Discrete Geometry Processing with Topological Guarantees

Dinesh Manocha

UNC Chapel Hill

http://gamma.cs.unc.edu

Collaborators

- Young Kim (Ewha University)
- Shankar Krishnan (AT & T Labs)
- Ming C. Lin (UNC Chapel Hill)
- Gokul Varadhan (Google)
- Liangjun Zhang (UNC Chapel Hill)

Geometry Processing for CAD

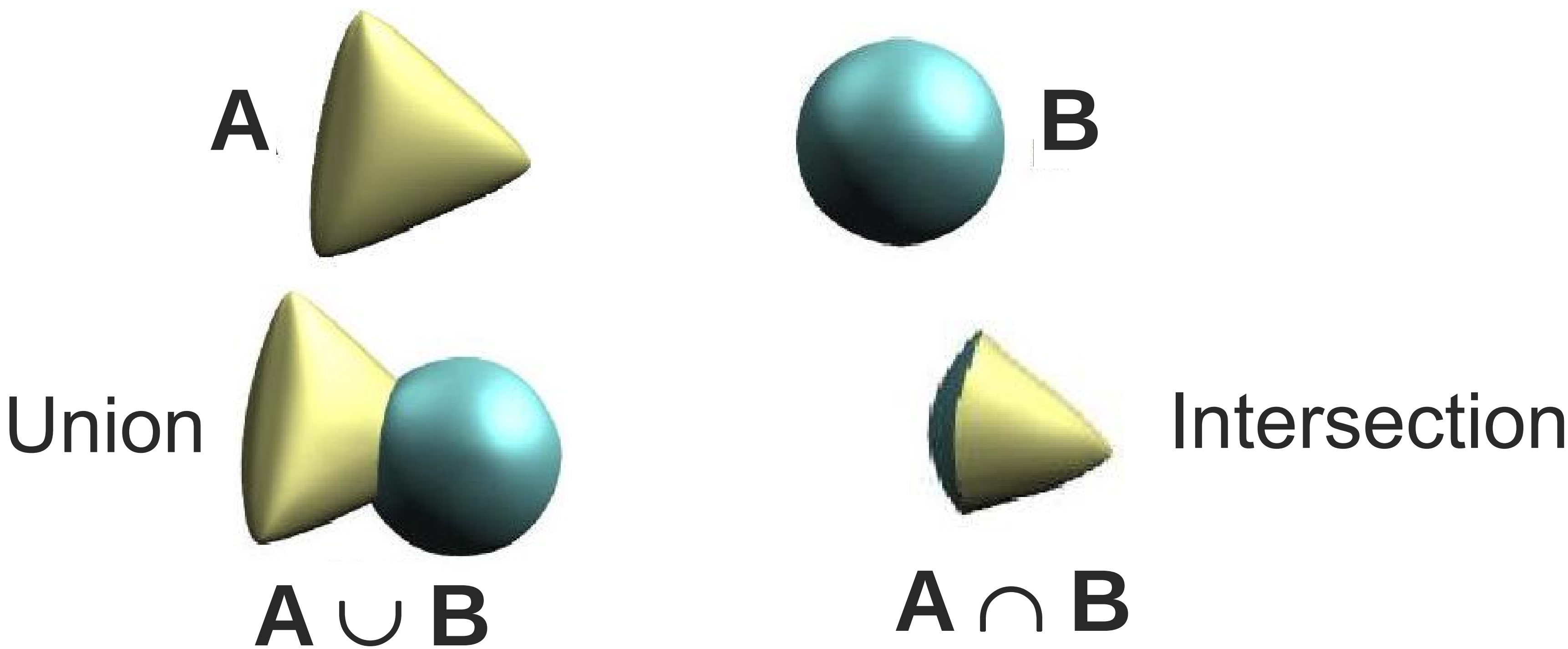
- Geometric problems arising in computer graphics, solid modeling, visualization, robotics,…..
- Given a collection of 3D primitives
- Perform operations on these primitives
  - Boolean operations
  - Minkowski sums
  - Implicit surface meshing
  - Configuration space computation & motion planning

Topology

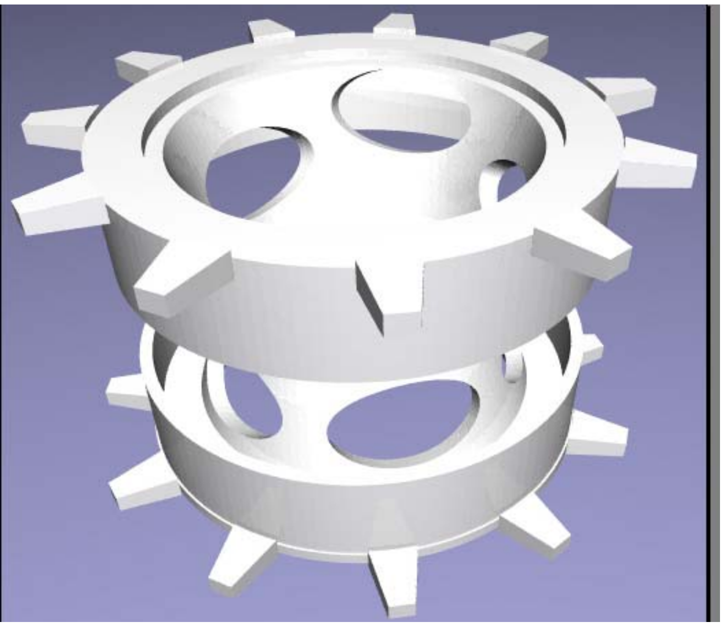
- Qualitative measure of underlying space
- Classification based on global connectivity
  - Connected components, tunnels, voids etc.
- Two spaces have the *same topology* if there exists a homeomorphism between them
  - Square and disc have same topology
  - Sphere and torus do not

Boolean Operations

- Union, intersection, difference, complement



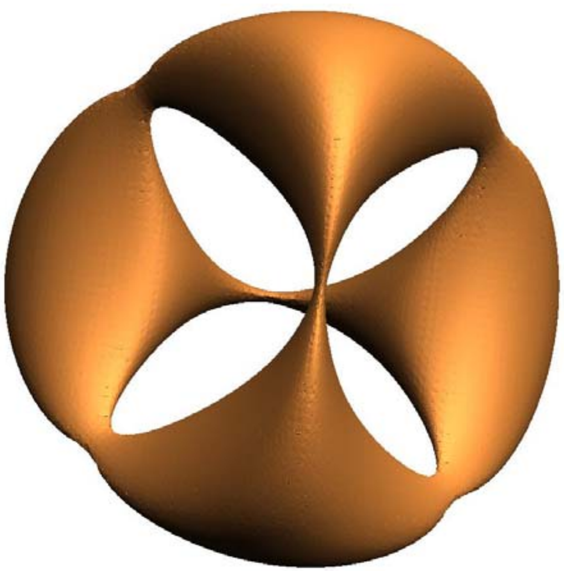
Boolean Operations



Complex shapes designed using tens or hundreds of Booleans on non-linear primitives

Implicit Surface Meshing - Polygonization

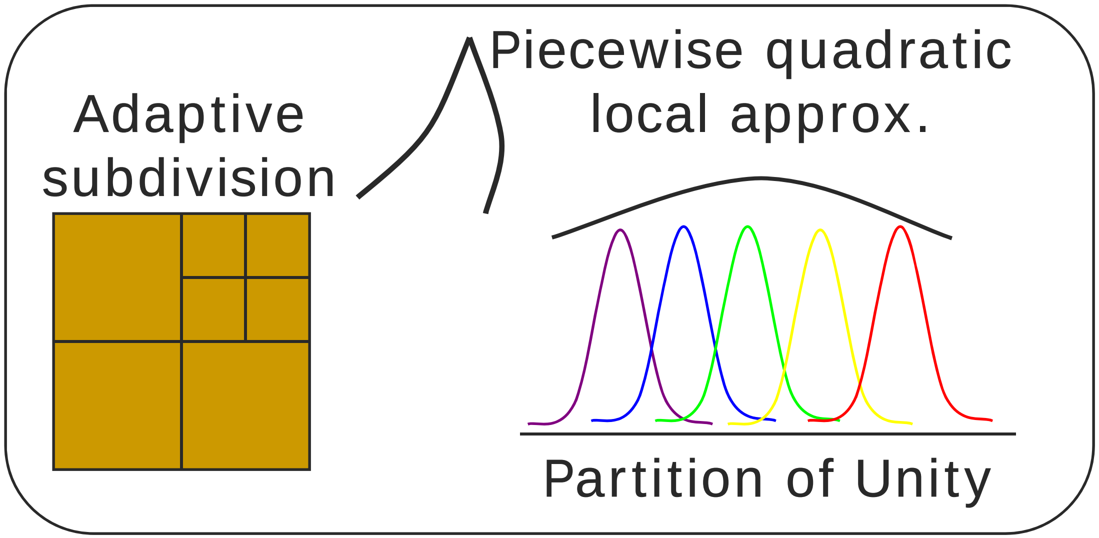
- Described as an implicit equation
  - $f(x,y,z) = 0$



Chair

$$10x^4-12x^2y^2+36x^2z^2-75x^2+y^4+36y^2z^2-75y^2+2z^4-75z^2-16y^2z+16x^2z+640.625 = 0$$

Implicit Surface Meshing - MPU Implicits



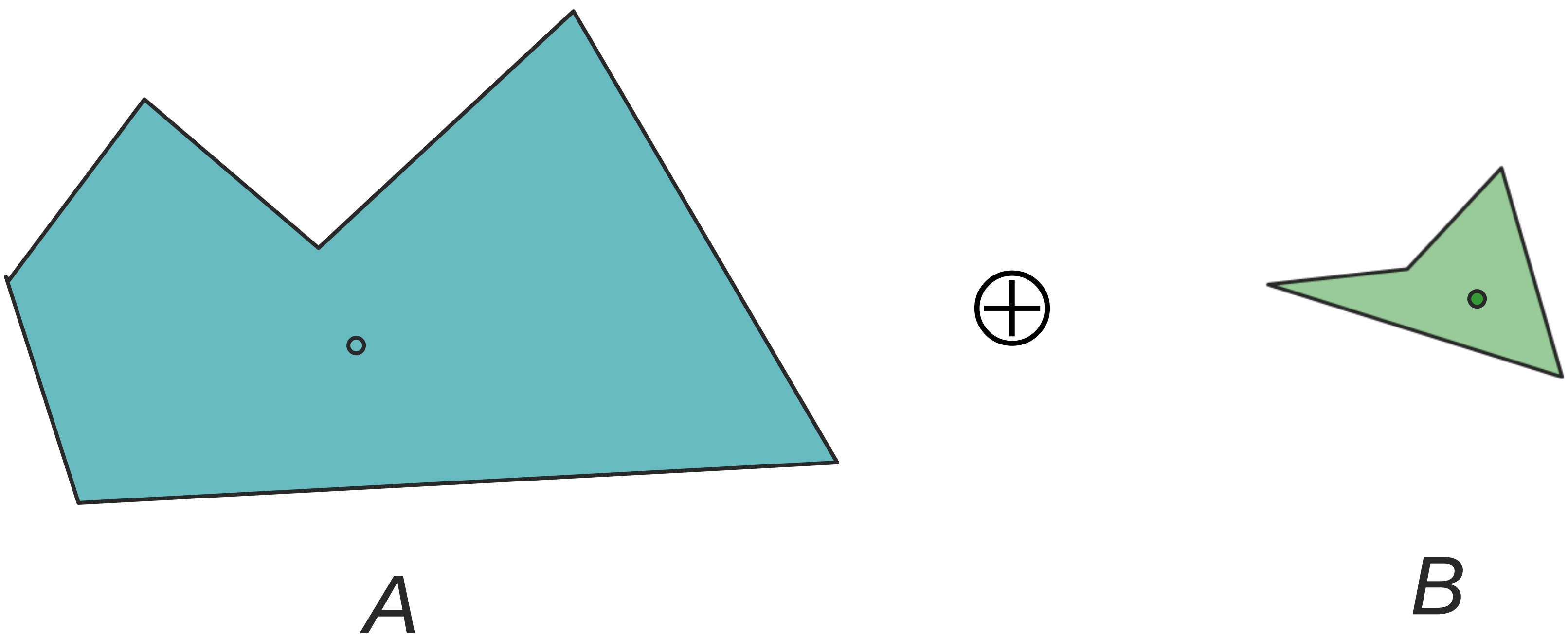
$$f(\mathbf{x}) = \frac{\sum w_i(\mathbf{x}) Q_i(\mathbf{x})}{\sum w_i(\mathbf{x})}$$

Approximation of 14 million points

Slide courtesy of Ohtake et al., Siggraph '03

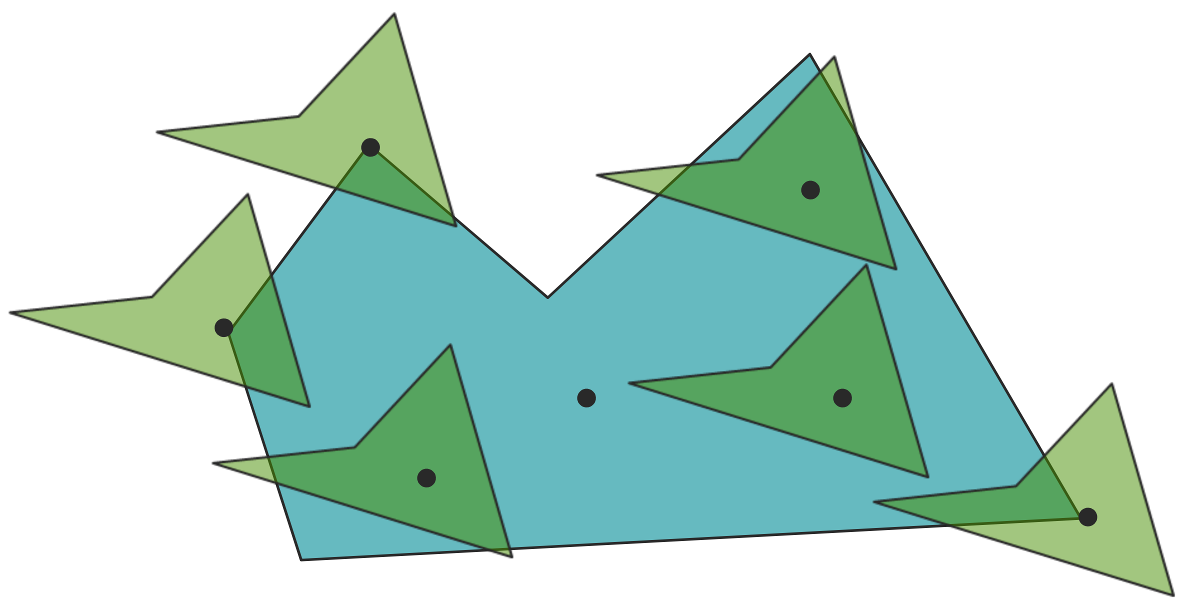
Minkowski Sum

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$

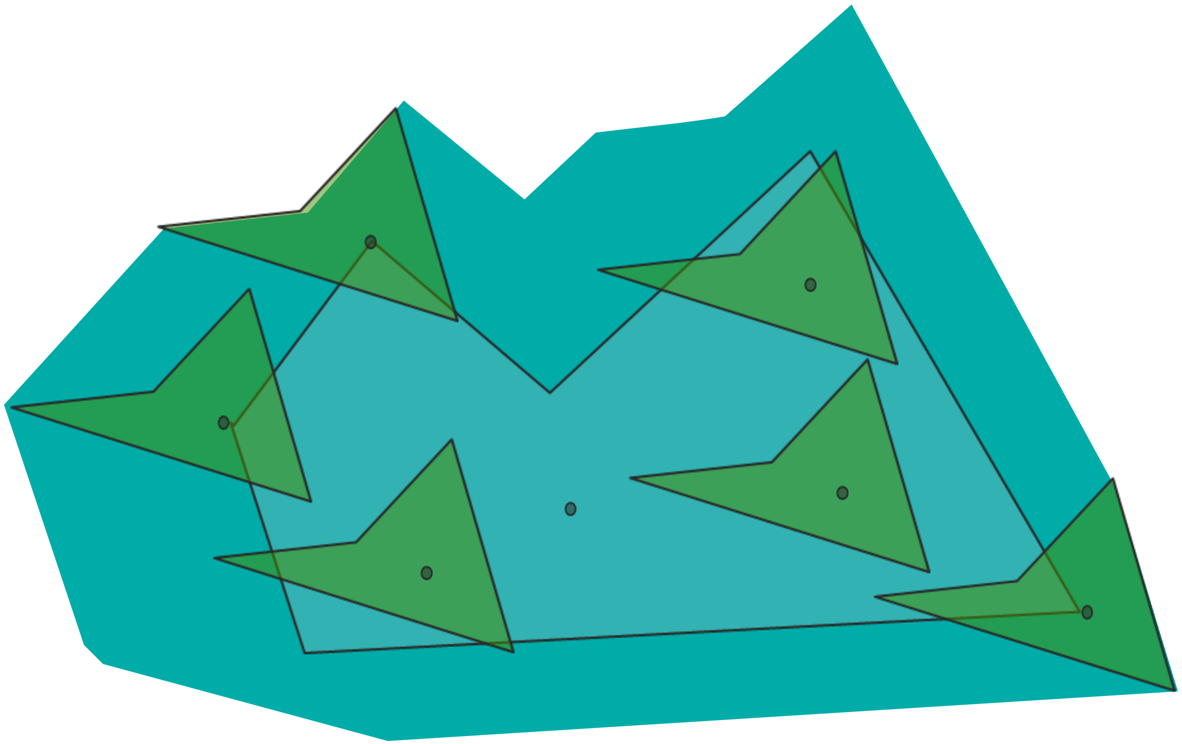


Minkowski Sum

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



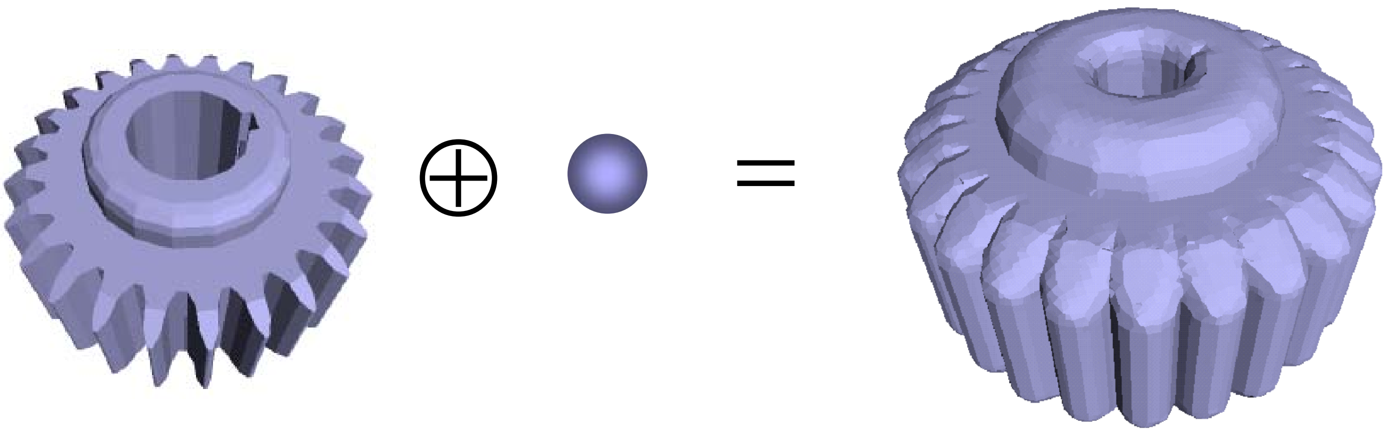
Minkowski Sum



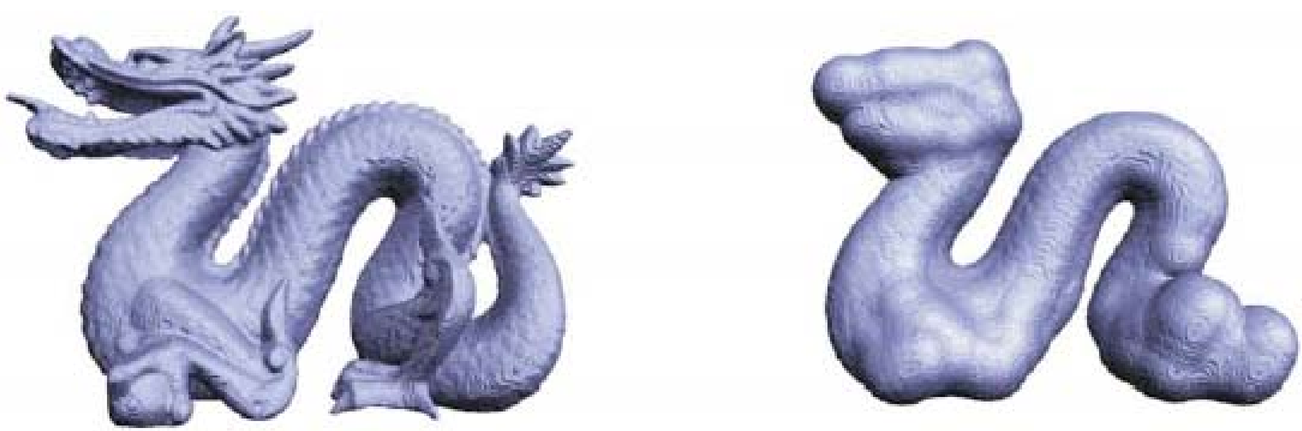
Minkowski Sum

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$

Minkowski Sum

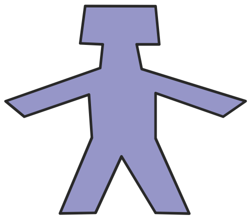
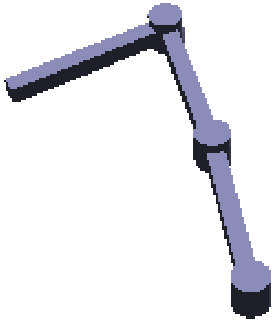


Offsetting



Mathematical Morphological Operations

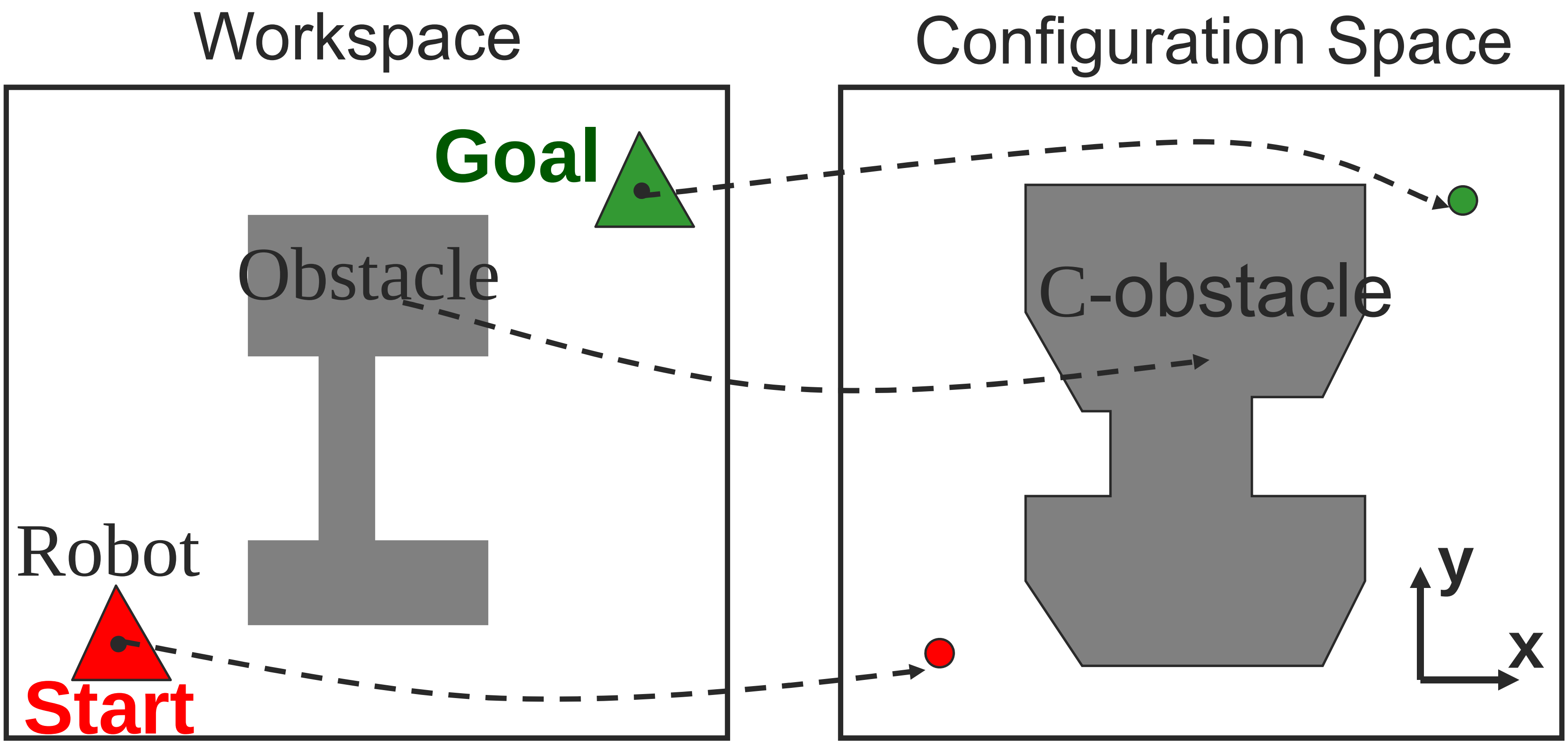
Basic Motion Planning

- Assumptions
  - Robot is either rigid  or articulated 
  - Obstacles are stationary
  - Geometry of both robot and obstacles is known

Configuration Space

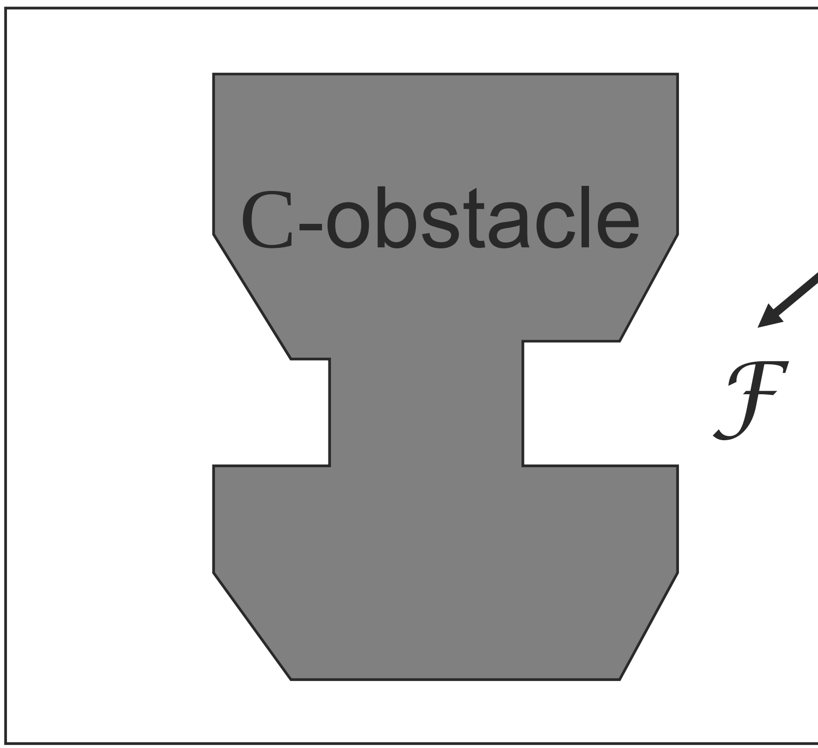
- Motion planning is often studied using a concept called
  - *Configuration space* [Lozano-Perez 1979]
- Set of all positions and orientations of the robot
  - Planar robot capable of only translation
    - (x, y)
  - Translation and rotation
    - (x, y, θ)
  - Its dimension is equal to the number of degrees of freedom (DOF)

Example: 2D Translation



Configuration Space Computation

- Goal is to compute the free space



Free Space

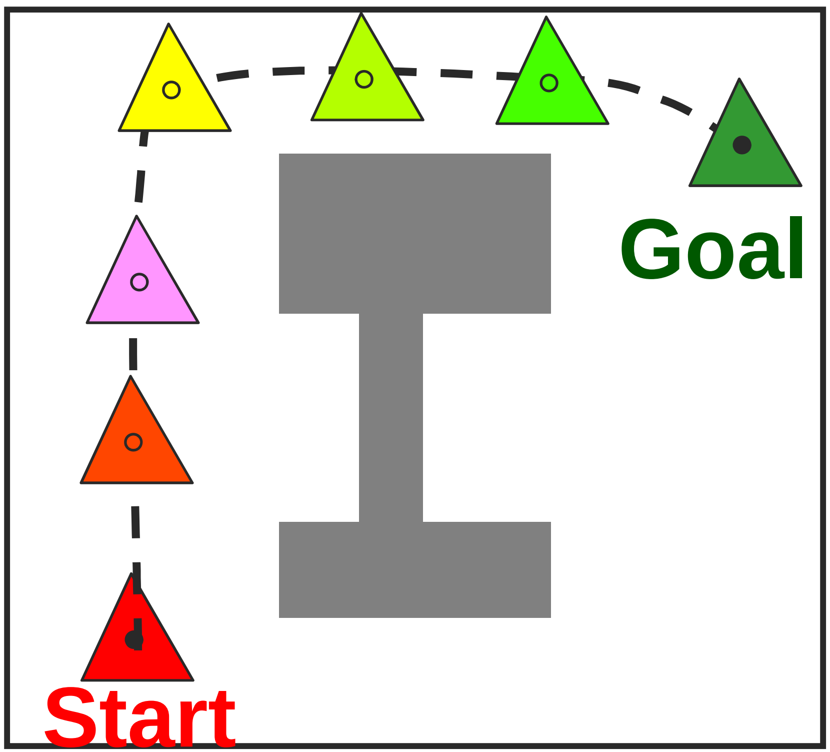
$$\mathcal{F} = \{\mathbf{q} \mid \mathcal{R}(\mathbf{q}) \cap O = \phi\}$$

where $\mathcal{R}(\mathbf{q})$ is the robot placed at configuration $\mathbf{q}$
O is the obstacle

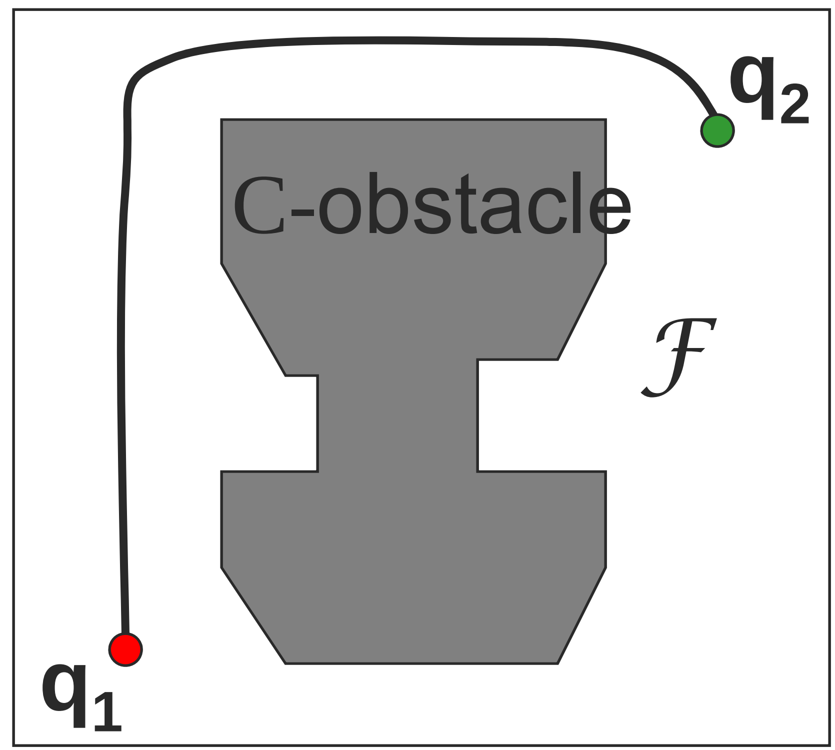
Definition of Motion Planning

Find a curve connecting $\mathbf{q}_1$ and $\mathbf{q}_2$ that lies completely in $\mathcal{F}$

Workspace



Configuration Space



Applications

- Minkowski sums and configuration spaces have also been used for

Interference Detection

Morphing

Penetration Depth

Tolerance Analysis

Packing

Knee/Joint Modeling

Problem Classification

- Surface Extraction
  - Compute a “surface of interest” $\mathcal{E}$
  - Boolean operations, Minkowski sum
    - $\mathcal{E}$ is the boundary of the final solid
  - Configuration space computation
    - $\mathcal{E}$ is the boundary of the free space

Prior Work

- Surface extraction problems
  - Exact approaches
  - Sampling based approaches

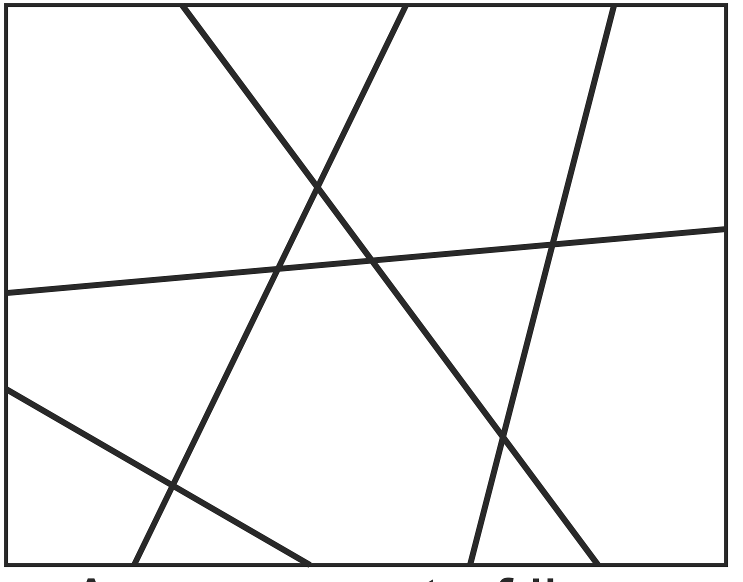
Arrangement Computation

- Arrangement $\mathcal{A}(S)$ of a set S of geometric objects [Halperin 1997; Agarwal & Sharir 2000]

Decomposition of space into relatively open connected *cells* of dimensions 0,...,d

A cell is a maximal connected set of points lying in the intersection of a fixed subset of S

$O(n^d)$ in d dimensions



Arrangement of lines

Exact Approaches for Surface Extraction

1. Enumerate a set S of surface primitives that contribute to $\mathcal{E}$

2. Compute “relevant cells” of $\mathcal{A}$(S) to obtain $\mathcal{E}$
   - Depends on the type of surface extraction problem

Main bottleneck – Arrangement computation

Issues

- Arrangement computation is difficult in practice
  - Need to compute surface-surface intersection
    - Prone to robustness problems
  - Number of primitives is high
  - Primitives could be non-linear

Outline

- Prior Work
  - Surface extraction problems
    - Exact approaches
    - **Sampling based approaches**

Sampling Based Approaches

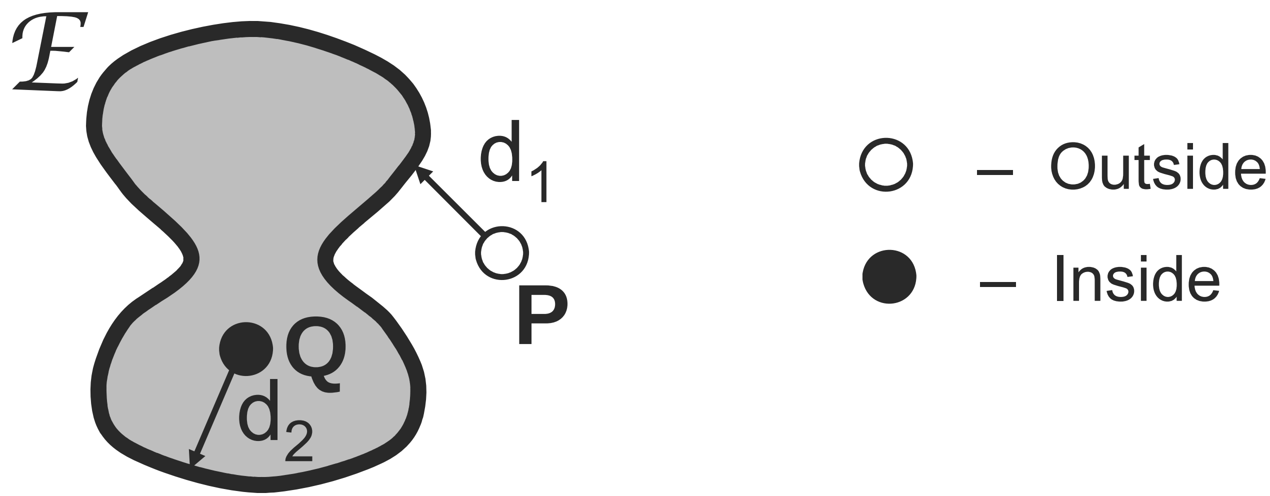
- Approximate $\mathcal{E}$ using
  - Implicit representations
  - Represent $\mathcal{E}$ as a zero-level *isosurface* of a function

$$f: \mathbb{R}^d \rightarrow \mathbb{R}$$

  - $\mathcal{E}$ is the set of points where $f(x,y,z) = 0$

Implicit Representation

- A popular choice of function is the signed distance function/field



- Compute a boundary representation of $\mathcal{E}$
  - extract the zero level isosurface

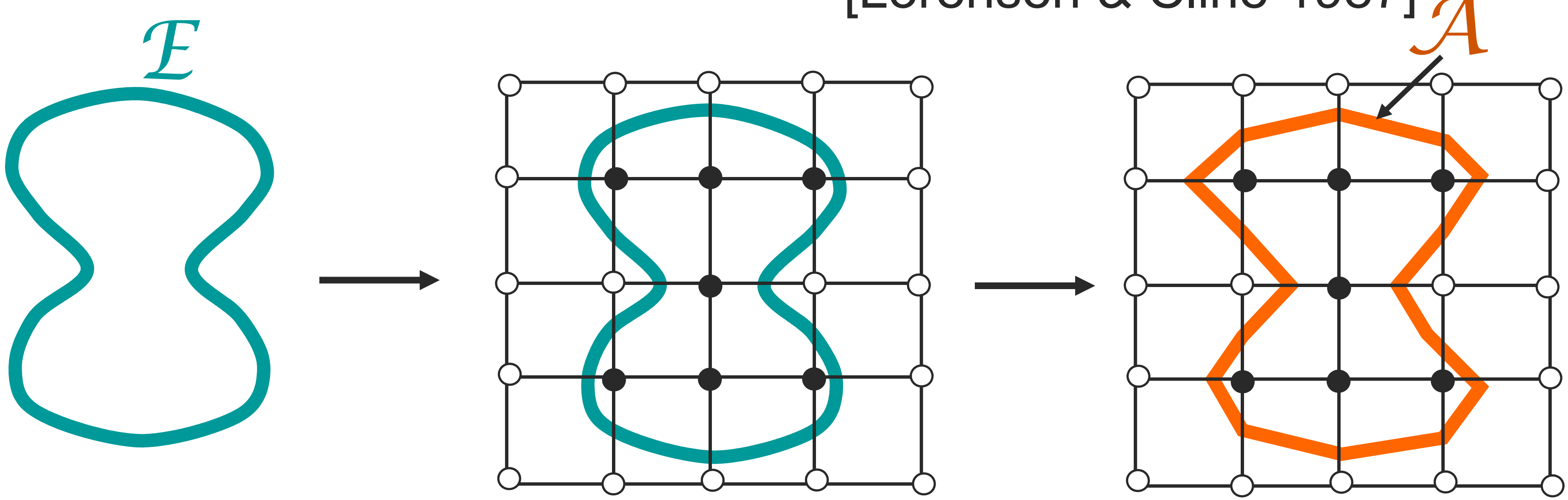
Sampling Based Approach

Sampling

Evaluate function f on a volumetric grid

Reconstruction

Extract an isosurface from the grid using Marching Cubes [Lorensen & Cline 1987]



Marching Cubes

- Determine the topology of surface within the cube – handle actual intersection later
- Each vertex of the cube is assigned a 1 if it lies outside the surface, 0 if inside
- *Intersection pattern*: 256 cases in all, can be reduced to 15 by symmetry
- Use an index into the table

Marching Cubes: Extensions

- Topological consistencies
- Sharp features
- Adaptive grids

Sampling Based Approach

- Simple to implement and efficient in practice
- Easy to perform Boolean operations
  - Perform min/max operations on distance fields
- Widely used for geometric modeling

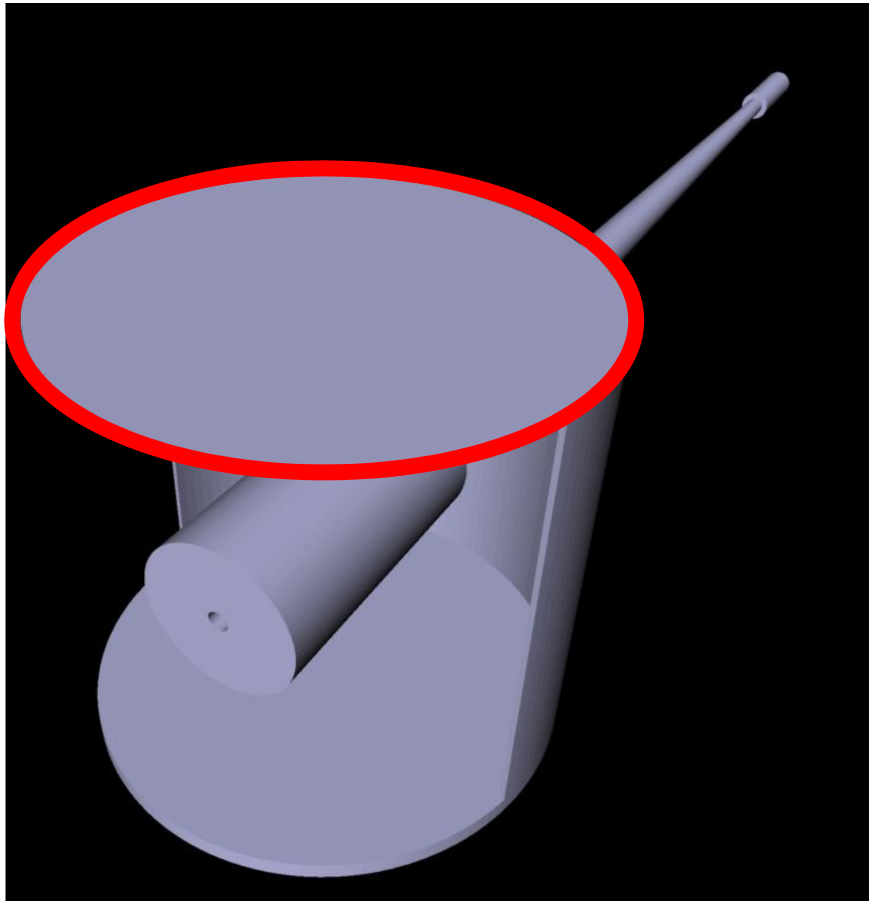
[Wyvill et al. 1986; Bloomenthal 1997; Breen et al. 1998; Frisken 2000]

Issues

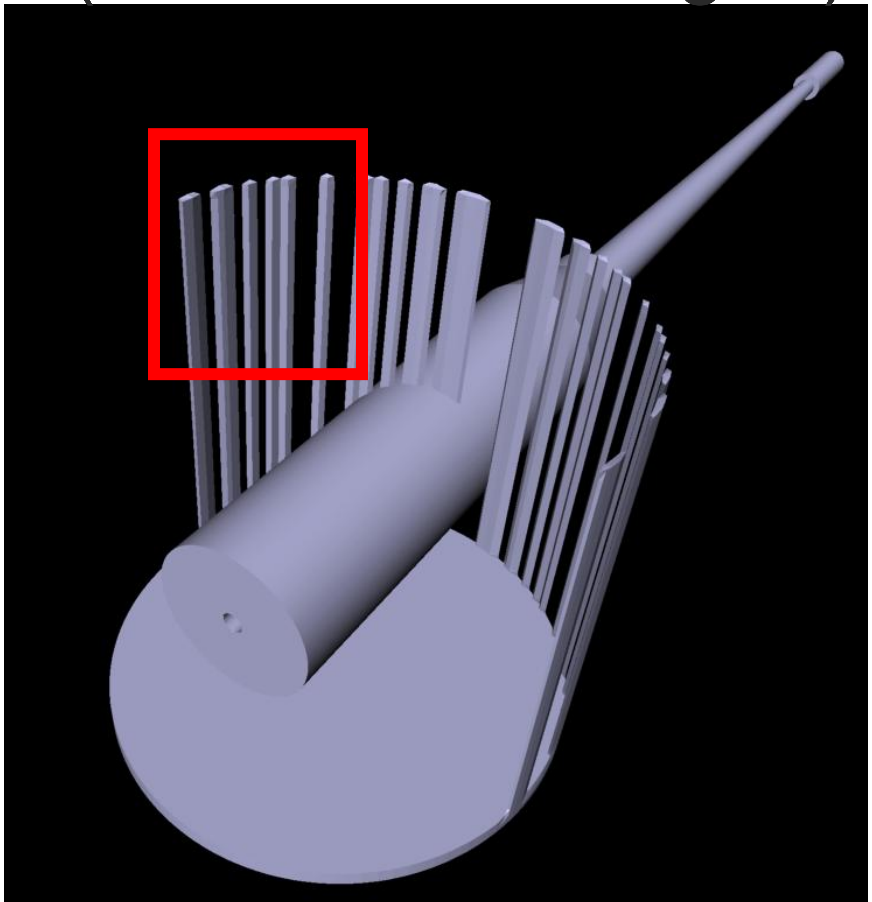
- Accuracy of Marching Cubes (MC) reconstruction?
  - Depends on the rate of sampling
    - Resolution of the underlying volumetric grid
- Insufficient sampling may produce a “poor approximation”

Errors in MC Reconstruction

Exact Surface



MC Reconstruction
(on a 64x64x64 grid)



Topology Preserving Surface Extraction: Limited cases

- Assume smooth implicit surface
- Critical point analysis
  - Stander & Hart 97; Boissonnat et al. 04
  - Require tracking of critical points
- Spatial subdivision and interval arithmetic
  - Plantinga & Vegter 04
- Restricted Delaunay triangulation
  - Topological ball property
  - Oudot & Boissonnat 03 and 05
  - Cheng, Dey & Ramos 04 and 06
    - Anisotropic Voronoi diagram

Prior Methods

- Prior exact and sampling based approaches do not capture both
  - Simplicity
  - Accuracy/completeness
- Goal is to develop sampling based algorithms that can guarantee accuracy/completeness

Our Goals: Sampling Conditions

- Sampling
  - Process of generating a volumetric grid in space using ***certain criteria***
  - Criteria determine sampling condition

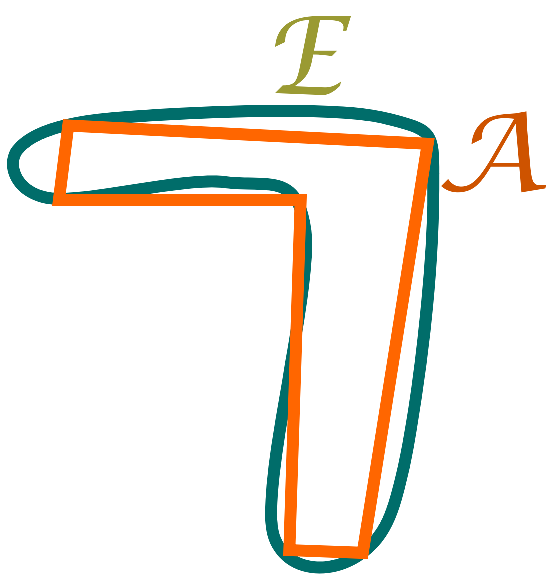
Surface Extraction

- Output is a polygonal approximation $\mathcal{A}$ to the exact surface $\mathcal{E}$

If the *sampling condition* is met, then we can guarantee a *geometrically accurate and a topologically correct* approximation.

Geometric Guarantees

- Bounded two-sided Hausdorff error
  - Given any $\varepsilon > 0$, we have $H(\mathcal{A}, \mathcal{E}) < \varepsilon$



Every point on $\mathcal{A}$

is **CLOSE** to a point on $\mathcal{E}$

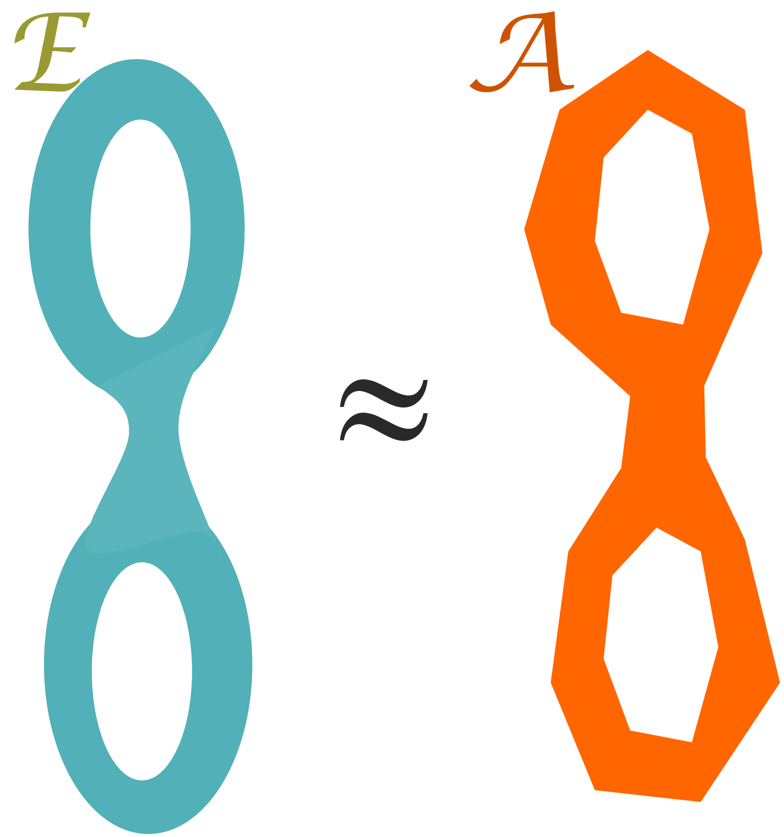
and **vice-versa**

$$H(\mathcal{A},\mathcal{E}) = \max\left\{ \max_{\mathbf{a}\in\mathcal{A}} d(\mathbf{a},\mathcal{E}) \,,\, \max_{\mathbf{b}\in\mathcal{E}} d(\mathbf{b},\mathcal{A}) \right\}$$

Topological Guarantee

- Homeomorphism
  - Continuous bijective mapping with a continuous inverse
- $\mathcal{E} \approx \mathcal{A}$

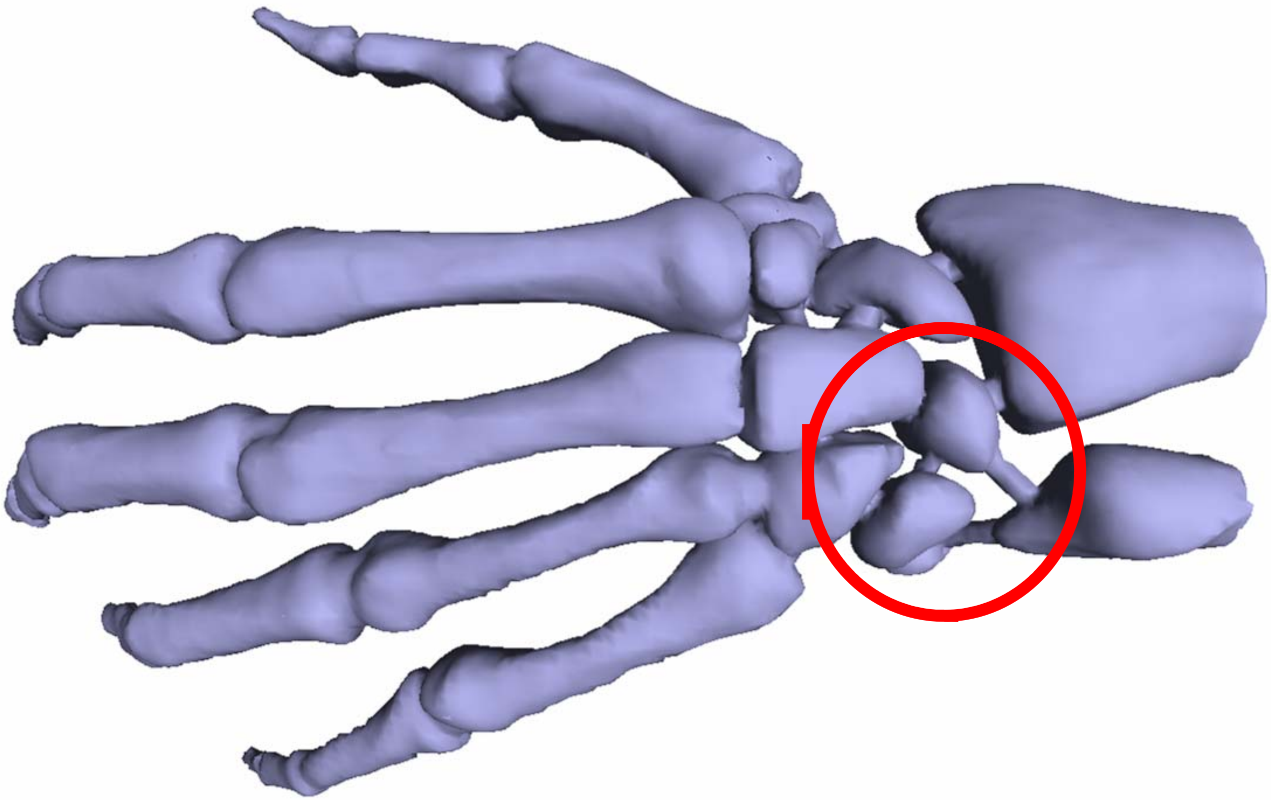
  if there exists a homeomorphism between them
  - Same number of connected components and handles



Topology Preservation

CAD

Medical Datasets



Outline

- Surface Extraction
- Minkowski sum approximation
- Configuration space approximation
- Motion Planning
- Conclusions

Outline

- **Surface Extraction**
  - Topological guarantee
    - Sampling condition
    - Sampling algorithm
  - Results
  - Analysis

Sampling Condition

- Every cell in the volumetric grid must satisfy

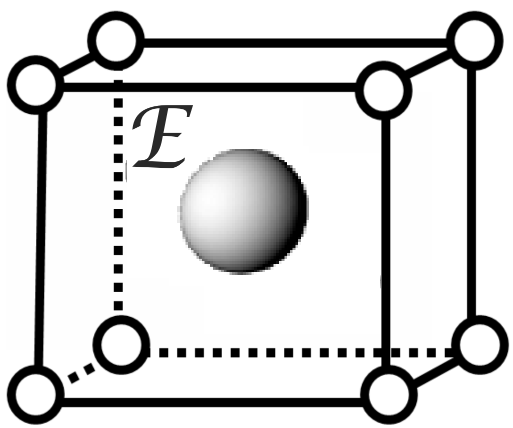
1. Simple-cell criterion

2. Star-shaped criterion

[Varadhan et al.'04; Varadhan et al.'06]

Simple-Cell Criterion

- Simple-cell criterion forbids certain cases

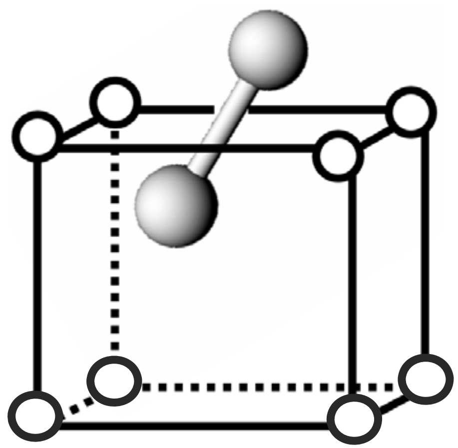


Complex Voxel

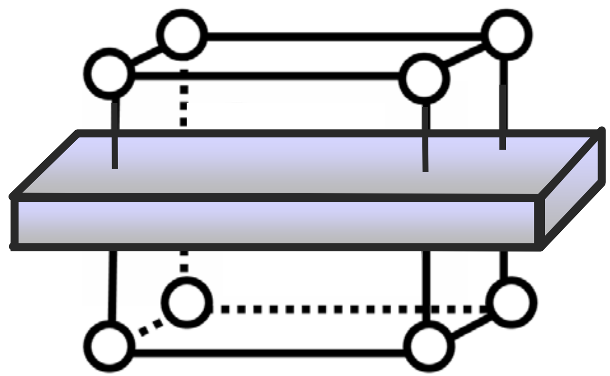
1. $\mathcal{E}$ intersects the voxel and
2. All the voxel vertices have the same sign

Simple-Cell Criterion

Complex Face

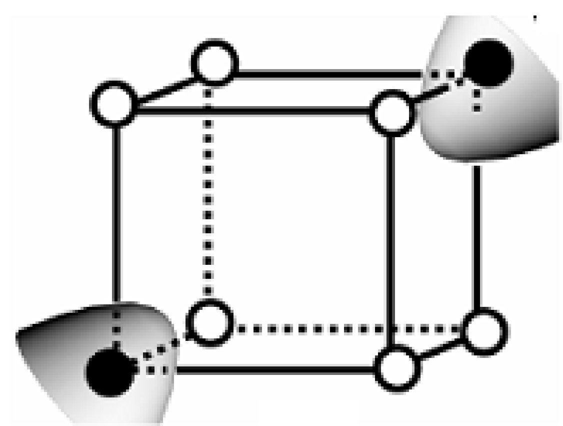


Complex Edge



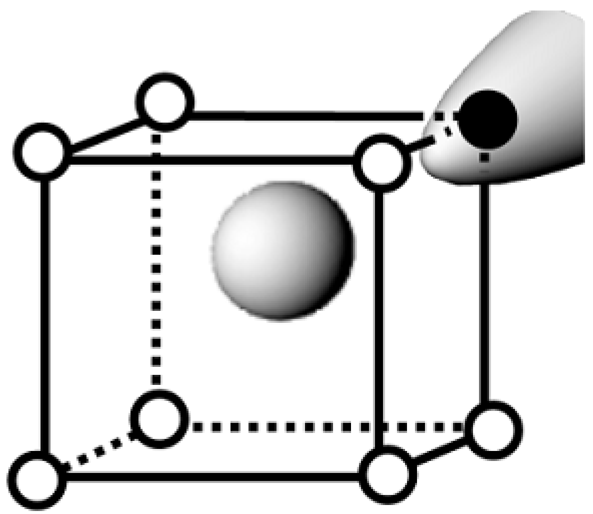
$\mathcal{E}$ intersects edge more than once

Topological Ambiguity



Simple-Cell Criterion

- A cell satisfies the simple-cell criterion if
  - No complex voxel/faces/edges
  - No ambiguity
- Not sufficient for topology preservation



Star-shaped Property

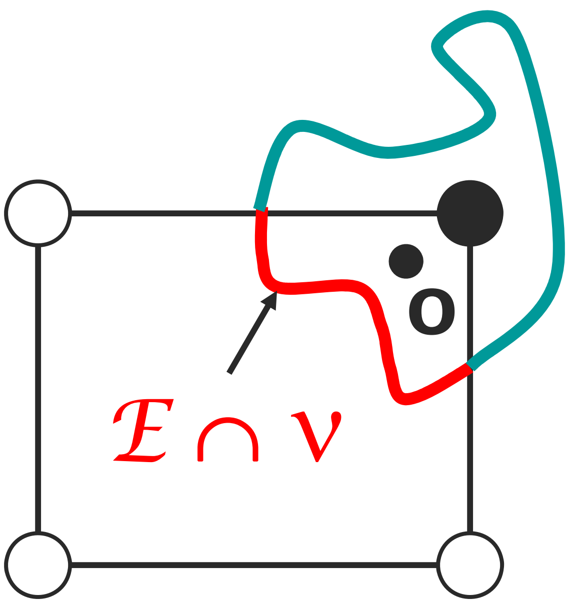
- A surface S is star-shaped if there exists a point **o** in $\mathbb{R}^3$ that can "see" every point in S

$$\mathbf{op} \cap S = \{\ \mathbf{p}\ \} \text{ for all } \mathbf{p} \text{ in } S$$

Point **o** is called a *guard*

Star-shaped Property

- $\mathcal{E}$ is star-shaped with respect to (w.r.t) a voxel ν if $\mathcal{E} \cap \nu$ is star-shaped w.r.t some point **o** in $\mathbb{R}^3$



Guard **o** may lie outside ν

Star-shaped Property

- $\mathcal{E}$ is star-shaped w.r.t a face f if

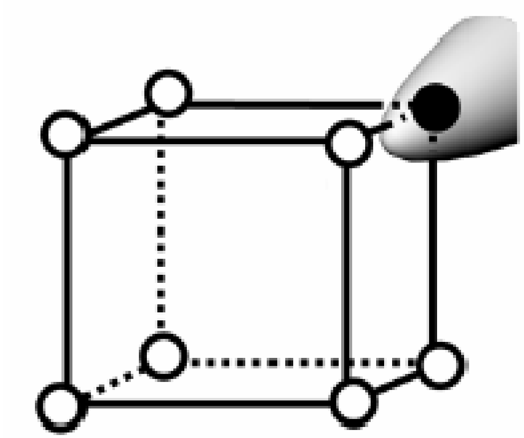
  $\mathcal{E} \cap f$ is star-shaped w.r.t some point **o** in the

  plane containing f

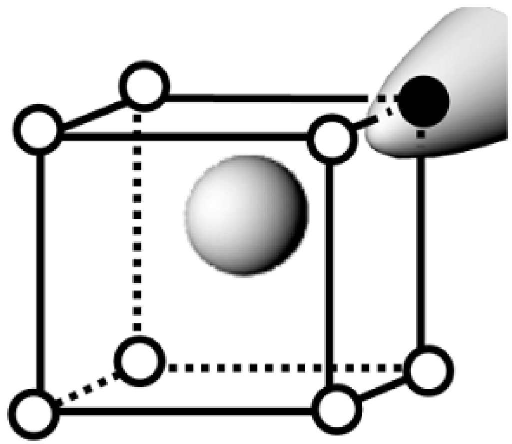
Star-shaped Criterion

- A cell satisfies the star-shaped criterion if

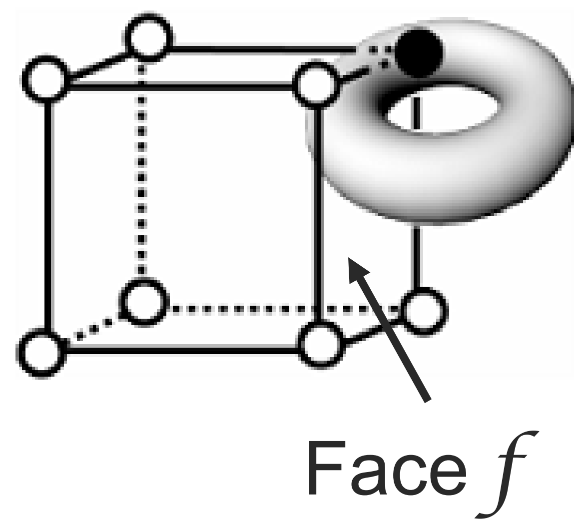
  $\mathcal{E}$ is star-shaped w.r.t its voxel and each face



Star-shaped



Not star-shaped w.r.t voxel



Not star-shaped w.r.t face f

Topological Equivalence

- **Theorem:** If every grid cell satisfies simple-cell and star-shaped criteria

  then Marching Cubes can be applied to the grid to produce $\mathcal{A}$ such that

$$\mathcal{A} \approx \mathcal{E}$$

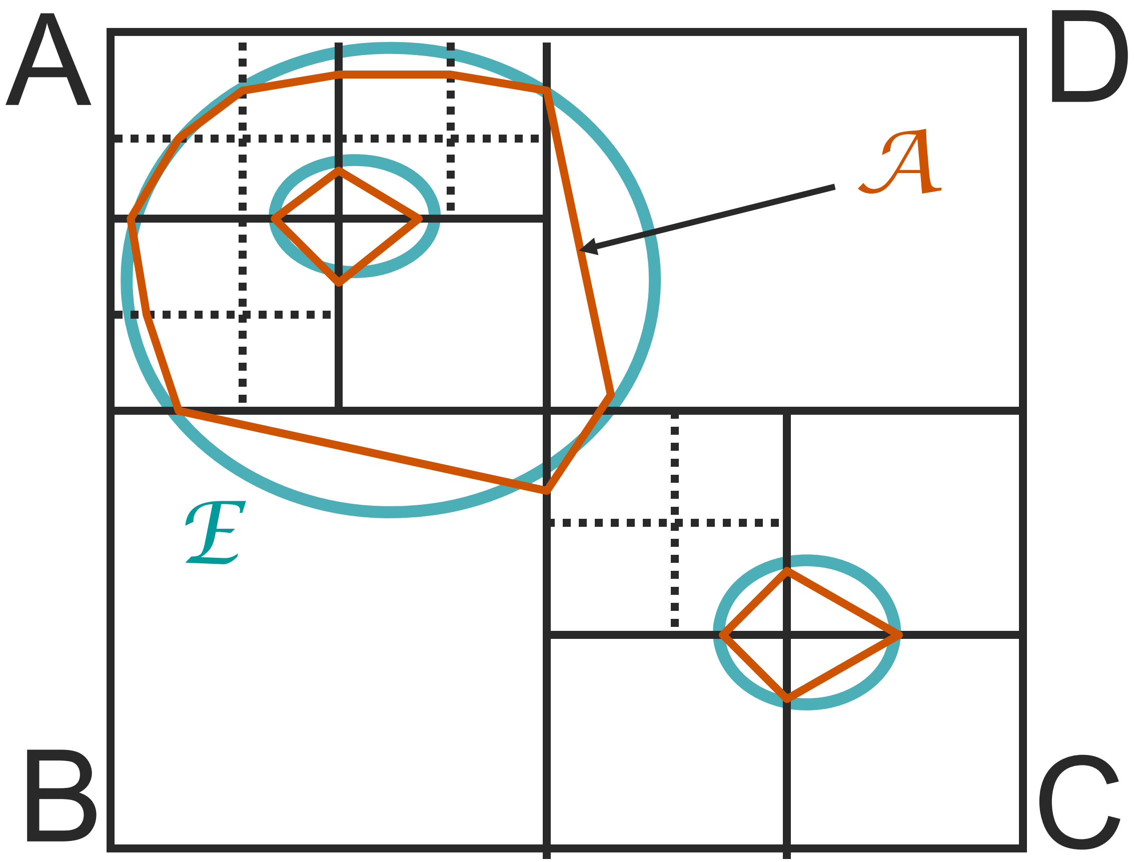
[Varadhan et al.'04]

Outline

- Surface Extraction
  - Topological guarantee
    - Sampling condition
    - **Sampling algorithm**
  - Results
  - Analysis

Sampling Algorithm

Perform adaptive subdivision by applying the sampling condition recursively



Sampling Algorithm

- Easy to verify the two sampling criteria
  - Simple-cell test
  - Star-shaped test

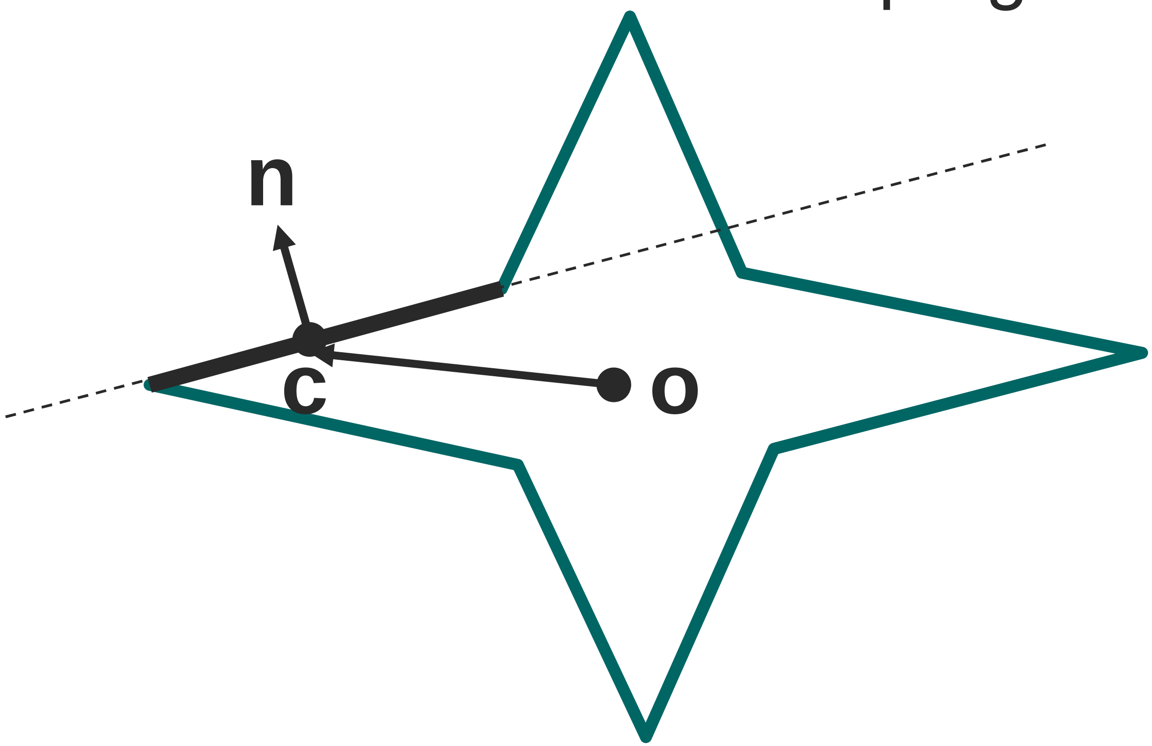
Simple-Cell Test

- Complex voxel

1. Check if $\mathcal{E}$ intersects the voxel
   - Use max-norm distance computation [Varadhan et al. 2003]
2. Check if the voxel vertices have the same sign

Star-shaped Test

- Polyhedral primitives
  - Reduces to linear programming

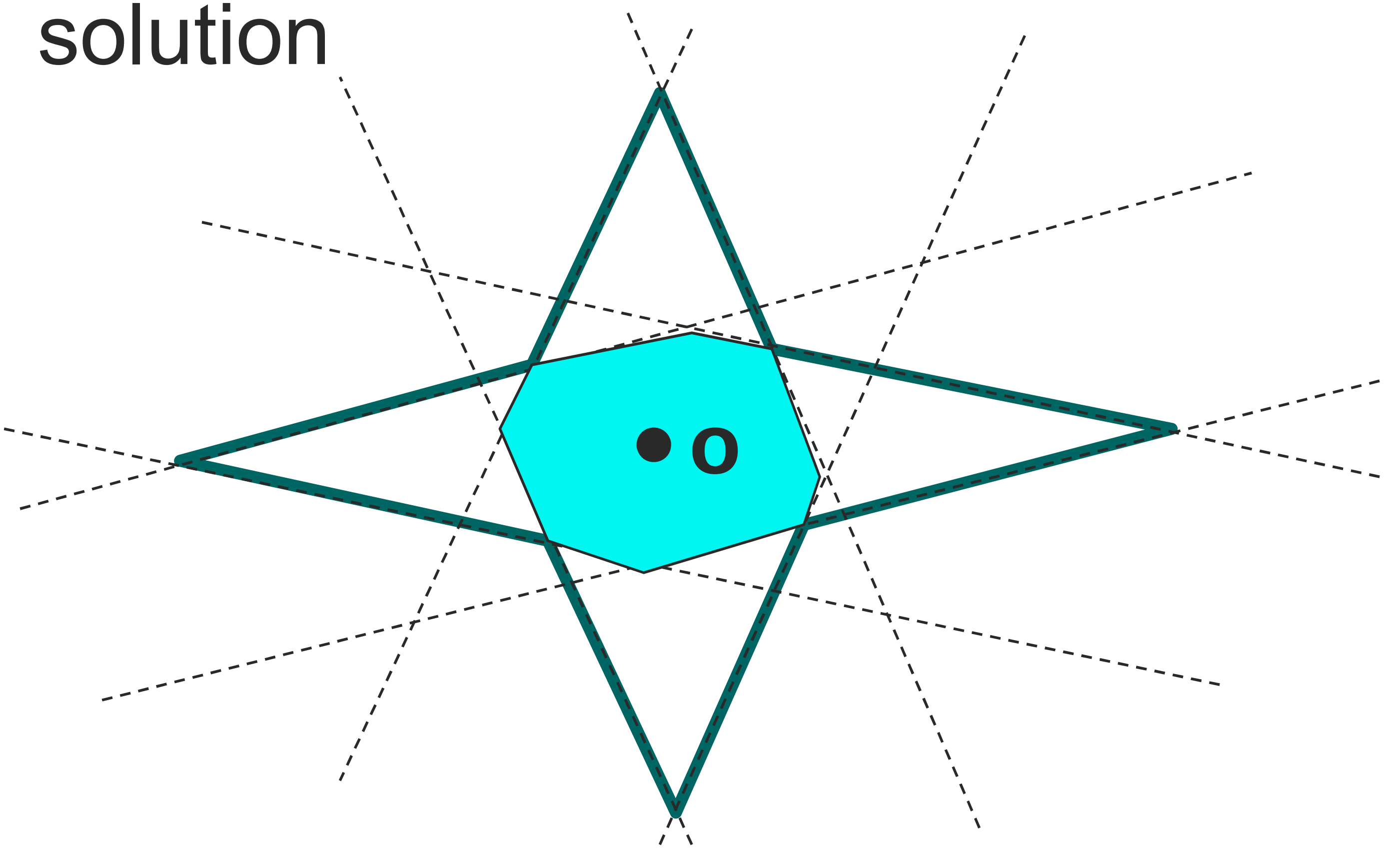


Linear constraint

$\mathbf{n} \cdot (\mathbf{c} - \mathbf{o}) > 0$

Star-shaped Test

- Check if the linear program has a feasible solution

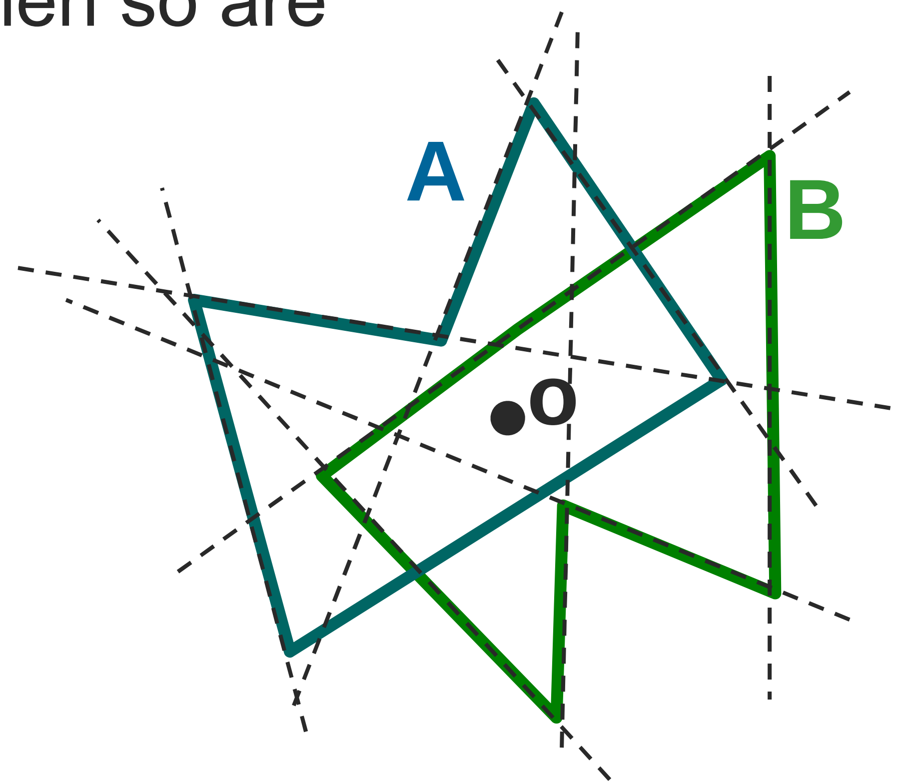


Star-shaped Test

- If $\mathcal{E}$ is defined as a Boolean combination
  - If both **A** and **B** are star-shaped w.r.t a common point **o**, then so are

    $A \cup B$ and $A \cap B$

  – Combine the linear constraints



Star-shaped Test

- Algebraic surfaces

- Check if there exists a point **o** in $\mathbb{R}^3$ such that

  $$\mathbf{n}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{o}) > 0 \quad \forall\, \mathbf{x} \in S$$

- Exact test is difficult

- Modified test based on linear programming and interval arithmetic

Outline

- Surface Extraction
  - Topological guarantee

  - **Results**
    - Boolean operations
    - Simplification
    - Remeshing
    - Implicit surface meshing

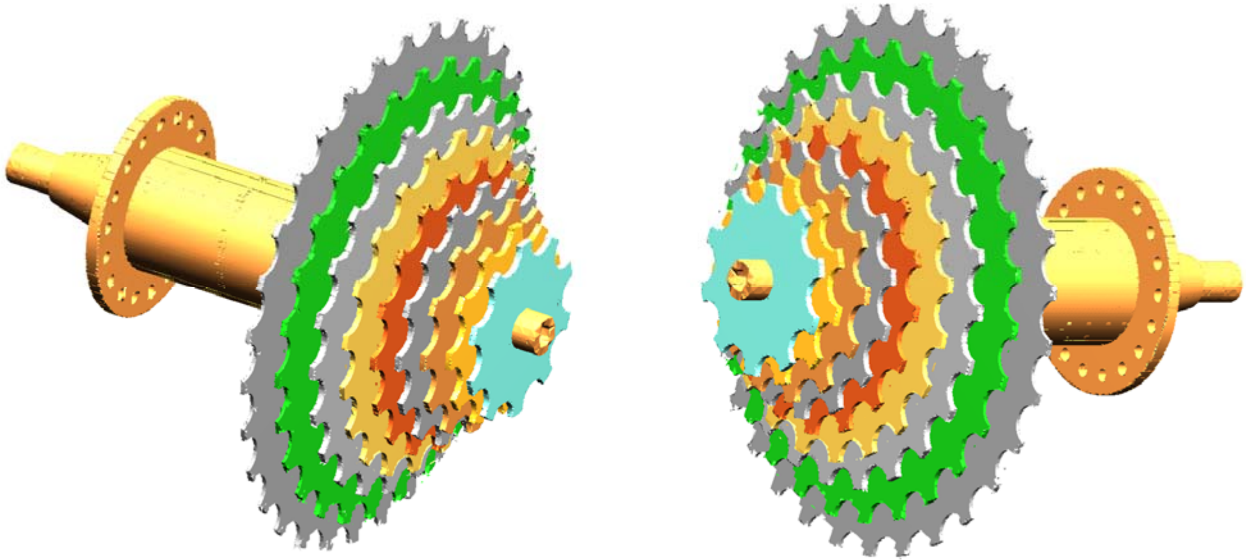
  - Analysis

Boolean Operations

Union of Two Dragons
Input: 1.7M tris
Output: 118K tris
95 secs

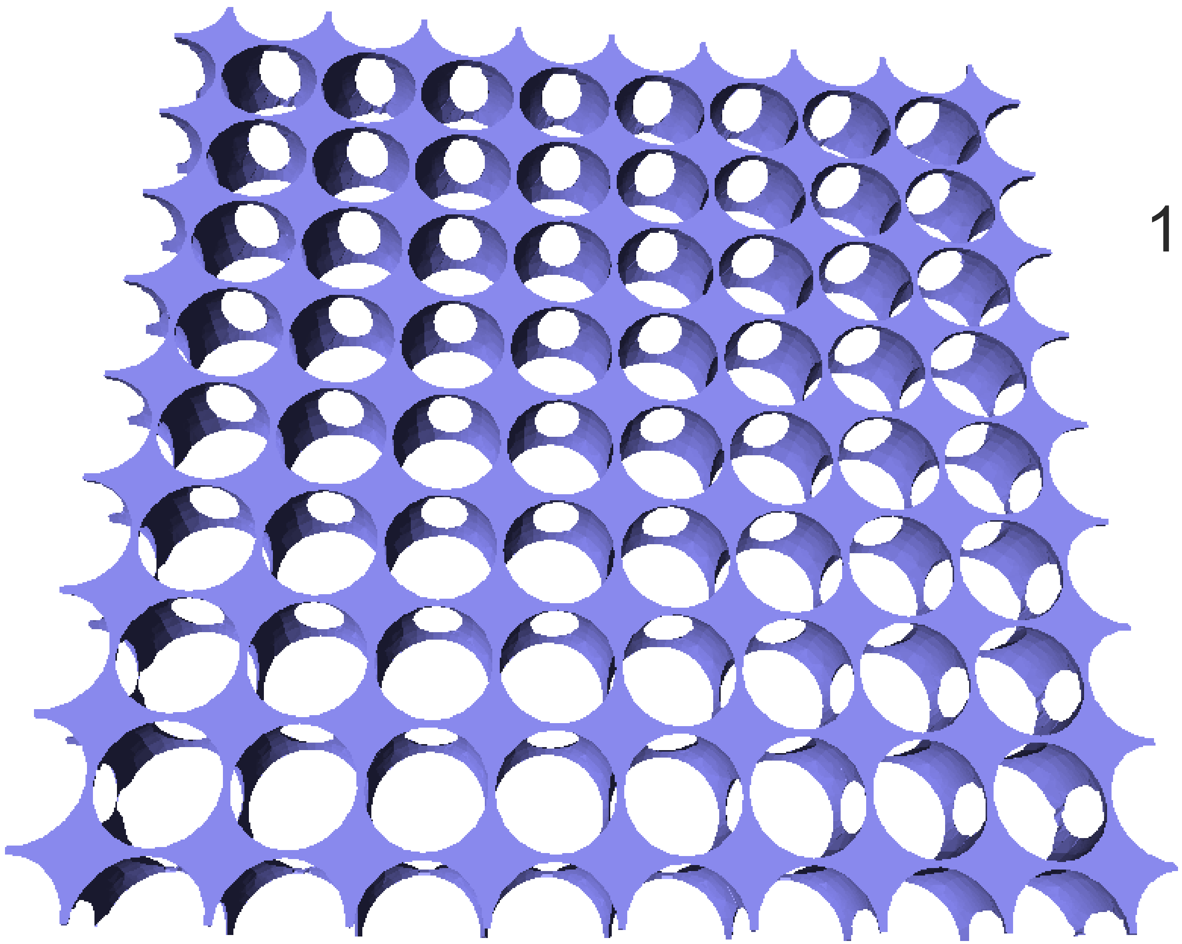
Difference on Turbine Blade
Input: 1.7M tris
Output: 319K tris
116 secs

Boolean Operations



CAD Model
14 solids
84 Boolean operations

Curved Primitives

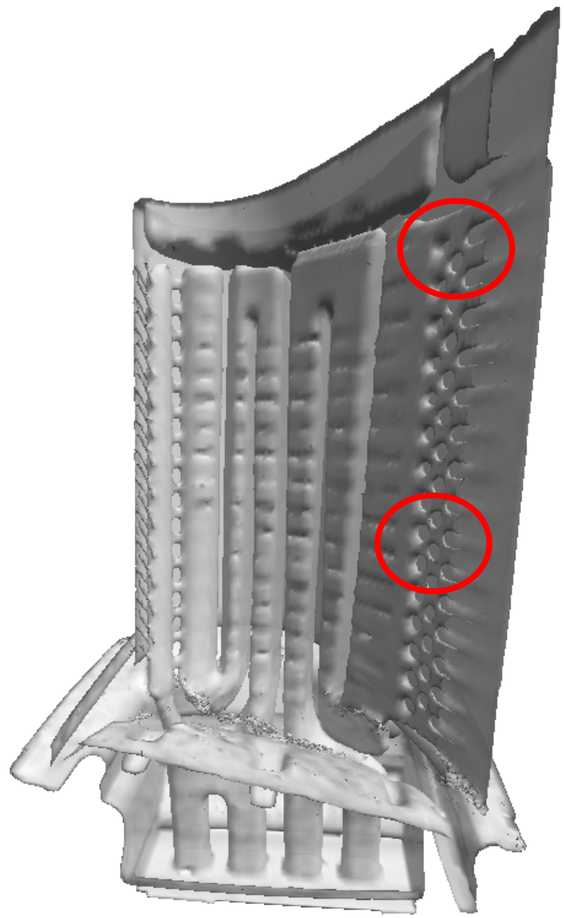


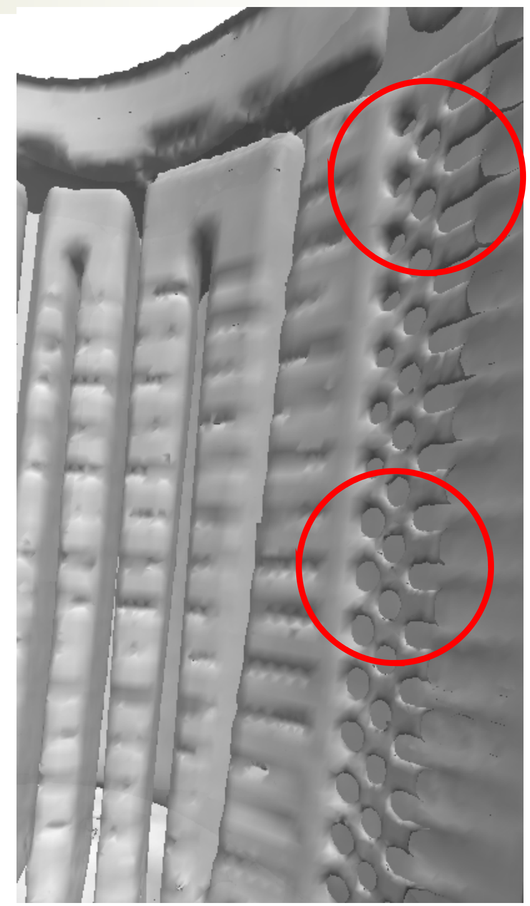
100 difference operations between a polyhedron and ellipsoids

Genus 208

16 secs

Volumetric Simplification





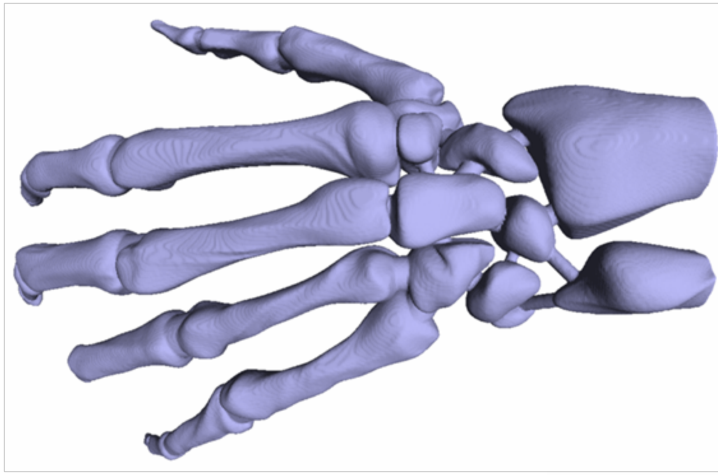
Turbine Exterior Interior

1.7M tris

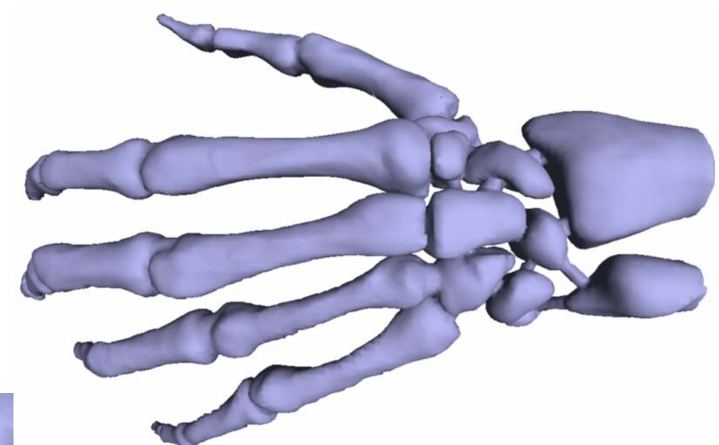
Simplified

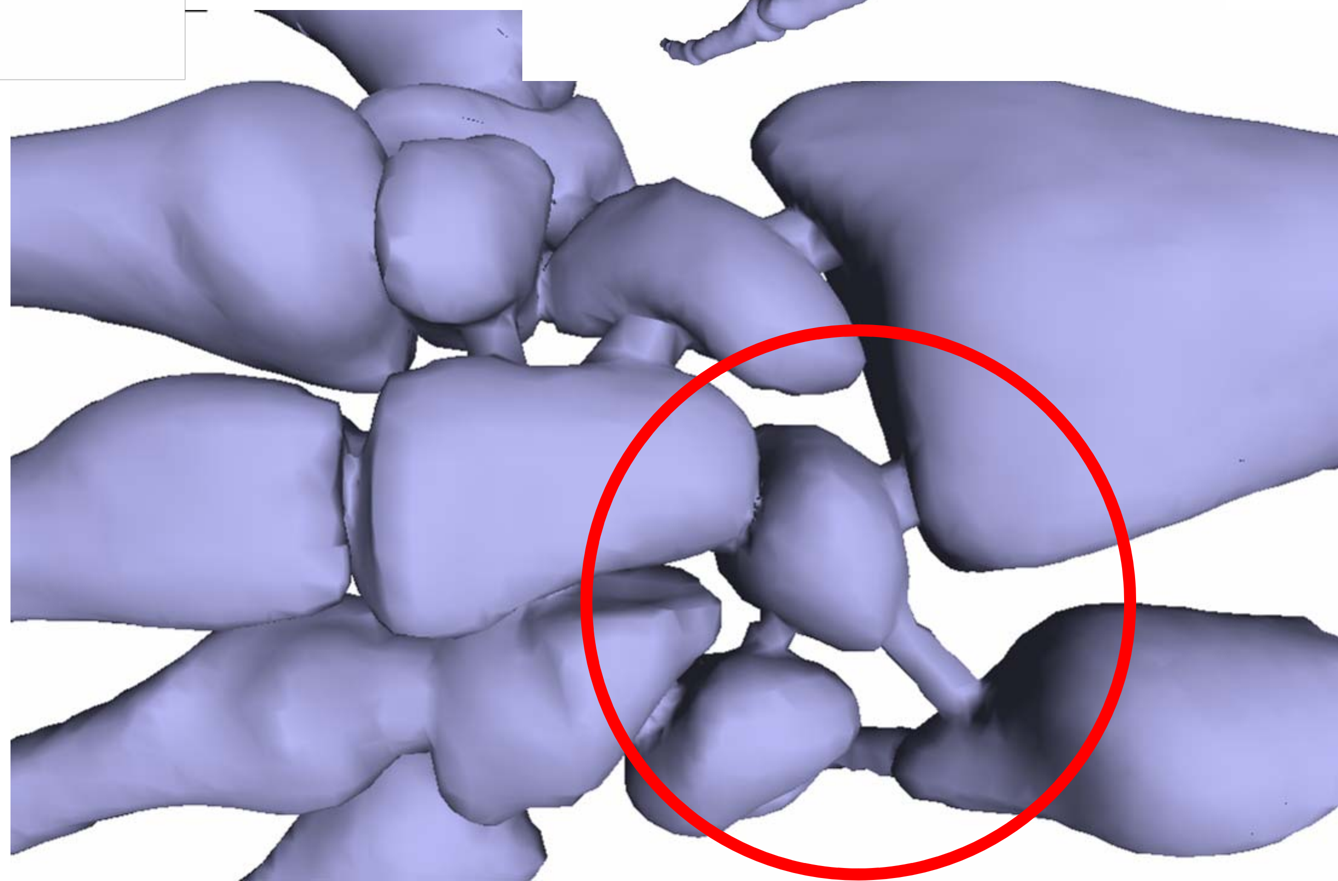
511K tris, 110 secs

Original
654K tris



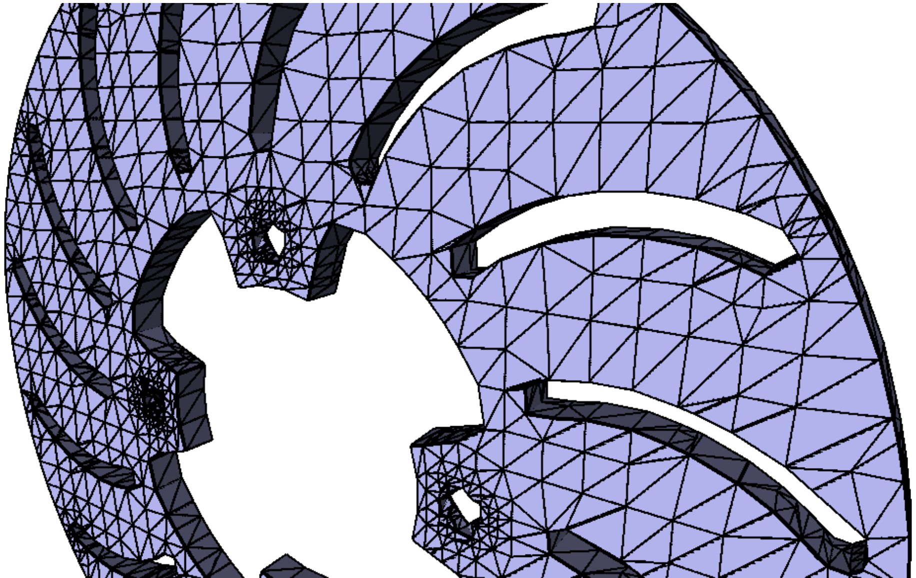
Simplified
58K tris, 36 secs





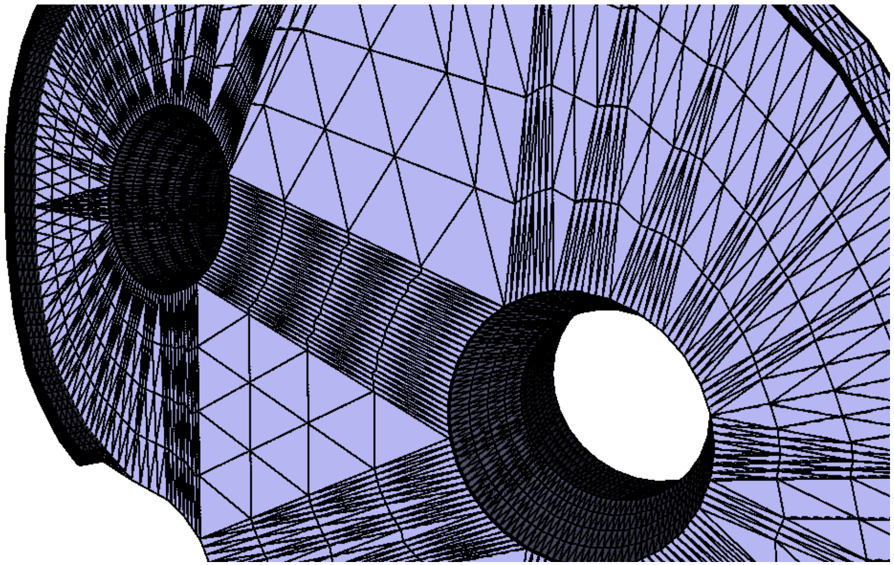
Remeshing

Original
14K tris

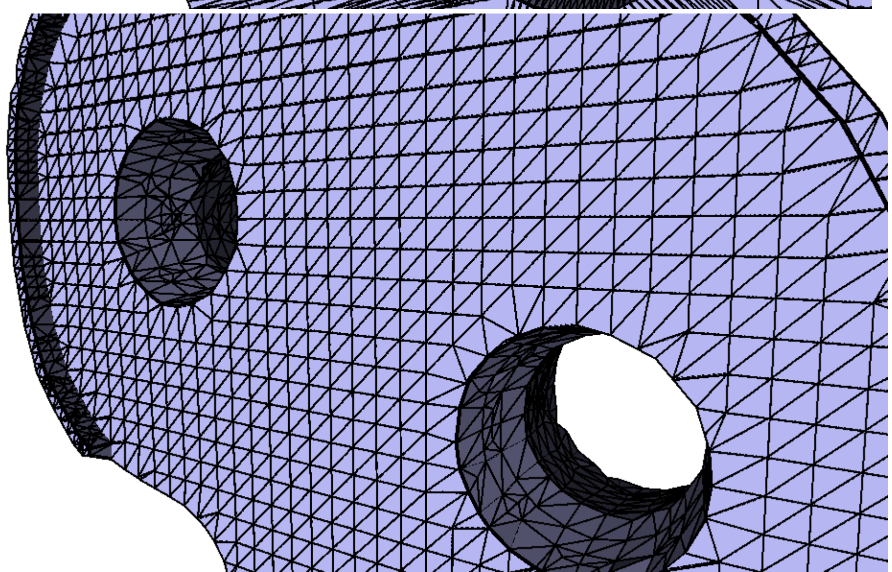


Remesh
7K tris
1.85 secs

Remeshing

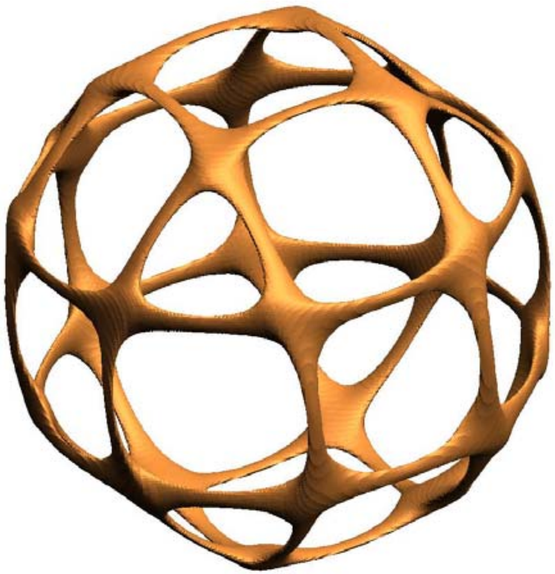


Original
41K tris

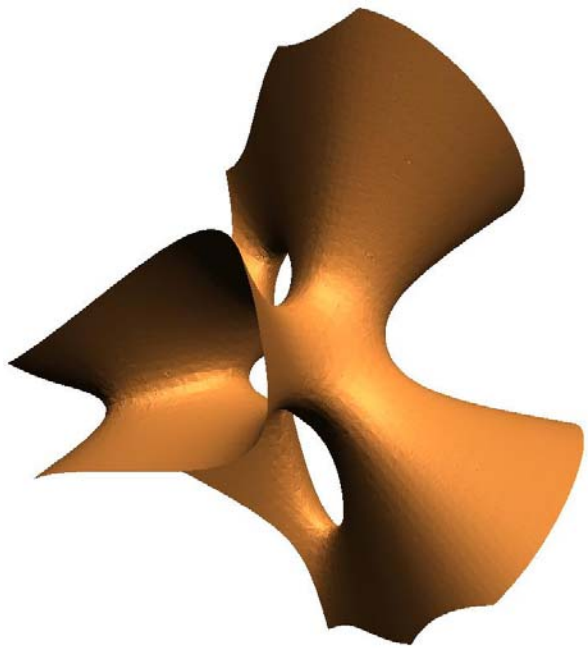


Remesh
16K tris
2.8 secs

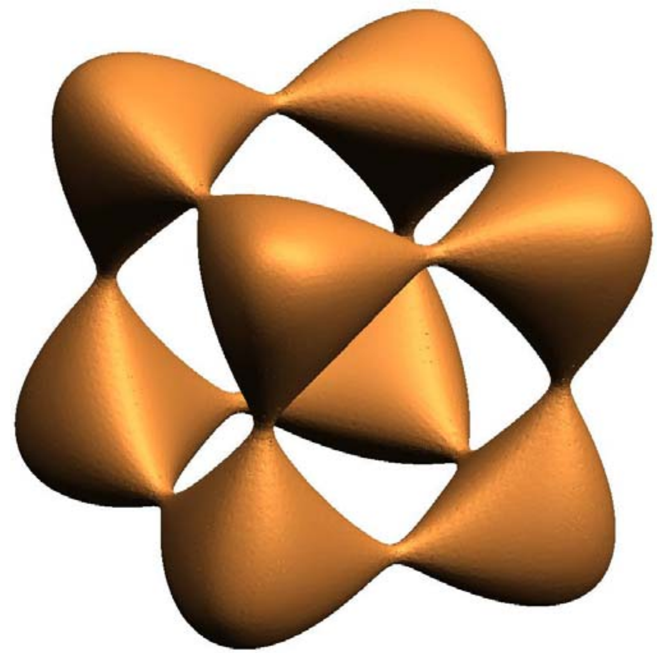
Implicit Surface Meshing – Algebraic Surfaces



Decocube
Degree 12

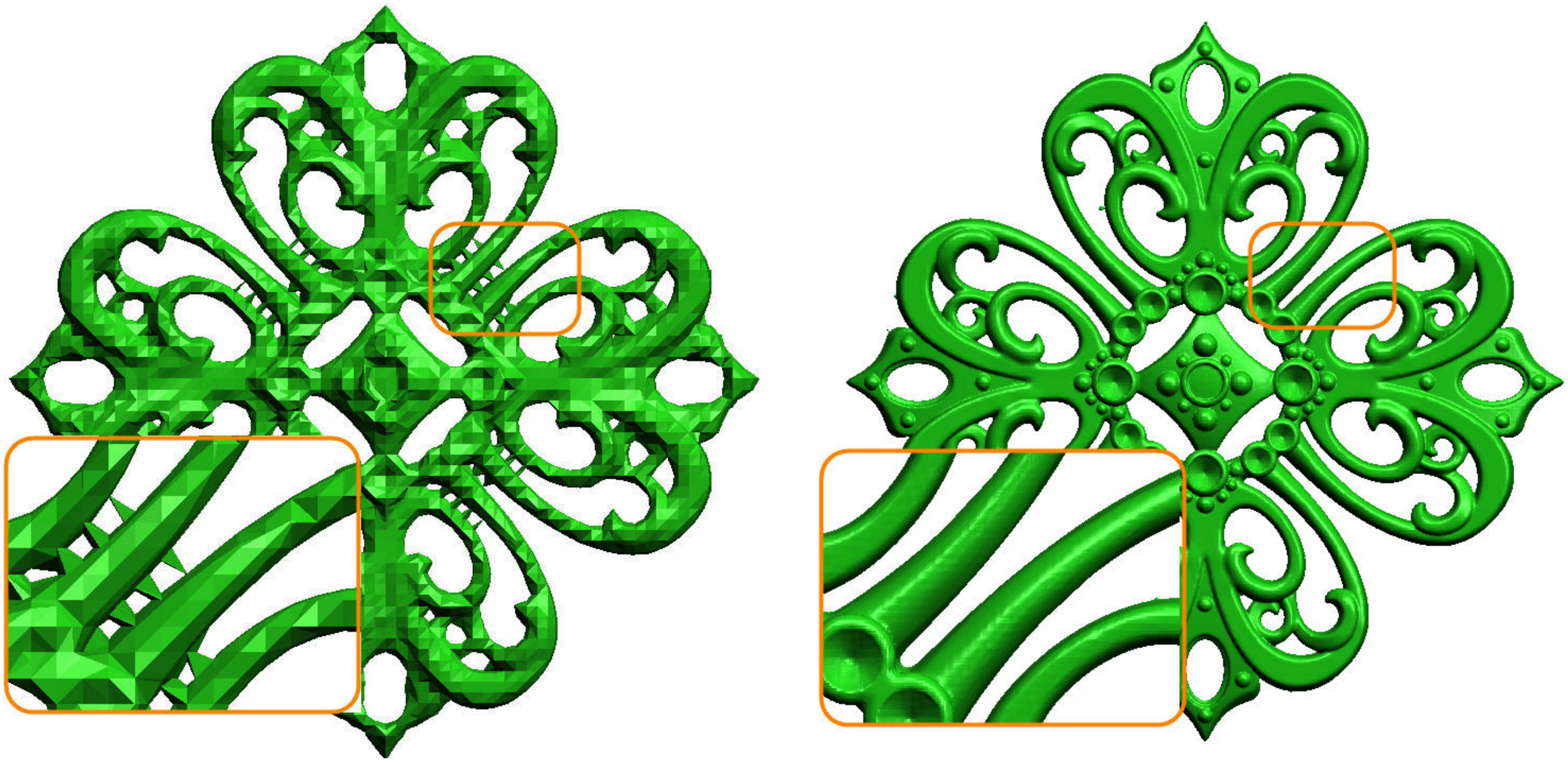


Diagonal
Degree 3



Tanglecube
Degree 4

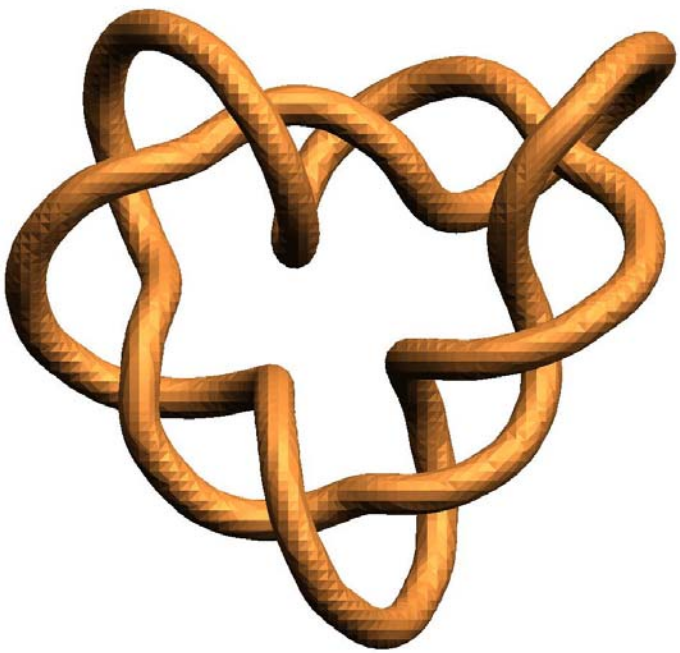
Implicit Surface Meshing – MPU Implicits

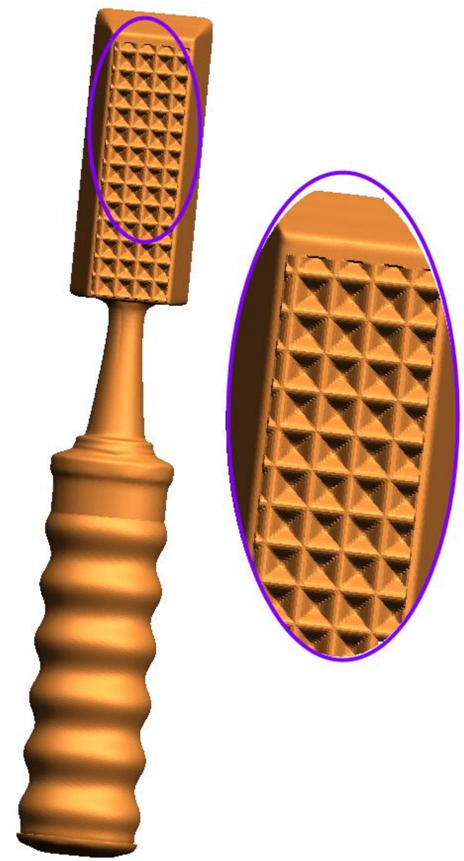


Filigree Model: genus 65, 514K points
Left: Bloomenthal's output, Right: our output

[Varadhan et al. 06]

Implicit Surface Meshing – MPU Implicits





<u>Knot</u>
28K points

<u>Chisel</u>
142K points

Implicit Surface Meshing – MPU Implicits

Vase Lion Model: 200K points

Results - Performance

Model	Bloomenthal	Boissonnat [BO05]	Our Method
Filigree	380	519	241
Chisel	47	79	58
VaseLion	205	13582	234
Chair	1.0	?	10.6
Tanglecube	0.35	17.6	11
Decocube	1.89	?	52
Diagonal	0.93	9.0	9.1

2 GHz Pentium IV, 1 GB RAM, time in secs

Outline

- Surface Extraction
  - Topological guarantee
  - Results
  - Analysis [Varadhan et al. 06]

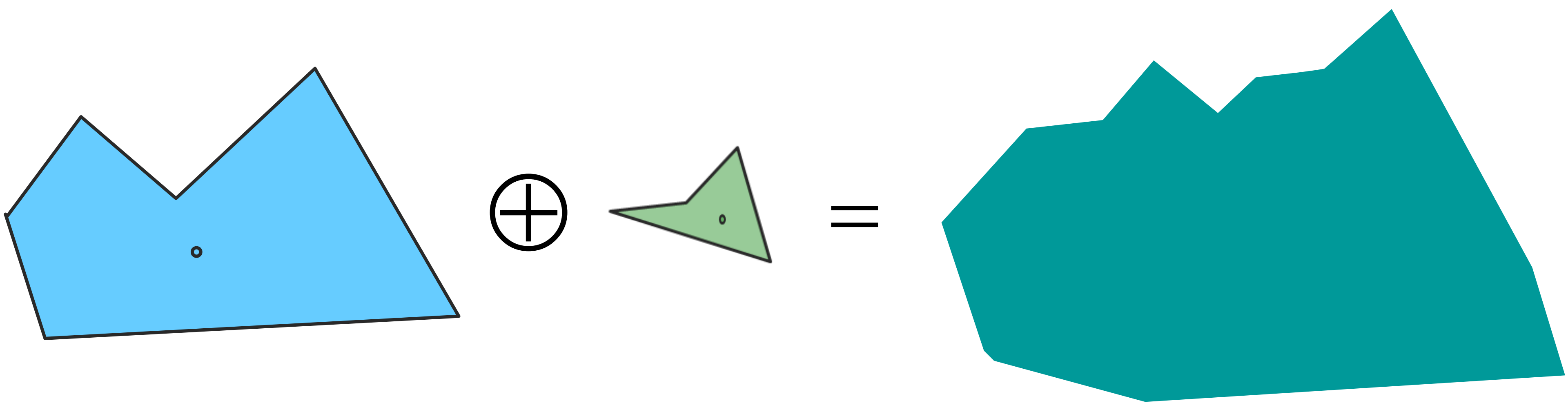
Outline

- Surface Extraction
- **Minkowski sum approximation**
  - Overall approach
  - Speedup techniques
  - Results
- Configuration space approximation
- Motion planning
- Conclusion

Minkowski Sum

- Our goal is to compute the 3D Minkowski sum of general polyhedral models

$$A \oplus B = \{a + b \mid a \in A, b \in B\}$$



Minkowski Sum Computation

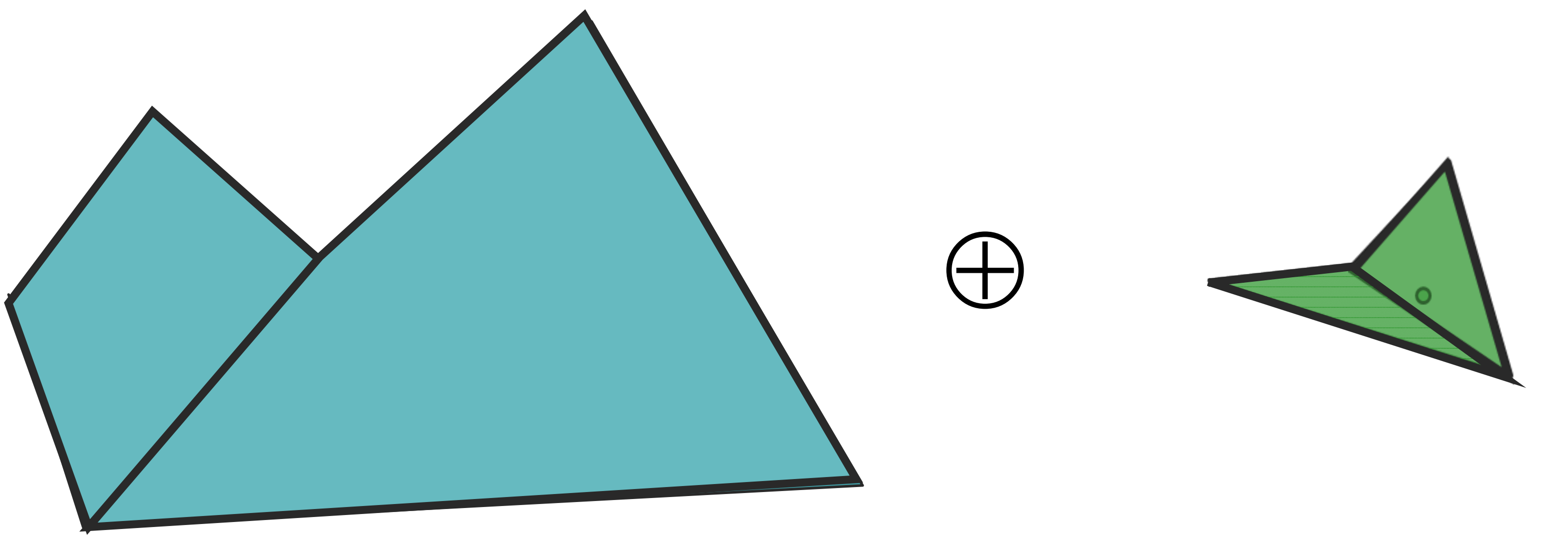
- Difficult to compute exactly
  - $O(n^6)$ - n is complexity of polyhedra

- Most prior methods are restricted to convex objects
  - [Lozano-Perez 83; Guibas & Seidel 87; Kaul & Rossignac 91]

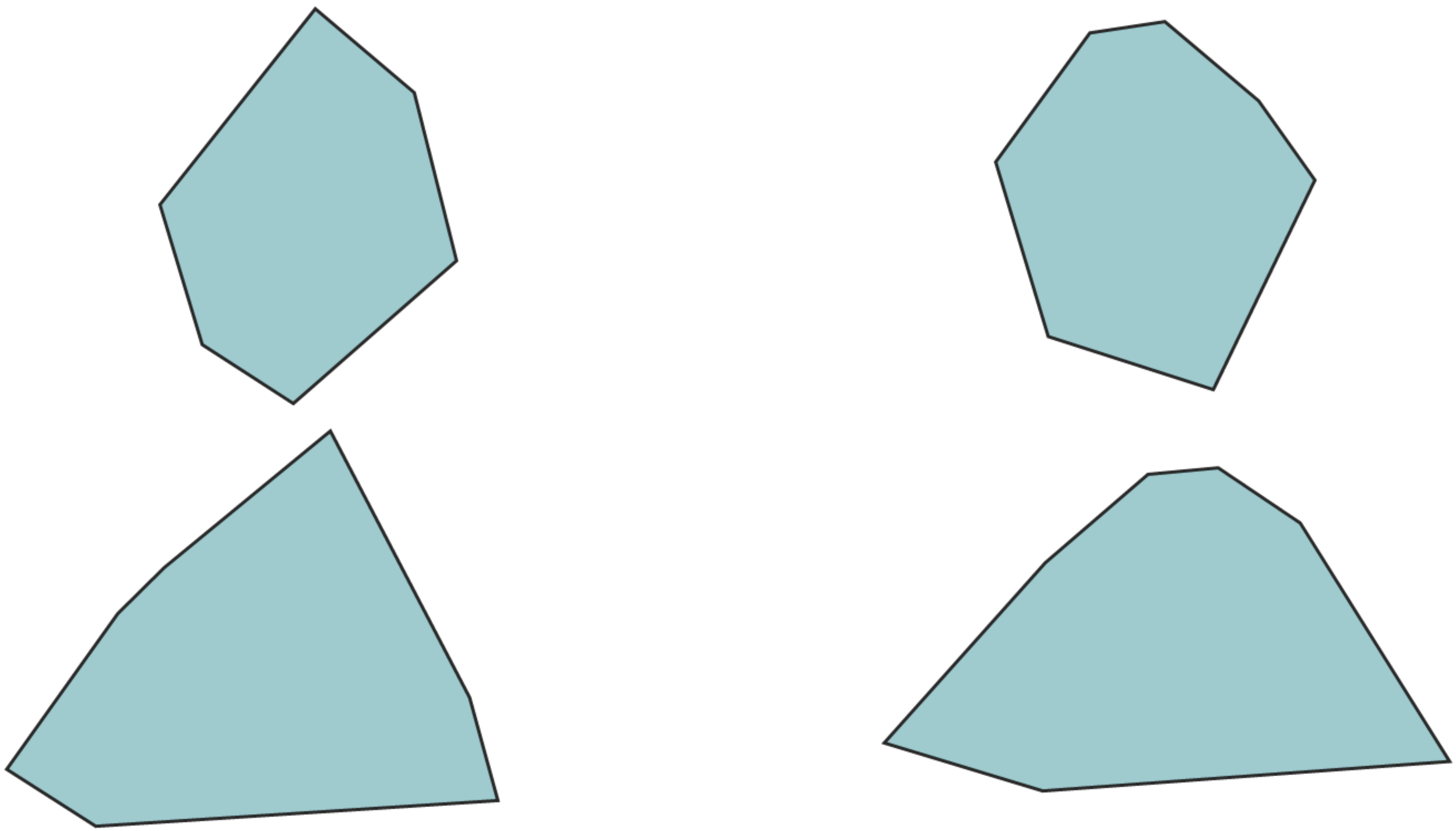
  - Convex case is much easier
    - $O(n^2)$ complexity

Overall Approach

A common approach for non-convex objects is to resort to convex decomposition

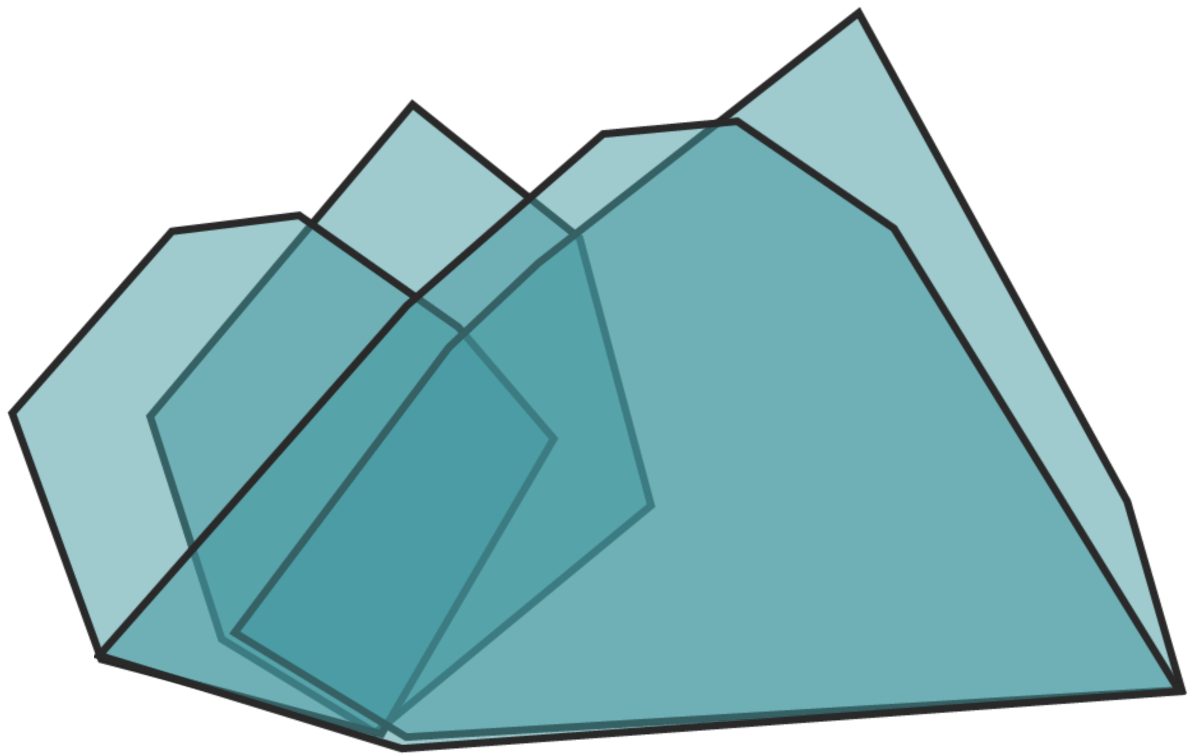


Overall Approach



Overall Approach

Compute union of pairwise Minkowski sums



Overall Approach

Final Minkowski sum

Overall Approach

$$\mathcal{M} = \bigcup_{ij} \mathcal{M}_{ij}$$

- Union computation is a bottleneck
  - Typically, tens of thousands of primitives

Our Approach

- Approximate the boundary of $\mathcal{M}$ using surface extraction algorithm
  - Geometric and topological guarantees hold

Outline

- Surface Extraction
- Minkowski sum approximation
  - Overall approach
  - **Speedup techniques**
  - Results
- Configuration space approximation
- Motion planning
- Conclusion

Speedup Techniques

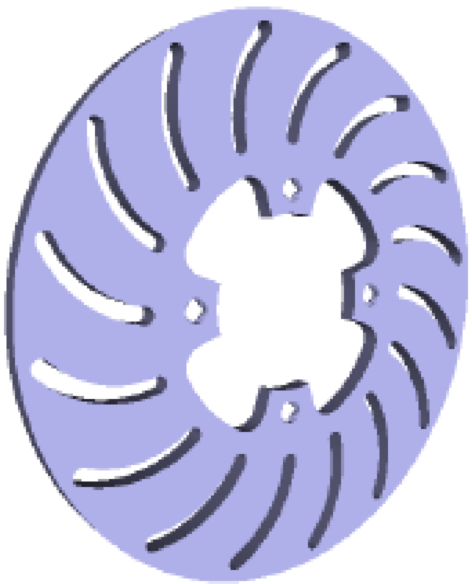
- Two speedup techniques:
  - Cell culling
    - Disregard cells that do not intersect boundary of $\mathcal{M}$ (max-norm computation)
  - Primitive culling
    - When performing a test, disregard primitives that do not contribute to the final answer

  [Varadhan & Manocha'05]

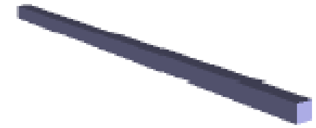
Outline

- Surface Extraction
- Minkowski sum approximation
  - Overall approach
  - Speedup techniques
  - **Results**
- Configuration space approximation
- Motion planning
- Conclusion

Brake Hub
(4,736 tris)



⊕



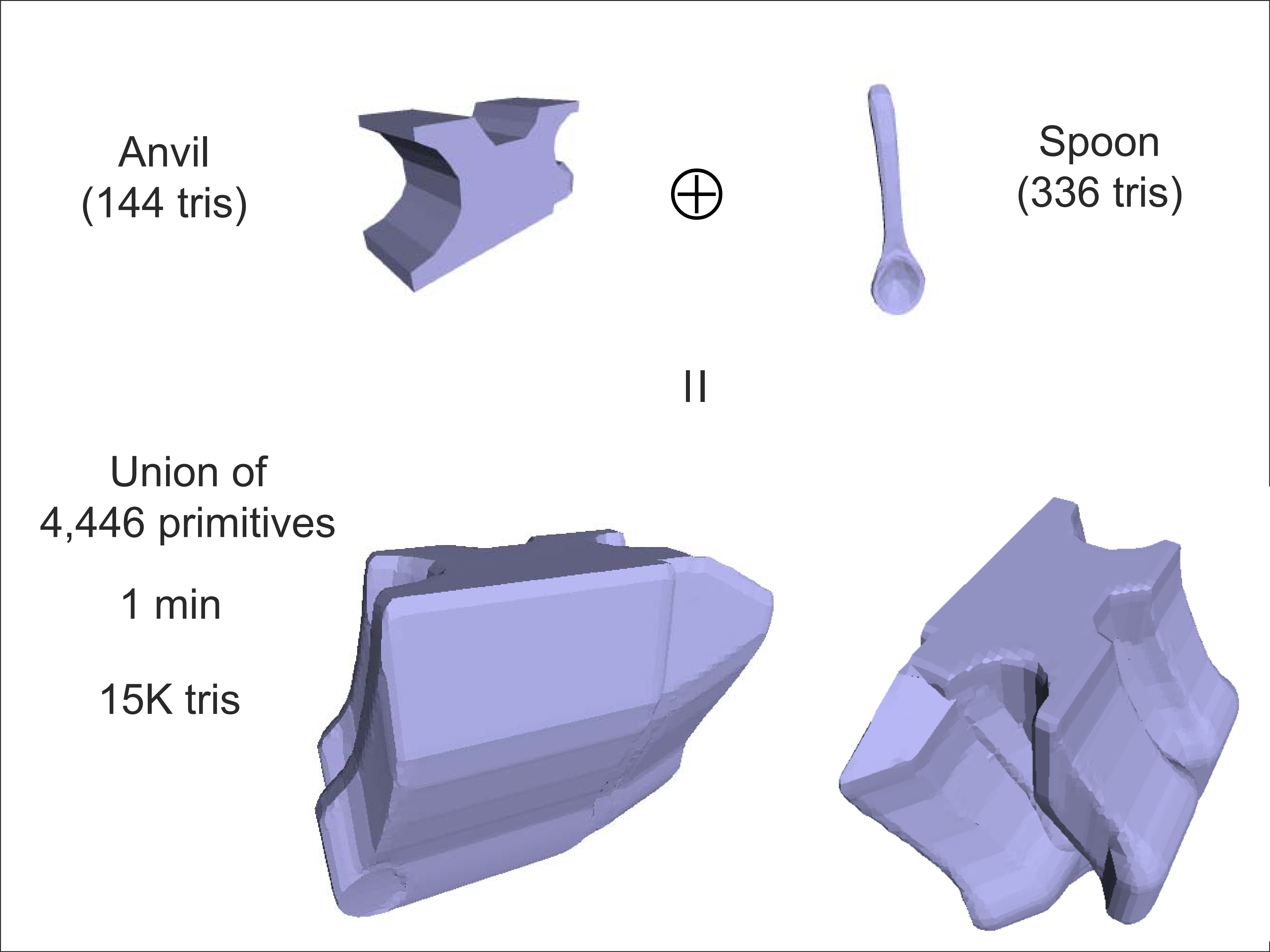
Rod
(24 tris)

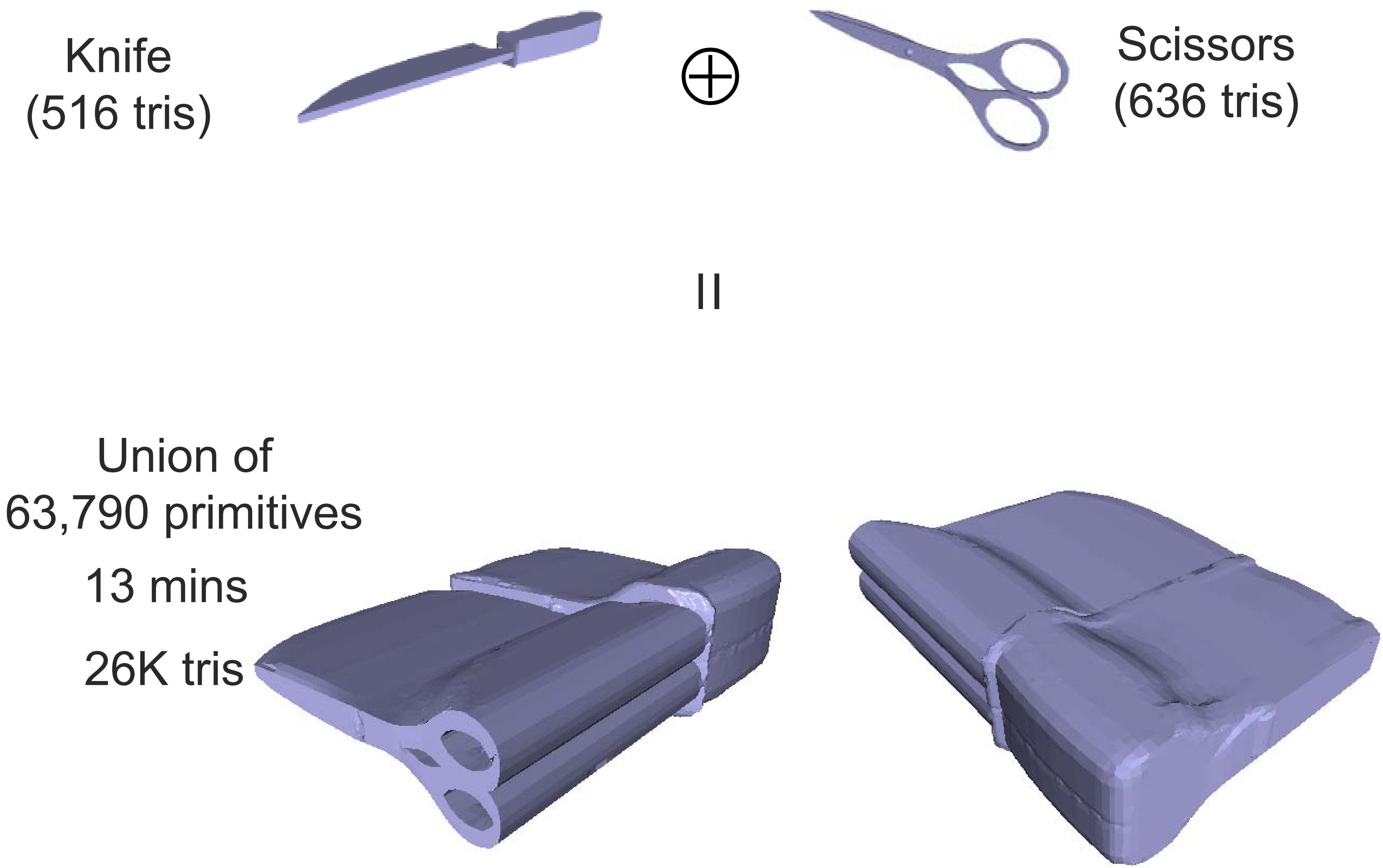
=

Union of
1,777 primitives

2 mins

45K tris


Anvil
(144 tris)
⊕
Spoon
(336 tris)
=
Union of
4,446 primitives
1 min
15K tris

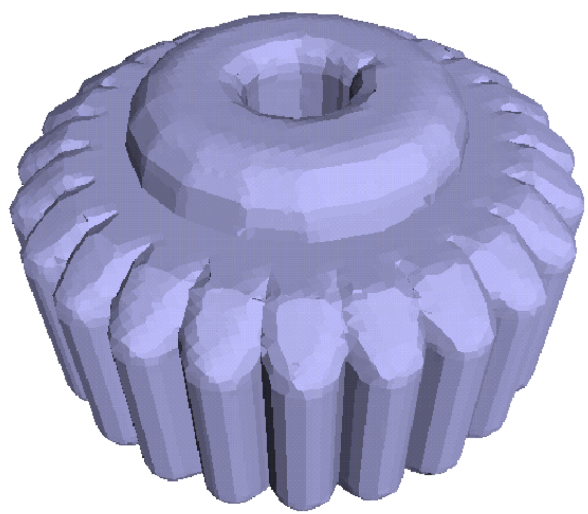

Knife
(516 tris)
⊕
Scissors
(636 tris)
=
Union of
63,790 primitives
13 mins
26K tris

Offsetting

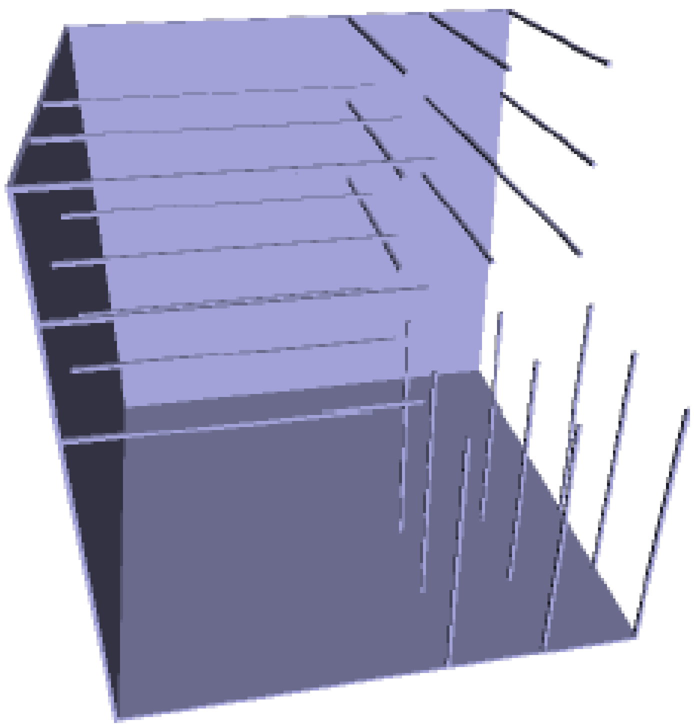
Cup
(1,000 tris)

Cup Offset
33 secs
14K tris

Gear
2,382 tris)

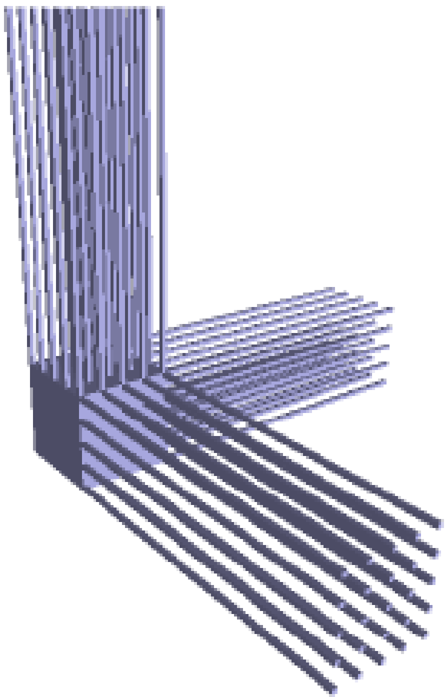


Gear Offset
84 secs
22K tris

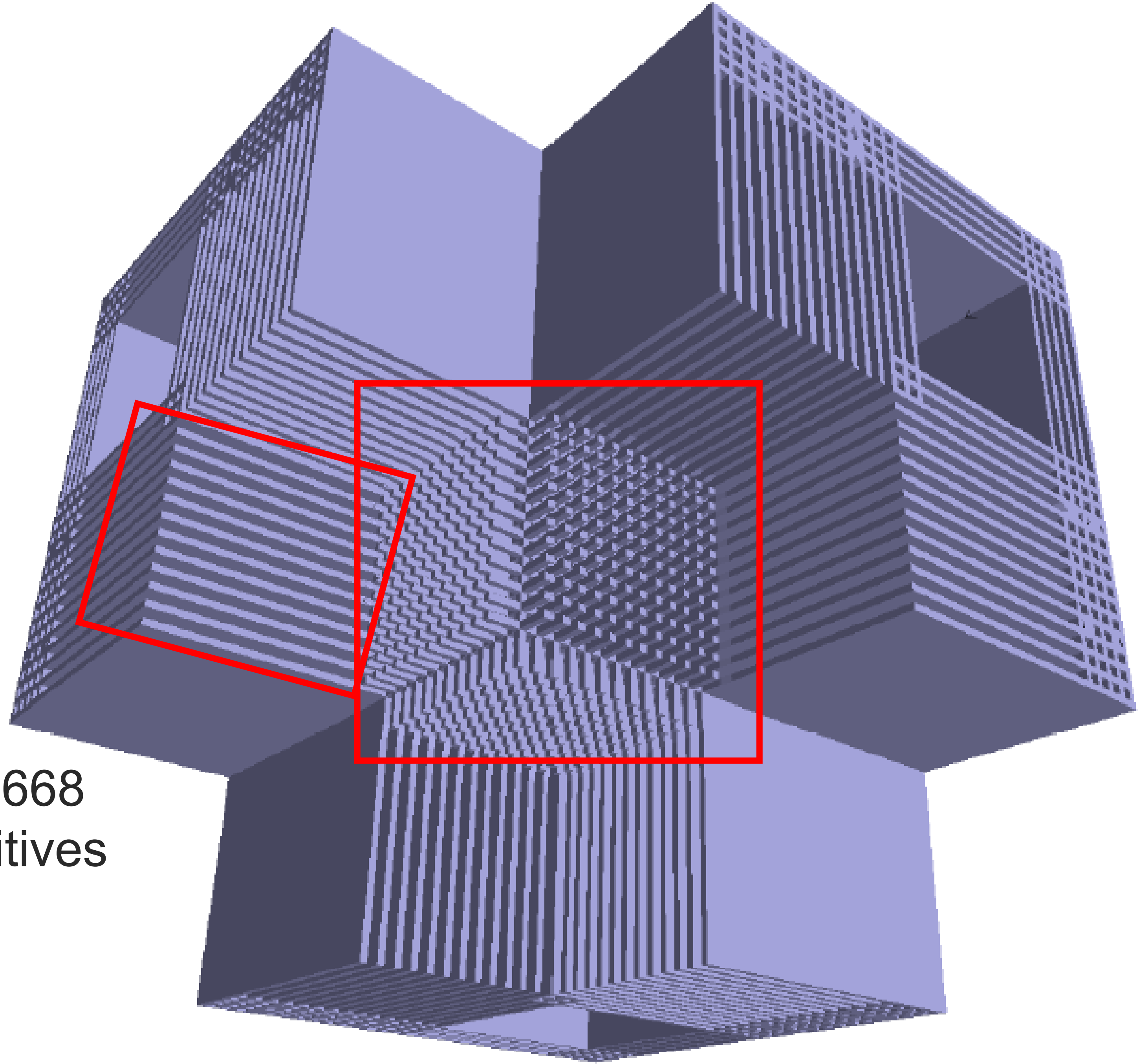


444 tris

⊕



1,134 tris



Union of 66,668 convex primitives

52 mins

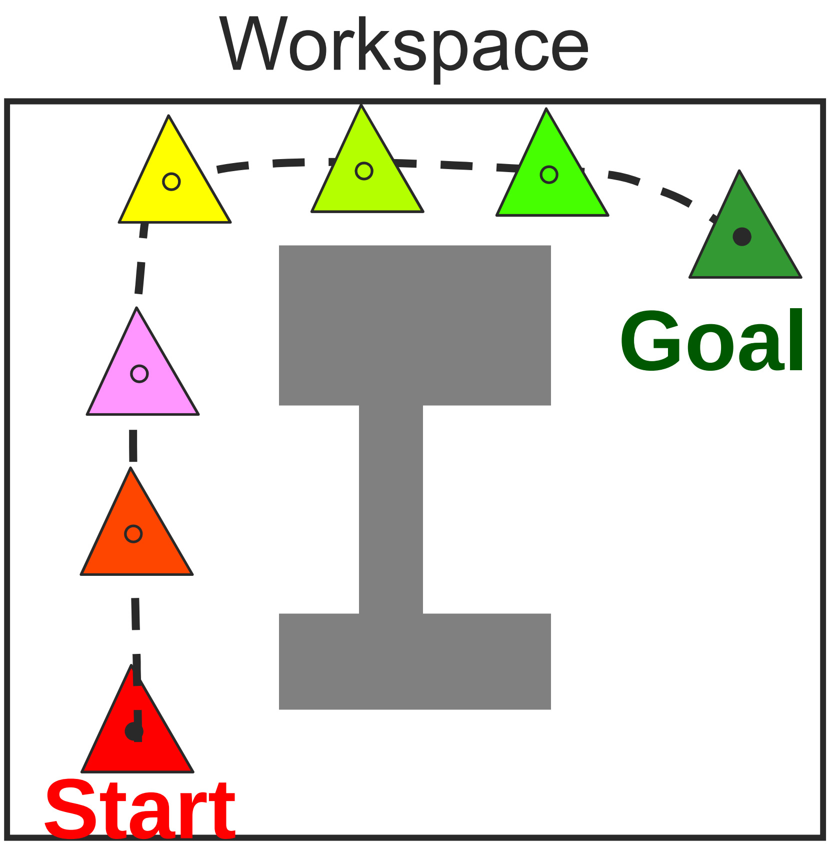
358K tris

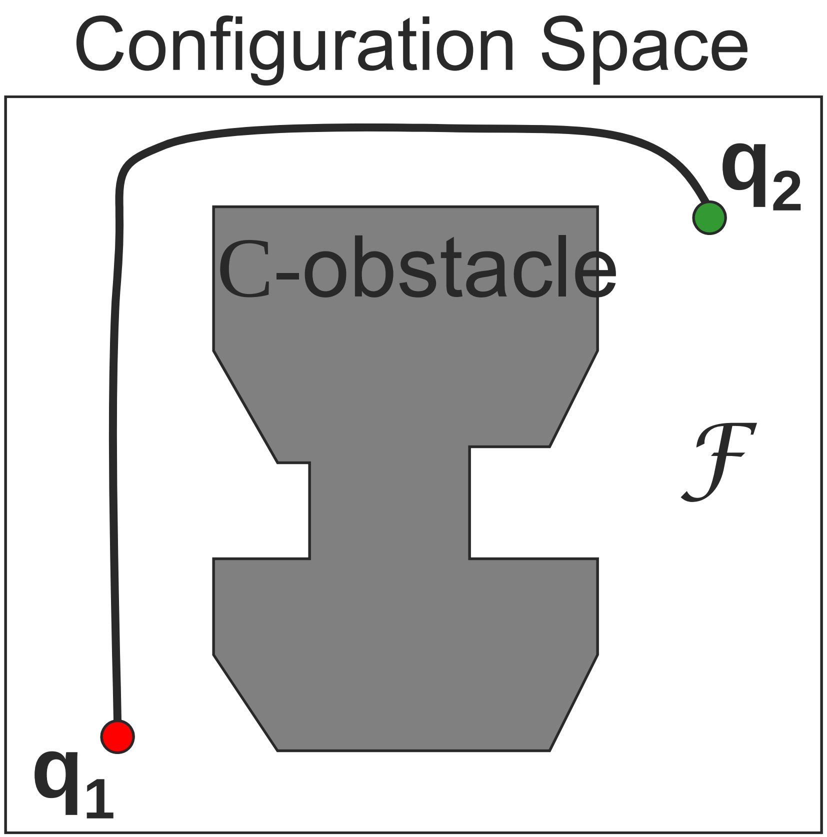
Outline

- Surface Extraction
- Minkowski sum approximation
- **Configuration space approximation**
  - Overall approach
  - Approximate Algorithm
  - Results
- Motion planning
- Conclusion

Configuration Space Computation

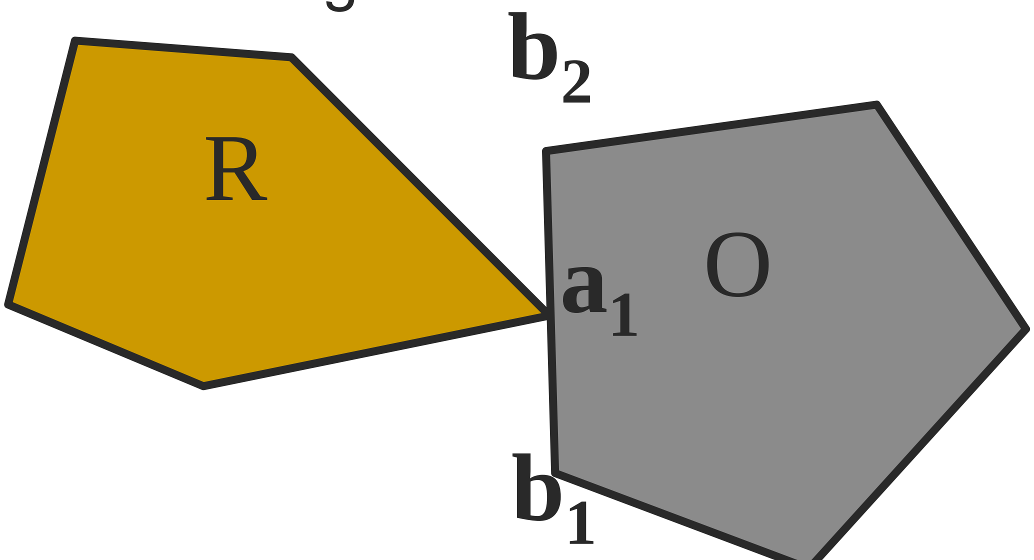
Goal is to compute the free space $\mathcal{F}$





Overall Approach

- Free space $\mathcal{F}$ is represented using
  - *Contact surfaces* (C-surfaces)
  - A C-surface arises from a contact between features of the robot and the obstacle
    - Portion of an algebraic surface



Outline

- Surface Extraction
- Minkowski sum approximation
- Configuration space approximation
  - Overall approach
  - **Approximate algorithm**
  - Results
- Motion planning
- Conclusion

Approximate Algorithm

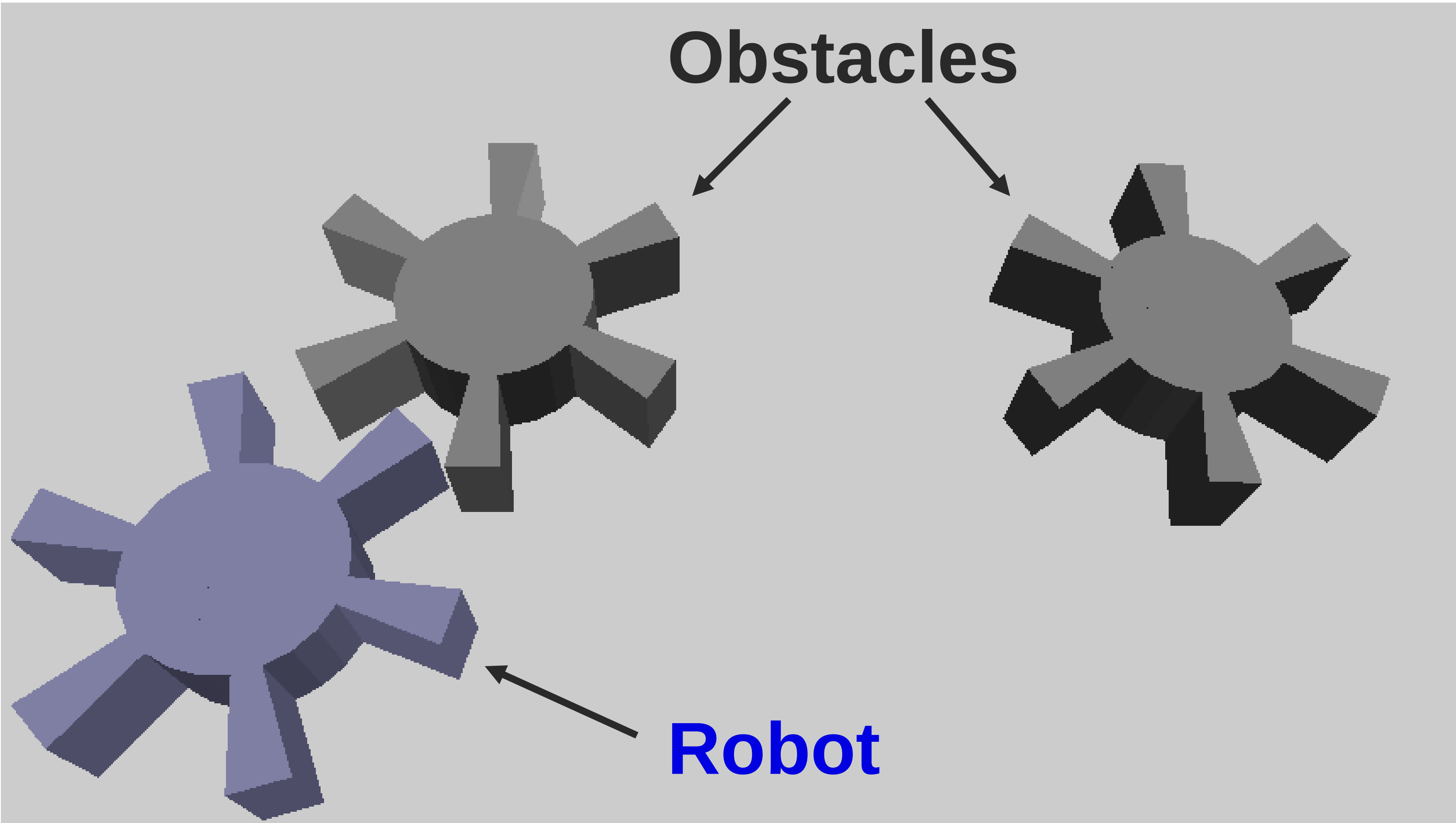
- C-surface condition
  - Allows us to perform both simple cell and star-shaped tests
  - Can be verified using linear programming and interval arithmeticV
- Rest of the surface extraction algorithm remains identical
  - Produces an approximation to boundary of $\mathcal{F}$
  - Geometric and topological guarantees hold

[Varadhan et al. 2006]

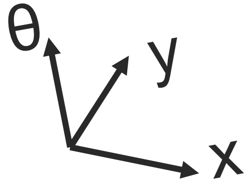
Outline

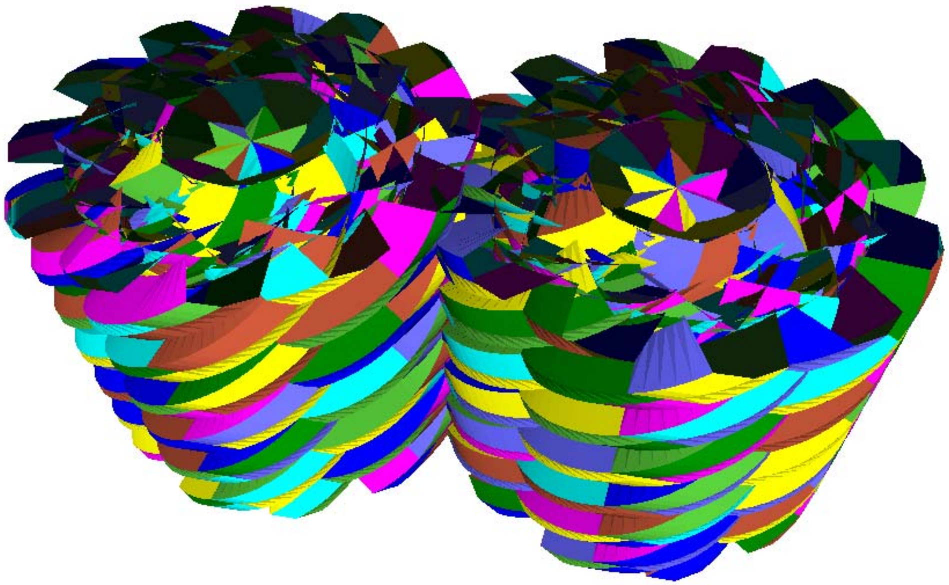
- Surface Extraction
- Minkowski sum approximation
- Configuration space approximation
  - Overall approach
  - Approximate algorithm
  - **Results**
- Motion planning
- Conclusion

2D Translation and Rotation

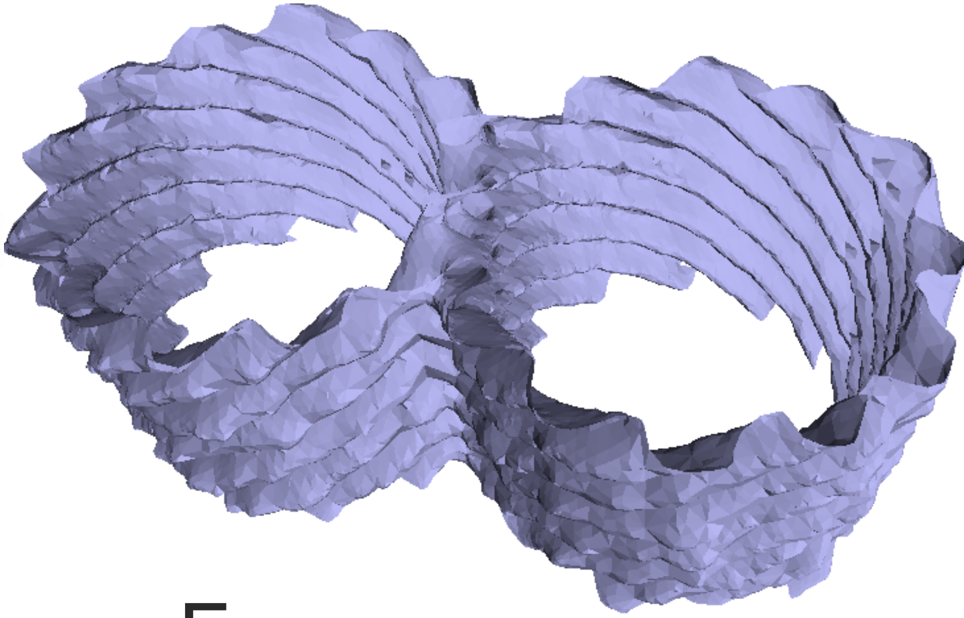


Free Space Approximation



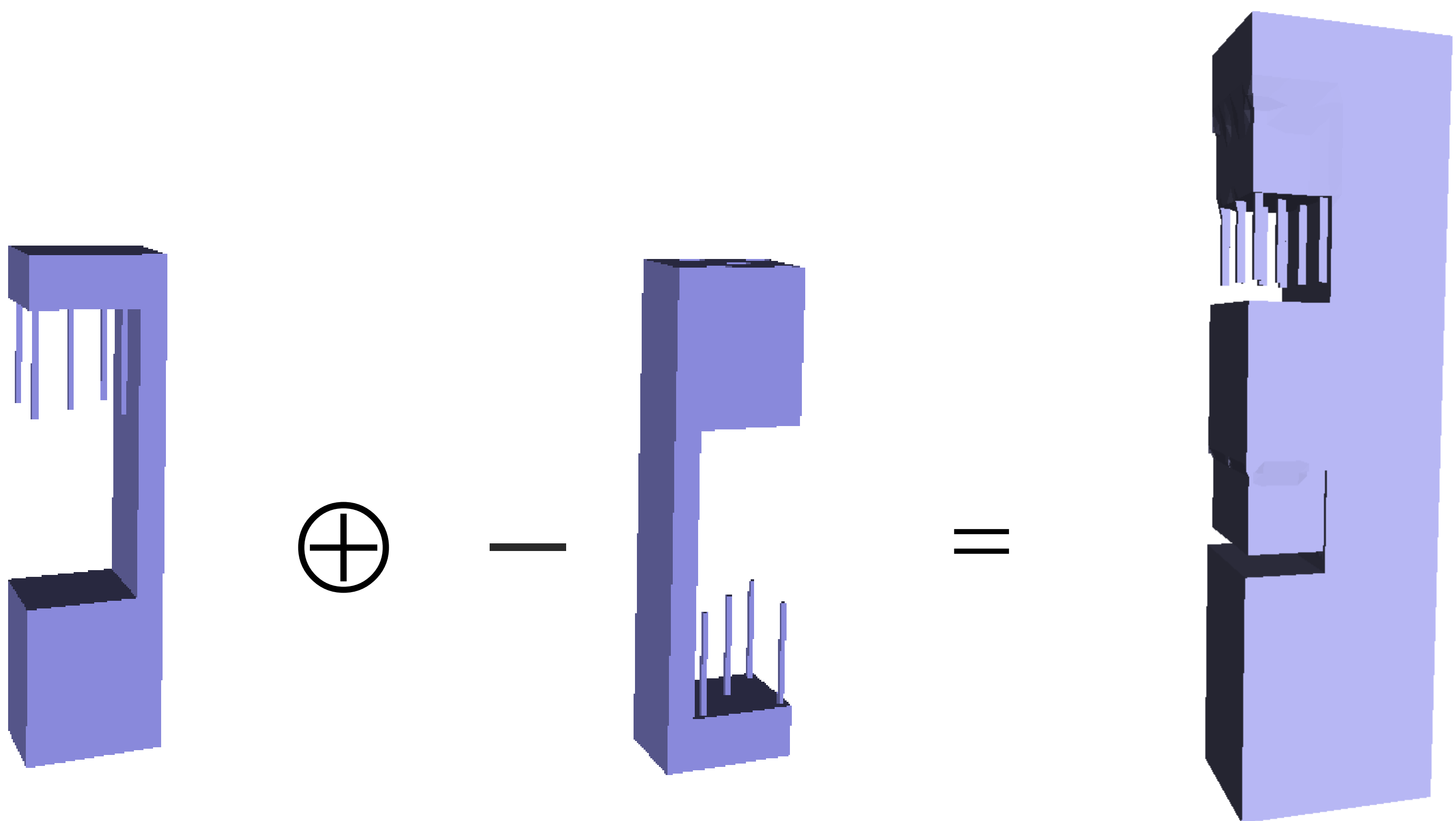


3,929
contact surfaces



Free space
approximation

3D Translation



Obstacle $\mathcal{O}$

Robot $\mathcal{R}$

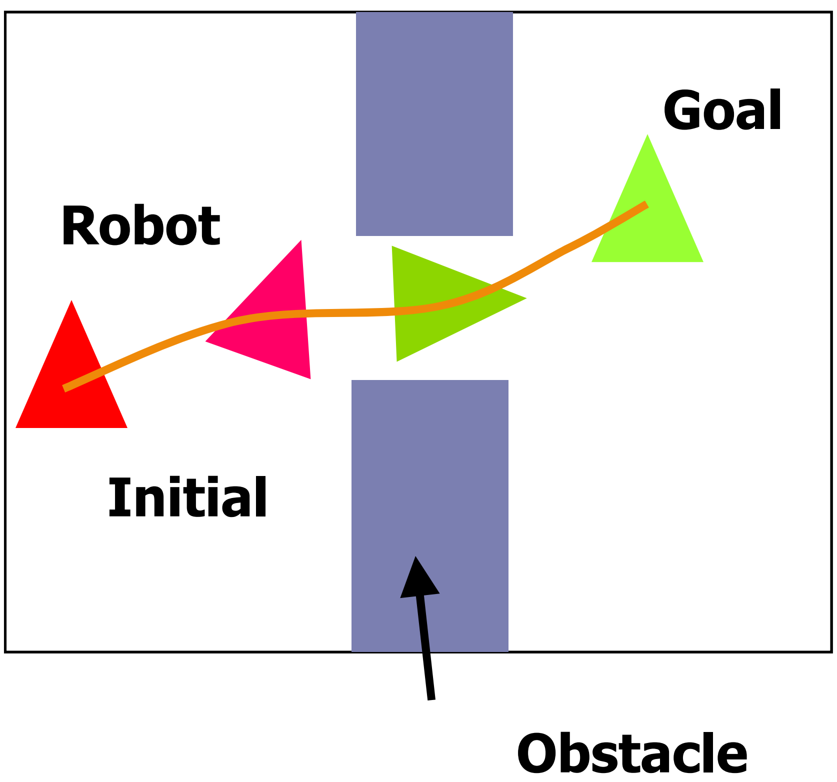
$\mathcal{O} \oplus -\mathcal{R}$

Outline

- Surface Extraction
- Minkowski sum approximation
- Configuration space approximation
- **Motion planning**
- Conclusion

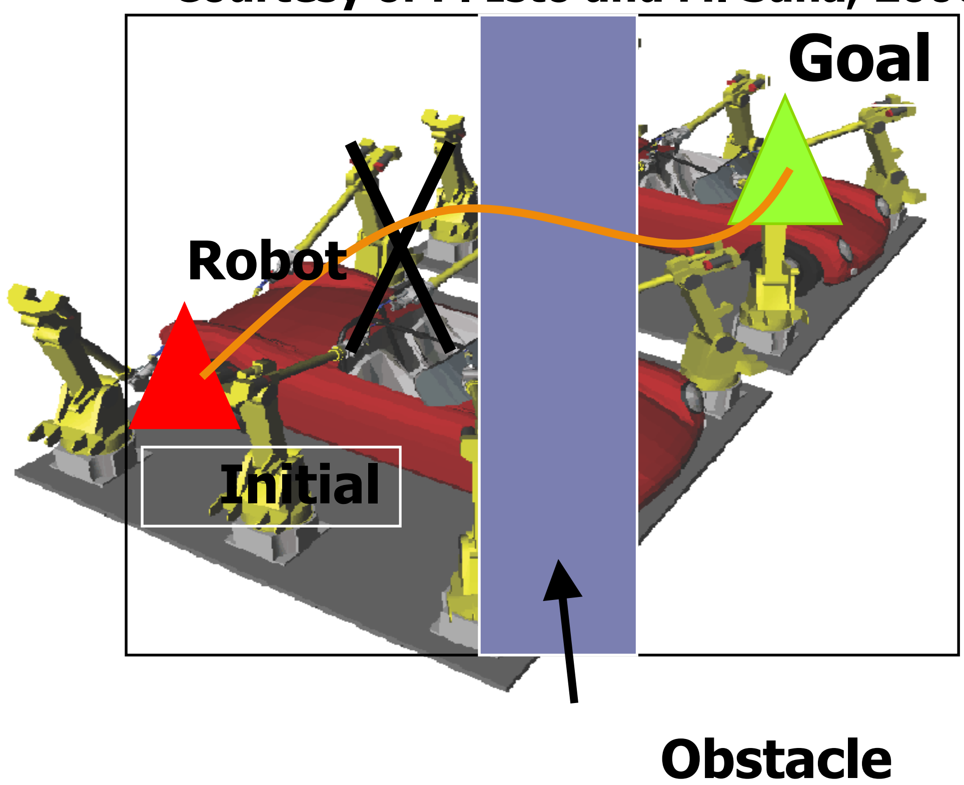
Motion Planning

To find a path



To report no path **72 DOF**

Courtesy of P. Isto and M. Saha, 2006



History of Motion Planning

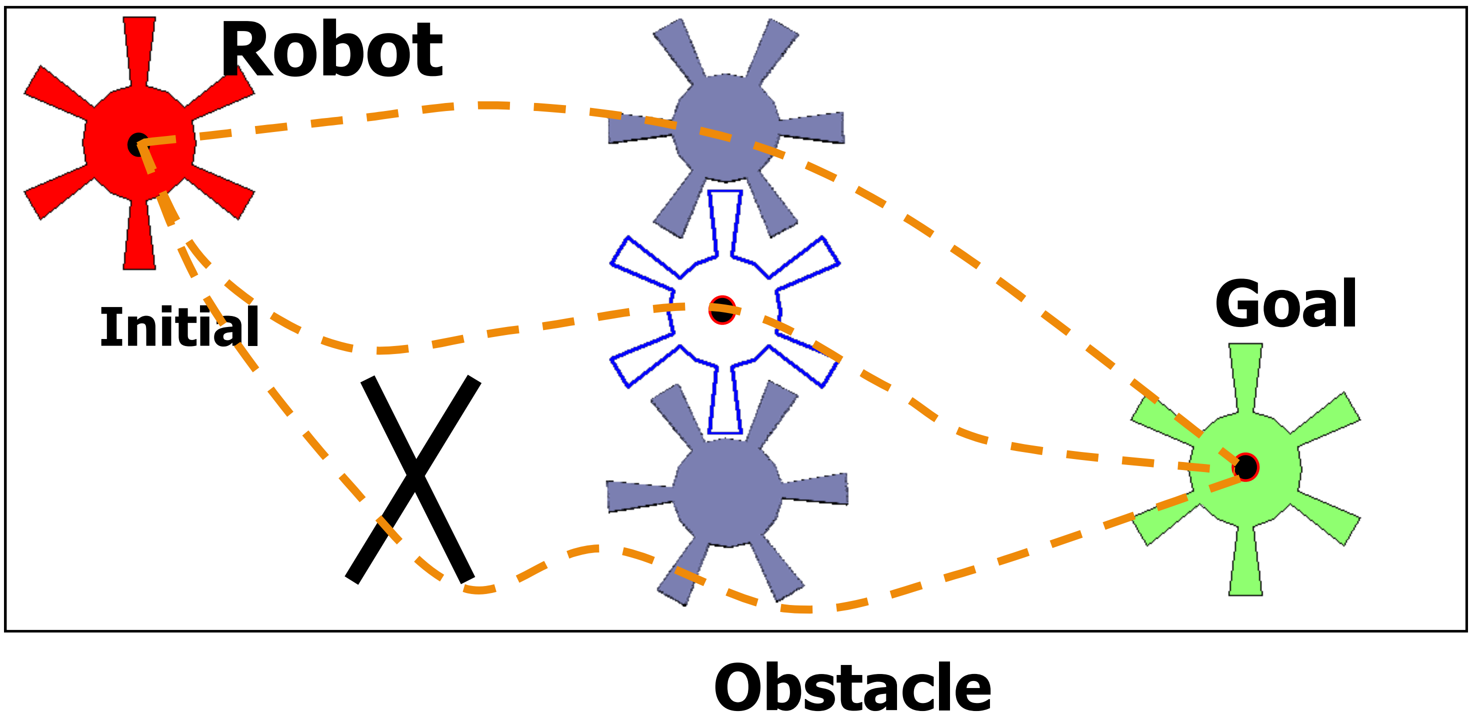
- Studied for more than 4 decades
  - 1970-80s: Great interest from theoreticians; complexity results
  - 1990 onwards: practical and heuristic approaches
  - Robot algorithms [Latombe'99]

Why Complete Motion Planning?

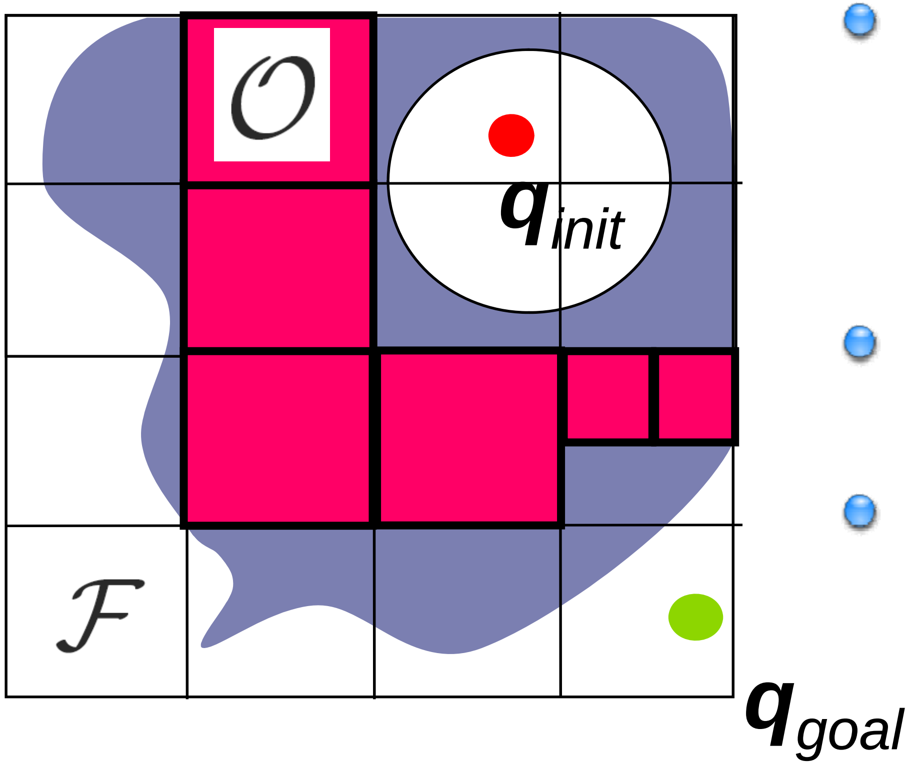
- **Sample-based motion planning**
  - Efficient
  - Work for complex problems with many DOF
  - Difficult for narrow passages
  - May not terminate when no path exists
- **Complete motion planning**
  - Always terminates
  - Not efficient
  - Not robust even for low DOF

Path Non-existence Problem

- Exponential complexity of DOF
- More difficult than finding a path
  - To check all possible paths



C-obstacle Query

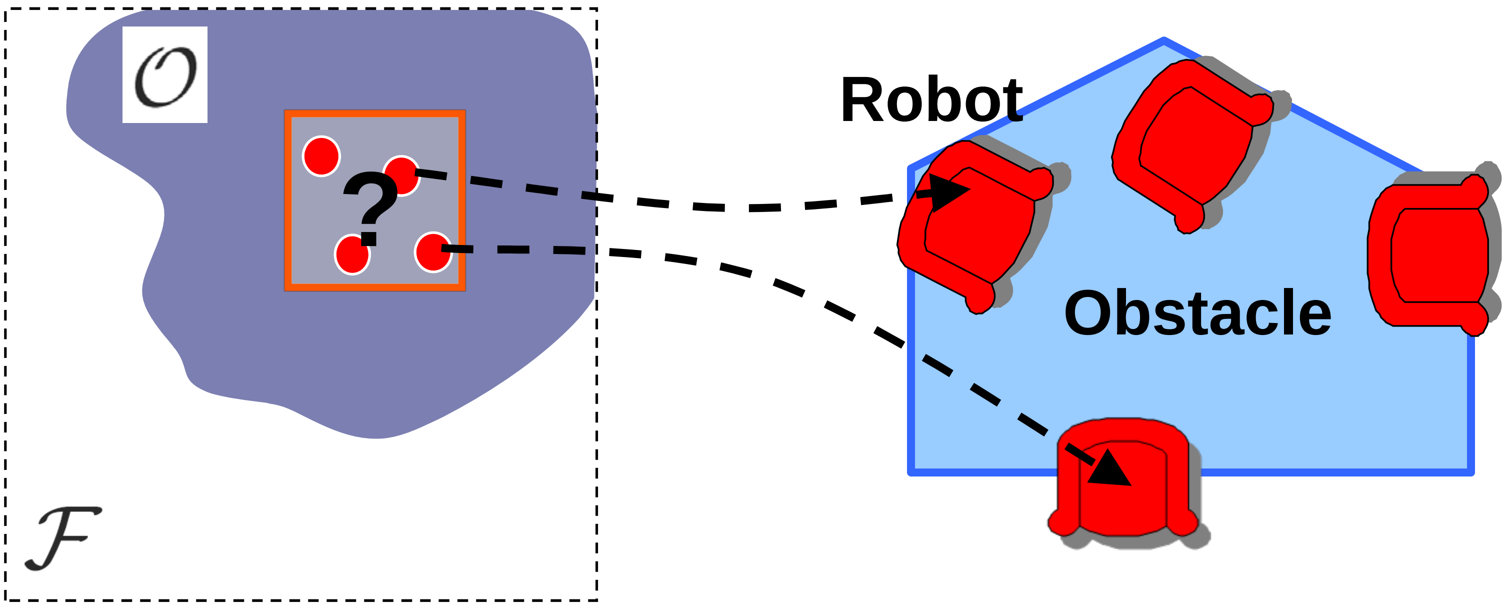


- Whether a primitive lies entirely in C-obstacle?
  - Usually a cell
- Useful for path non-existence
- Uses results from adaptive sampling

[Zhang et al. 2007]

C-obstacle Query: Proximity Query Problem

- Does the cell lie inside C-obstacle?
- Do robot and obstacle intersect at all configurations?



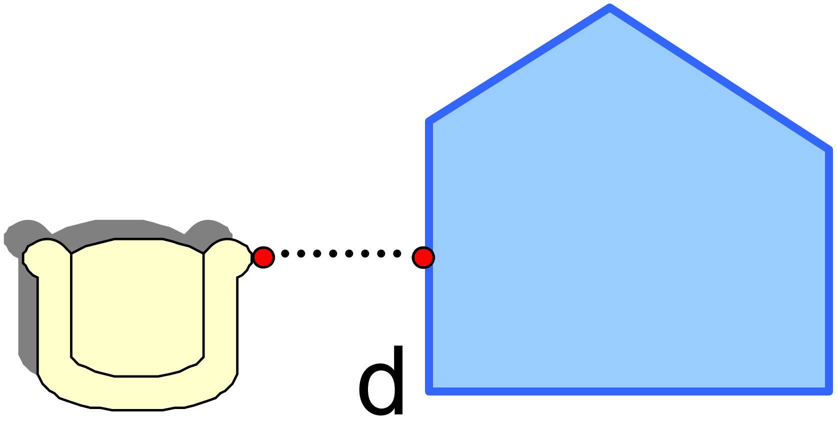
Configuration space

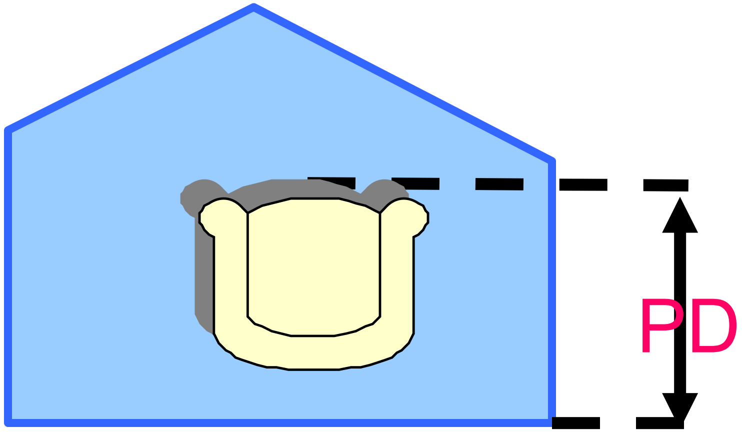
Workspace

Clearance VS 'Forbiddance'

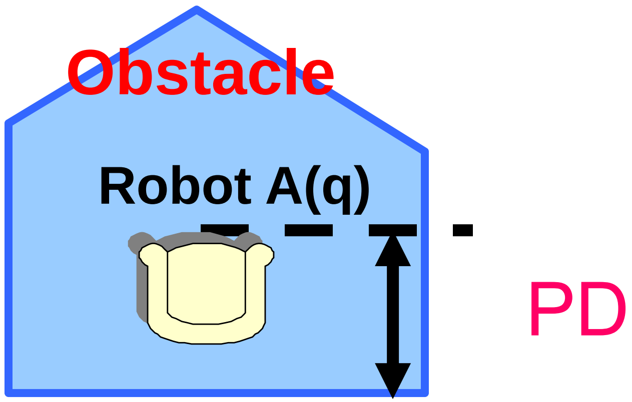
- Separation distance
- Clearance

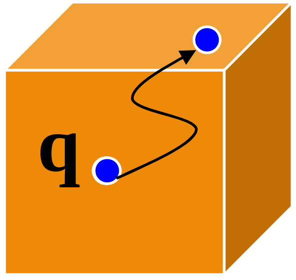
- Penetration Depth
- 'Forbiddance'





C-obstacle Query Algorithm

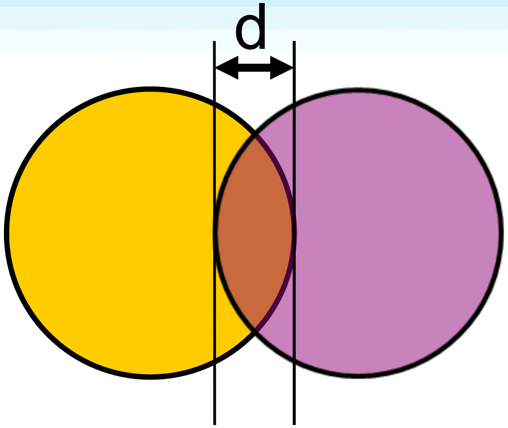


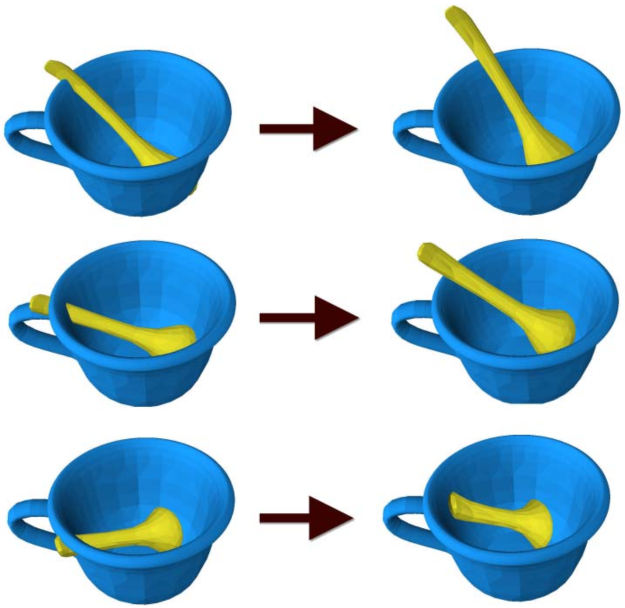


- Penetration Depth
  - Extent of interpenetration between robot and obstacle
- Motion Bound
  - Extent of the motion that robot can make.
- Is Penetration Depth > Motion Bound?

Generalized Penetration Depth

- **Distance measure for interpenetration**
- **PD^g formulation**
  - Generalize from translational penetration depth
  - Take into account both translational and rotational motion
- **PD^g computation**
  - Practical lower bound and upper bound algorithms



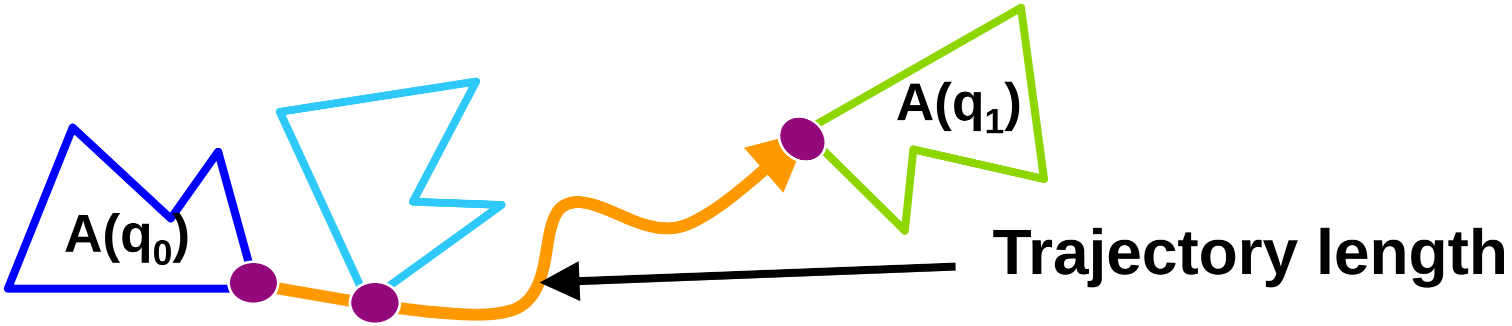


Generalized Penetration Depth: PD^g

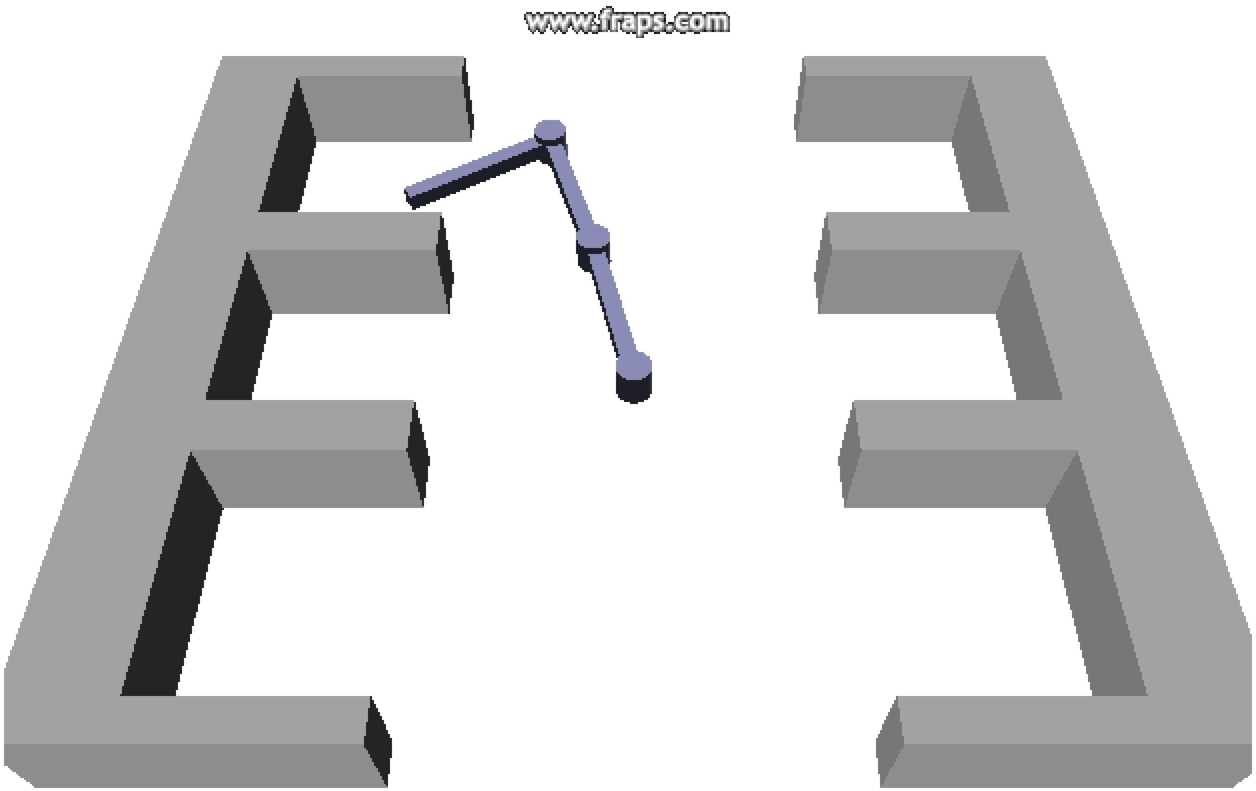
- Take into account translational and rotational motion

  [Zhang et al.'06; Zhang et al.'07]

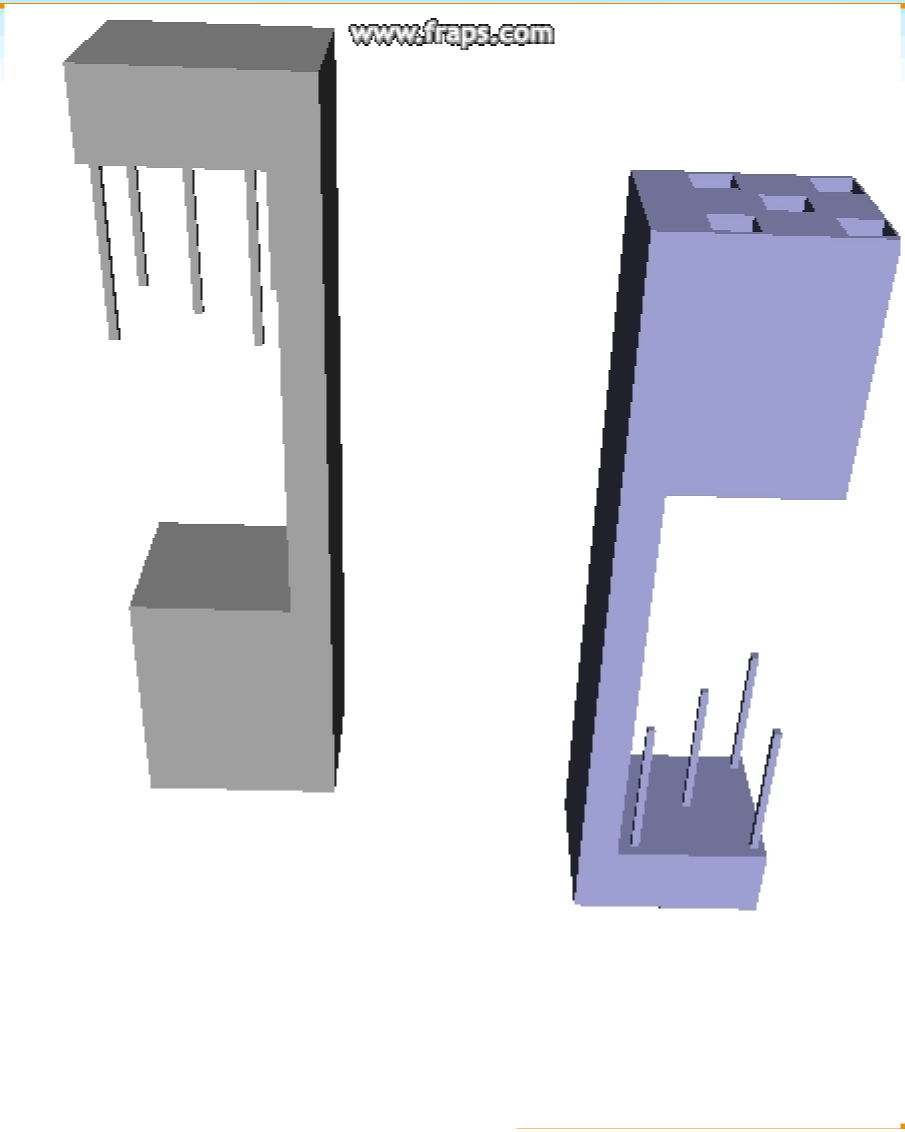
  - Trajectory length
  - *Distance Metrics: DISP metric*
  - Min/Max operations



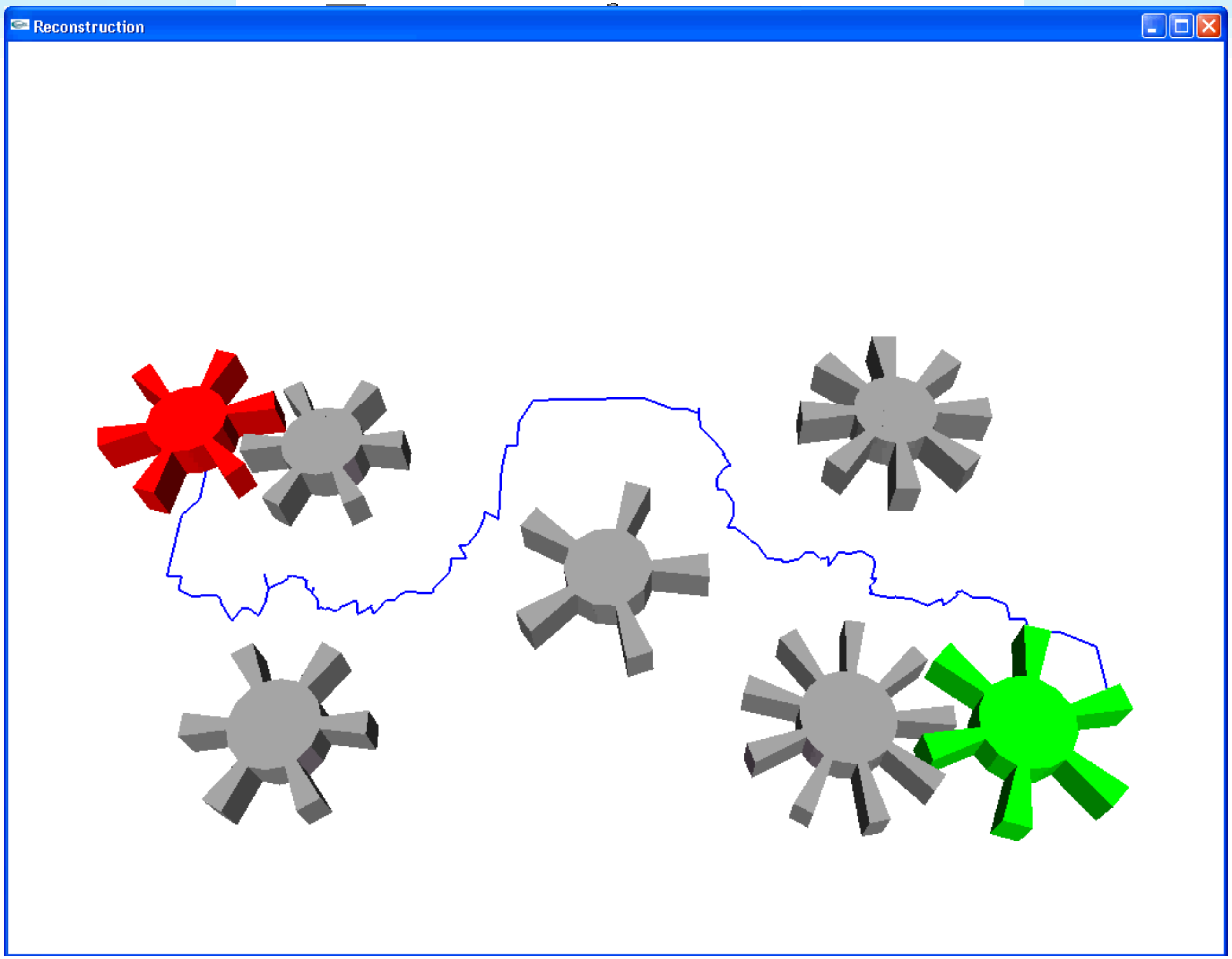
Complete Motion Planning



Complete Motion Planning



Complete Motion Planning



Hybrid Planning

- **Sample-based Motion Planning**
  - \+ Efficient
  - \+ Many DOFs
  - \- Narrow passages
  - \- Path non-existence
- **Complete Motion Planning**
  - \+ Complete
  - \- Not efficient

Can we combine them together?

Results of Hybrid Planning

A Hybrid Approach for Complete Motion Planning

Liang-Jun Zhang, Young K. Jim*, Dinesh Manocha

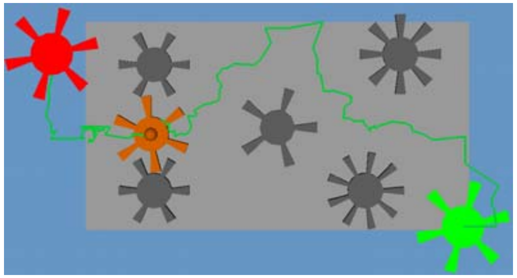
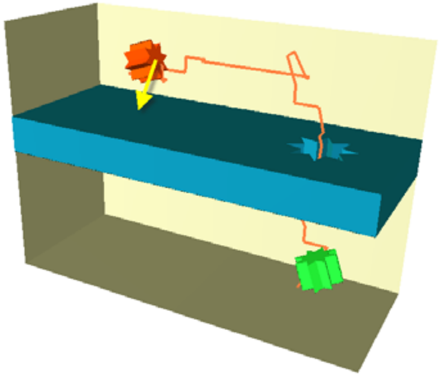
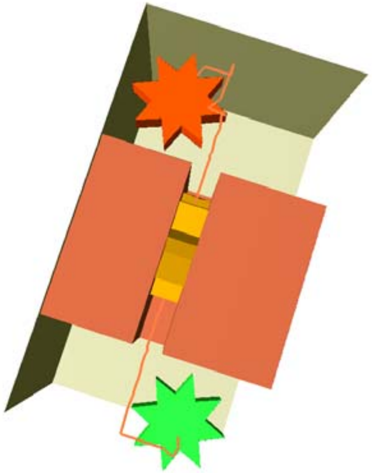
University of North Carolina at Chapel Hill

*EWHA University, Korea

http://gamma.cs.unc.edu/NOPATH

Results of Hybrid Planning

- 2.5 - 10 times speedup

	3 DOF 		4 DOF 		4 DOF 	
	timing	cells #	timing	cells #	timing	cells #
Hybrid	34s	50K	16s	48K	102s	164K
ACD	85s	168K	?	?	?	?
Speedup	2.5	3.3	≥10	?	≥10	?

Outline

- Surface Extraction
- Minkowski sum approximation
- Configuration space approximation
- Motion planning
- **Conclusions**

Main Results

- Surface extraction with geometric and topological guarantees
- Boolean operations, simplification, implicit surface meshing and remeshing
- Minkowski sum approximation
- Configuration space approximation
- Complete motion planning w/o explicit C-space computation

Advantages

- Able to handle challenging scenarios
  - Large number of Boolean operations
  - Minkowski sum of polyhedra with hundreds of triangles
  - Path planning in narrow passages
  - Complete motion planning algorithms

Advantages

- Simplicity
  - Basic components

    Spatial subdivision

    Sign computation

    Distance computation

    Linear programming

    Interval arithmetic

    Marching Cubes

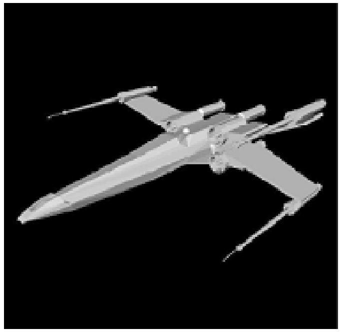
  - Easy to implement
    - Robustness of entire approach depends on these components and some non-degeneracy assumptions

Limitations

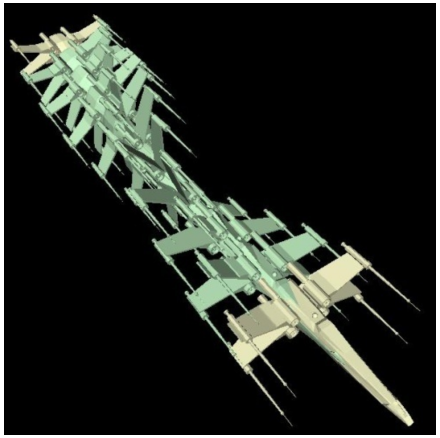
- Spatial subdivision is susceptible to degeneracies
  - May not terminate for tangential intersections
  - Perturbation methods
- Both sampling condition and the tests to verify the sampling condition are conservative
- Analysis
  - Assumes a smooth surface

Future Work

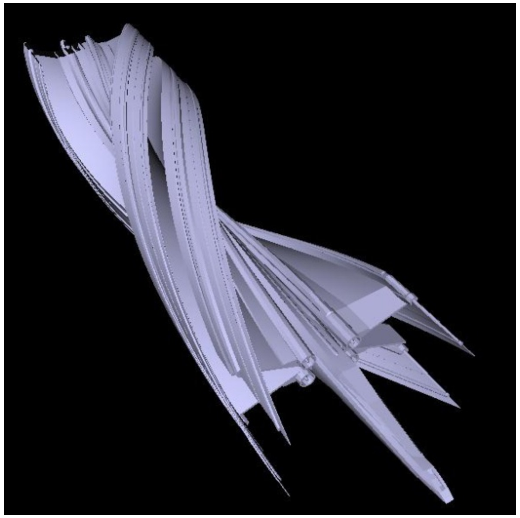
- Other surface extraction problems
  - Swept volume computation



X-wing Model



Trajectory



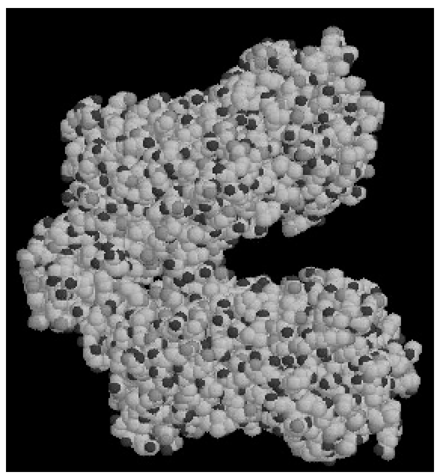
Swept volume

Future Work

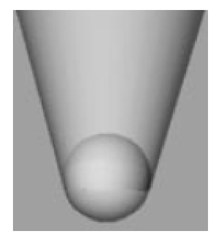
- Minkowski sum

Applications to molecular modeling

Investigate star-shaped decomposition



$\oplus$



Lac Repressor protein model

Model of a AFM microscope tip

Future Work

- Motion Planning
  - Extend to higher DOF robots: 3**T**+3**R**
  - High DOF articulated models

Future Work

- Lots of interesting applications of discrete geometric computations, but provide topological and geometric accuracy bounds!

Acknowledgements

- Army Research Office
- DARPA
- Disruptive Technology Office
- National Science Foundation
- Office of Naval Research
- RDECOM

Recruiting Good Graduate Students

Good graduate students in:

- Computer graphics
- Geometric computing
- Robotics
- Related areas….

Thank you!

dm@cs.unc.edu