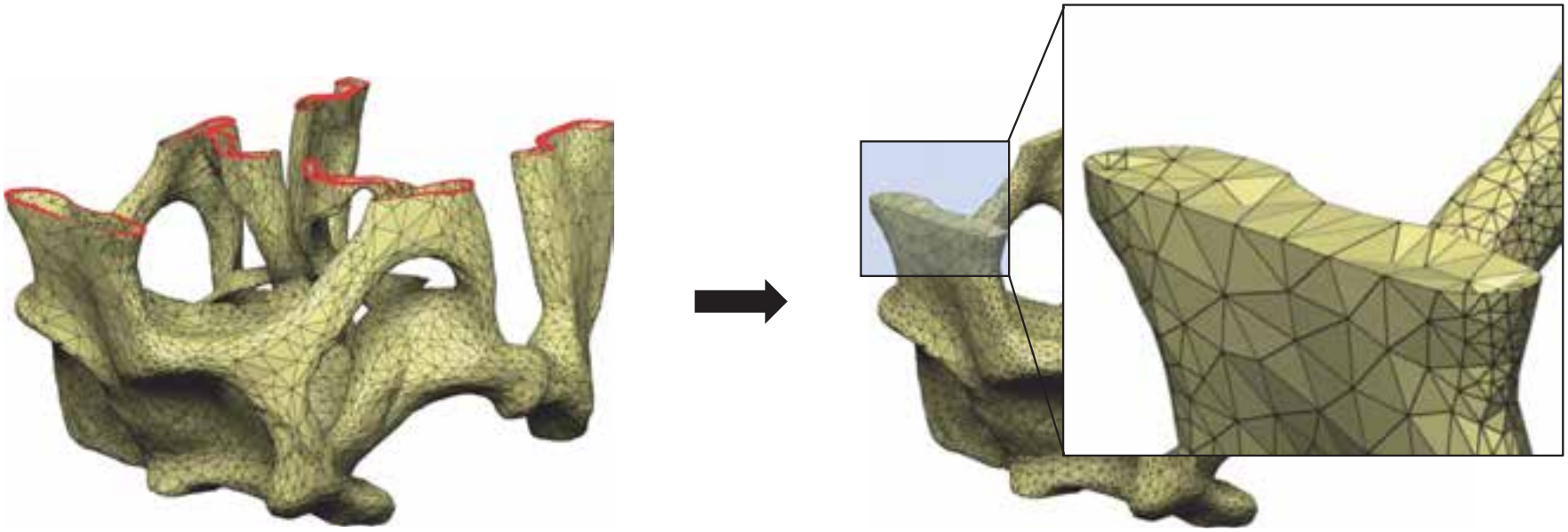
Input **Surface of Tetmesh** **Quadmesh**

Structured Meshing: Hexmesh



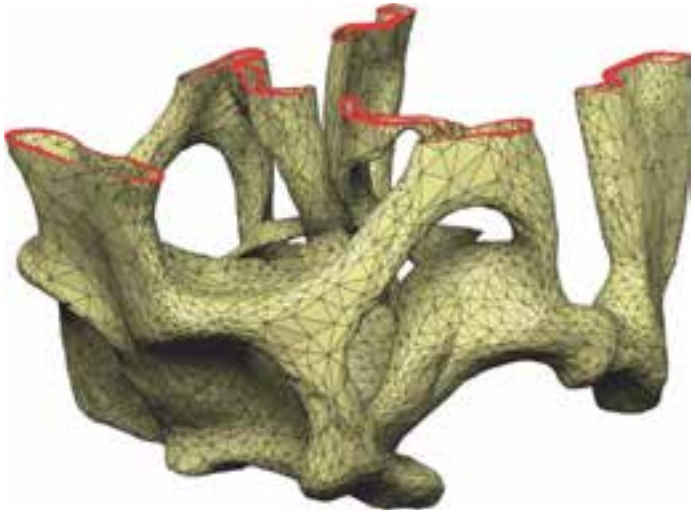
Input **Tetmesh** **Hex-dominant Mesh**

Input with Open Boundary

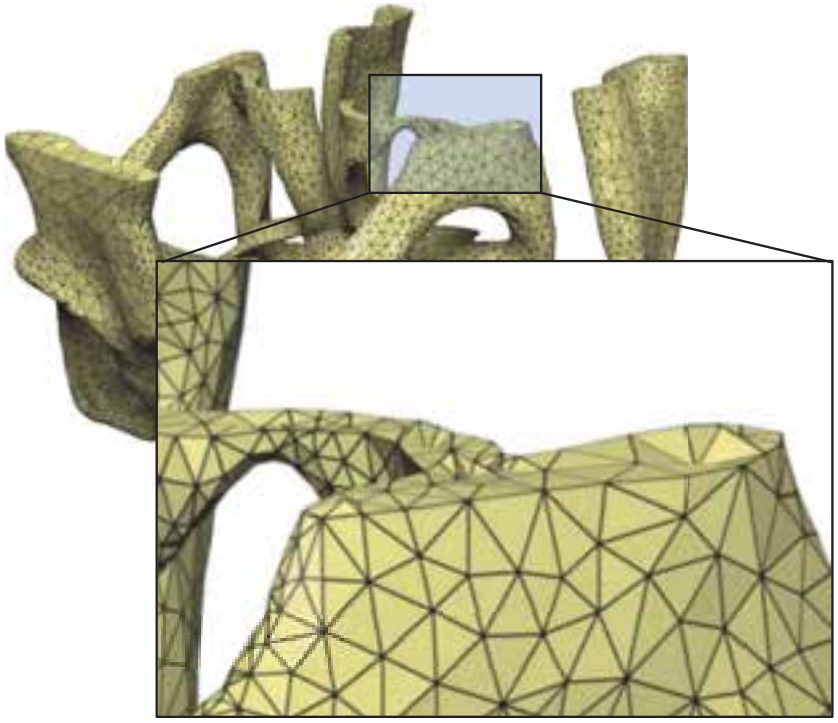


Input **Output w/ smoothing**

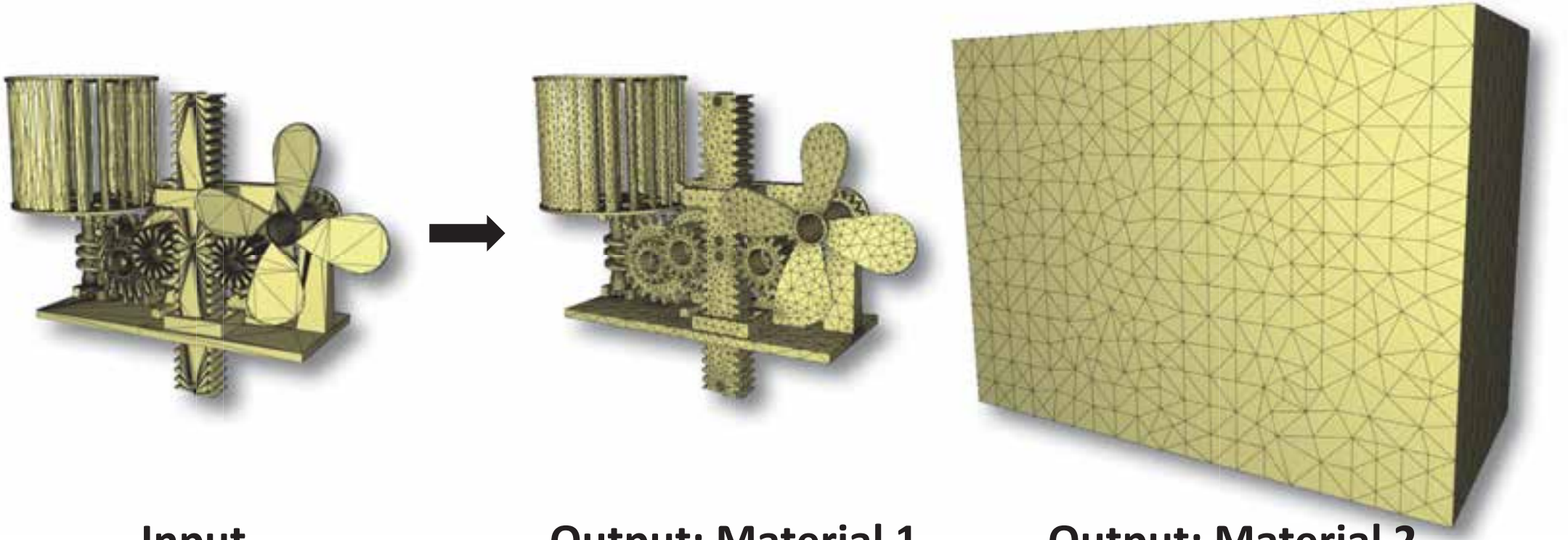
Input with Open Boundary



Input

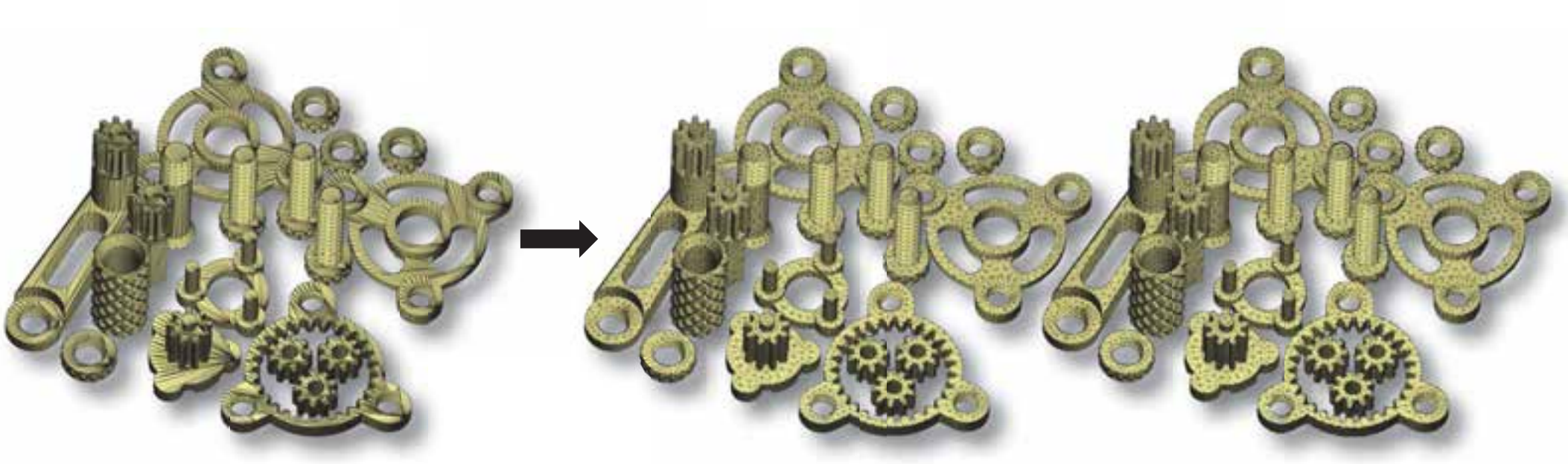


Meshing for Multi-material Solids



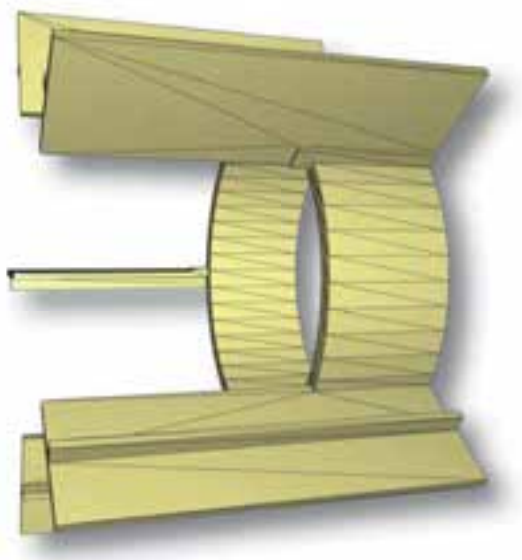
Input **Output: Material 1** **Output: Material 2**

Multi-component

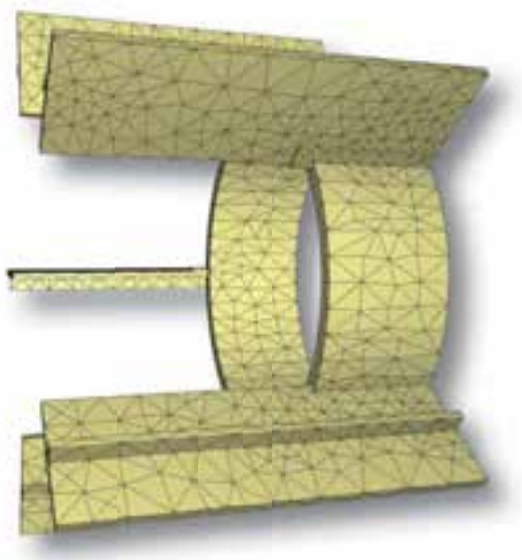


Input **Output**

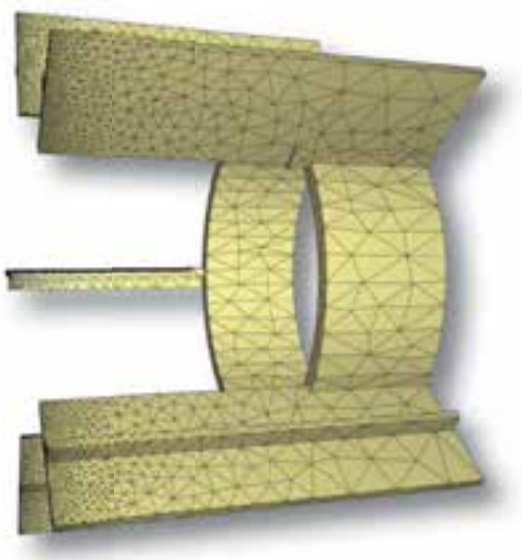
Spatially Varying Sizing Field



Input

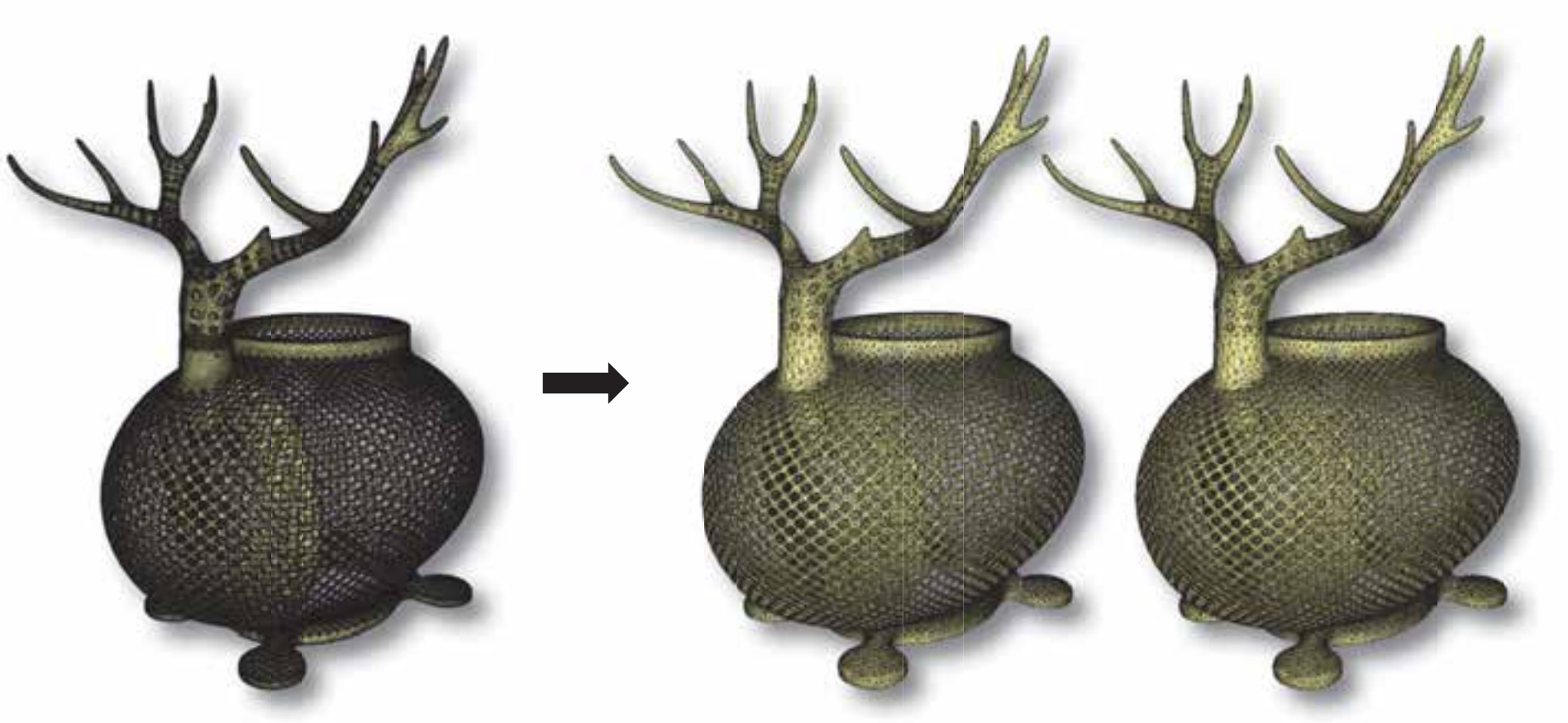


Without Sizing Field

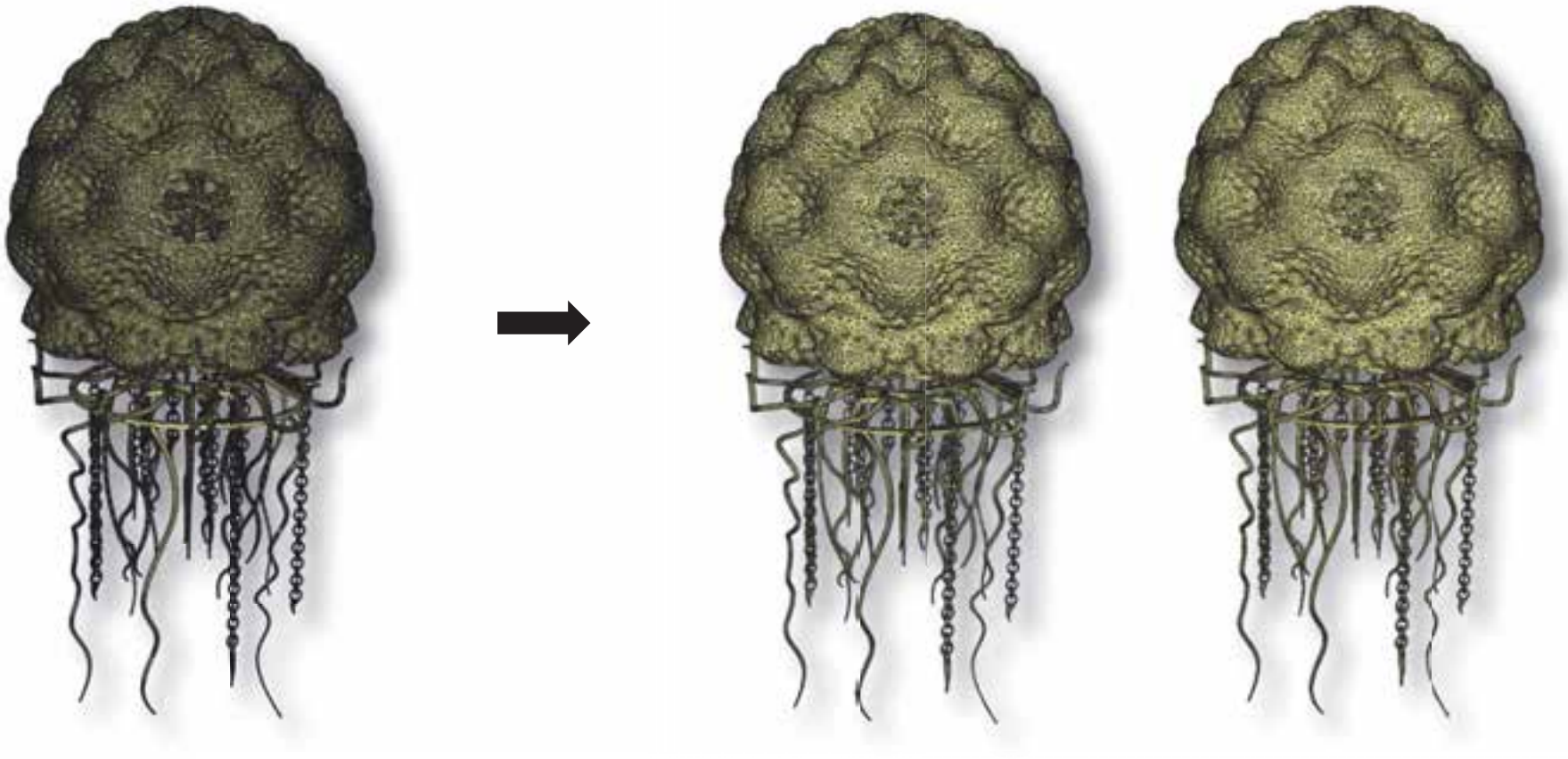


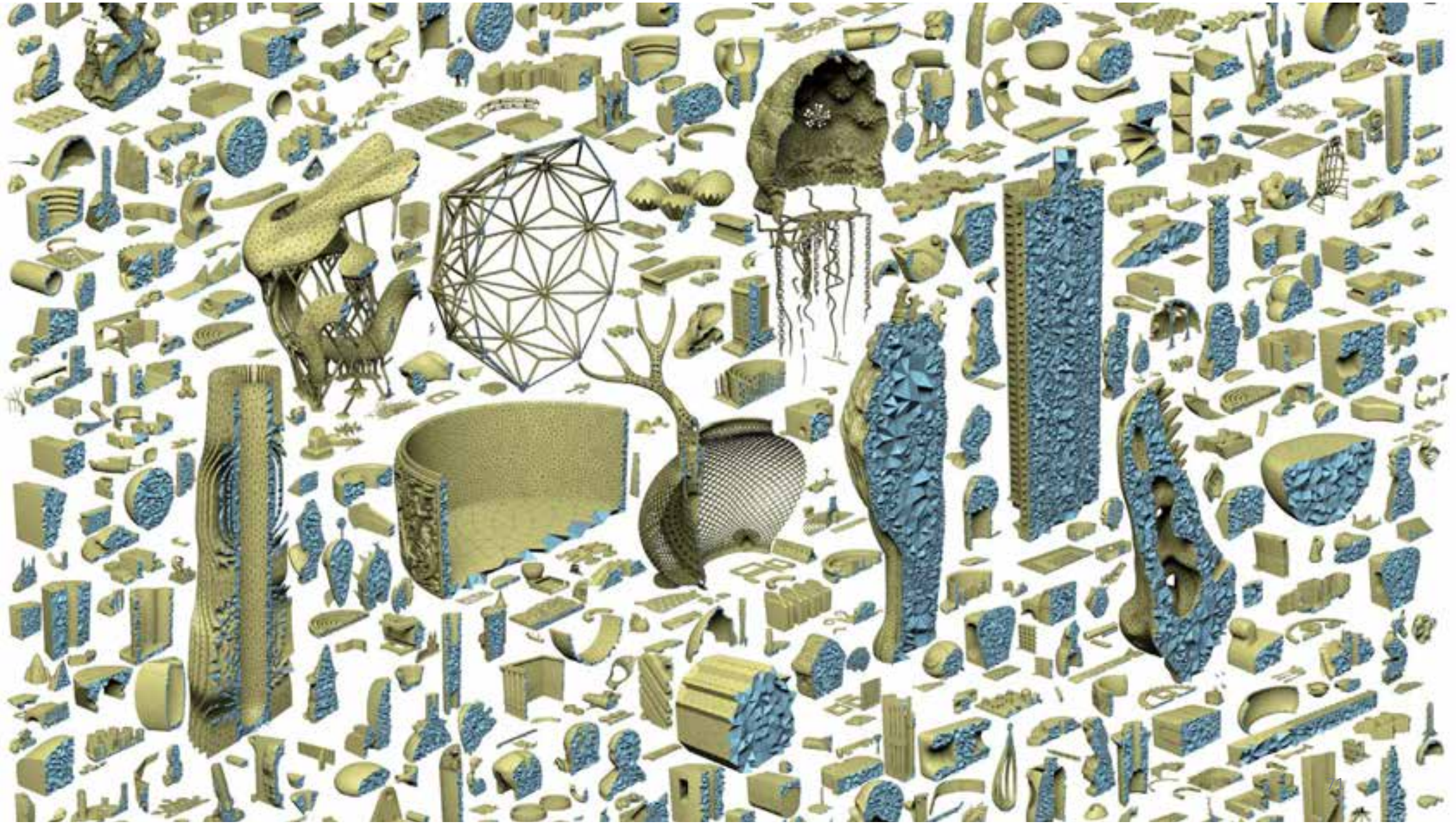
With Sizing Field

More Examples: Thin Structure



More Examples: Thin Structure

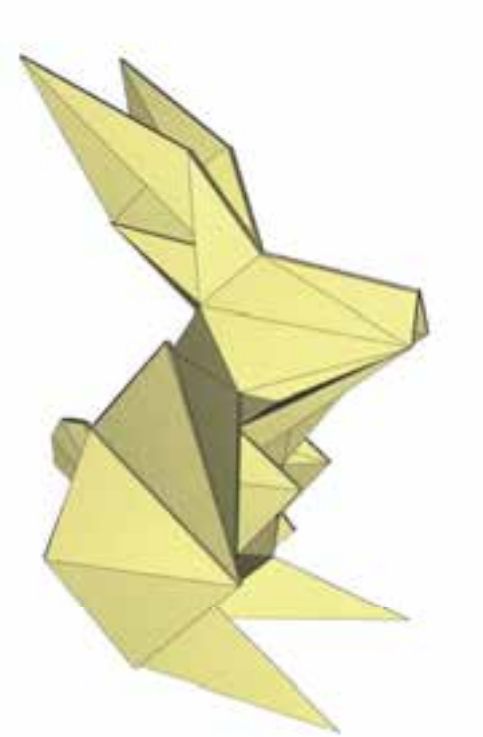




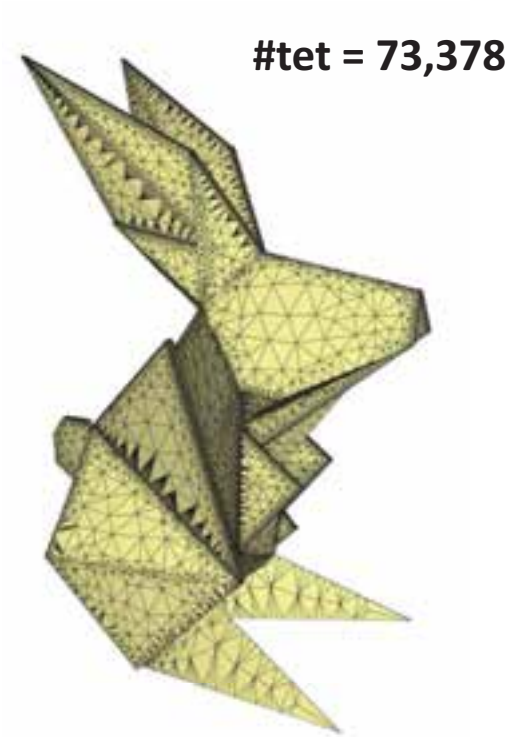
Limitations

- Our algorithm handles sharp features in a soft way.
  - There always a balance between how well the features are preserved and how much time it takes.
- For repairing surface purpose, our method is limited to surface without large open regions.
- Rational-based computation and envelop check are time-consuming.

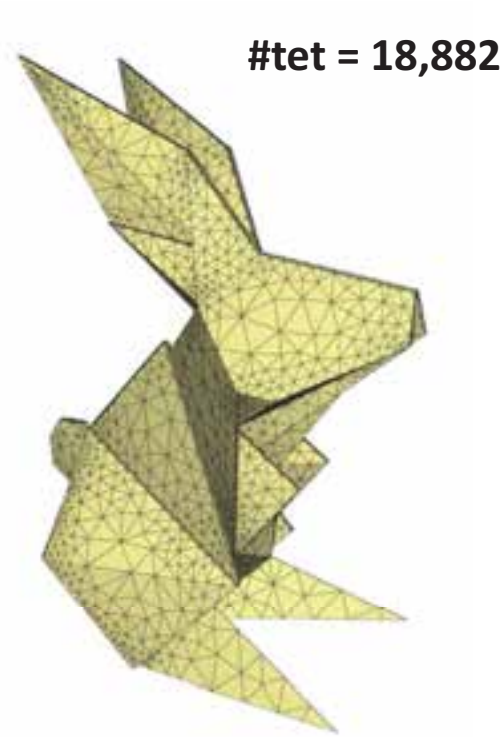
CGAL with Feature vs. Ours



Input

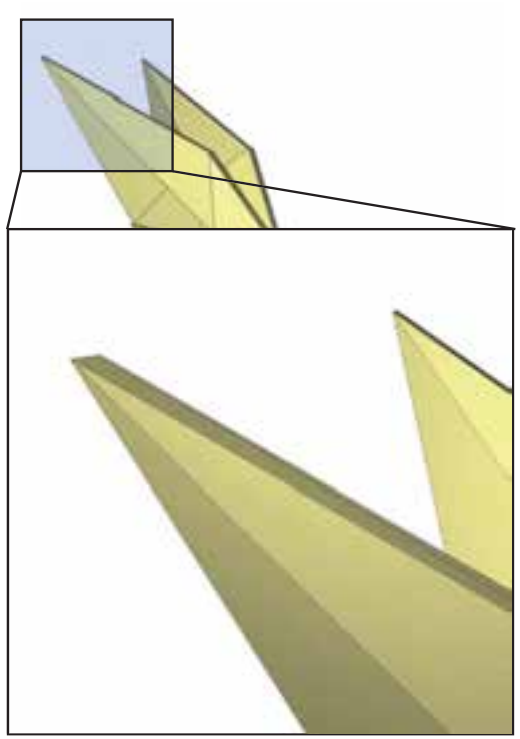


CGAL with feature

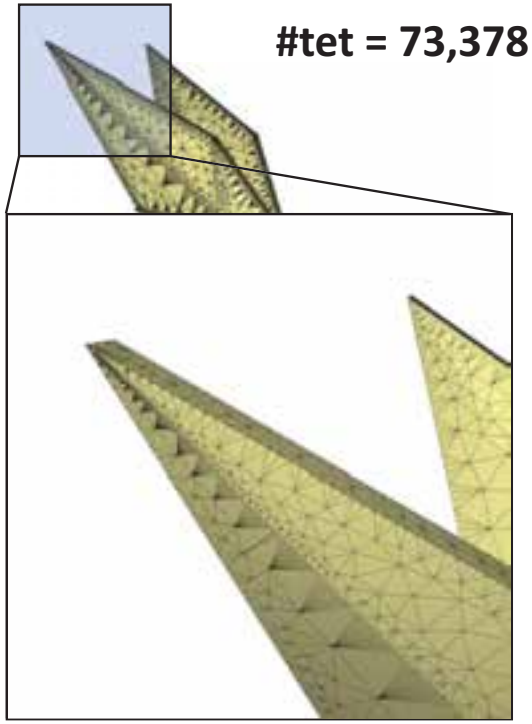


TetWild

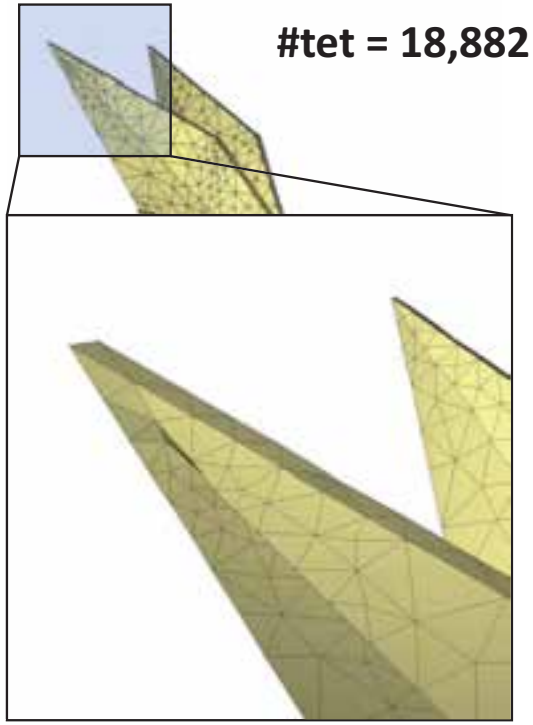
CGAL with Feature vs. Ours



Input



CGAL with feature



TetWild

Limitations

- Our algorithm handles sharp features in a soft way.
  - There always a balance between how well the features are preserved and how much time it takes.
- For repairing surface purpose, our method is limited to surface without large open regions.
- Rational-based computation and envelop check are time-consuming.

Conclusion

- **Robust:**
  - No assumption about the input.
  - Verified on 10,000 models and achieved 100% success rate.
- **Automatic:**
  - No complex user-controlled parameters.
  - Pre-set general parameters tested on 10k dataset.
- **High-quality output.**

Open-source Implementation

- GitHub Repository: **https://github.com/Yixin-Hu/TetWild**

Acknowledgements

- We are grateful to Jérémie Dumas for illuminating discussions, to Wenzel Jakob for the Mitsuba renderer, to Thingiverse and the Stanford 3D Scanning Repository for the datasets, and to NYU HPC staff for providing computing cluster service.
- This work was supported in part by NSF CAREER award 1652515, NSF grant (IIS-1320635 & DMS-1436591), NSERC Discovery Grants (RGPIN-2017-05235 & RGPAS-2017-507938), Canada Research Chair award, Connaught Fund, and a gift from Adobe Systems and nTopology Inc..

Thank you!

- Email: yixin.hu@nyu.edu
- GitHub Repository: **https://github.com/Yixin-Hu/TetWild**