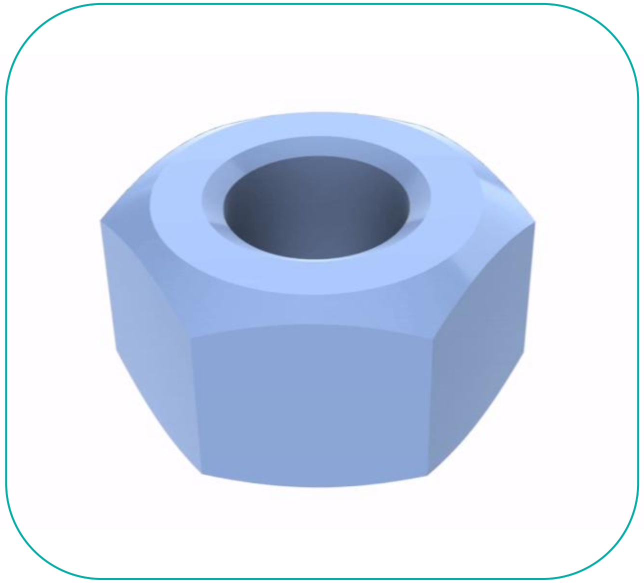
SIGGRAPH 2025

The Premier Conference & Exhibition on Computer Graphics & Interactive Techniques

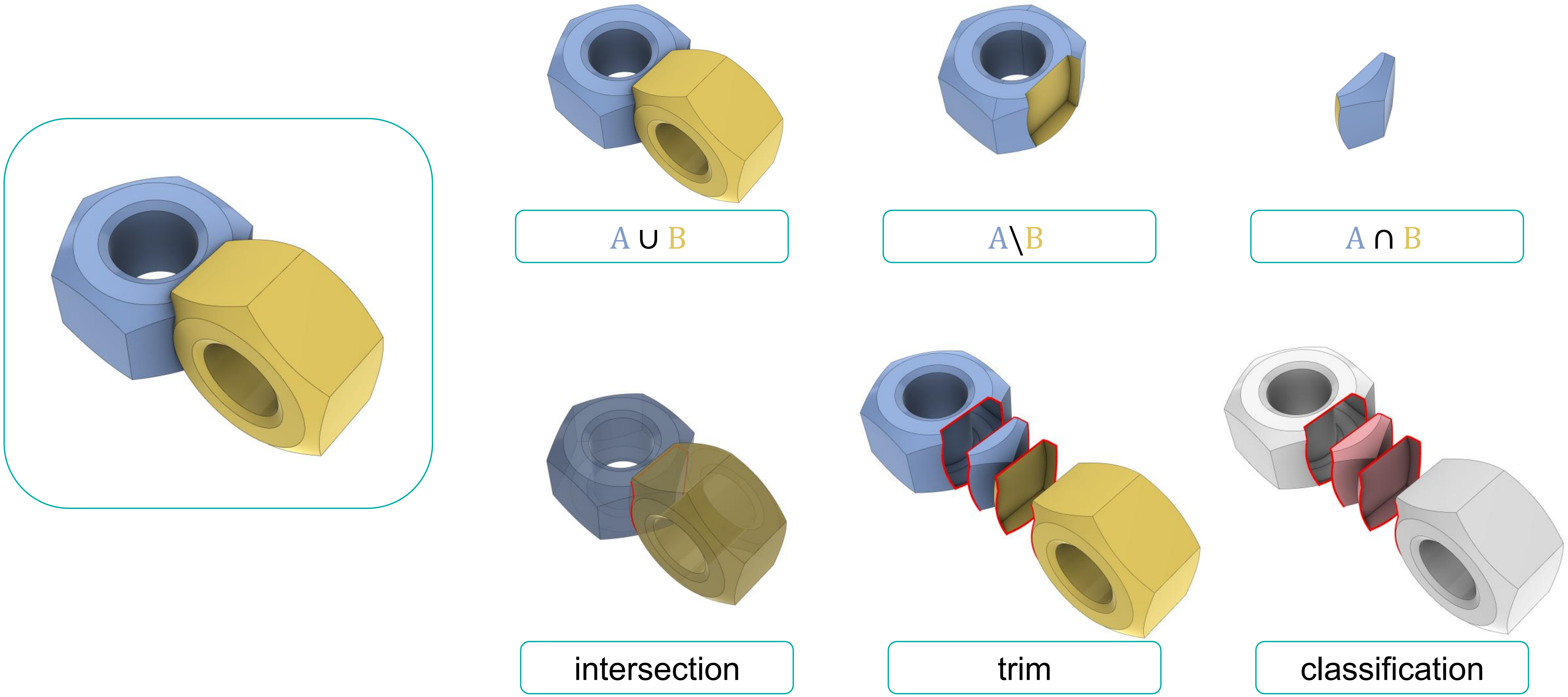
BOOLEAN OPERATION FOR CAD MODELS USING A HYBRID REPRESENTATION

YINGYU YANG, XIAOHONG JIA, BOLUN WANG, JIEYIN YANG, SHIQING XIN, DONG-MING YAN

B-REP BOOLEANS

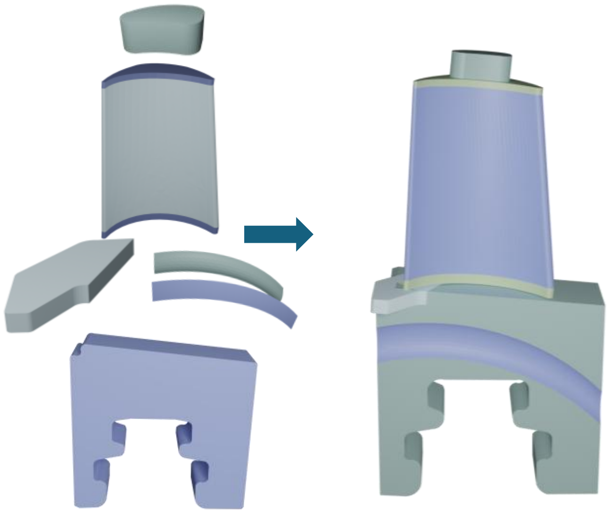


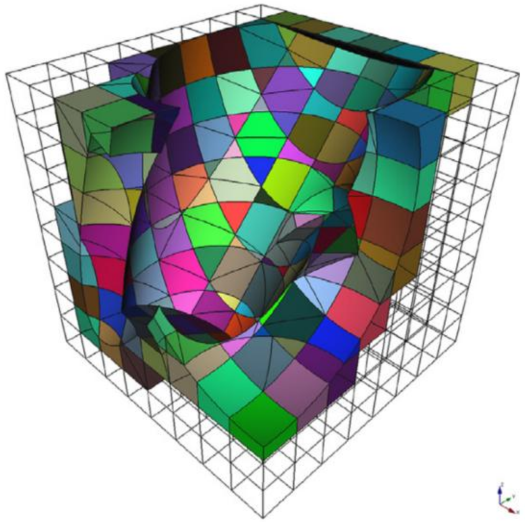
B-REP BOOLEANS

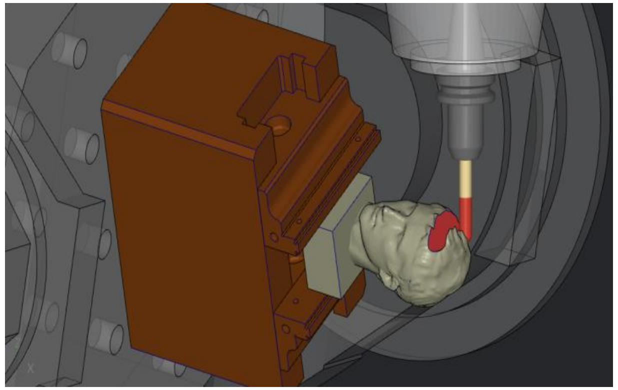


MOTIVATION

SIGGRAPH 2025
Vancouver+ 10-14 August







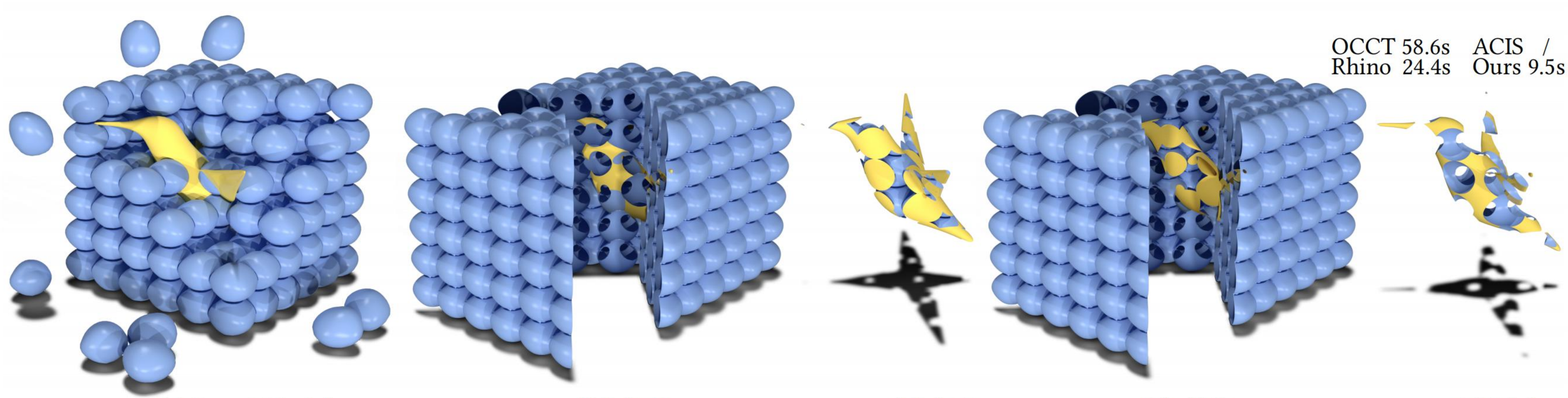
CAD

CAE

CAM

B-Rep Booleans

MOTIVATION



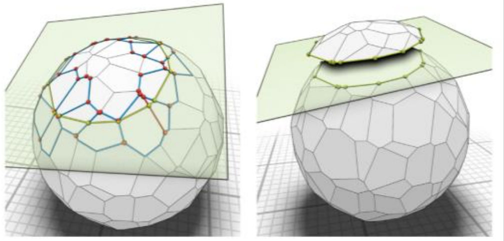
(a) Input Models (b) A∪B (c) A∩B (d) A\B (e) B\A

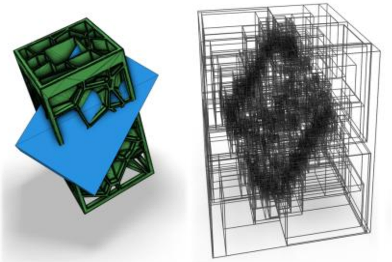
RELATED WORK

SIGGRAPH 2025
Vancouver+ 10-14 August

Mesh Booleans

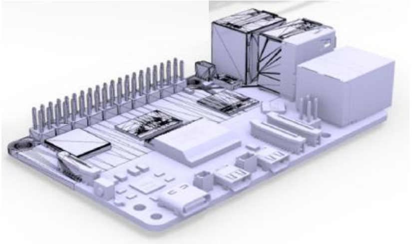
- plane-based





[Nehring-Wirxel et al. 2021] [Trettner et al. 2022]

- vertex-based

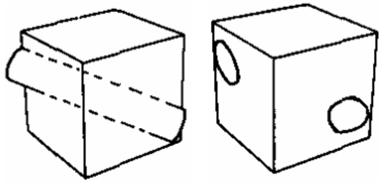


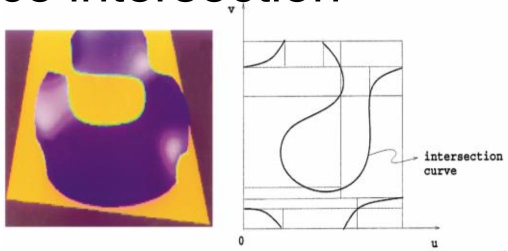
[Cherchi et al. 2022]   [Guo and Fu 2024]

Fast (interactive level), robust, mesh-based input

B-Rep Booleans

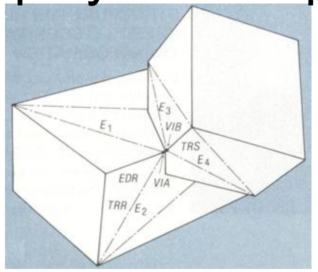
- parametric surface intersection

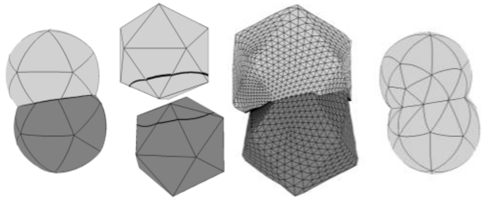




[Braid 1975]   [Krishnan et al. 2001]

- polyhedral approximation

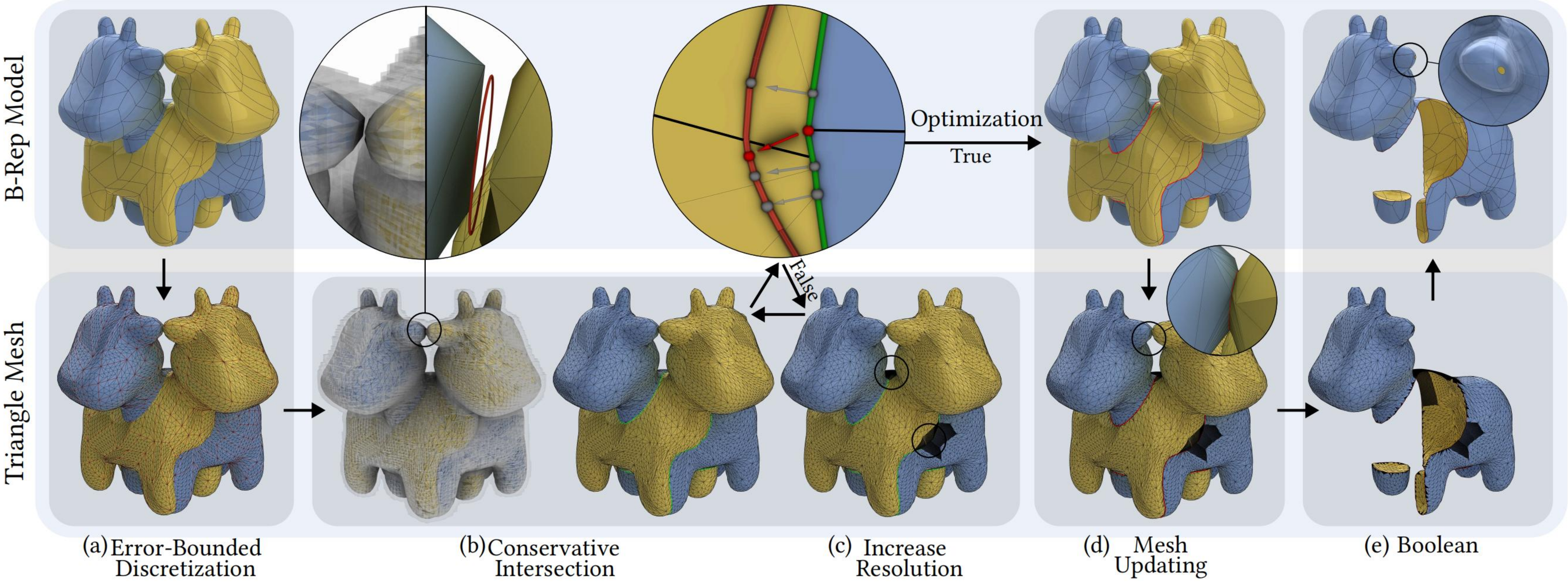




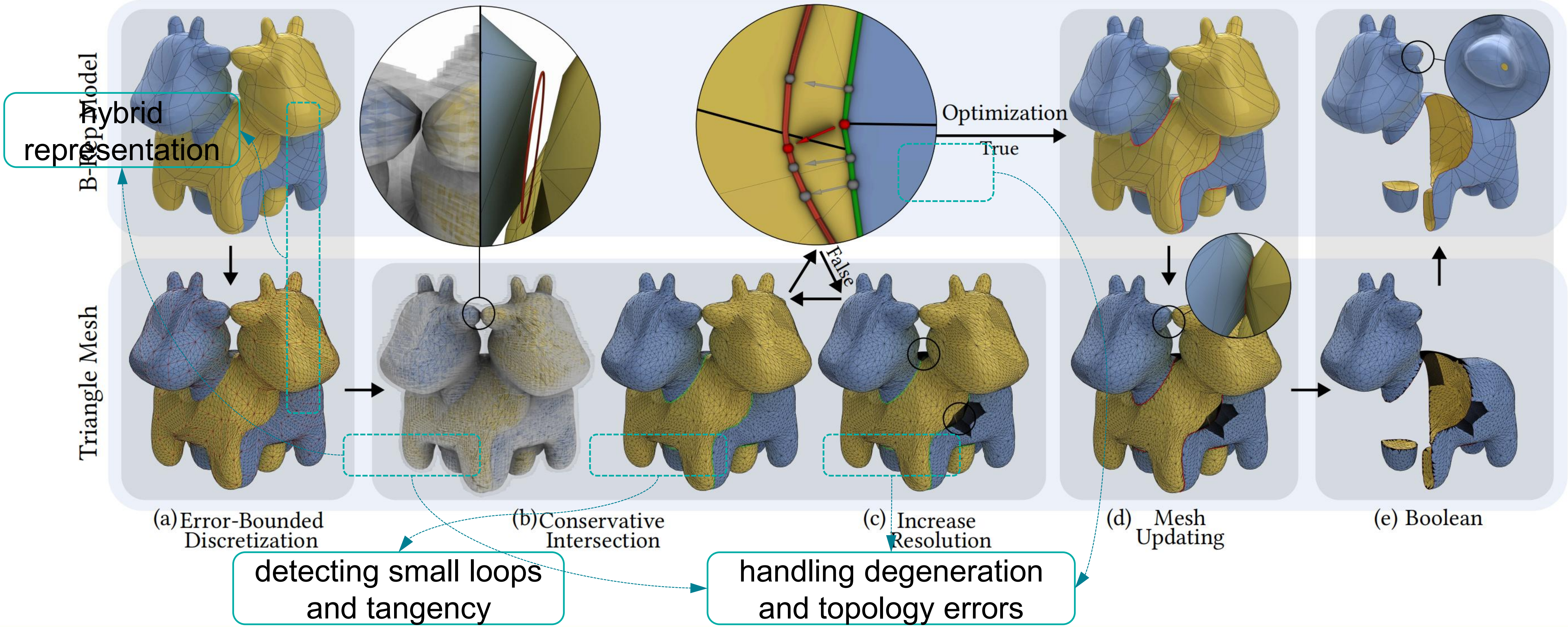
[Yamaguchi et al. 1984] [Biermann et al. 2001]

Limited speed, weak robustness, lack of general algorithms

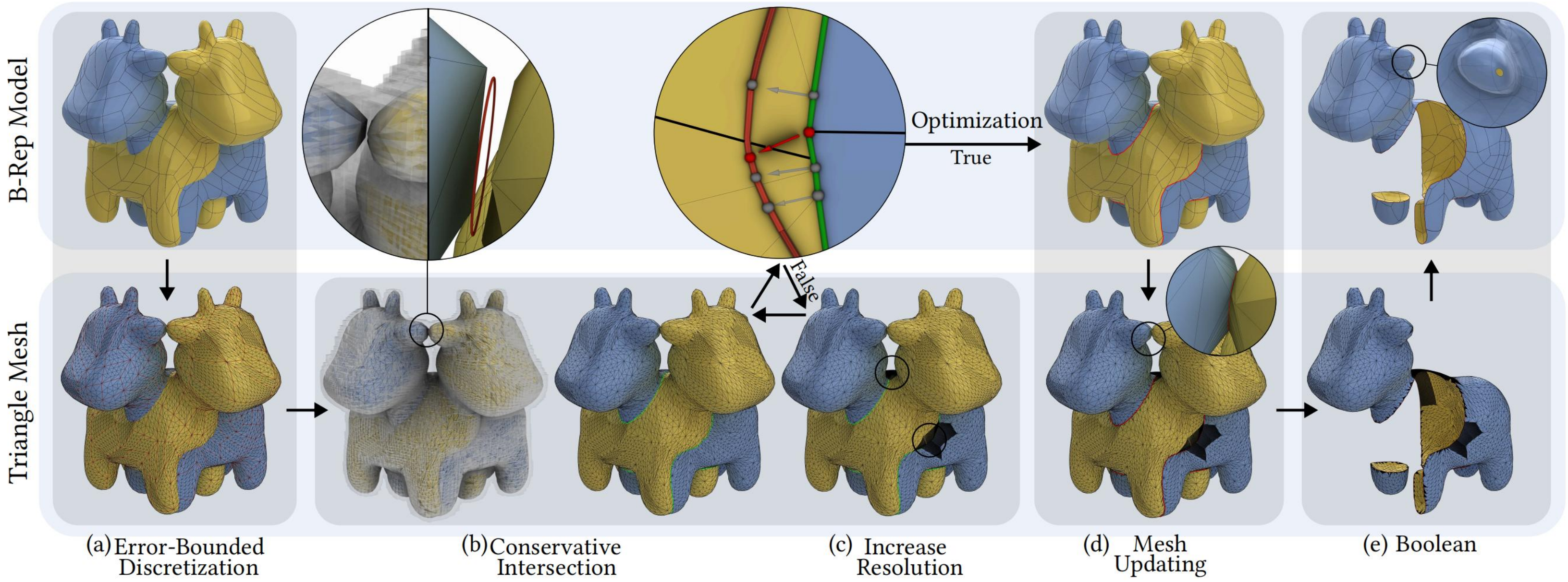
OUR PIPELINE



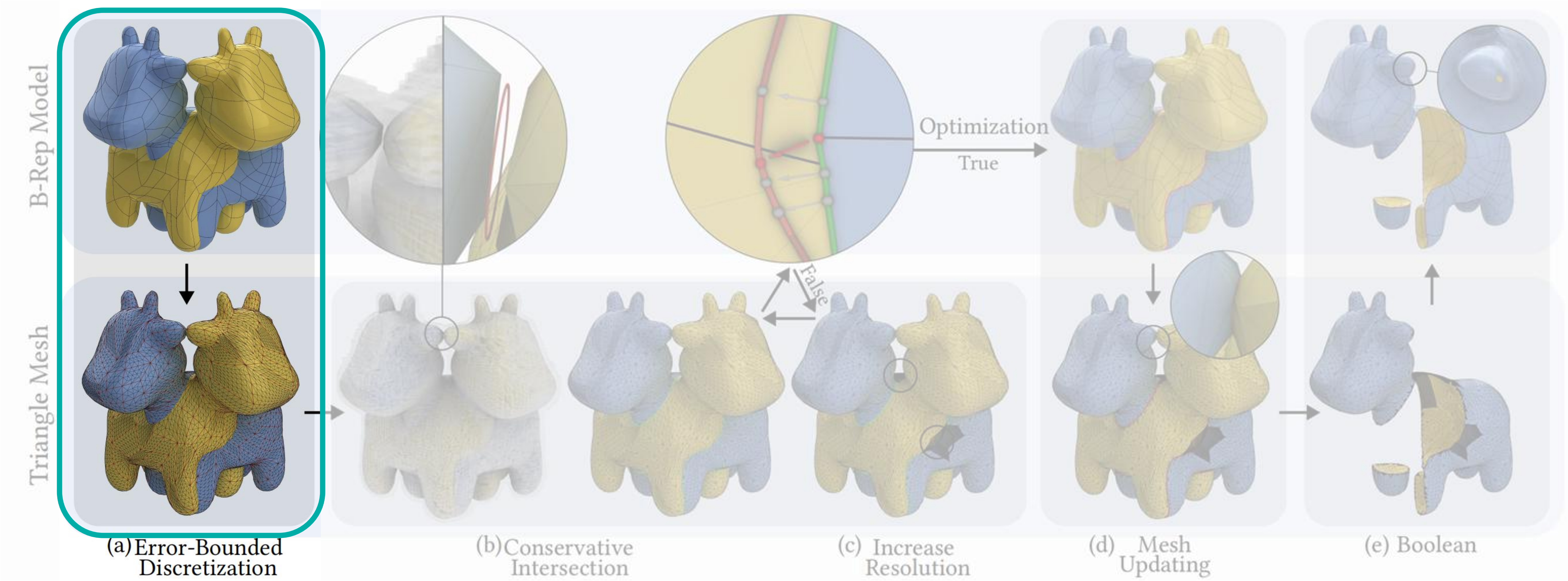
MAIN CONTRIBUTIONS



OUR PIPELINE

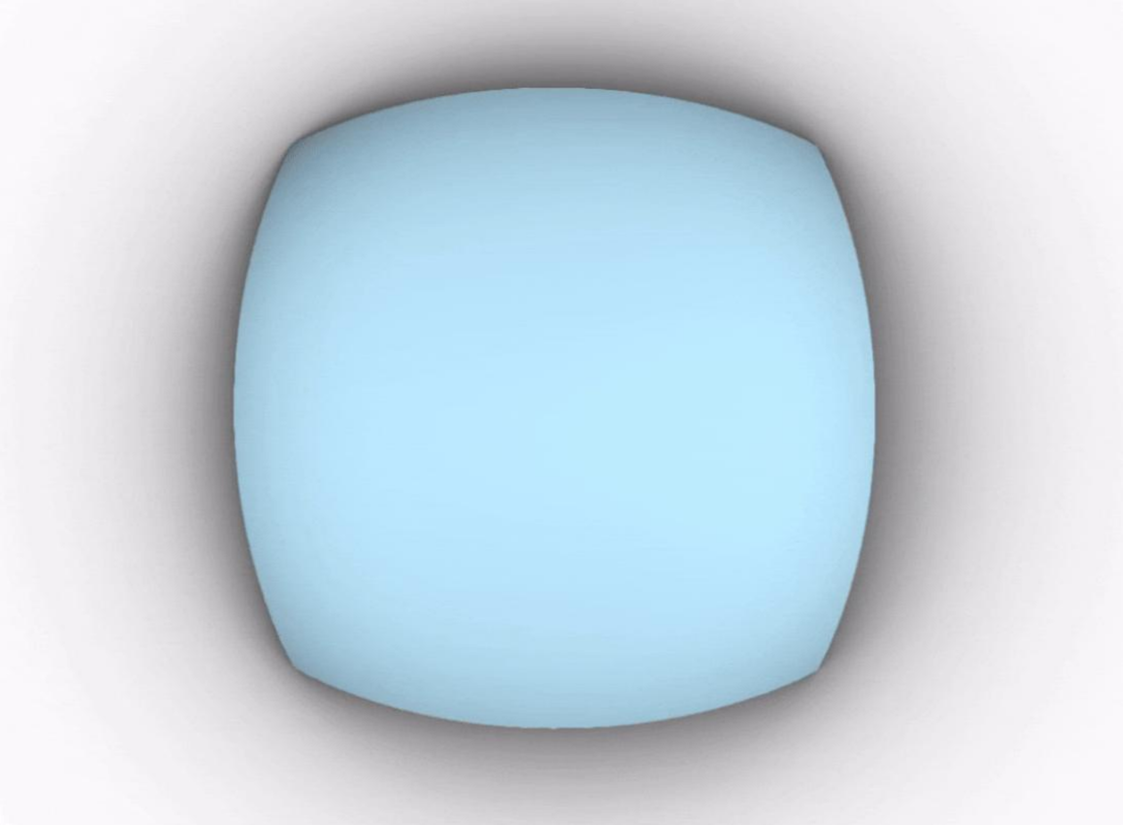


OUR PIPELINE



ERROR-BOUNDED DISCRETIZATION

NURBS →Bézier patches → Subdivide Bézier to distance tolerance

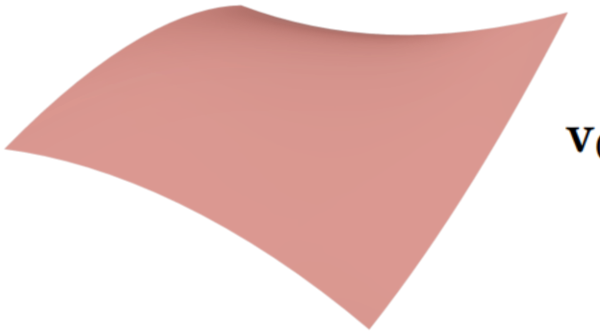


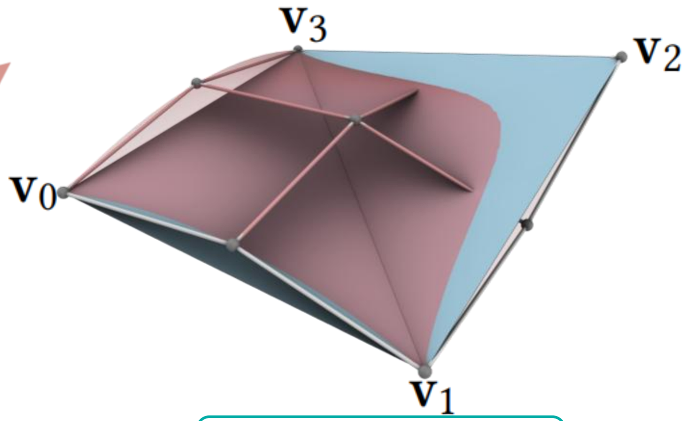
knot insertion

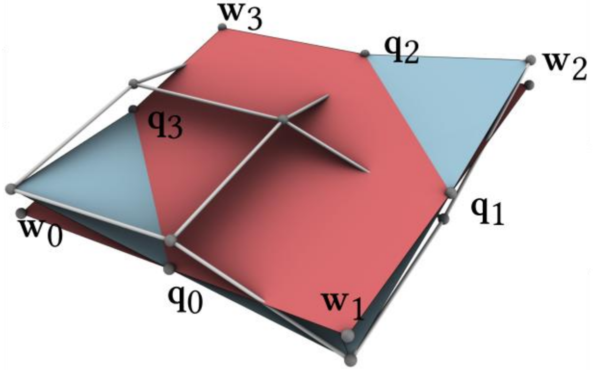
ERROR-BOUNDED DISCRETIZATION

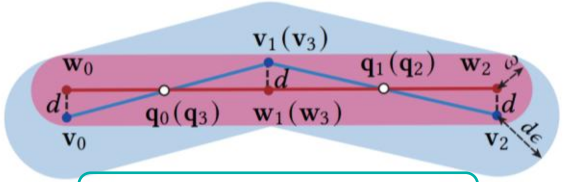
SIGGRAPH 2025
Vancouver+ 10-14 August

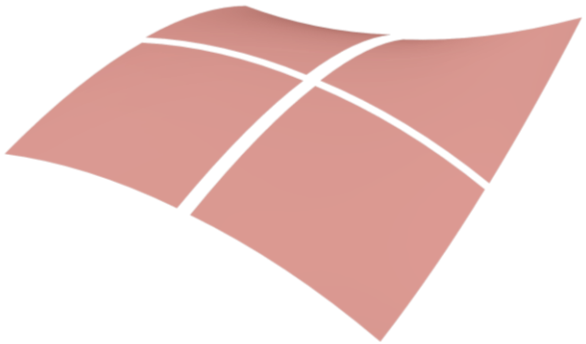
NURBS →Bézier patches → **Subdivide Bézier to distance tolerance**











Bézier patch → discretization → distance error → (large error) subdivision → Bézier patch

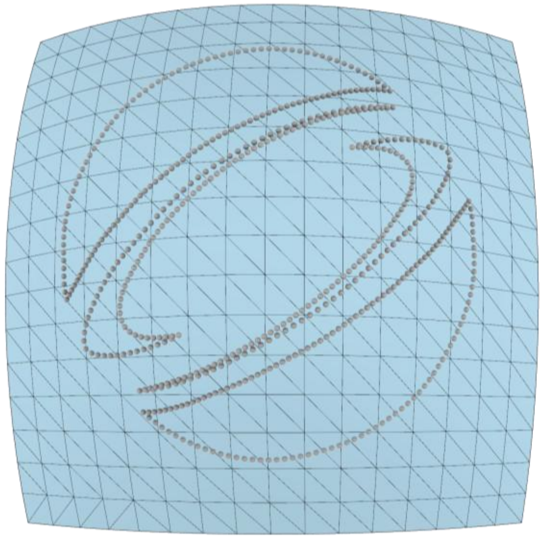
all control points are within distance $d_\varepsilon - d$ from $\square\mathbf{w}_0\mathbf{w}_1\mathbf{w}_2\mathbf{w}_3$

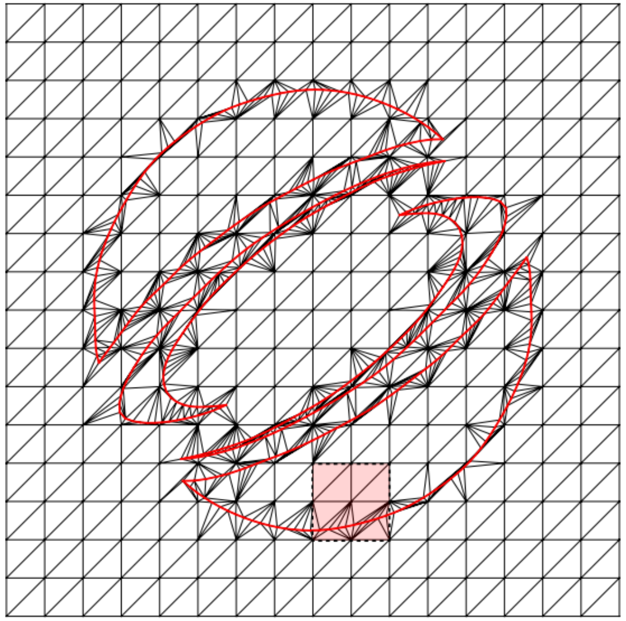
↓

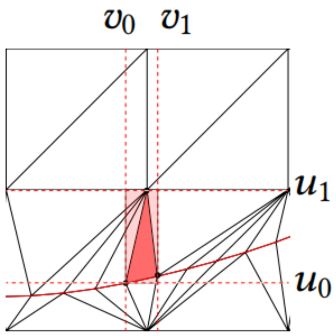
distance from Bézier patch to triangle mesh $\triangle\mathbf{v}_0\mathbf{v}_1\mathbf{v}_3$ $\underline{\triangle\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3}$ is less than d_ε

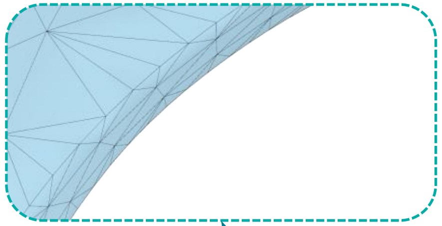
ERROR-BOUNDED DISCRETIZATION

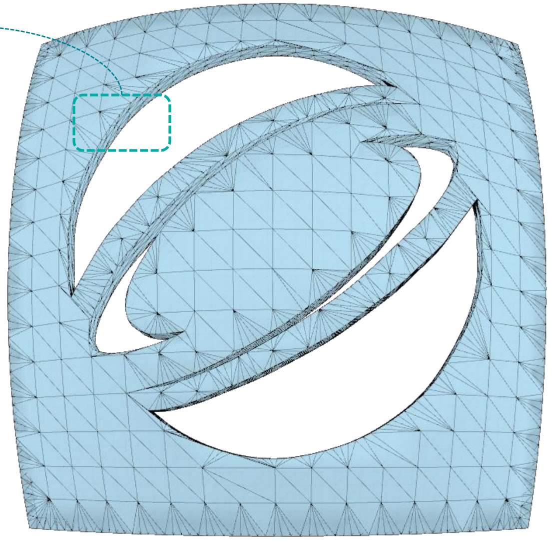
Handle trimming boundaries





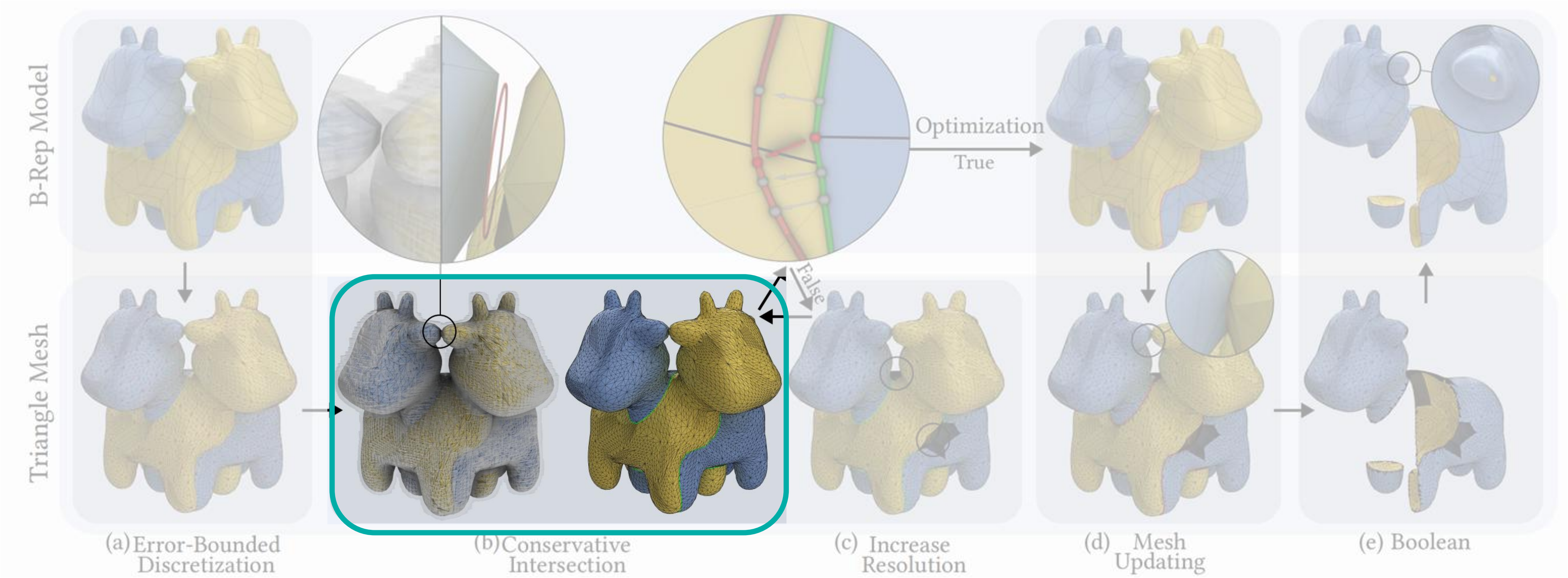






boundary curve sampling → triangulation and boundary error calculation → remove exterior region

OUR PIPELINE



CONSERVATIVE INTERSECTION

SIGGRAPH 2025
Vancouver+ 10-14 August

Is it reliable to replace surface intersections with mesh intersections?

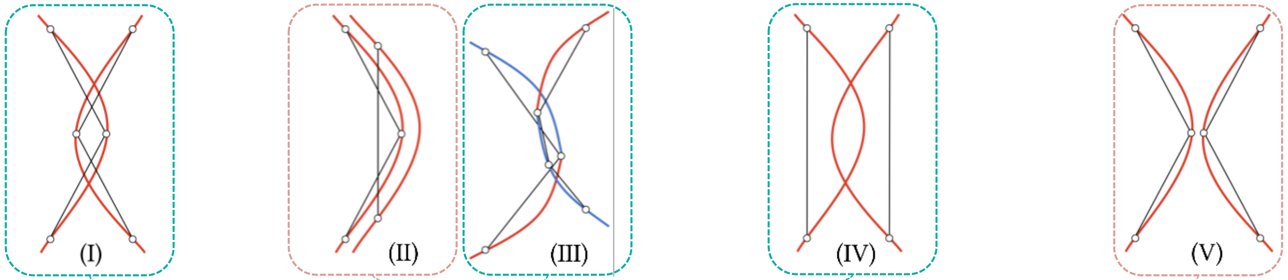
mesh intersection

intersecting | non-intersecting

conservative intersection detection

optimization

success | fail | success | fail

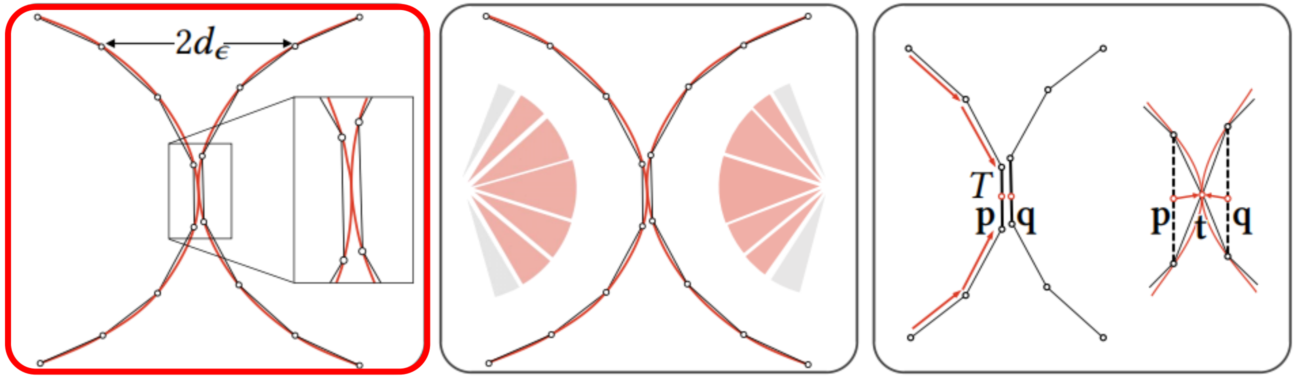


surfaces intersect

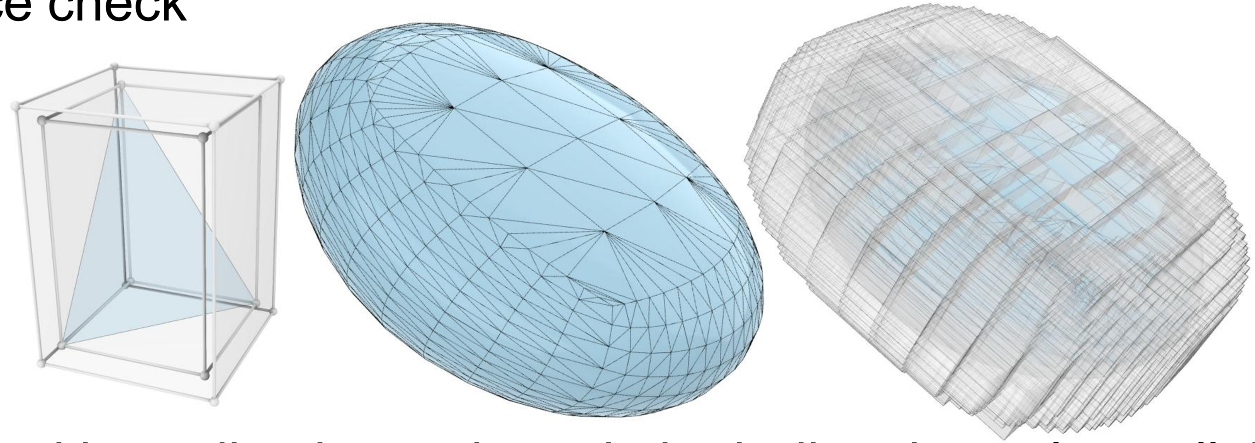
surfaces do not intersect

CONSERVATIVE INTERSECTION

conservative intersection detection



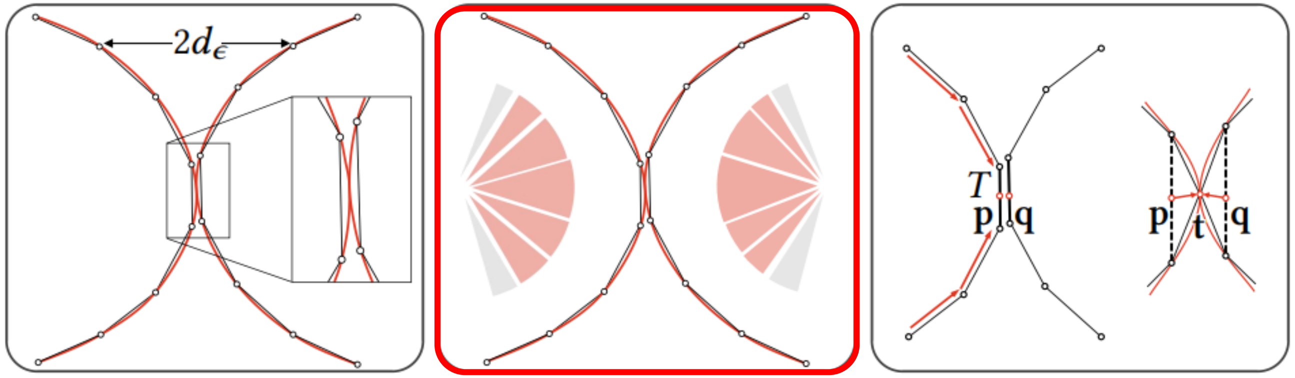
- Condition 1: Distance check



Expand bounding boxes by d_ε in both directions along all three axes
The intersection region after expansion may contain B-Rep intersection curves

CONSERVATIVE INTERSECTION

conservative intersection detection



- Condition 2: Normal check

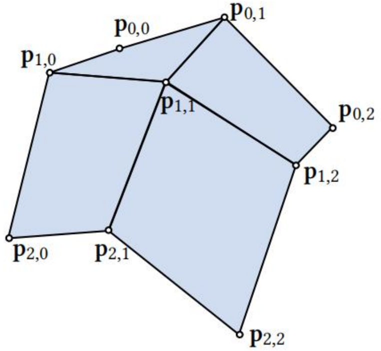
A small loop always contains a pair of points with parallel normals → Restrict to bounding box intersection regions that may contain small loops

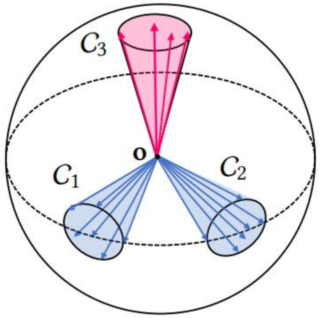
Conservatively estimate the normal range of the subdivided Bézier patch corresponding to each triangle

C_1 is the minimal enclosing cone of $\{p_{i+1,j} - p_{i,j}\}$

C_2 is the minimal enclosing cone of $\{p_{i,j+1} - p_{i,j}\}$

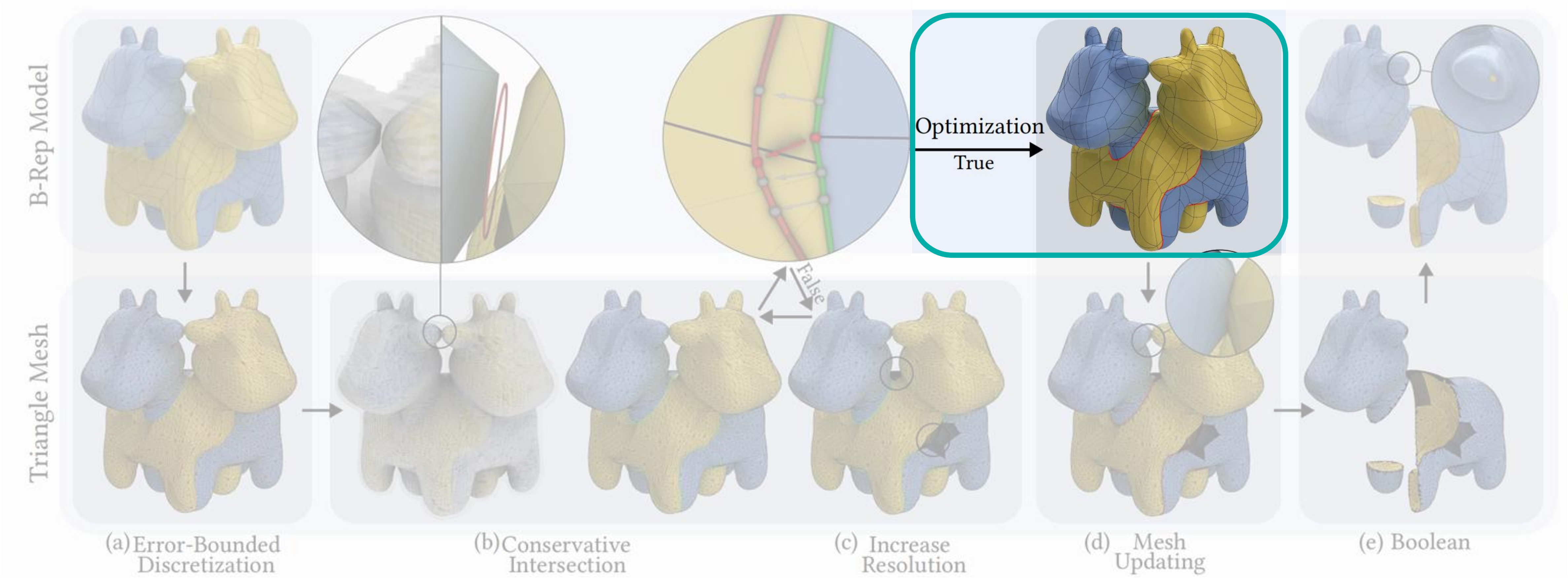
C_3 is the estimated normal range



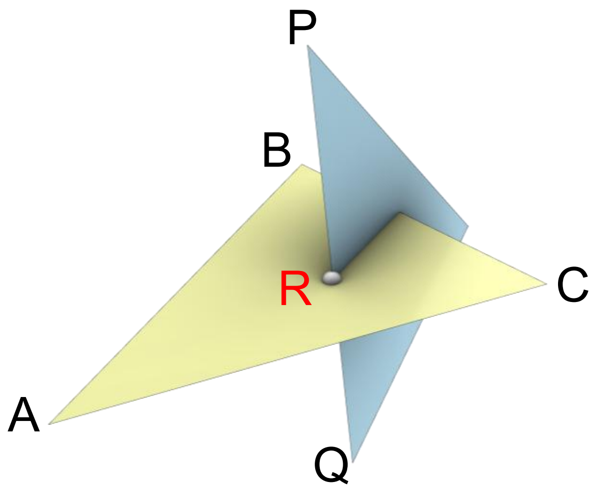


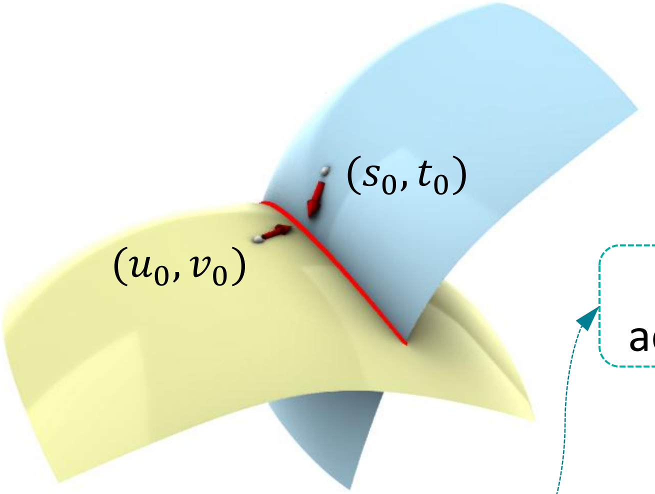
OUR PIPELINE

SIGGRAPH 2025
Vancouver+ 10-14 August



OPTIMIZATION





Initial guess to optimize for accurate parametric coordinates

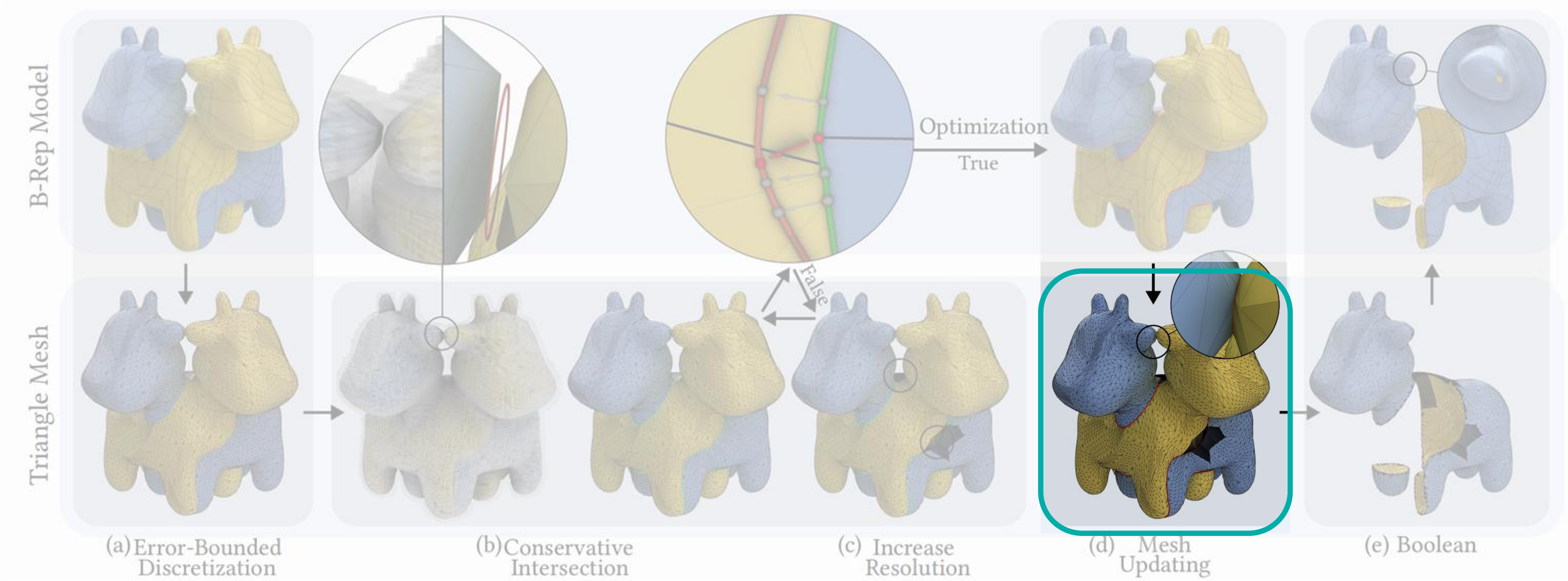
Compute approximate parametric coordinates $(u_0, v_0), (s_0, t_0)$ of intersection point R via linear interpolation

Newton's method

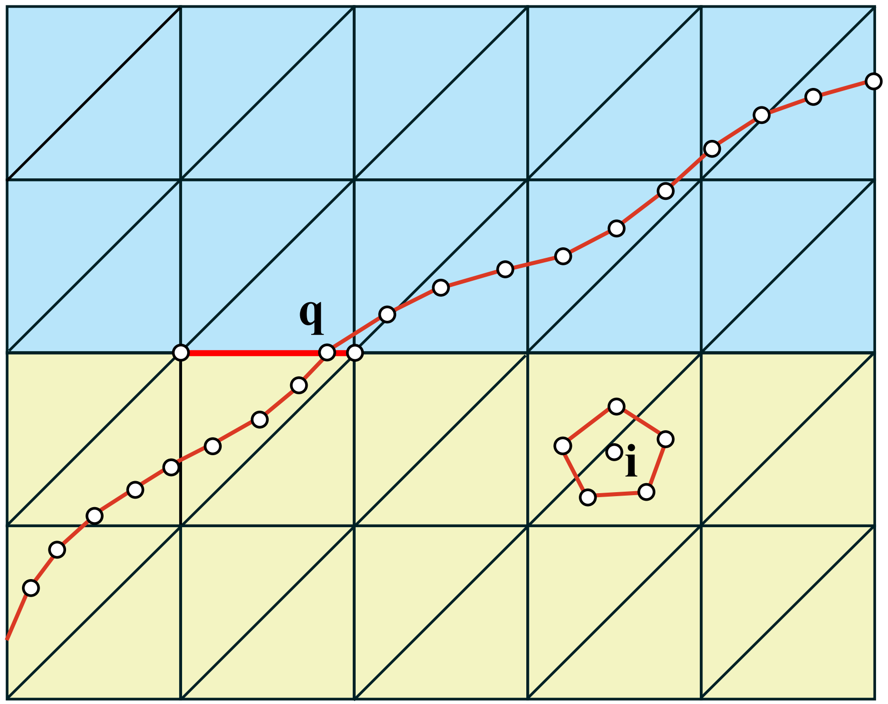
Use Newton's iteration to find the zero of $\mathbf{D}(u, v, s, t) = \mathbf{s}_1(u, v) - \mathbf{s}_2(s, t)$

Iteration equation: $\nabla \mathbf{D} \Delta_k = \mathbf{s}_2(s_k, t_k) - \mathbf{s}_1(u_k, v_k)$

OUR PIPELINE



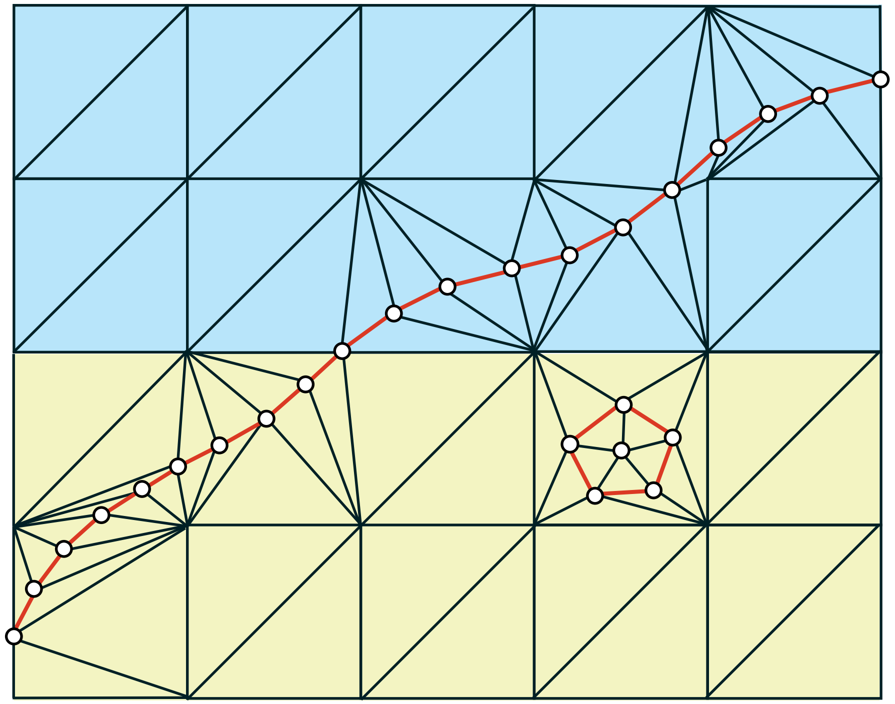
MESH UPDATING



- Insert one point **i** into small loop
- Process boundary intersection point

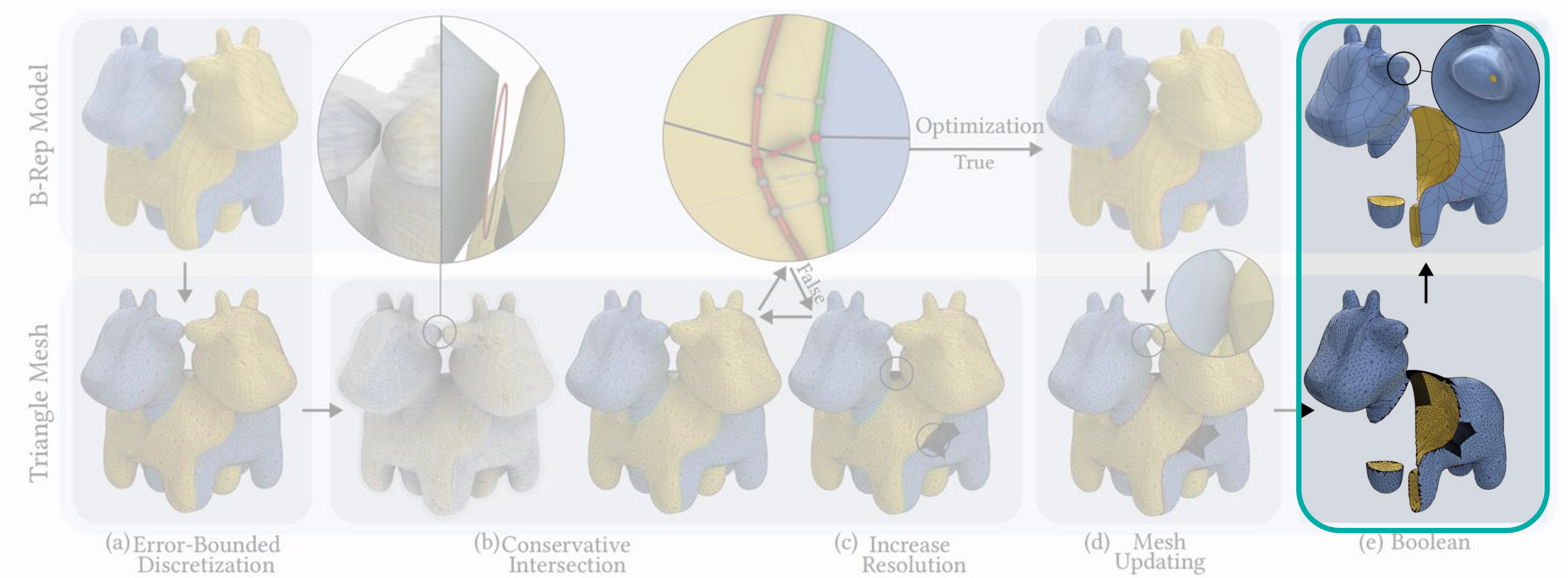
MESH UPDATING

SIGGRAPH 2025
Vancouver+ 10-14 August

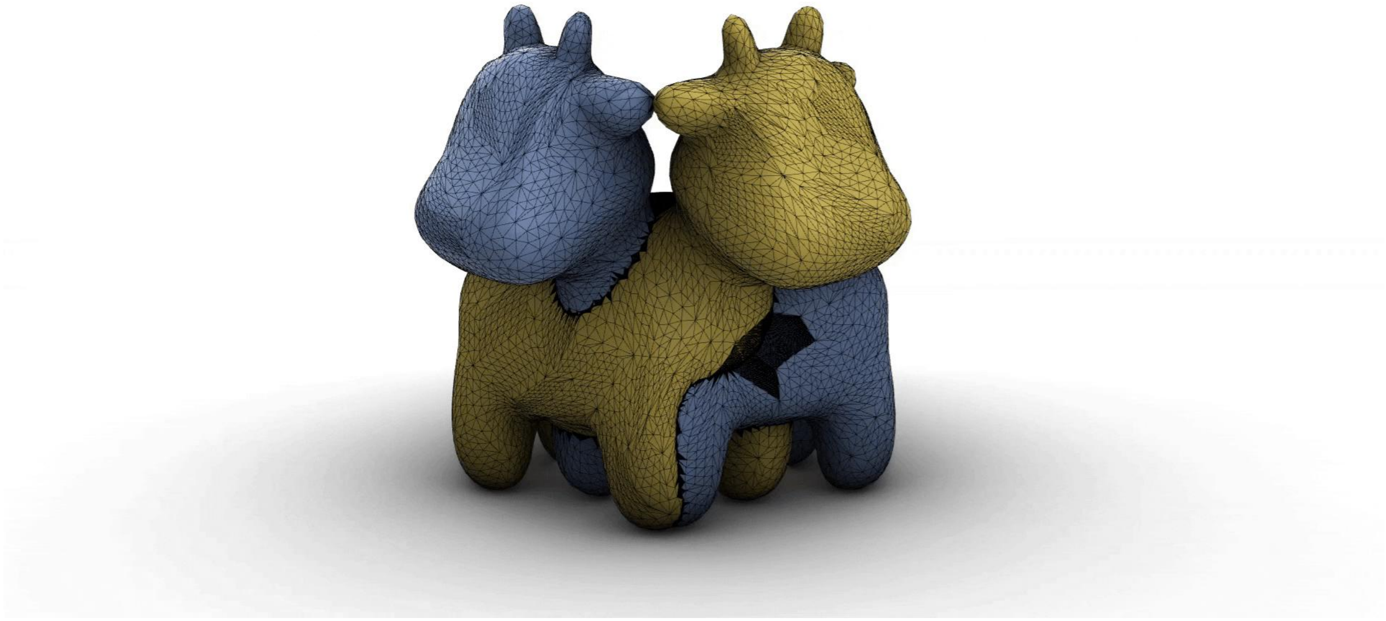


- Insert one point i into small loop
- Process boundary intersection point
- Perform constrained Delaunay triangulation

OUR PIPELINE

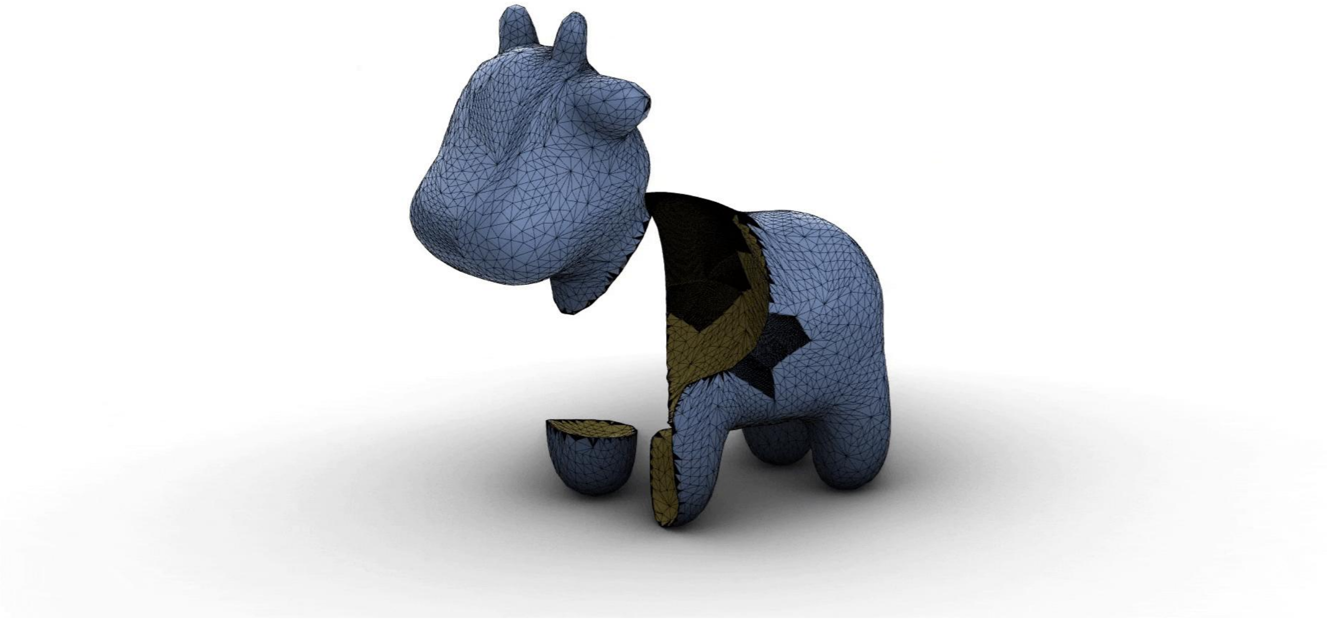


BOOLEAN



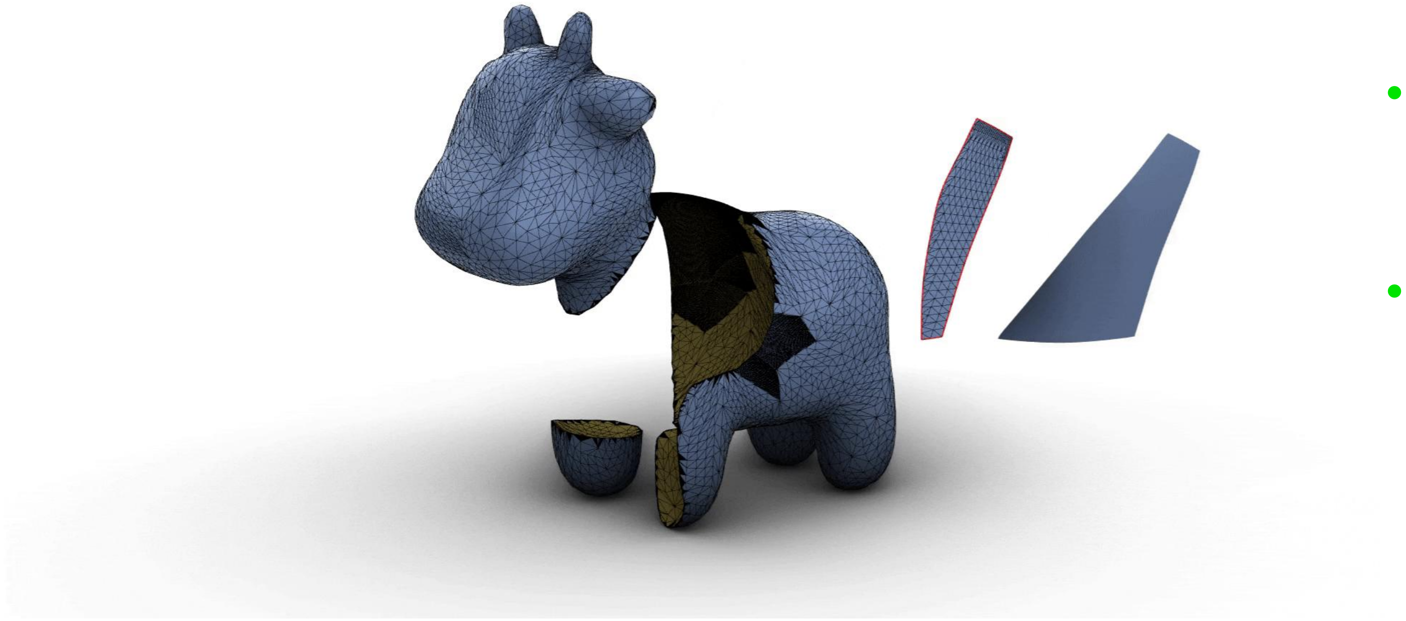
- Perform mesh Boolean

BOOLEAN



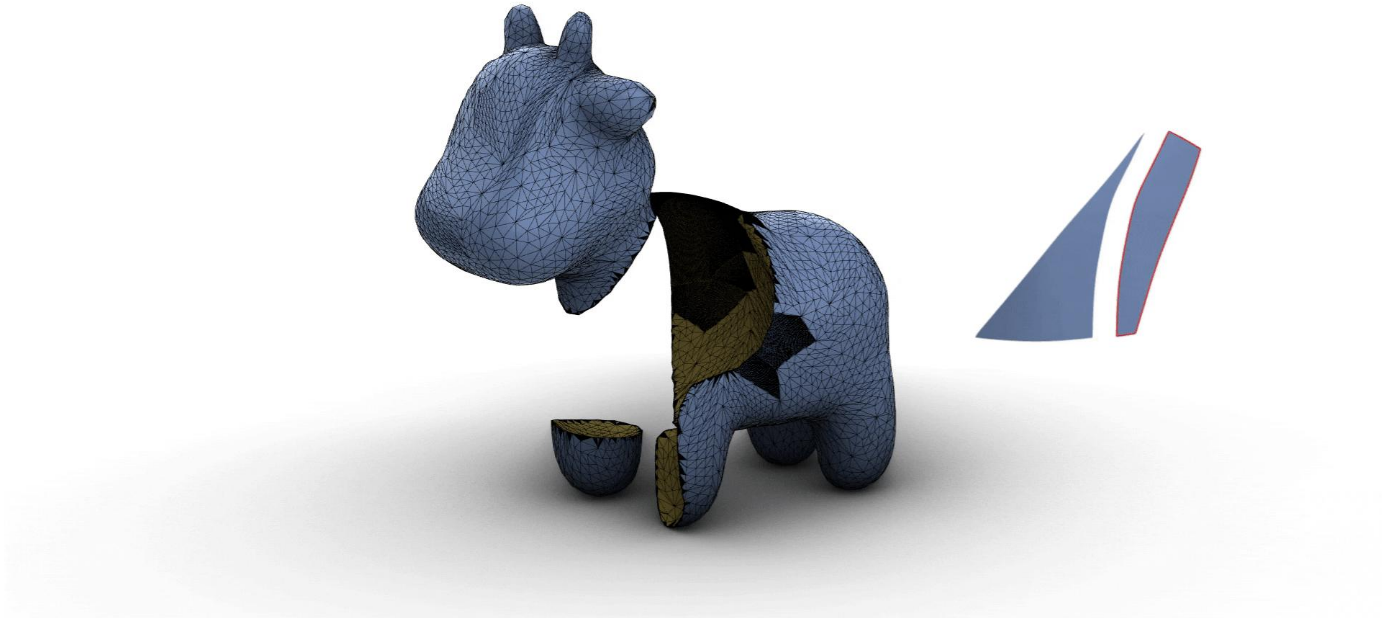
- Perform mesh Boolean
- Segment the mesh Boolean result

BOOLEAN



- Perform mesh Boolean
- Segment the mesh Boolean result
- Collect boundary curves

BOOLEAN



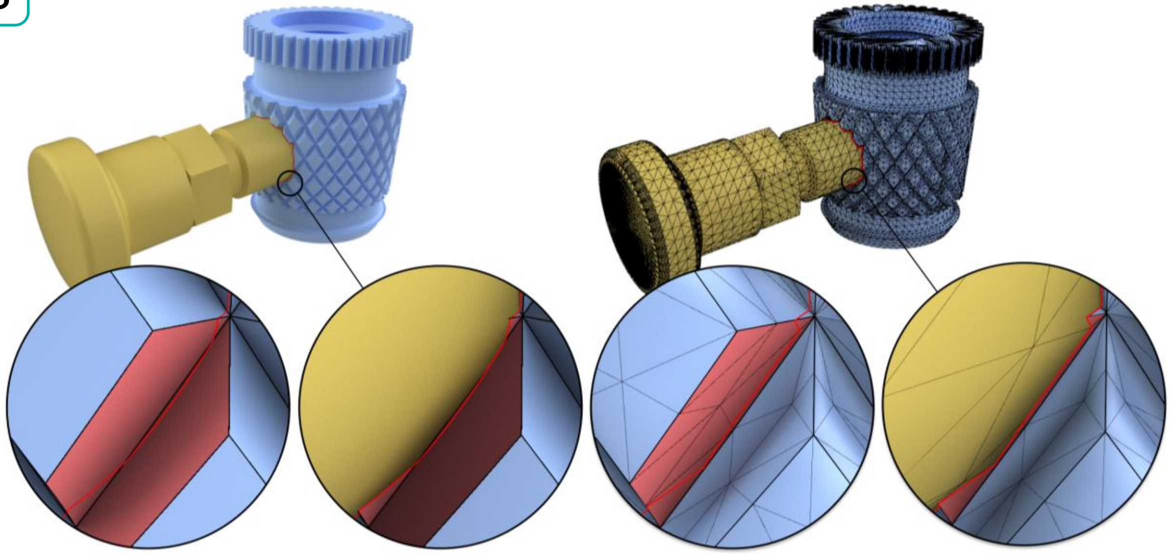
- Perform mesh Boolean
- Segment the mesh Boolean result
- Collect boundary curves
- Map back to parametric surfaces

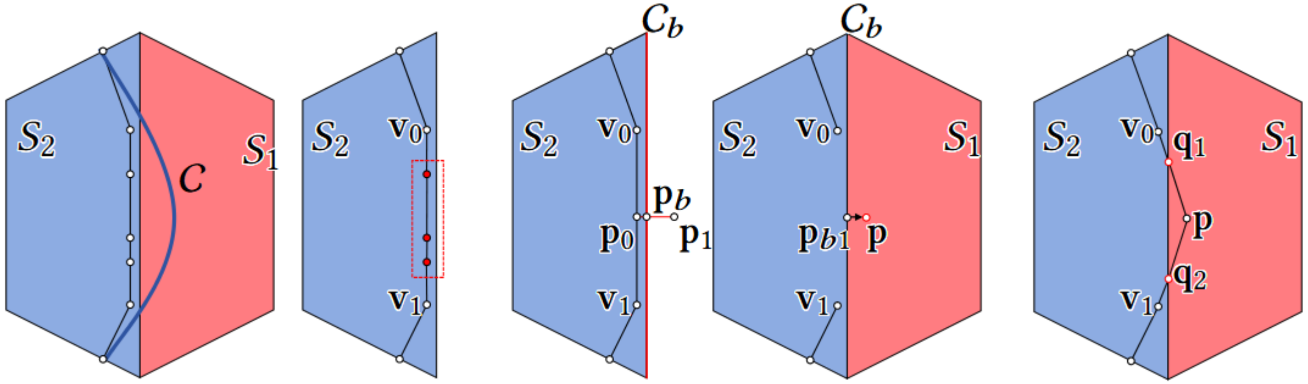
SOLVING FAILURES HANDLING

SOLVING FAILURES HANDLING

SIGGRAPH 2025
Vancouver+ 10-14 August

Optimize across boundaries

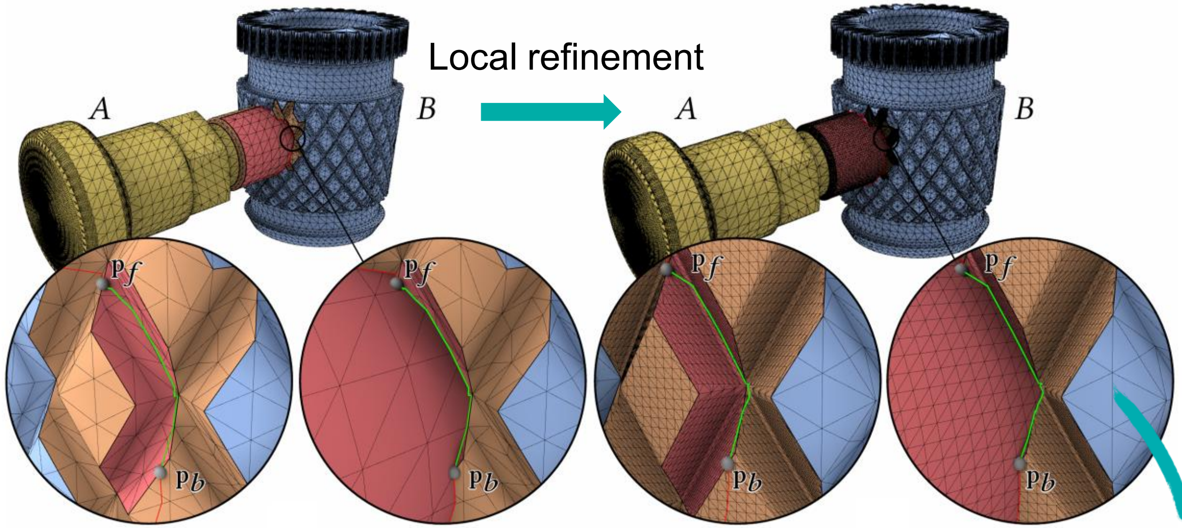


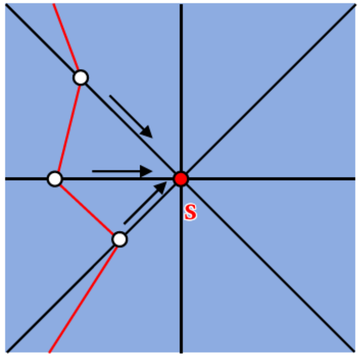


SOLVING FAILURES HANDLING

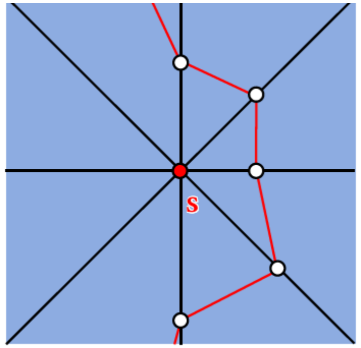
SIGGRAPH 2025
Vancouver+ 10-14 August

Local refinement

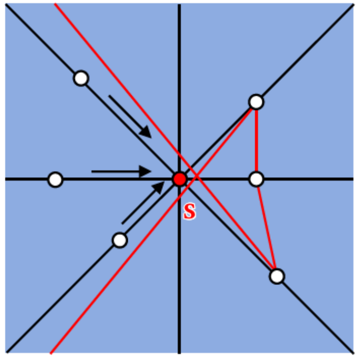




mesh intersection



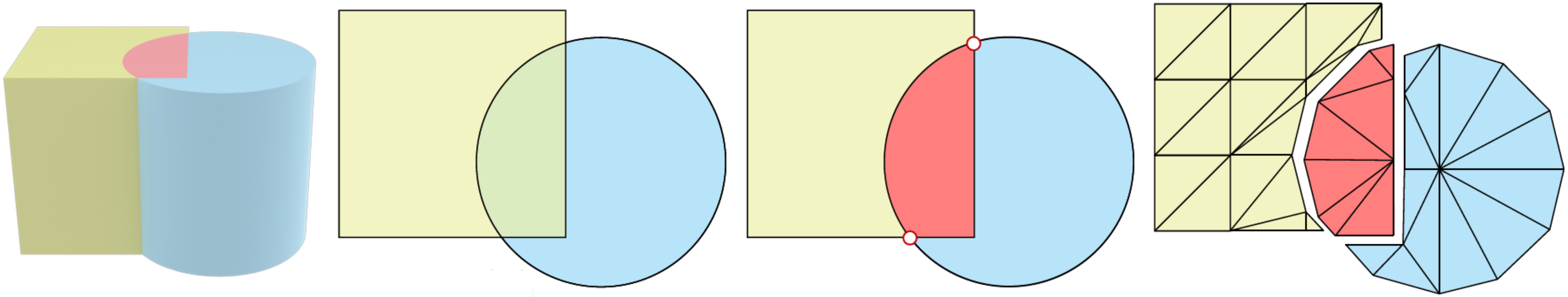
surface intersection

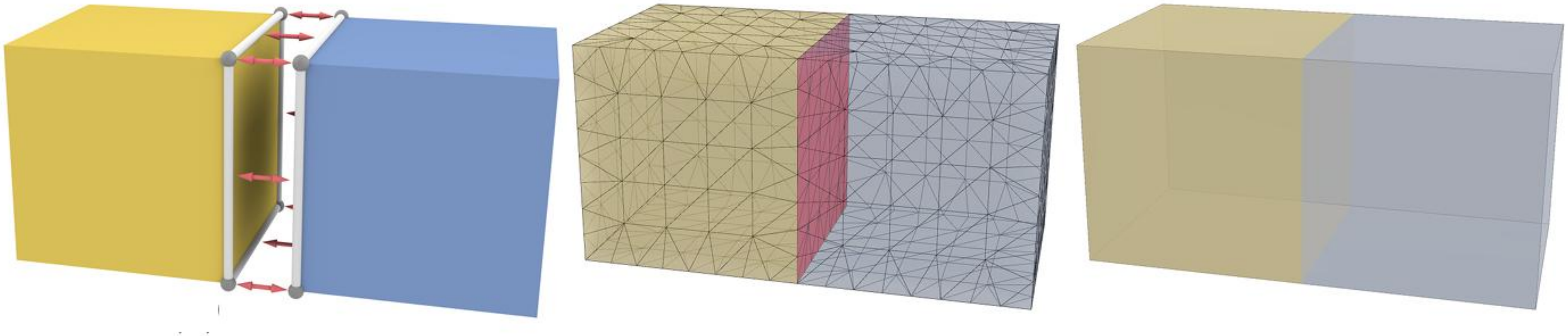


optimize across boundaries

SOLVING FAILURES HANDLING

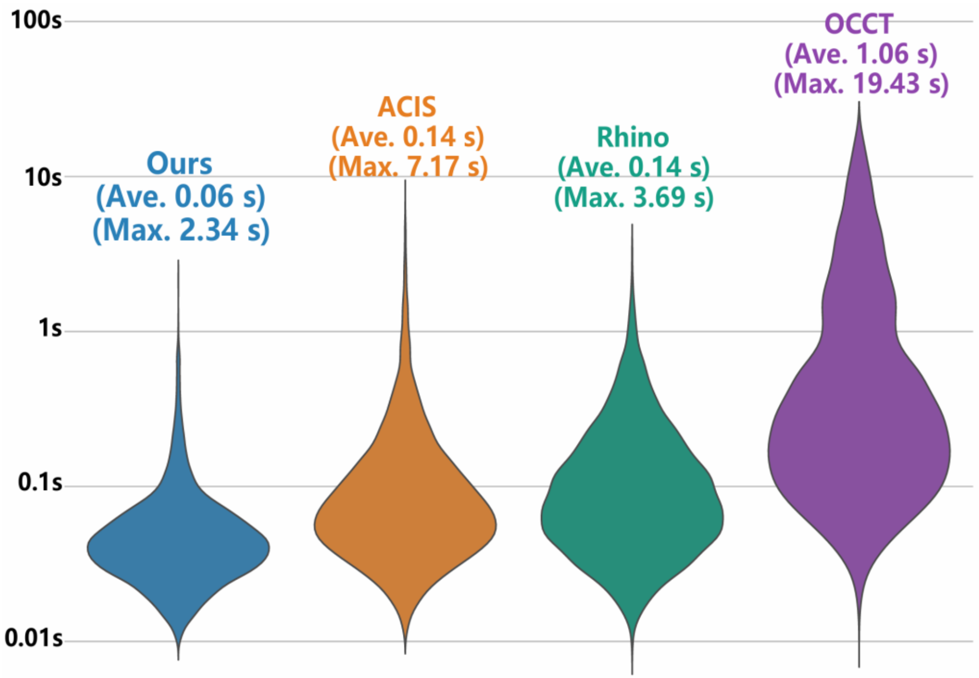
Handling coplanarity



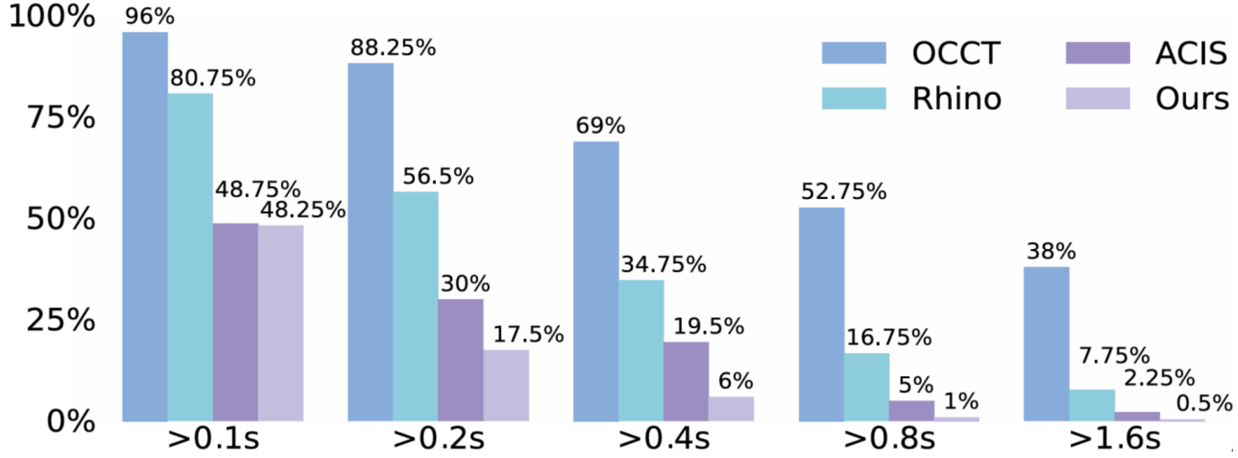


RESULTS AND DISCUSSION

RUMTIME COMPARISON



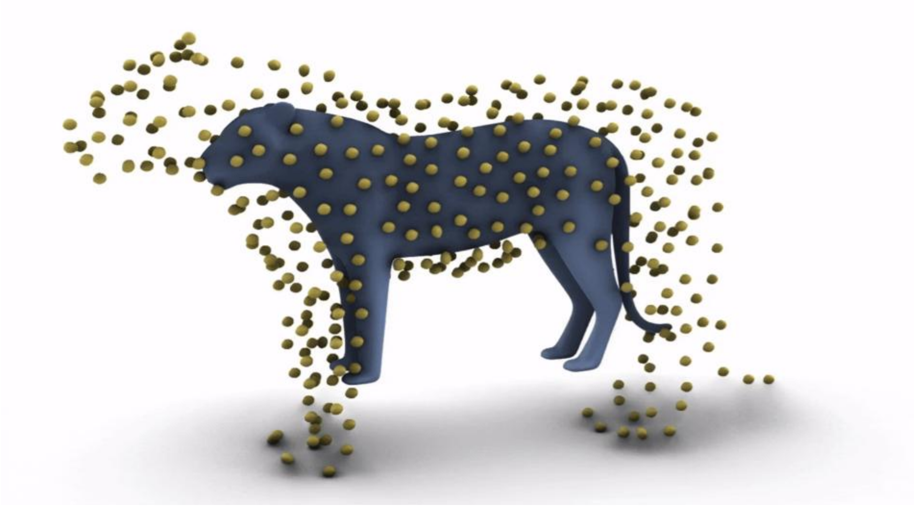
10,000 random cases



100 complex cases

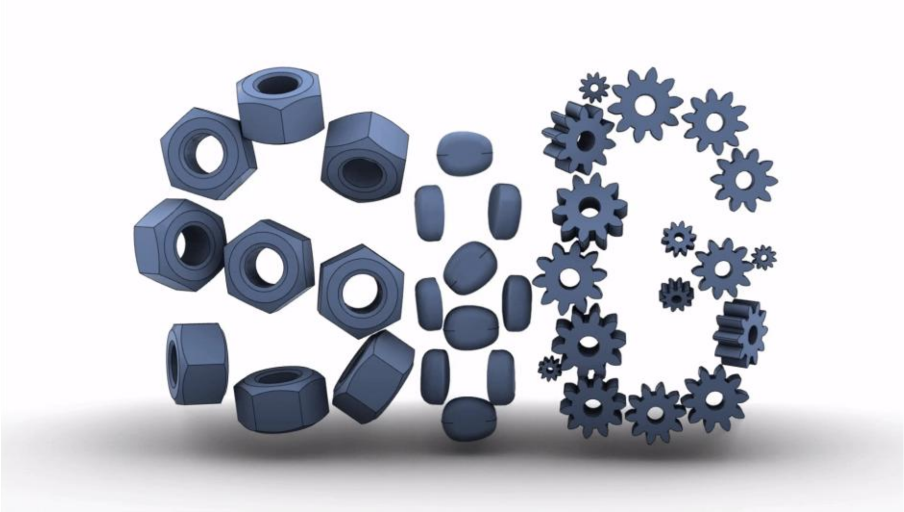
17× OCCT
2.3×Rhino&ACIS

RUMTIME COMPARISON



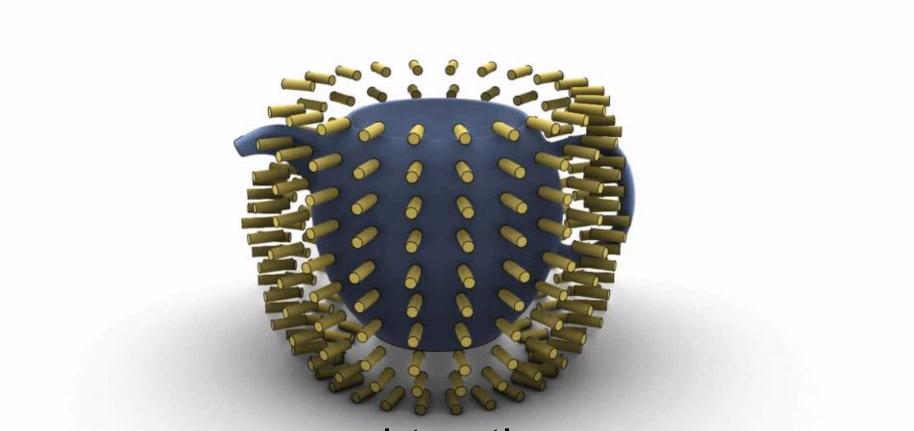
subtraction

Ours	3.5s
ACIS	7.6s
Rhino	13.1s
OCCT	47.8s



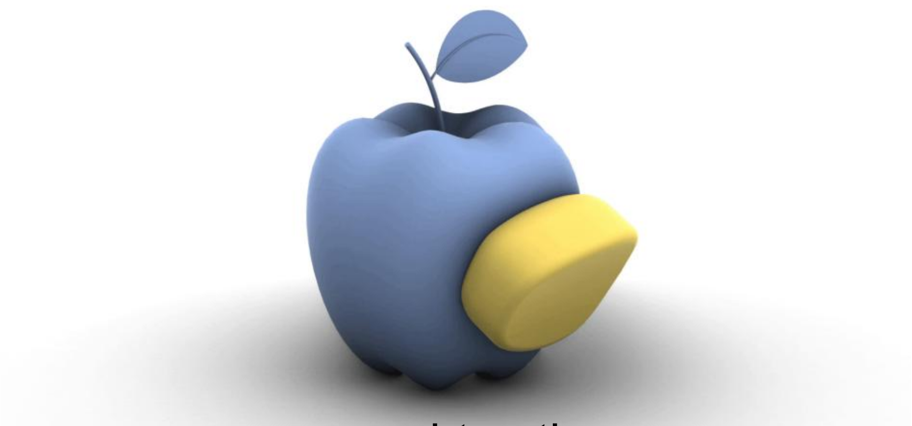
union

Ours	1.4s
ACIS	2.0s
Rhino	2.8s
OCCT	387.5s



subtraction

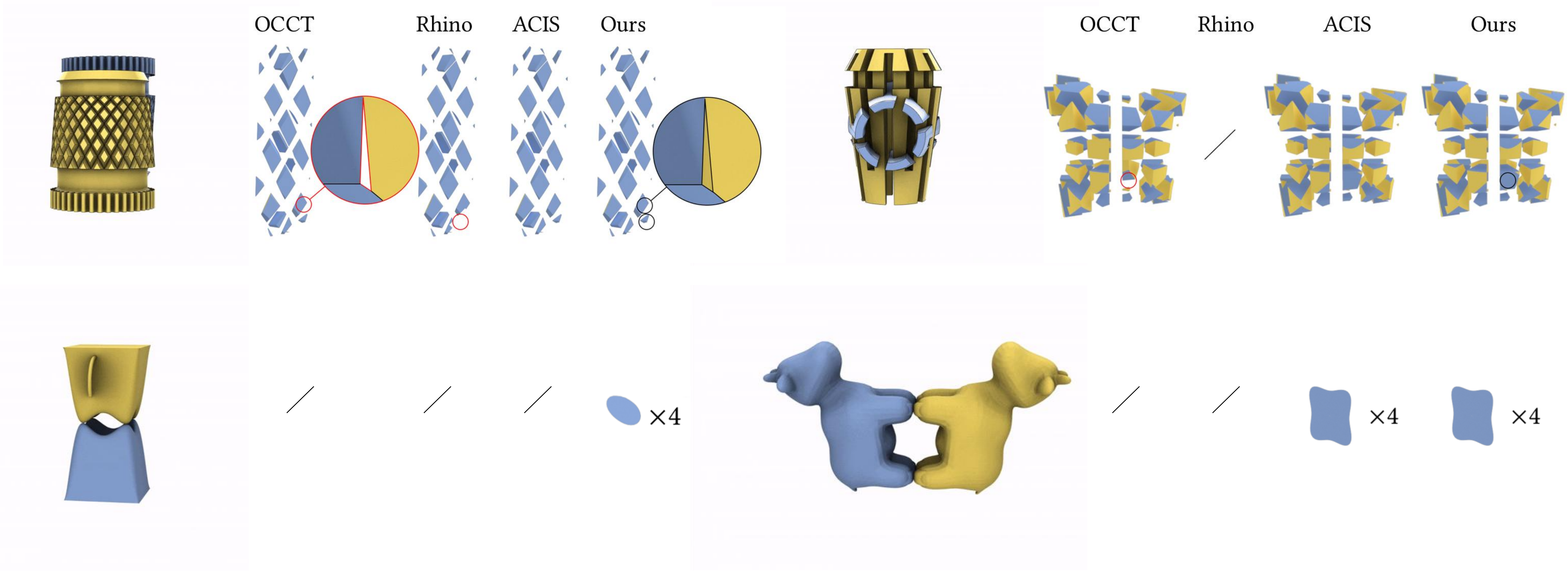
Ours	1.2s
ACIS	3.3s
Rhino	3.9s
OCCT	10.7s



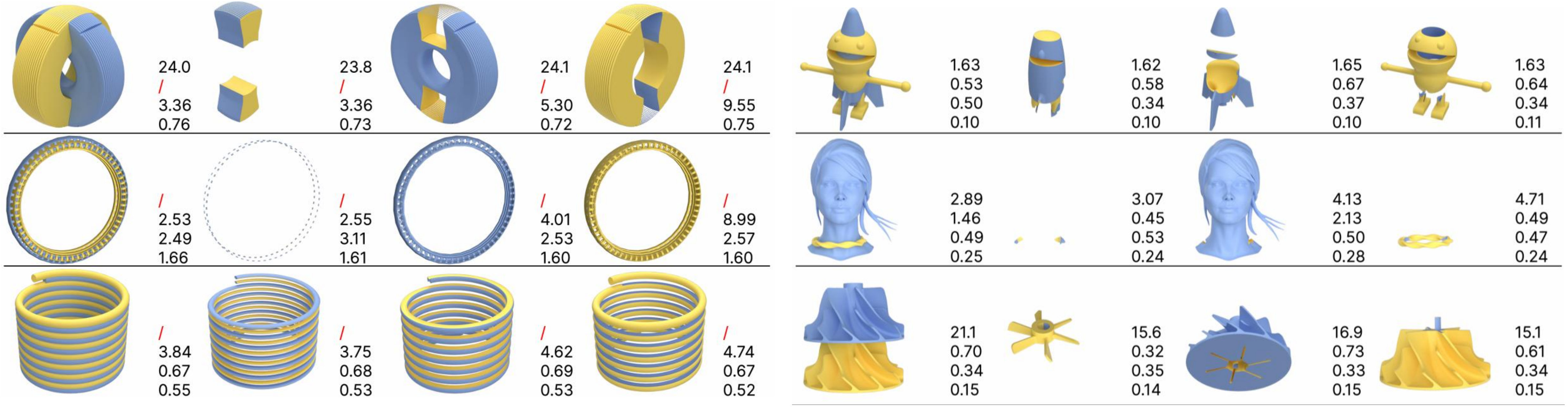
subtraction

Ours	0.4s
ACIS	0.9s
Rhino	2.1s
OCCT	189.5s

CORRECTNESS



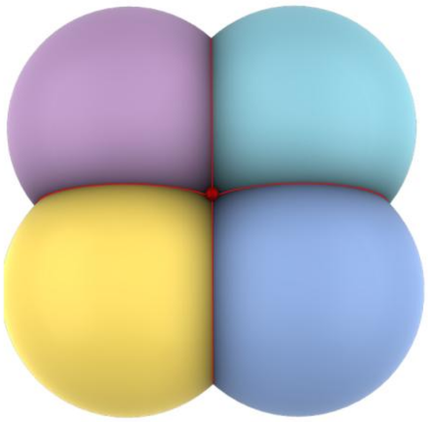
MORE EXAMPLES



The four numbers listed are the runtime in seconds for OCCT, Rhino, ACIS and our method.

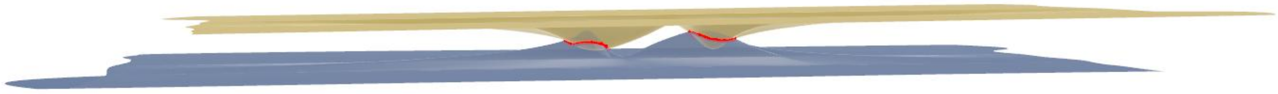
LIMITATIONS

CSG

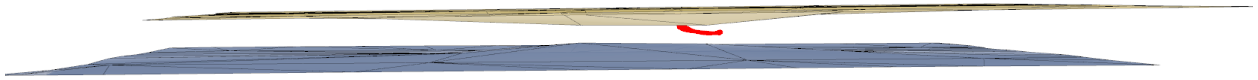


Common intersections and
more singular cases

Multiple small loops



surfaces



meshes

Multiple small loops within a
single triangle region

THANKS FOR YOUR ATTENTION!