SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

CGL

Computational Assemblies for Digital Fabrication

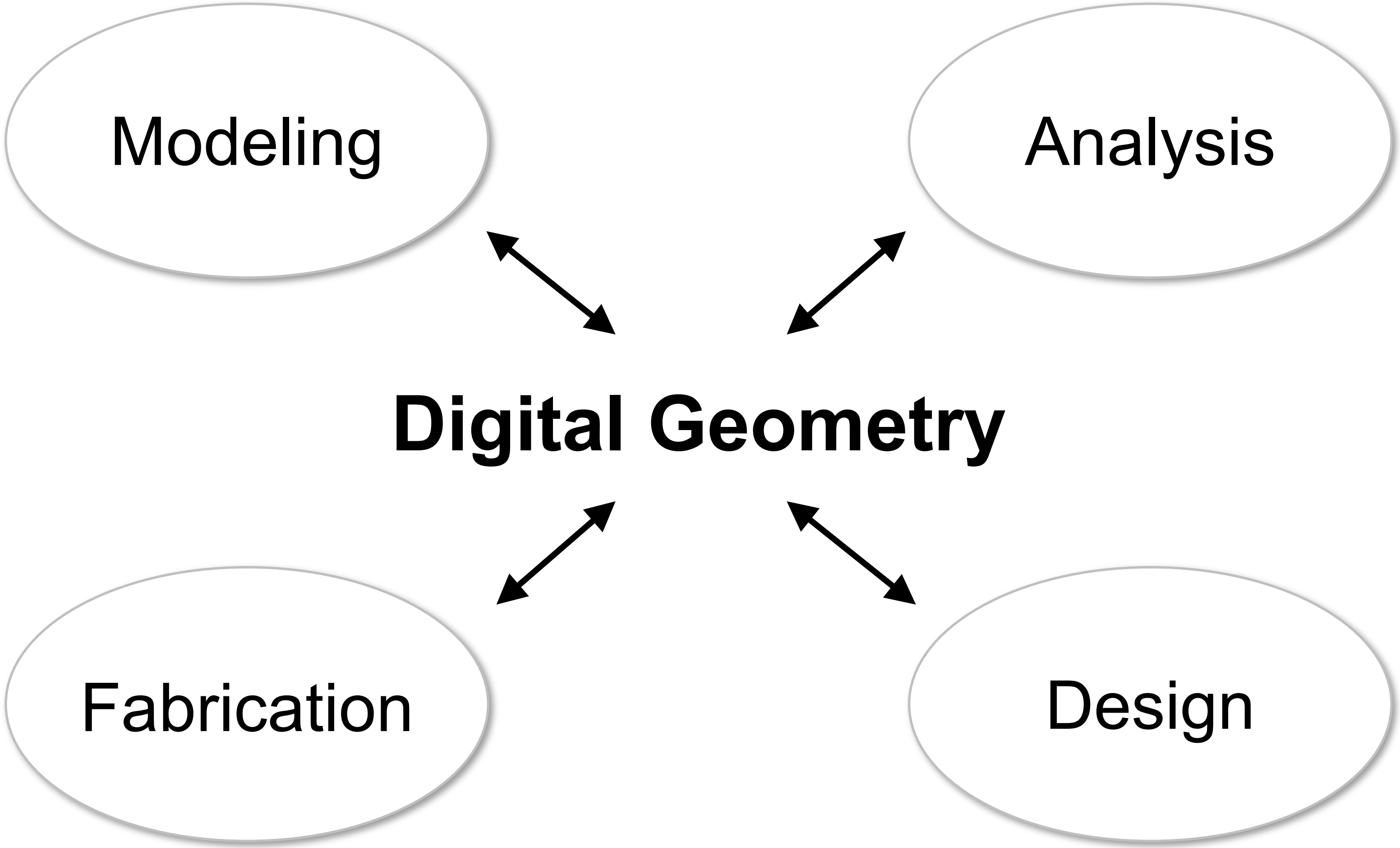
Peng SONG, SUTD

https://sutd-cgl.github.io/

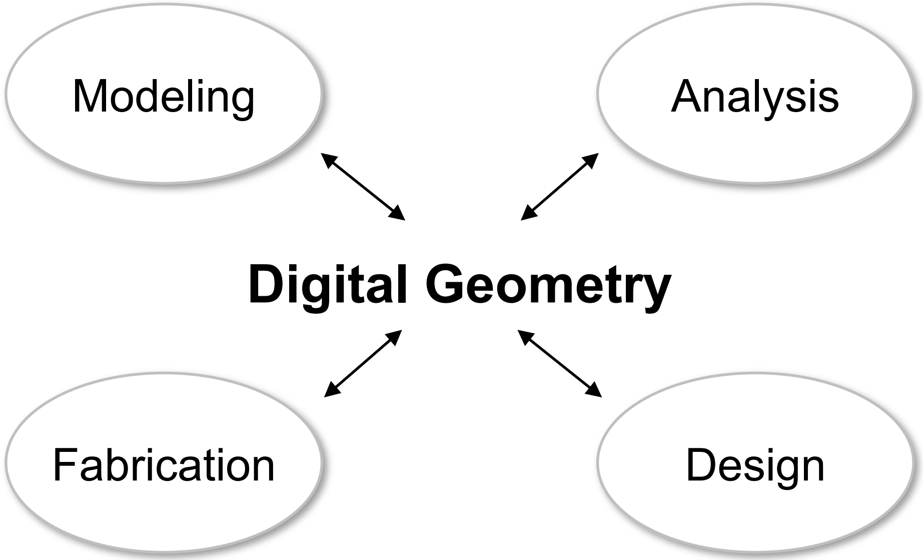
Singapore University of Technology and Design

- **Fourth** public autonomous university in Singapore established in **2009**
- Focus on **design** and **multi-disciplinary** curriculum and research

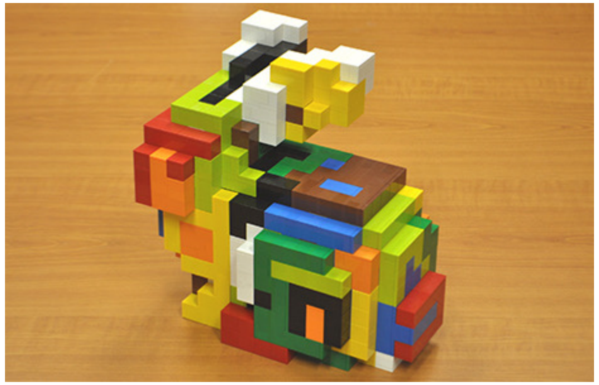
Computer Graphics Group in SUTD



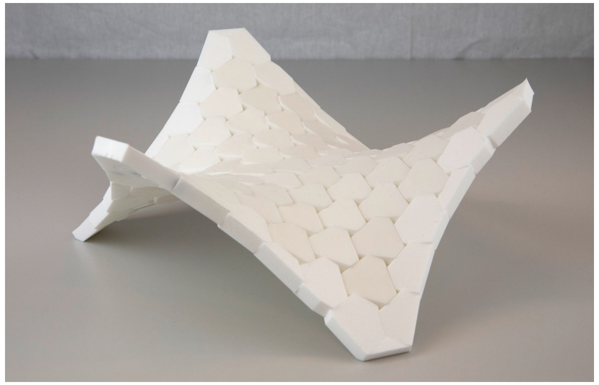
Computer Graphics Group in SUTD



Recreational

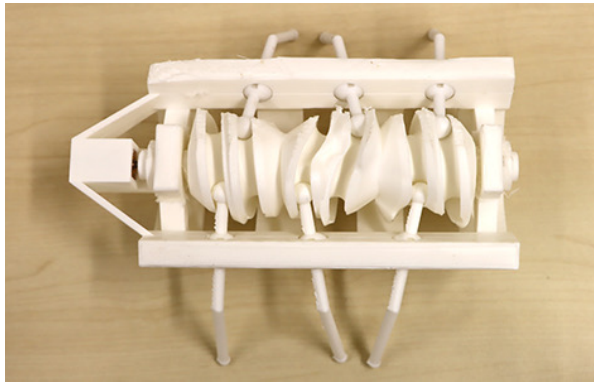


Architectural



Mechanical

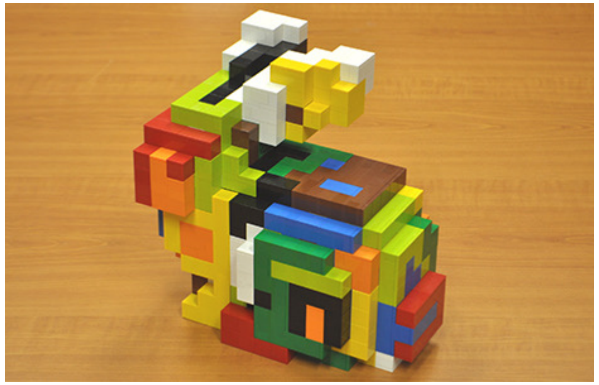
Robotic



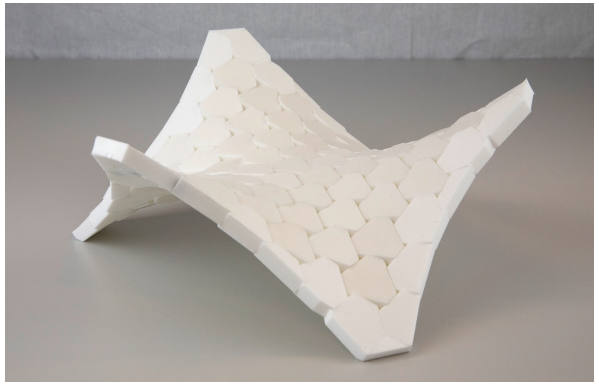
Computer Graphics Group in SUTD

Assemblies

Recreational

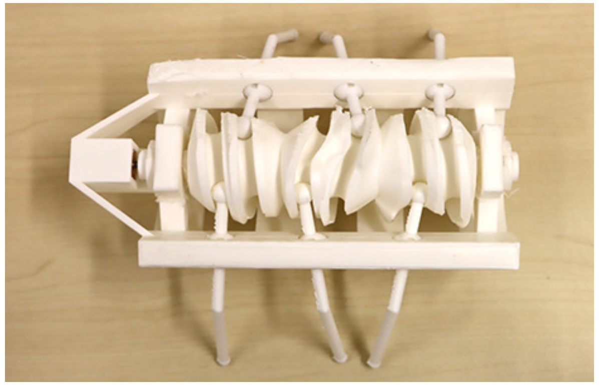


Architectural



Mechanical

Robotic

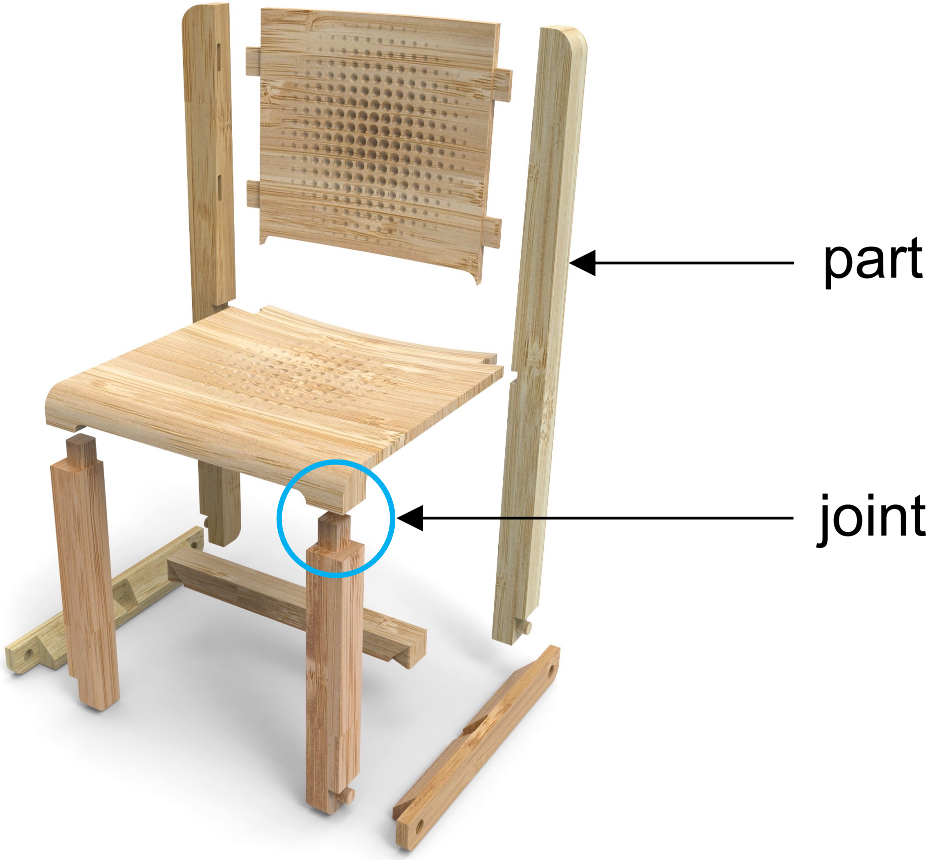


Assemblies

An assembly is an arrangement of parts connected by joints to have a specific form and functionality.

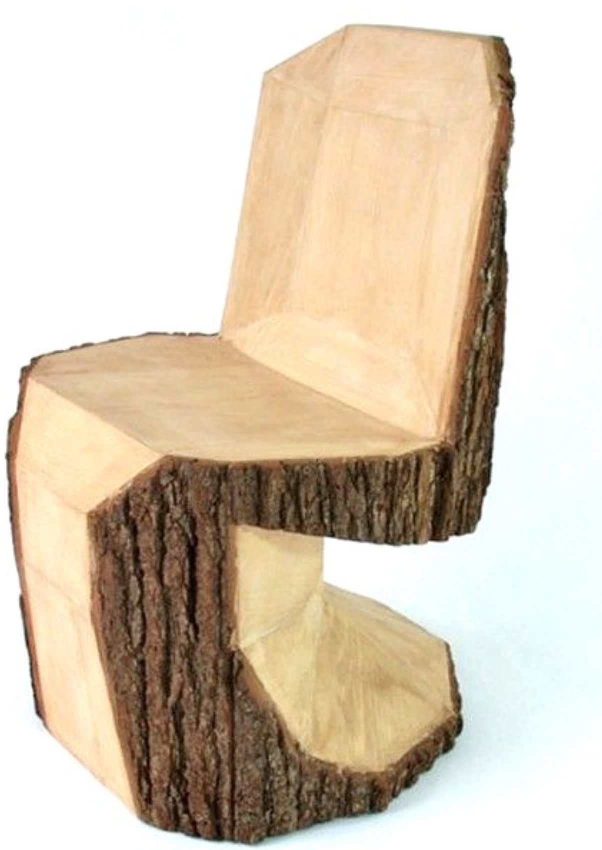
Assemblies

An assembly is an arrangement of parts connected by joints to have a specific form and functionality.



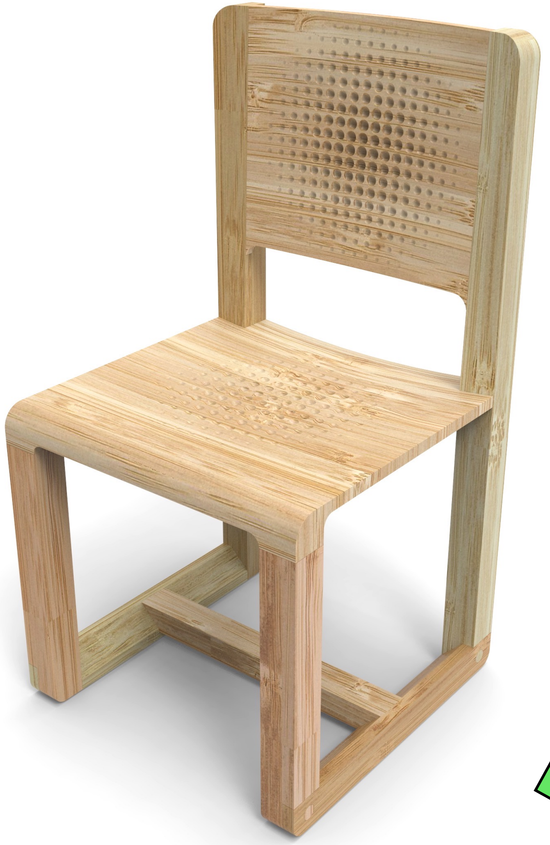
Assemblies

Monolithic object



VS

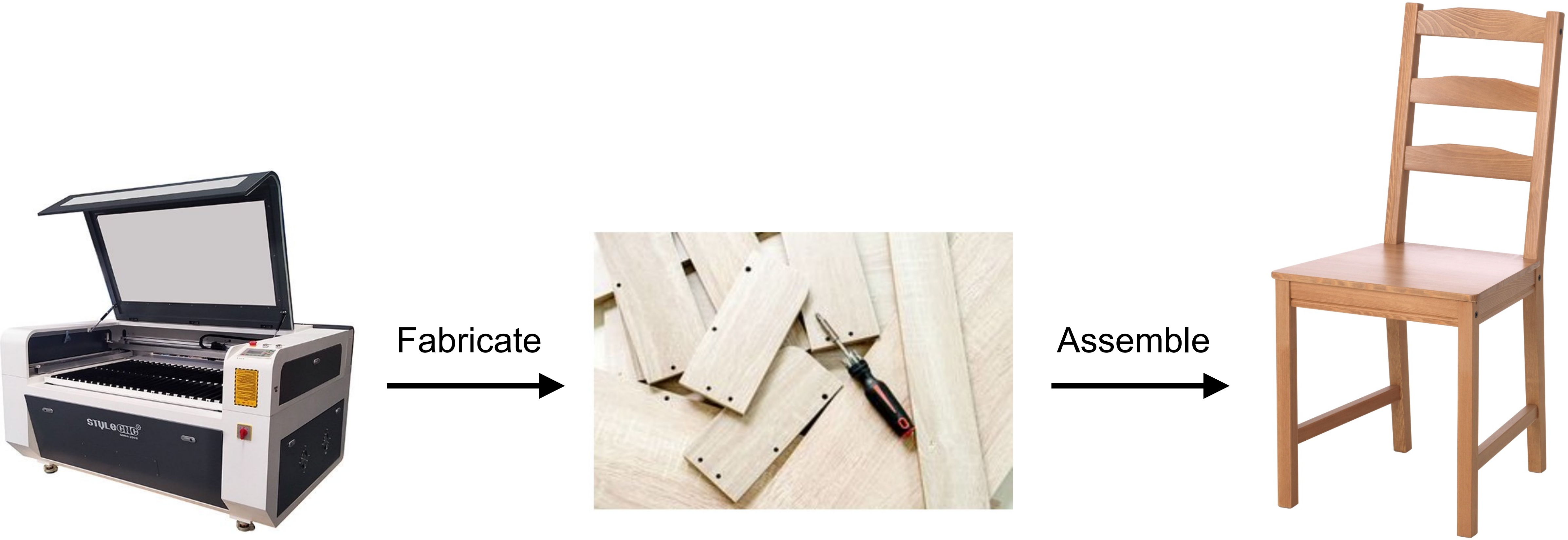
Assembly



✓

Assemblies: Advantages

- Simplify fabrication

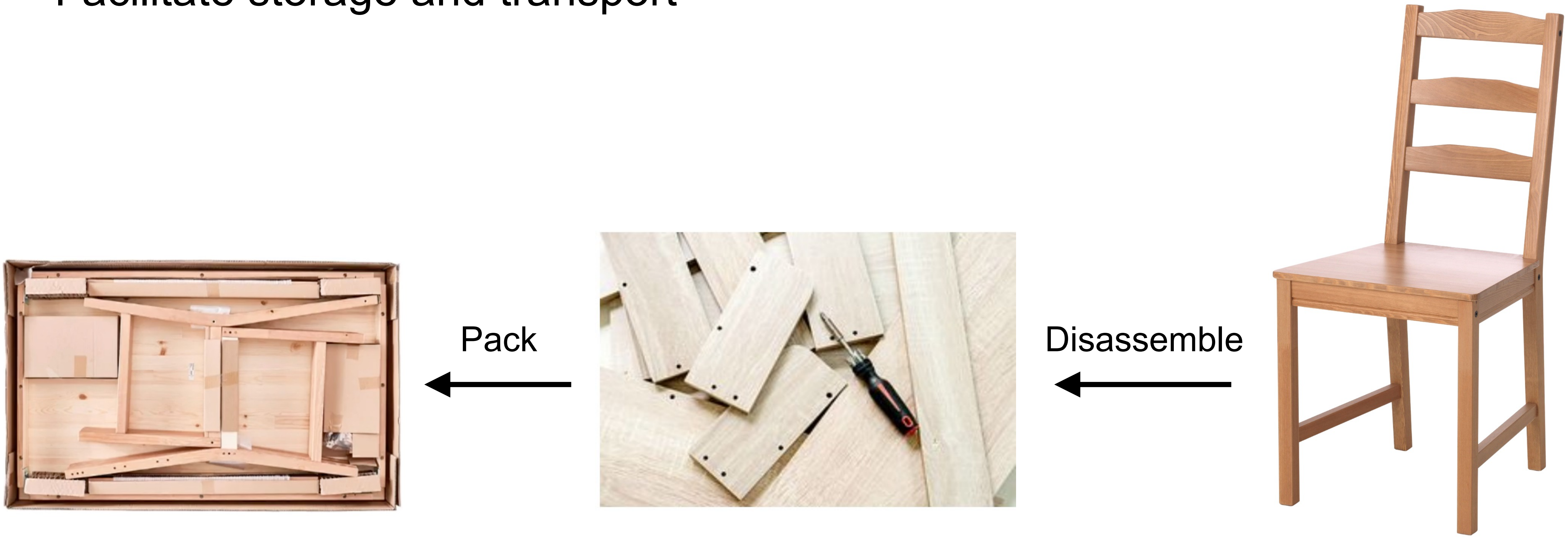


Fabricate

Assemble

Assemblies: Advantages

- Simplify fabrication
- Facilitate storage and transport

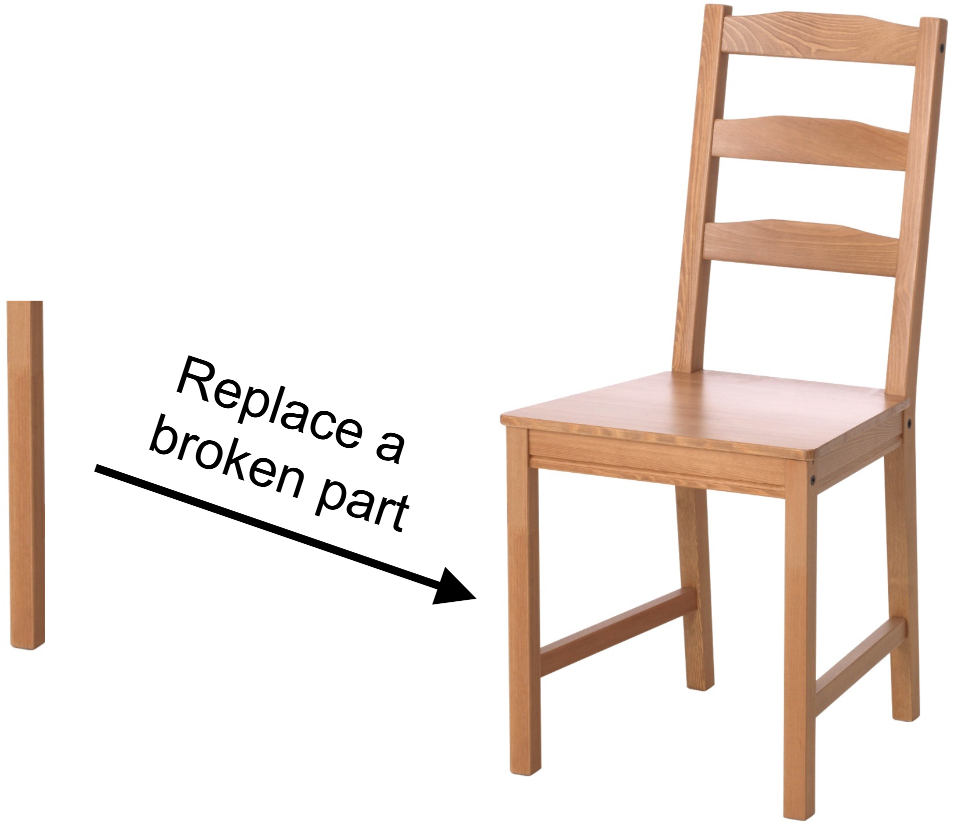


Pack

Disassemble

Assemblies: Advantages

- Simplify fabrication
- Facilitate storage and transport
- Facilitate maintenance

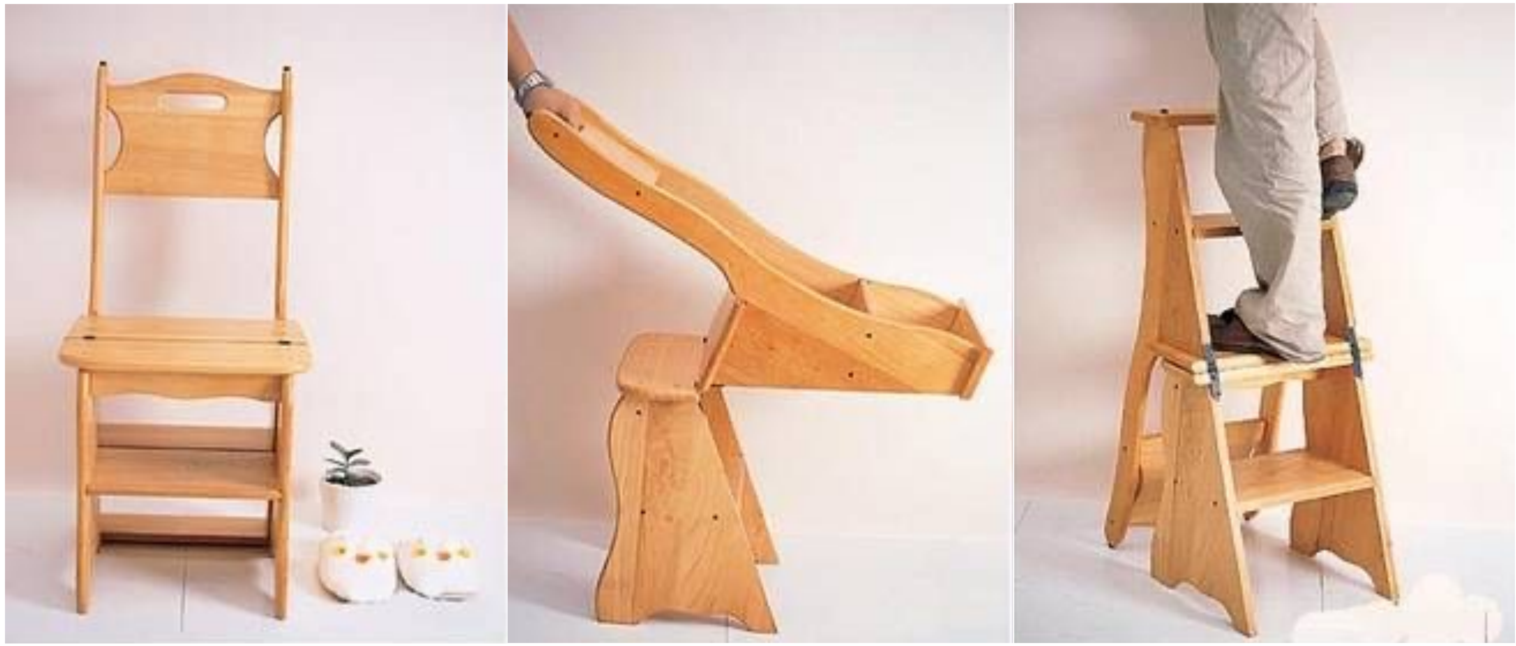


Assemblies: Advantages

- Simplify fabrication
- Facilitate storage and transport
- Facilitate maintenance
- Multiple forms

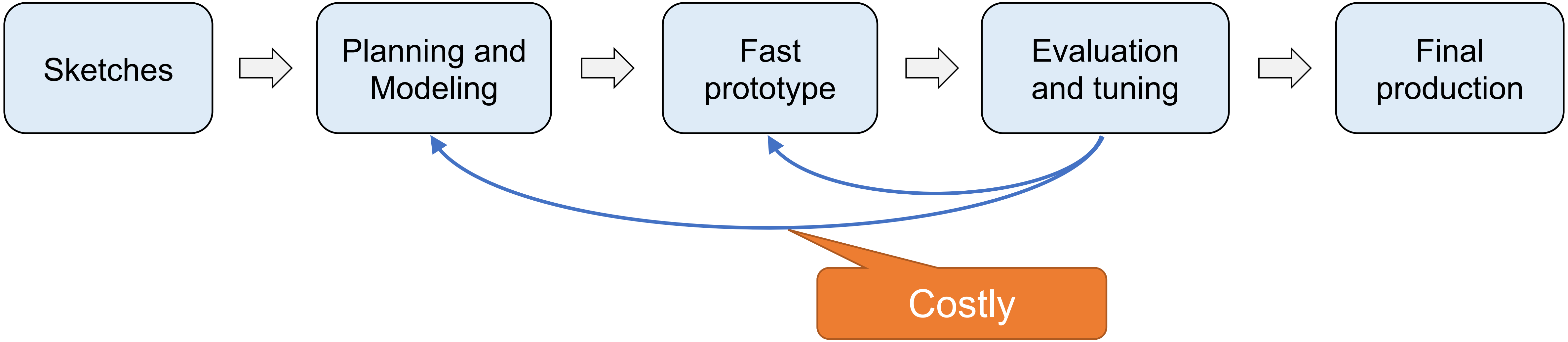
Assemblies: Advantages

- Simplify fabrication
- Facilitate storage and transport
- Facilitate maintenance
- Multiple forms
- Multiple functionalities



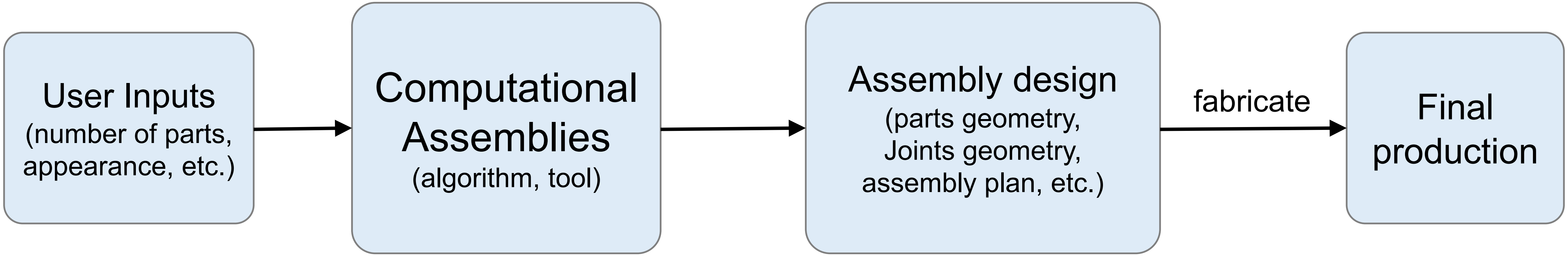
Traditional Design Process

- Traditionally, designing assemblies is a challenging task restricted to the professionals.



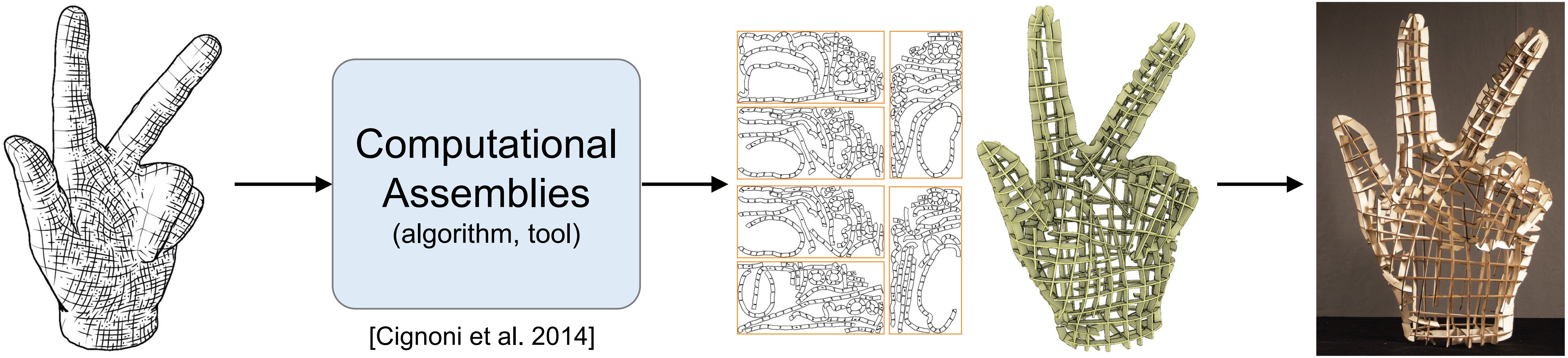
Computational Assemblies

- Currently, there is a trend to study and develop computational techniques for **analyzing**, **designing**, and **fabricating** assemblies
- We name this emerging research area **computational assemblies**
  - enable general users to design **personalized assemblies**
  - enable to generate designs with **optimized performance**



Computational Assemblies

- Currently, there is a trend to study and develop computational techniques for **analyzing**, **designing**, and **fabricating** assemblies
- We name this emerging research area **computational assemblies**
  - enable general users to design **personalized assemblies**
  - enable to generate designs with **optimized performance**



Computational Assemblies

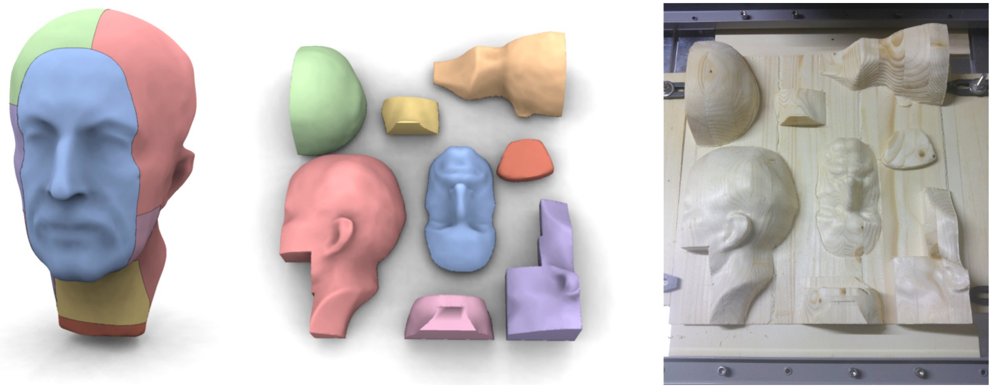
(algorithm, tool)

[Cignoni et al. 2014]

Research in Computational Assemblies

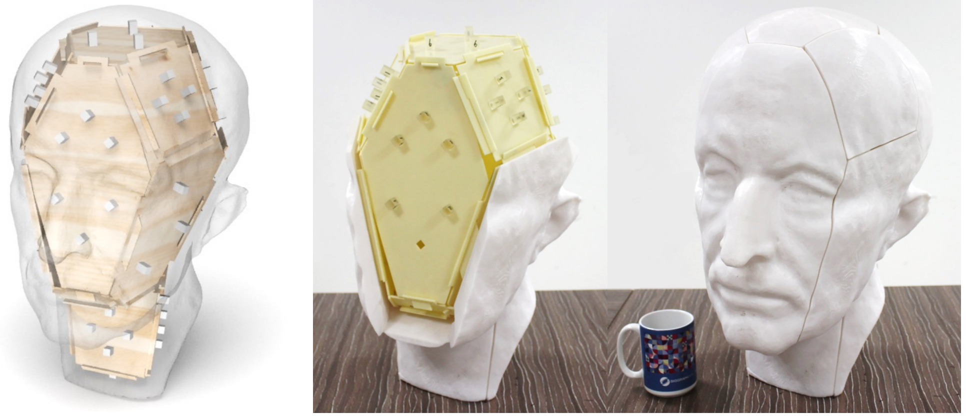
- Parts fabricability

CNC milling



[Muntoni et al. 2018]

laser cutting
+ 3D printing

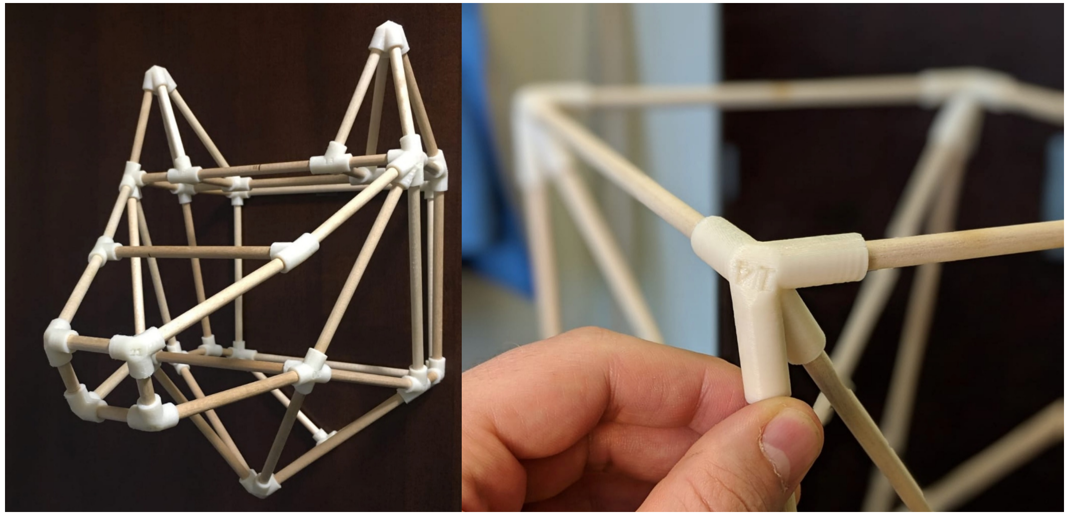


[Song et al. 2016]

Research in Computational Assemblies

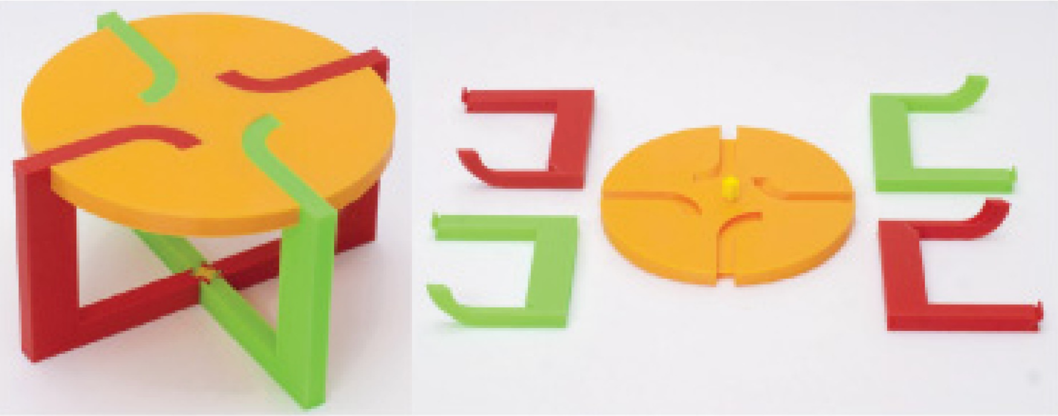
- Parts fabricability
- Parts joining

external joint



[Jacobson 2019]

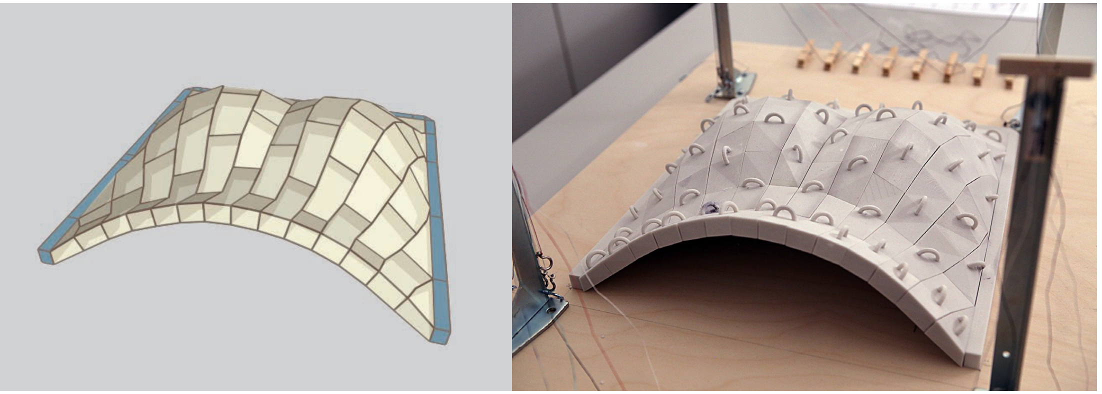
integral joint



[Yao et al. 2019]

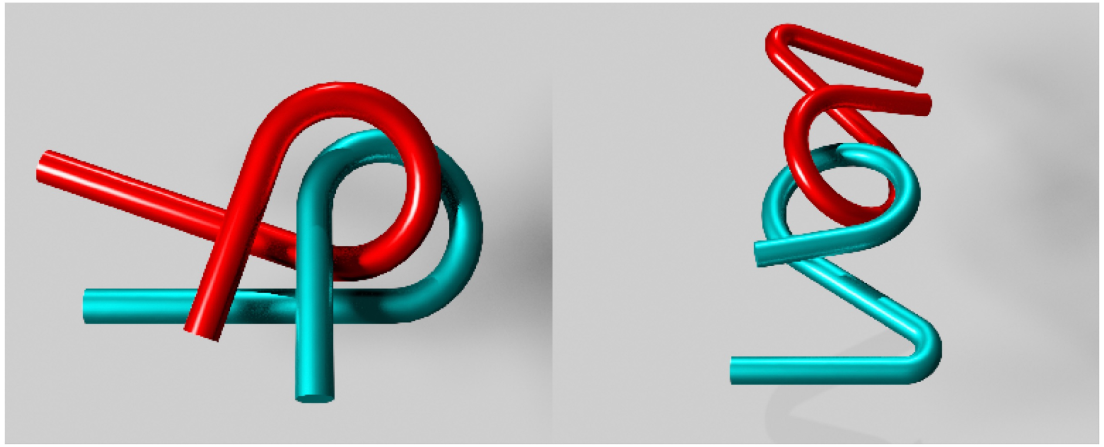
Research in Computational Assemblies

- Parts fabricability
- Parts joining
- Parts assembly



assembly sequence

[Deuss et al. 2014]



assembly motion

[Zhang et al. 2020]

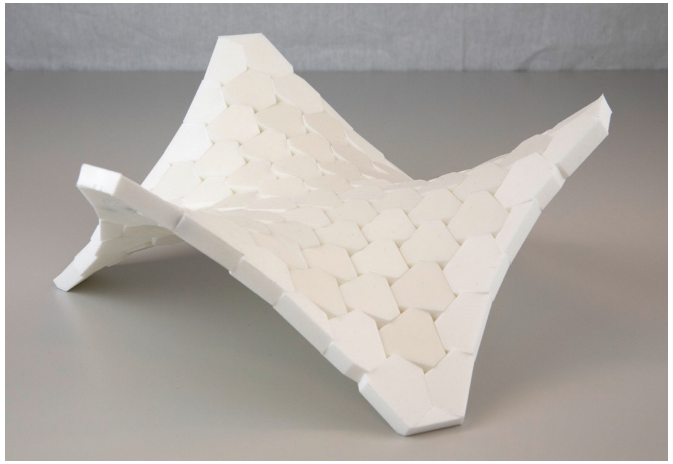
Research in Computational Assemblies

- Parts fabricability
- Parts joining
- Parts assembly
- Structural stability

equilibrium

[Panozzo et al. 2013]

interlocking



[Wang et al. 2019]

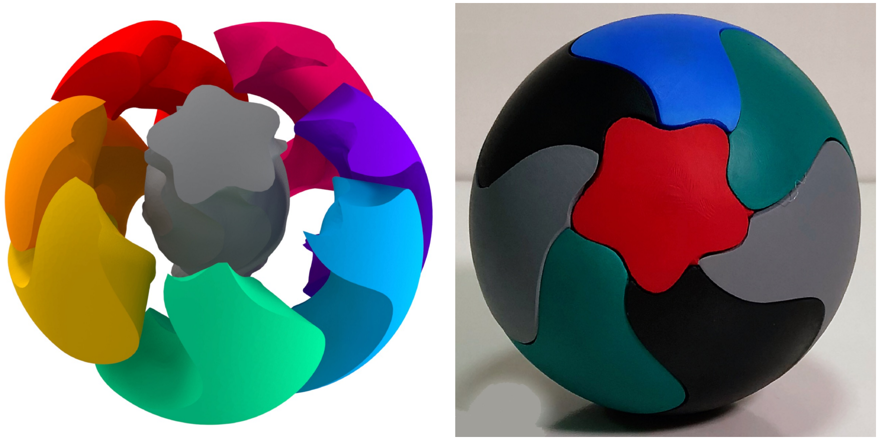
Research in Computational Assemblies

- Parts fabricability
- Parts joining
- Parts assembly
- Structural stability
- Assembly aesthetics

cutting seams

[Filoscia et al. 2020]

multi-color



[Araújo et al. 2019]

Research in Computational Assemblies

- Parts fabricability
- Parts joining
- Parts assembly
- Structural stability
- Assembly aesthetics
- Reconfigurability

[Song et al. 2017]

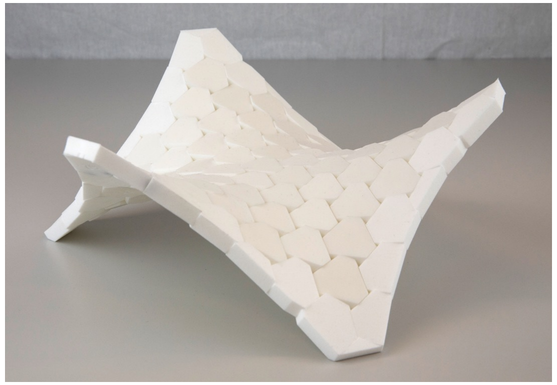
free
reconfig.

[Yuan et al. 2018]

hinged
reconfig.

Research in Computational Assemblies

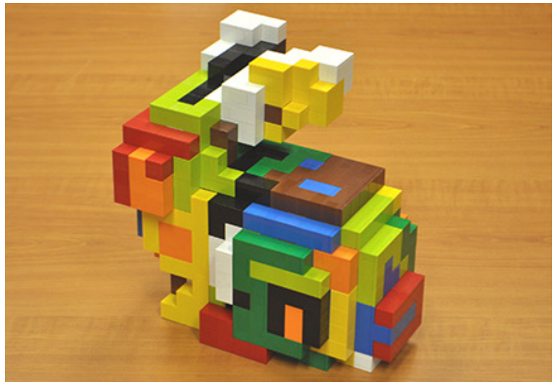
- Parts fabricability
- Parts joining
- Parts assembly
- **Structural stability**
- Assembly aesthetics
- Reconfigurability



[Wang et al. 2019]

[Wang et al. 2018]

Computational Design of Interlocking Assemblies



[Song et al. 2012]

Structurally Stable

- An assembly with rigid parts is structurally stable if it can preserve its form under external forces without collapse
- Structurally stable is a necessary condition for using many real-word objects

unstable

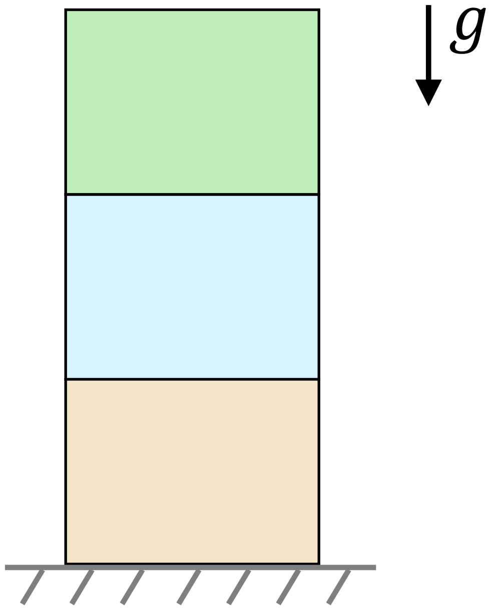
unstable

unstable

Structural Stability Analysis

Static analysis

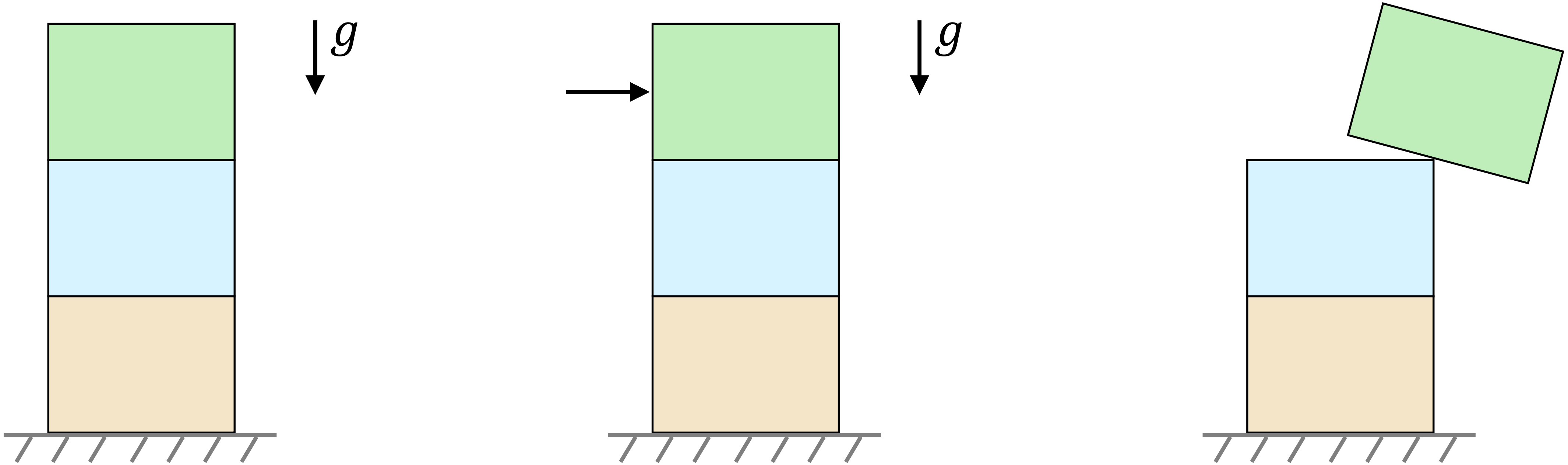
(equilibrium under **a certain external** force)



Structural Stability Analysis

Static analysis

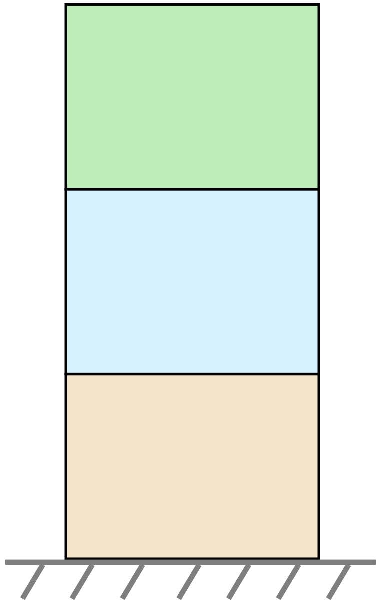
(equilibrium under
a certain external force)



Structural Stability Analysis

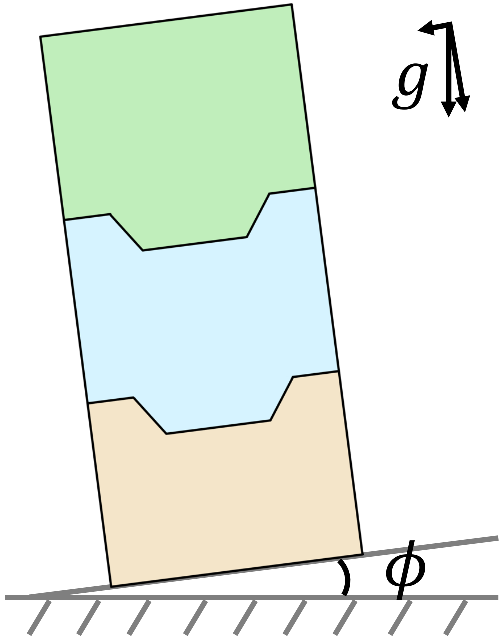
Static analysis

(equilibrium under
a certain external force)



Tilt analysis

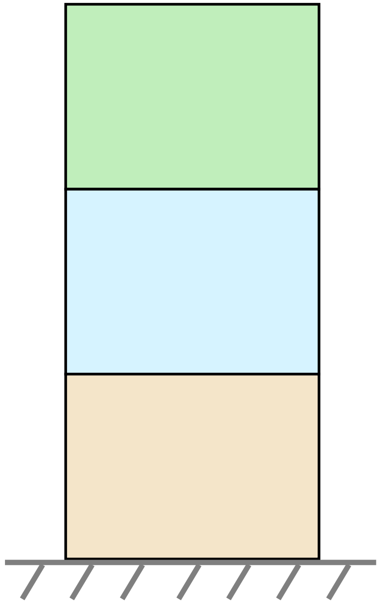
(equilibrium under
a set of external forces)



Structural Stability Analysis

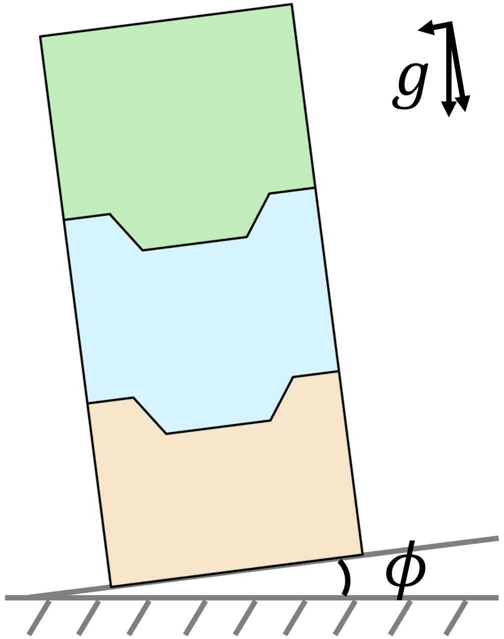
Static analysis

(equilibrium under
a certain external force)



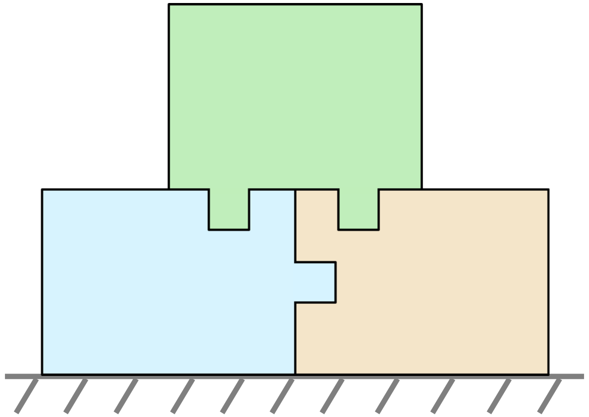
Tilt analysis

(equilibrium under
a set of external forces)



Interlocking test

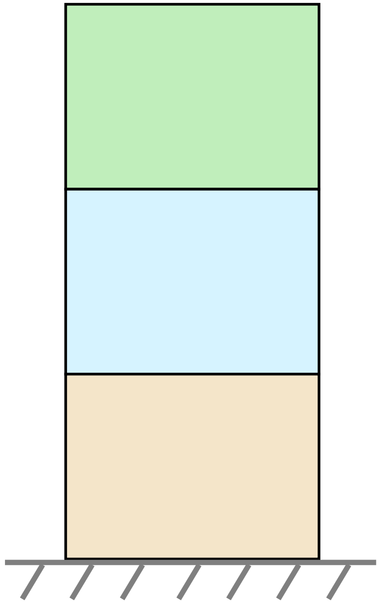
(equilibrium under
arbitrary external forces)



Structural Stability Analysis

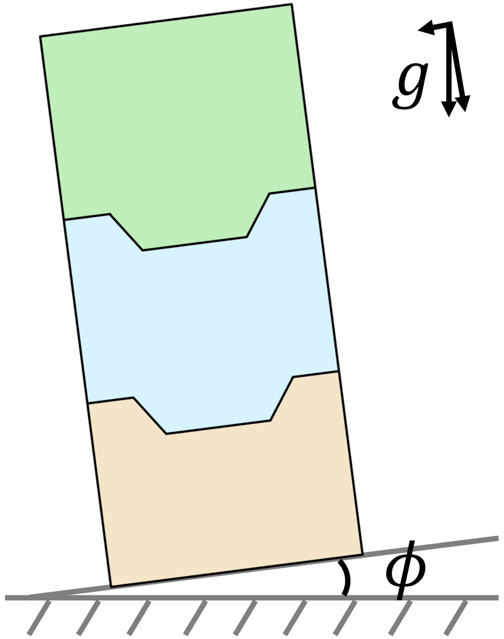
Static analysis

(equilibrium under **a certain external** force)



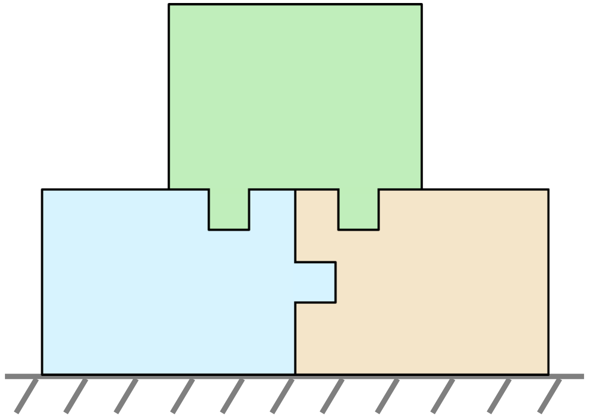
Tilt analysis

(equilibrium under **a set of external** forces)



Interlocking test

(equilibrium under **arbitrary external** forces)



more structurally stable (with more restrictive joints)

Interlocking Assemblies

Interlocking assemblies have been used for a variety of applications, where the assemblies need to bear forces from many unpredictable directions.

Puzzle

[Song et al. 2012]

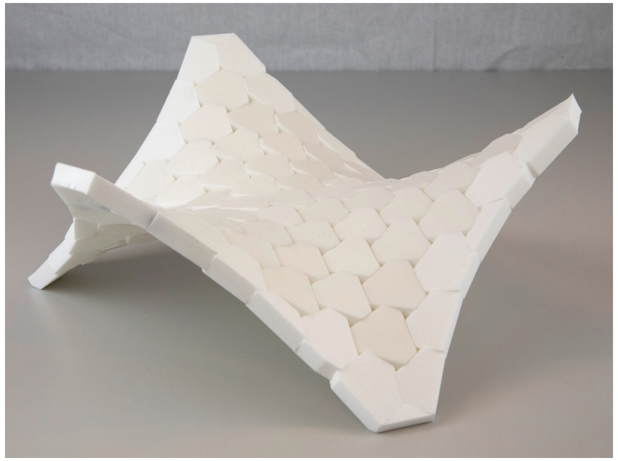
3D printing

[Yao et al. 2017]

Furniture

[Song et al. 2017]

Architecture

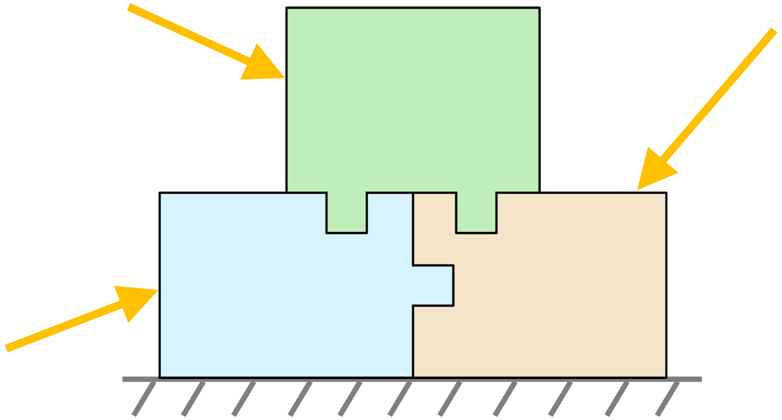


[Wang et al. 2019]

Interlocking Assemblies

Static

The assembly is in equilibrium under arbitrary external forces
(the key is held by other means)



Challenge:
We cannot enumerate all possible external force configurations!!!

Interlocking Assemblies

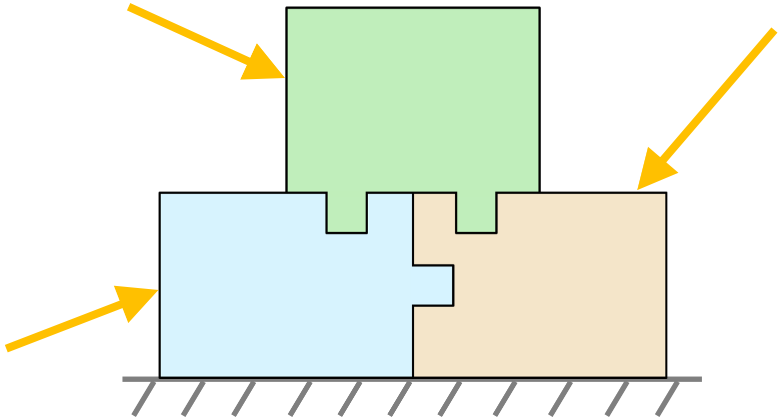
Dual

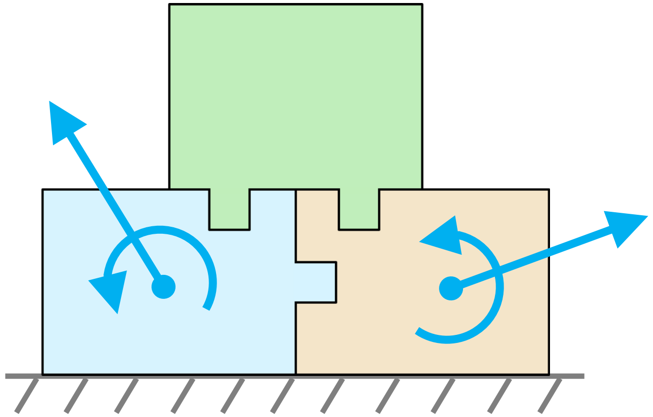
Static ↔ **Kinematic**

The assembly is in equilibrium under arbitrary external forces
(the key is held by other means)

=

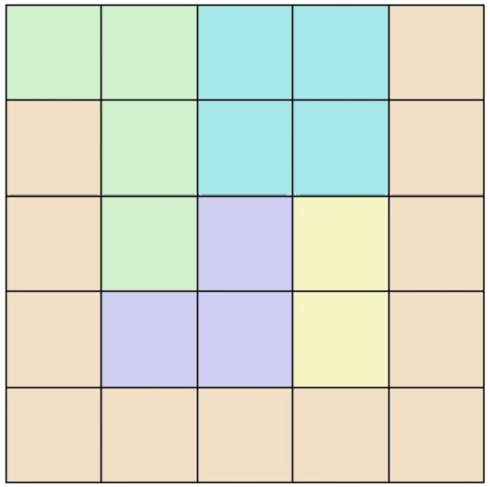
No movable part or subassembly
(except the key)



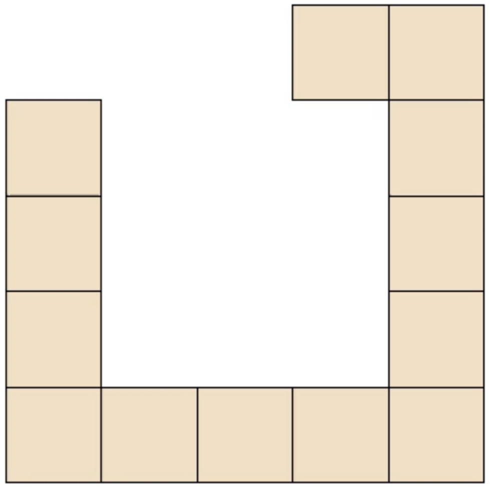


Interlocking Assemblies

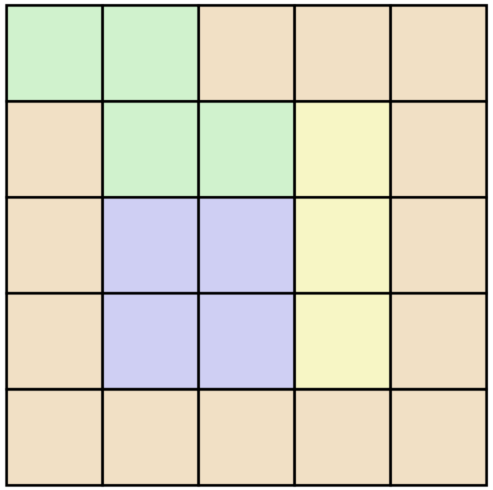
An assembly is interlocking if only one movable part (key), while all other parts, as well as any subset of the parts, are immobilized



non-interlocking



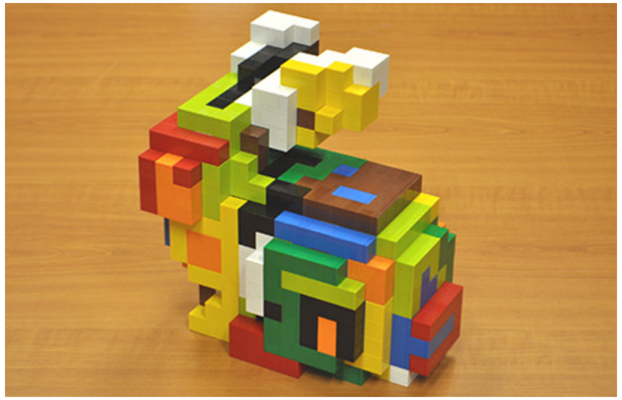
interlocking



deadlocking
(not disassemblable)

Three Research Projects

Enumeration-based Interlocking Test



Recursive Interlocking Puzzles

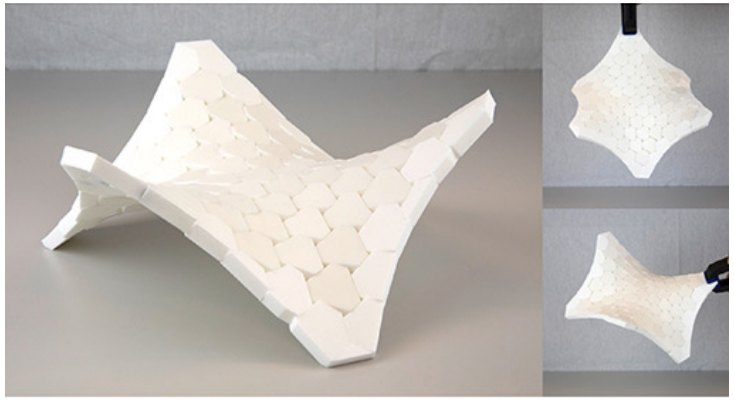
SIGGRAPH Asia 2012

DBG-based Interlocking Test

DESIA: A General Framework for Designing Interlocking Assemblies

SIGGRAPH Asia 2018

Inequality-based Interlocking Test

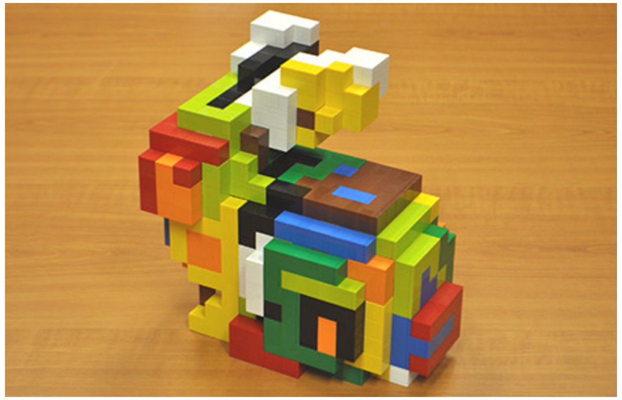


Design and Structural Optimization of Topological Interlocking Assemblies

SIGGRAPH Asia 2019

Three Research Projects

Enumeration-based Interlocking Test



Recursive Interlocking Puzzles

SIGGRAPH Asia 2012

DBG-based Interlocking Test

DESIA: A General Framework for Designing Interlocking Assemblies

SIGGRAPH Asia 2018

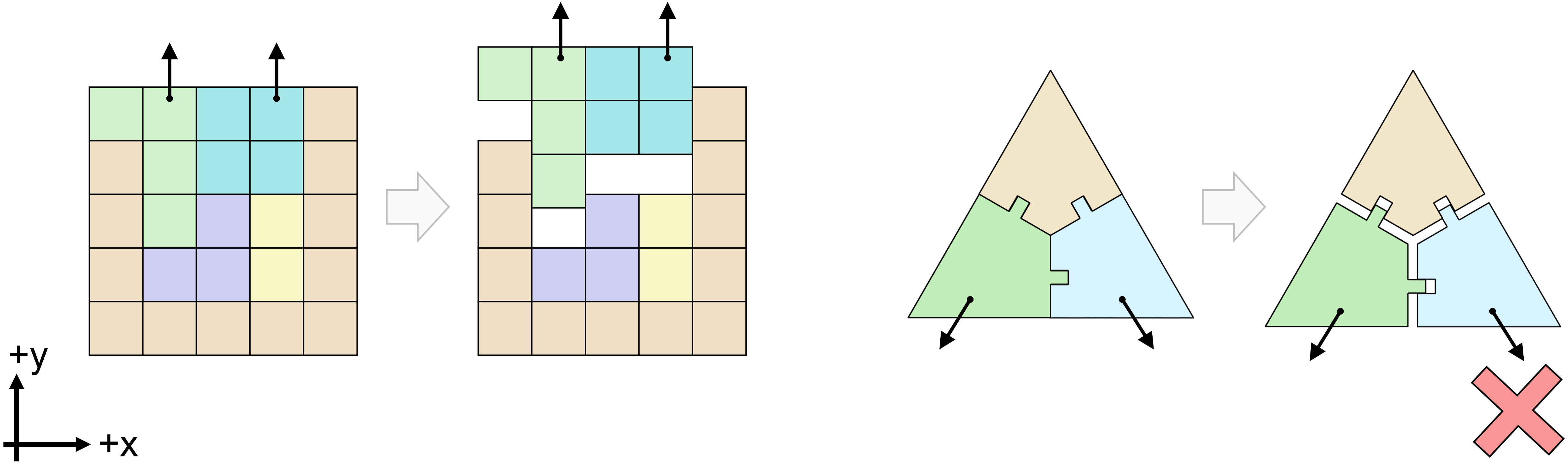
Inequality-based Interlocking Test

Design and Structural Optimization of Topological Interlocking Assemblies

SIGGRAPH Asia 2019

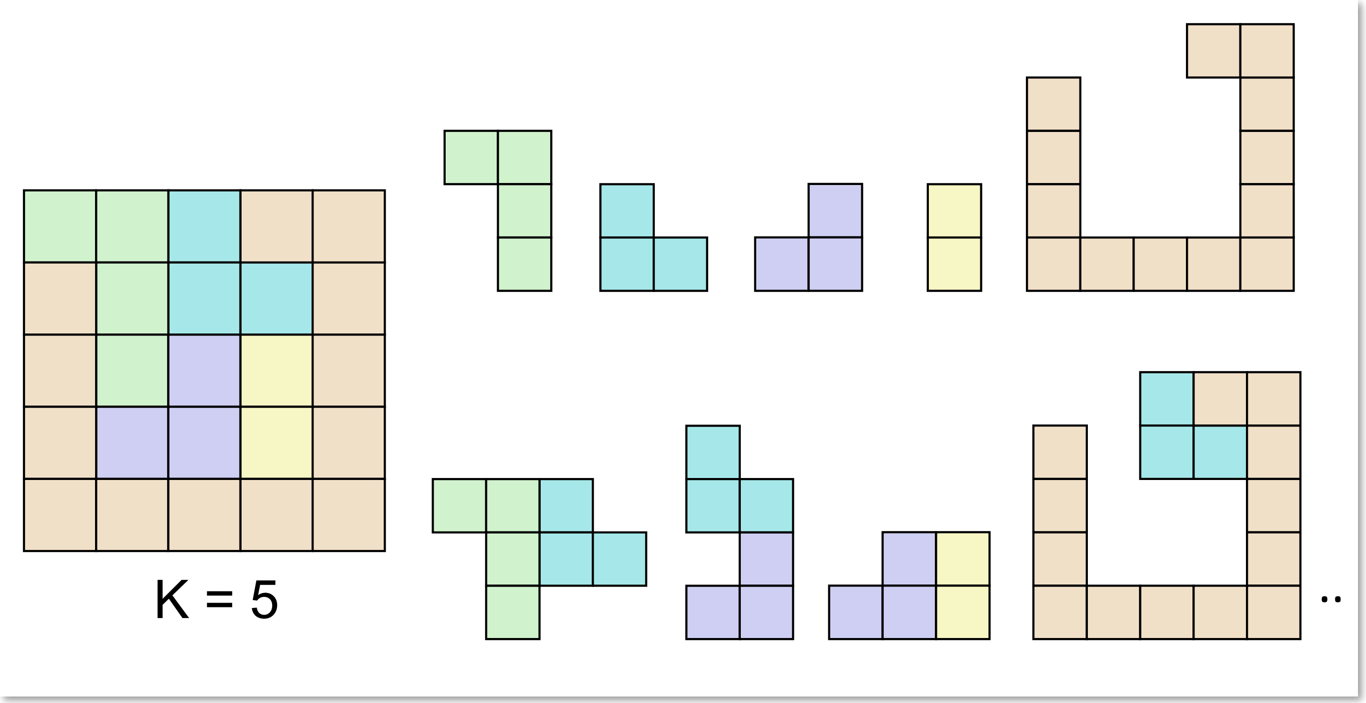
Enumeration-based Interlocking Test

- Method: exhaustively check mobility of each part and each subset of the parts
- Assumption: parts in a subassembly **translate** along the **same direction**



Enumeration-based Interlocking Test

- Method: exhaustively check mobility of each part and each subset of the parts
- Assumption: parts in a subassembly **translate** along the **same direction**
- Limitation: **exponential** computation complexity



C(K, 1)

+

C(K, 2)

…

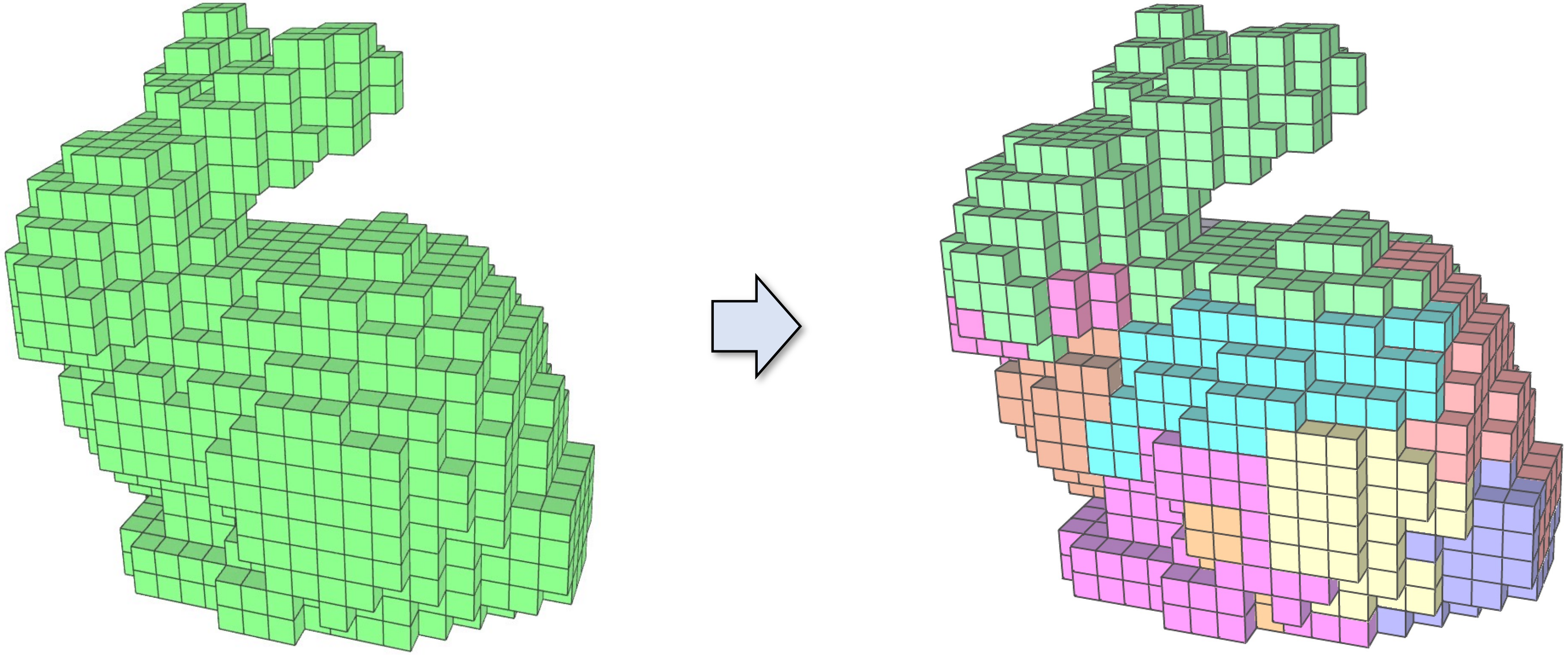
+

C(K, $\lfloor K/2 \rfloor$)

$= 2^{(K-1)}$ subsets of parts

Interlocking Puzzle Design

- Challenge: enumeration-based interlocking test can only handle assemblies with **a few parts** (e.g., <20) due to its computational complexity.

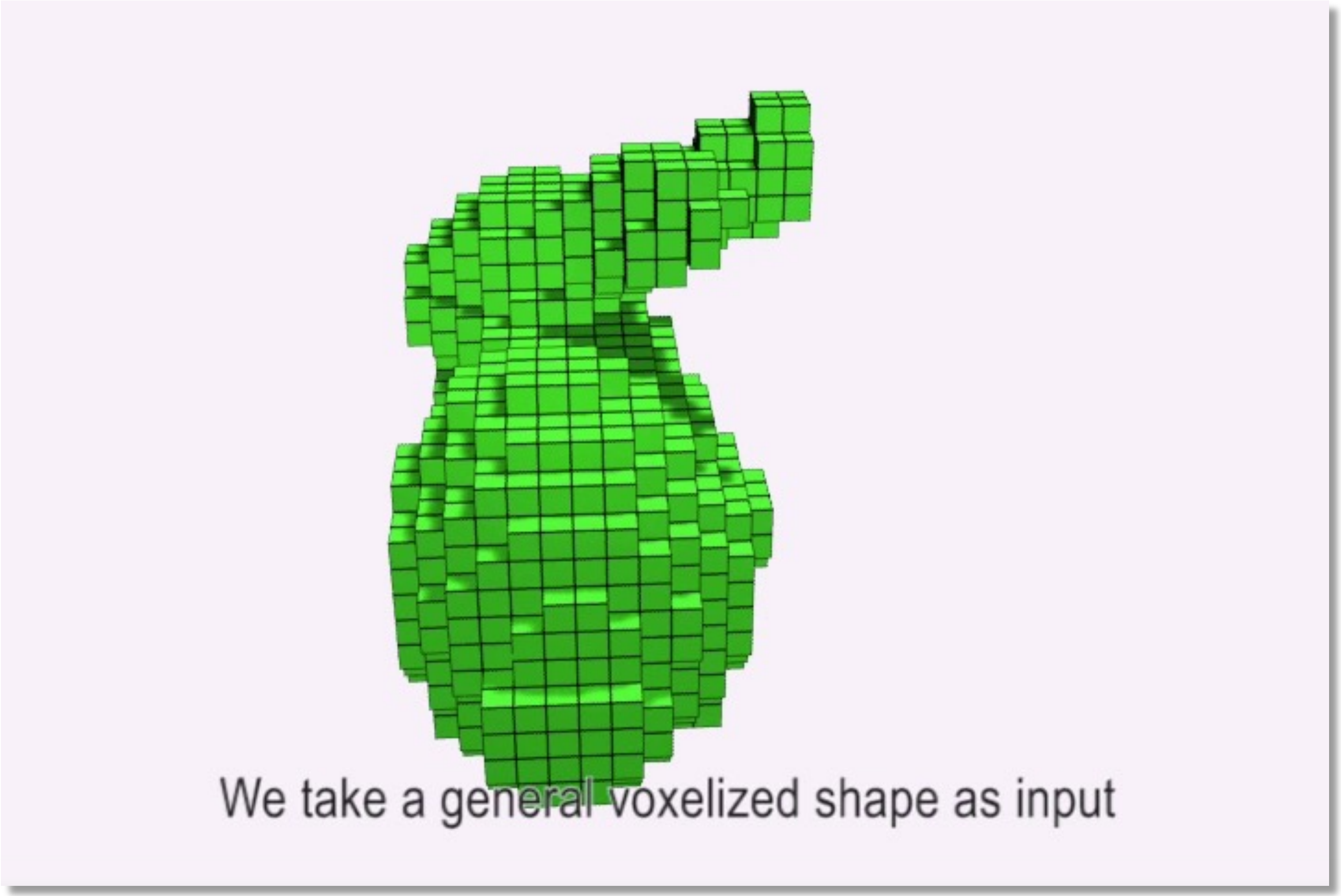


Input: target shape,
number of piece K

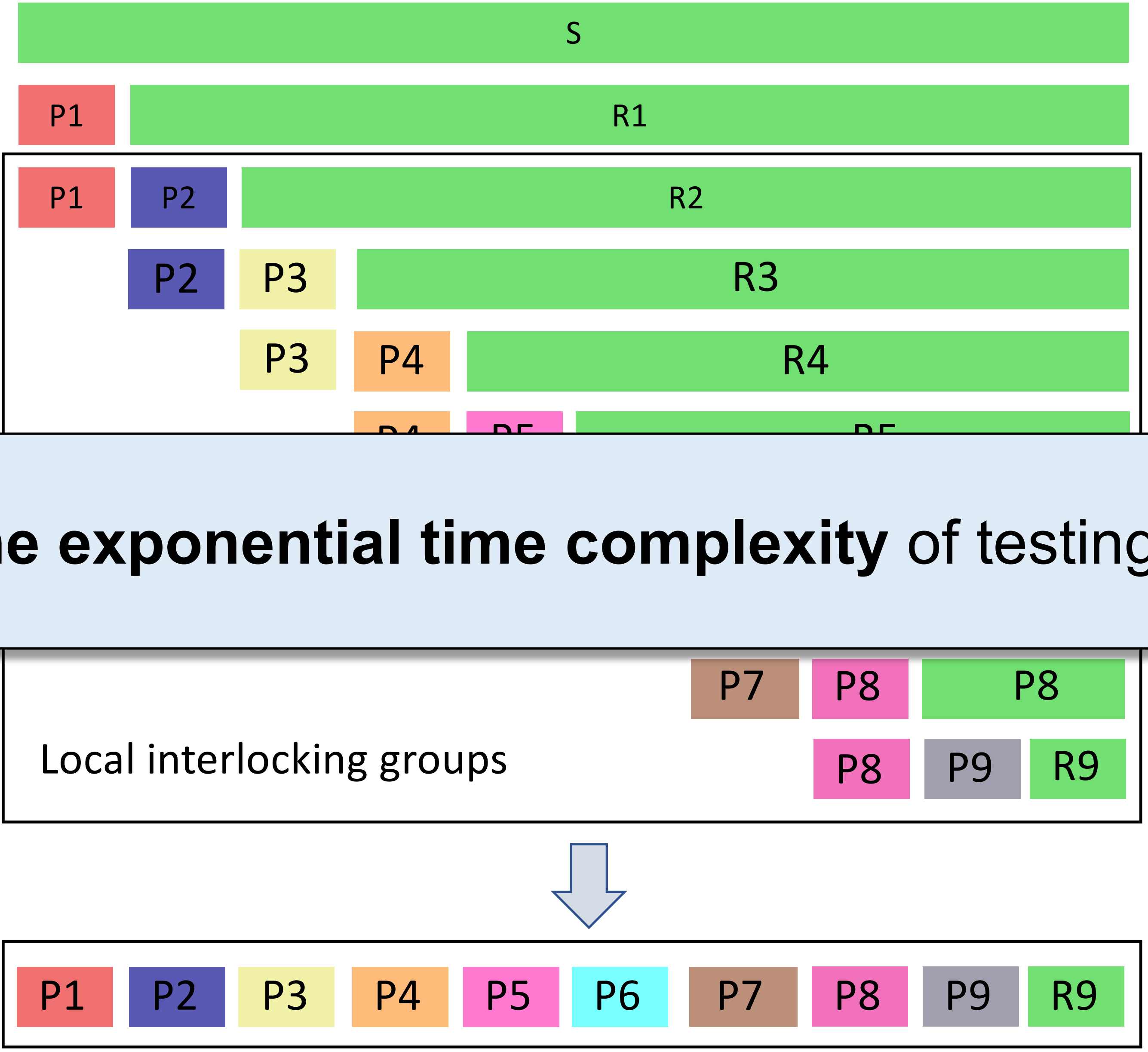
Output: interlocking puzzle

Interlocking Puzzle Design

- Idea: recursive interlocking to skip exponential complexity of interlocking test



We take a general voxelized shape as input



S

P1 R1

P1 P2 R2

P2 P3 R3

P3 P4 R4

P4 P5 R5

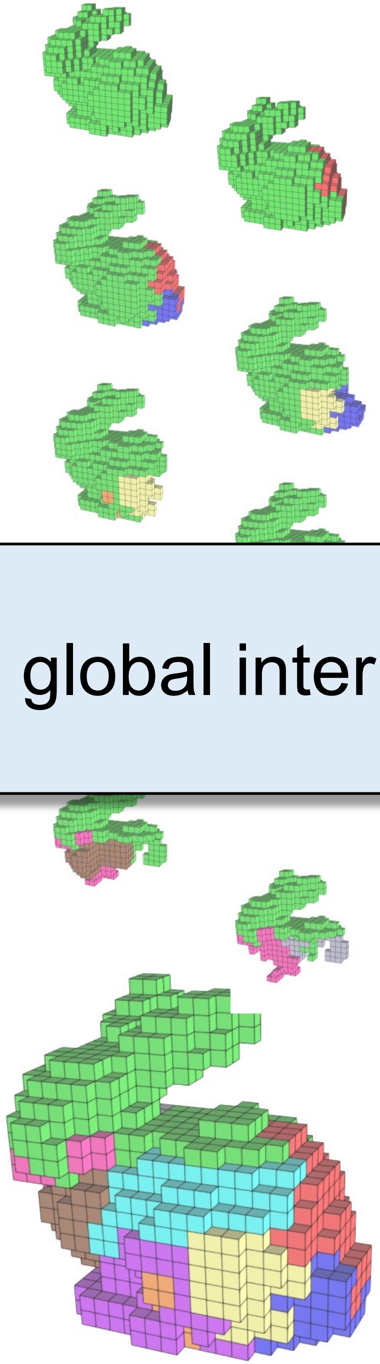
P7 P8 P8

P8 P9 R9

Local interlocking groups

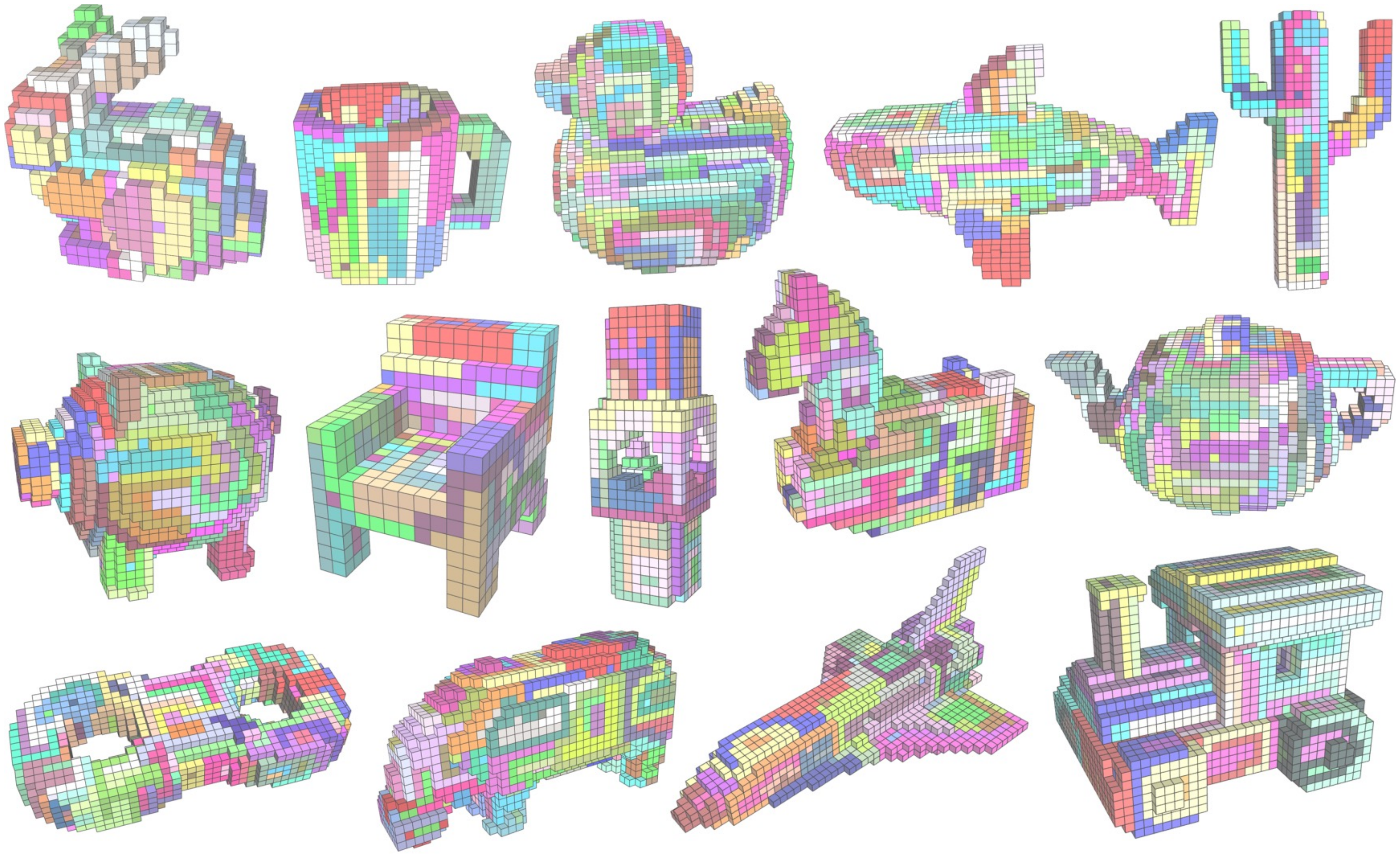
P1 P2 P3 P4 P5 P6 P7 P8 P9 R9

Global Interlocking

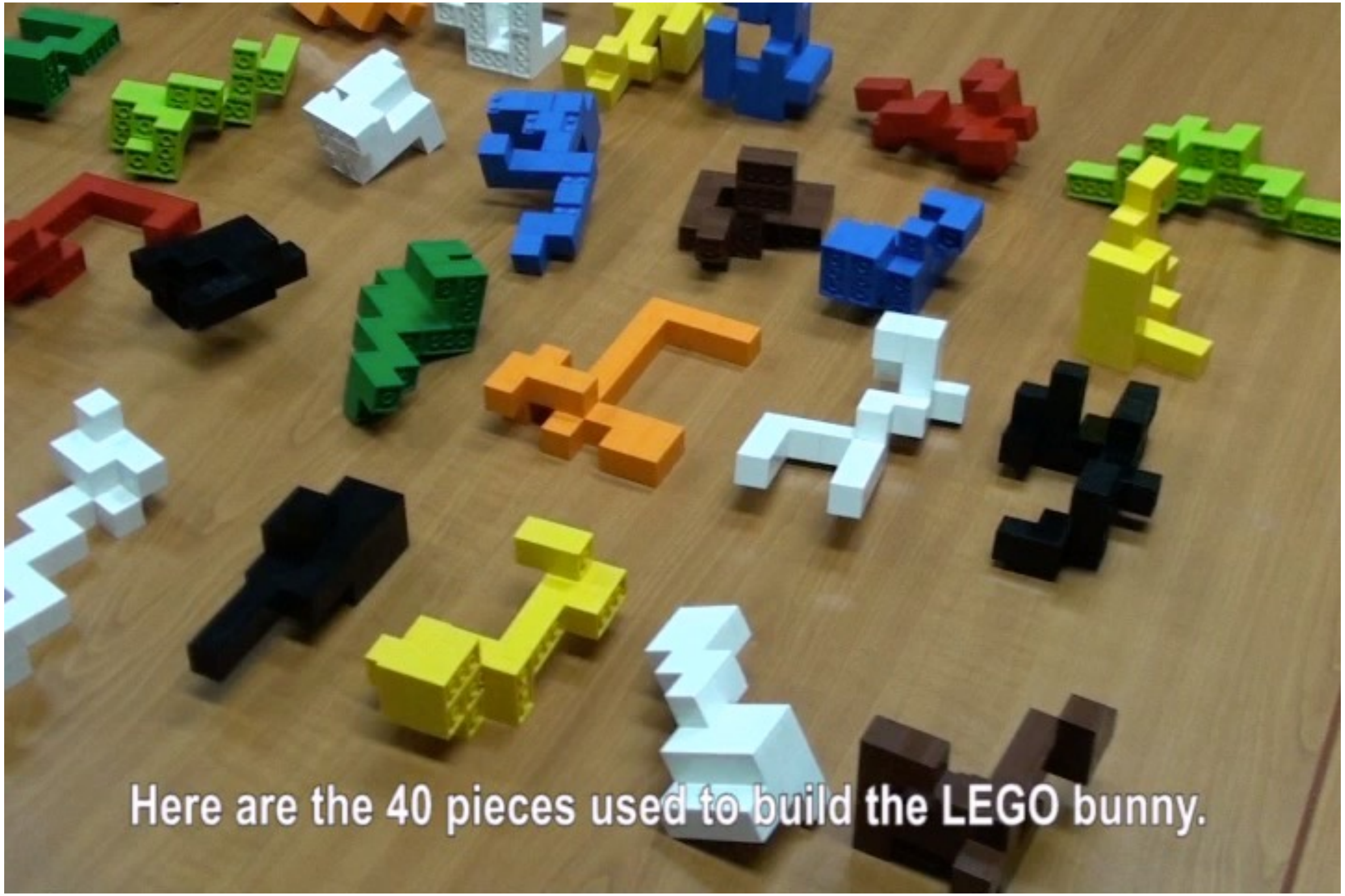


Skip **the exponential time complexity** of testing global interlocking!!!

Our Result



Our Result: 40-piece Bunny



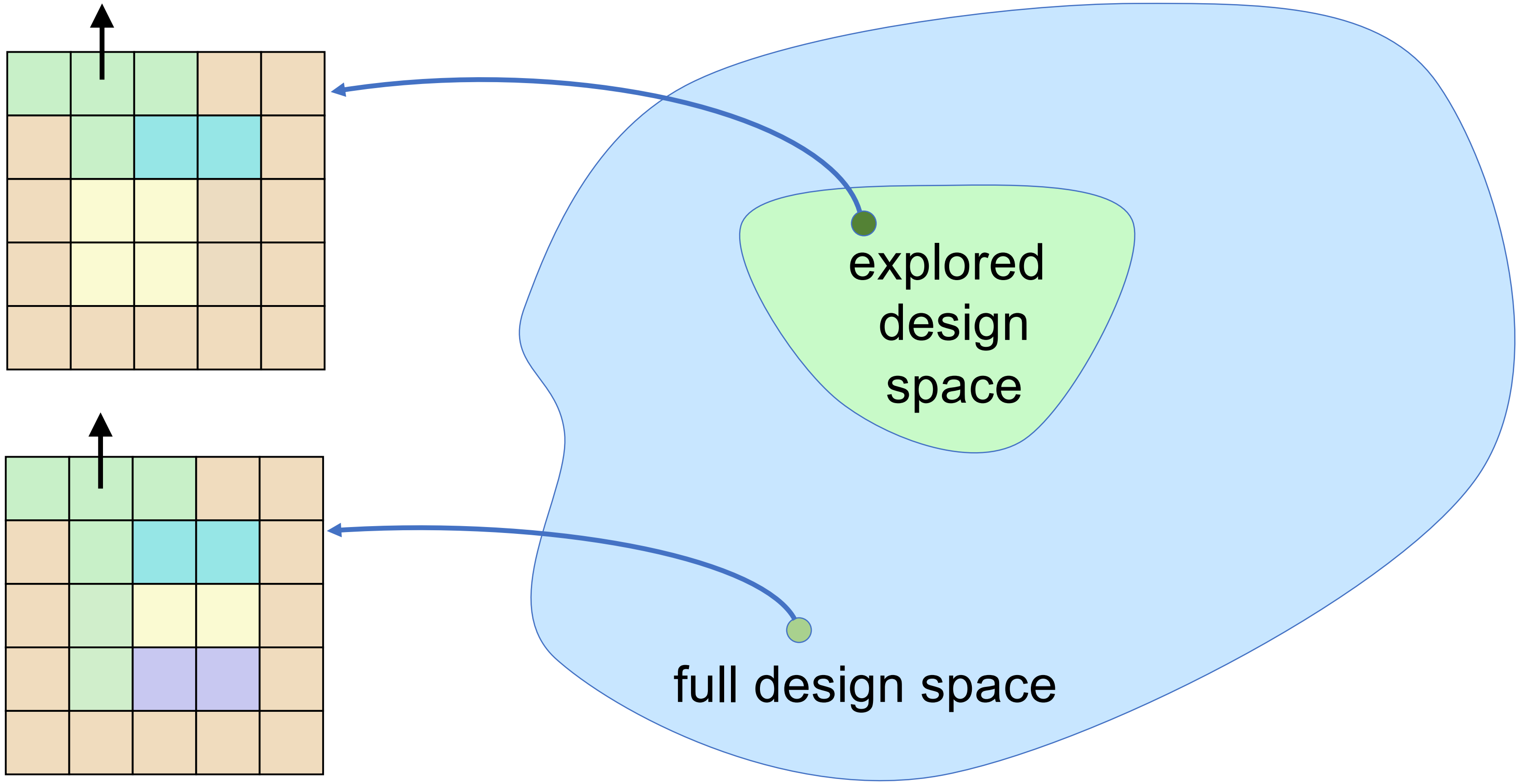
Here are the 40 pieces used to build the LEGO bunny.

Limitation

- The approach can only explore a limited design space

Recursive interlocking

General interlocking



Three Research Projects

Enumeration-based
Interlocking Test

Recursive Interlocking Puzzles

SIGGRAPH Asia 2012

DBG-based
Interlocking Test

DESIA: A General Framework for Designing Interlocking Assemblies

SIGGRAPH Asia 2018

Inequality-based
Interlocking Test

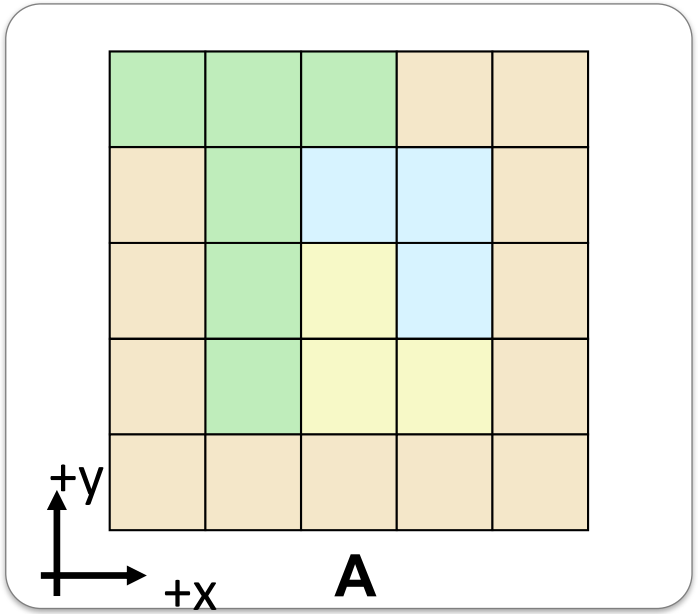
Design and Structural Optimization of Topological Interlocking Assemblies

SIGGRAPH Asia 2019

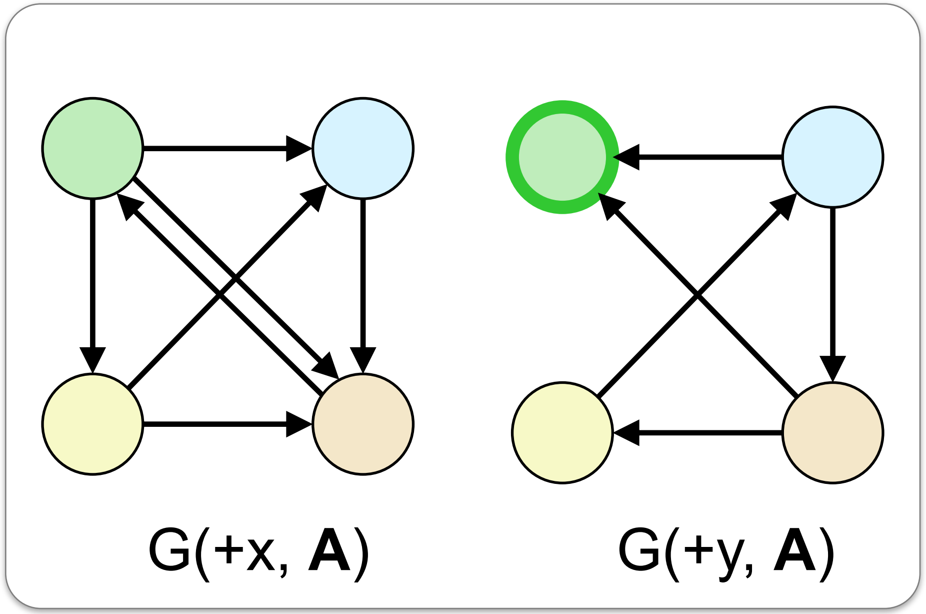
DBG-based Interlocking Test

- Method: check connectivity of direction blocking graphs (DBGs)
- Assumption: parts in a subassembly **translate** along the **same direction**

Test of global interlocking



Directional blocking graphs

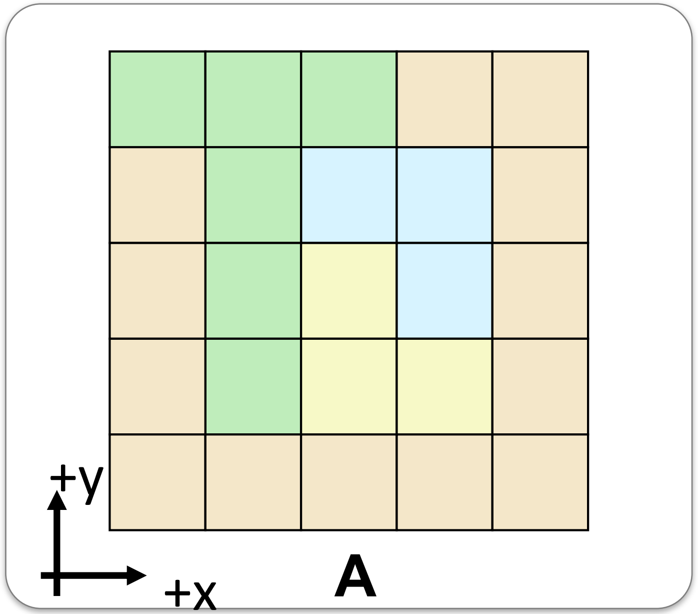


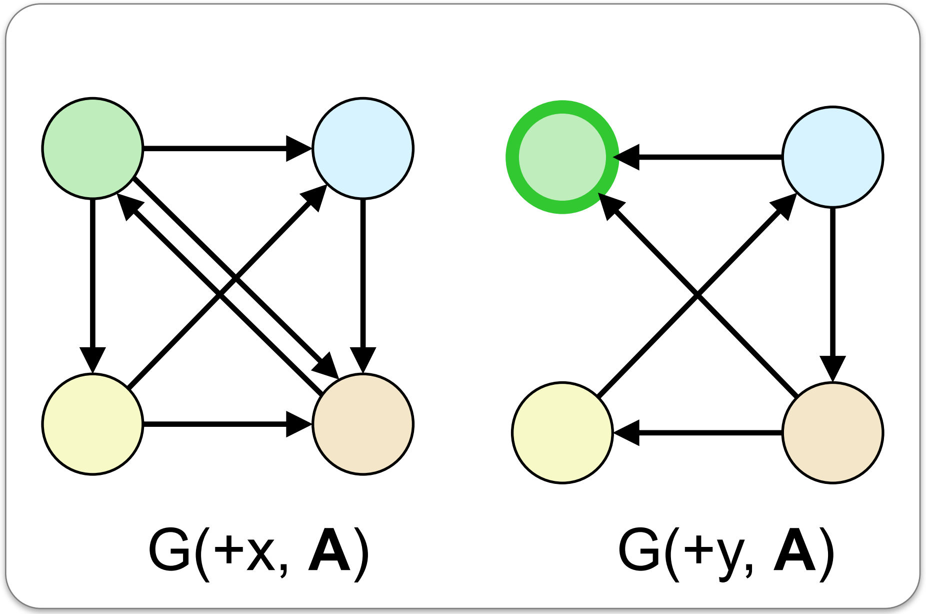
[Wang et al. 2018]

DBG-based Interlocking Test

- Method: check connectivity of direction blocking graphs (DBGs)
- Assumption: parts in a subassembly **translate** along the **same direction**
- Advantage: **polynomial** time complexity

Assembly is interlocking = All DBGs are strongly connected (except the key part)



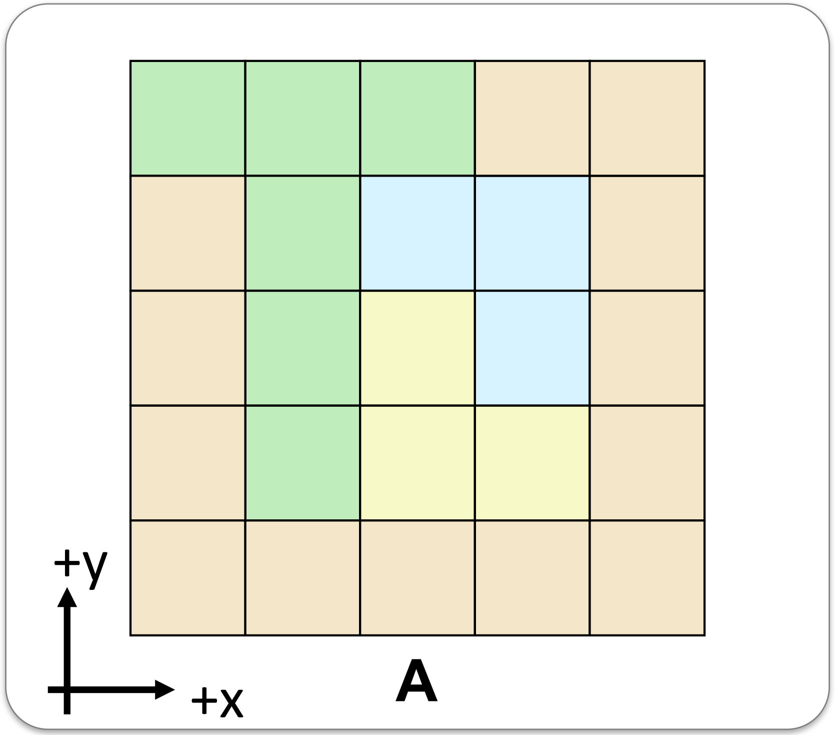


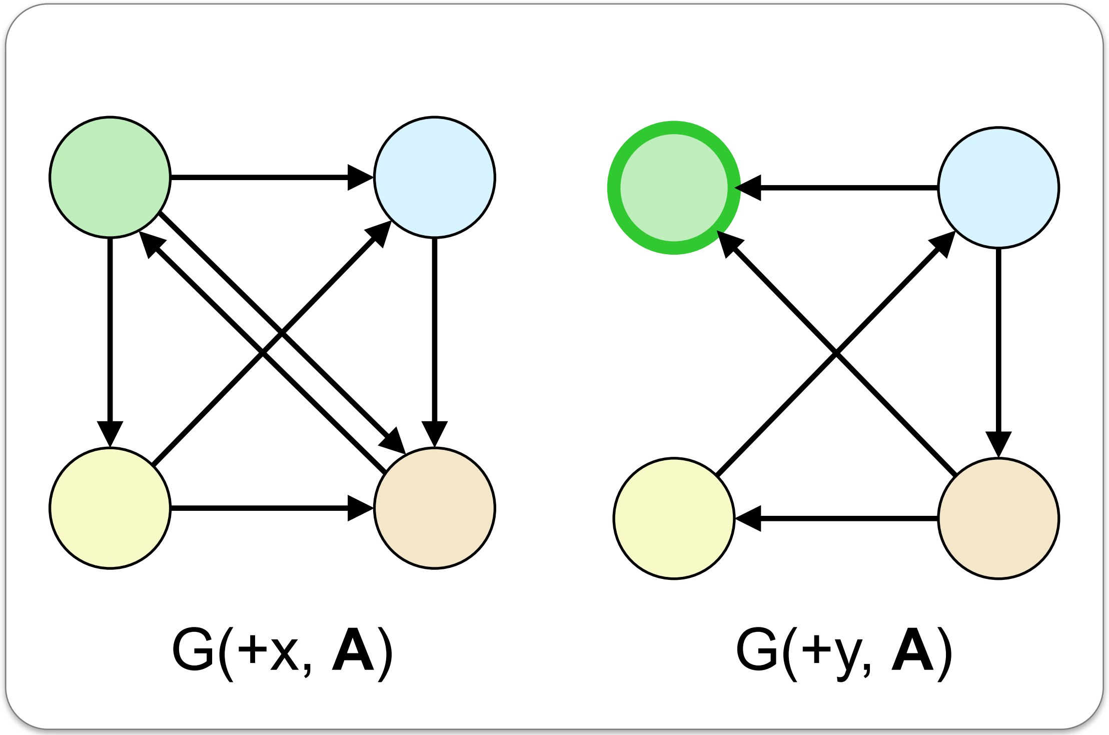
[Wang et al. 2018]

DBG-based Design Approach

Construct interlocking parts geometry

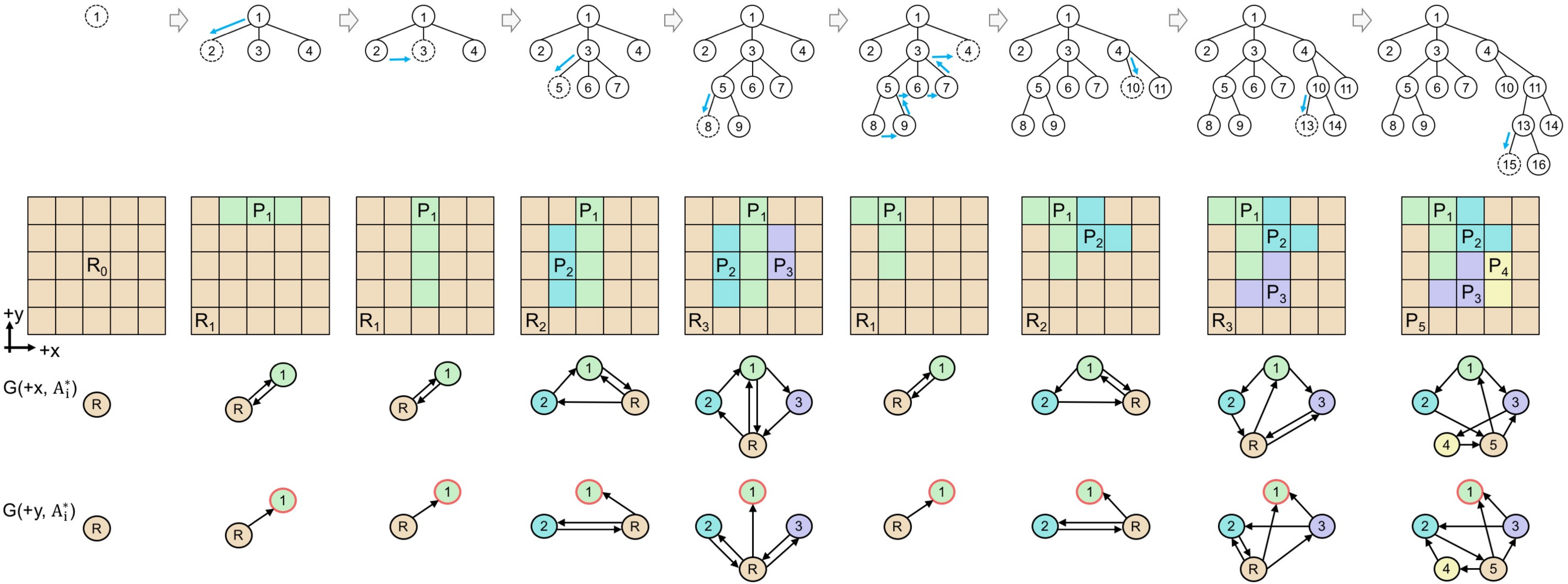
Make all graphs strongly connected



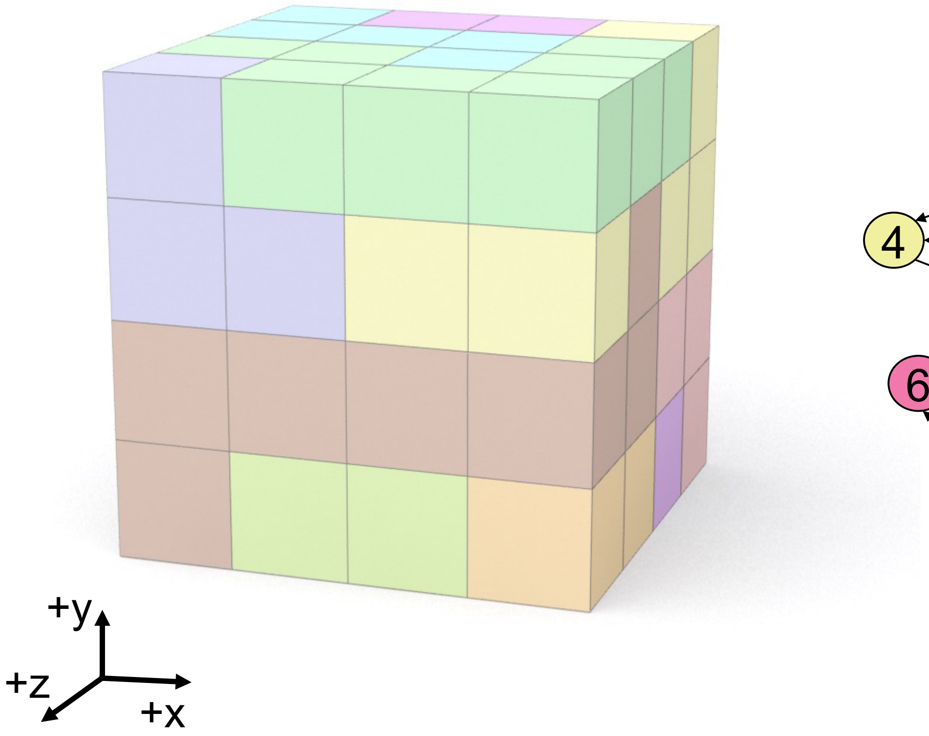


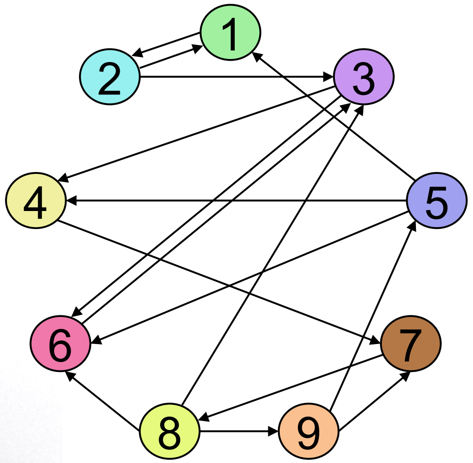
DBG-based Design Approach

Search space is explored in a tree traversal process with automatic backtracking

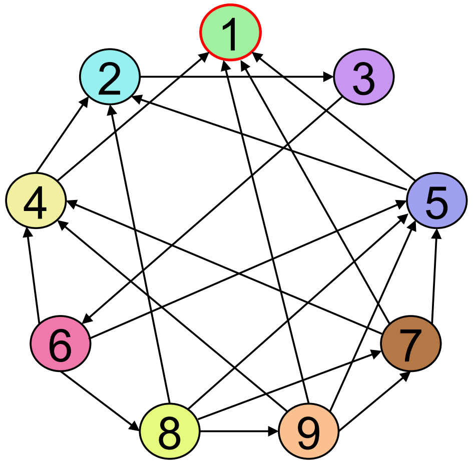


Results: Interlocking Cube

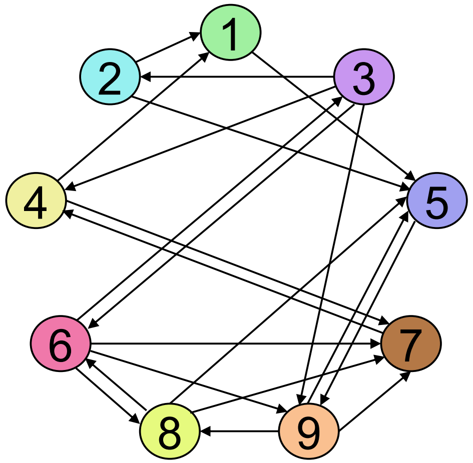




G(+x, A)



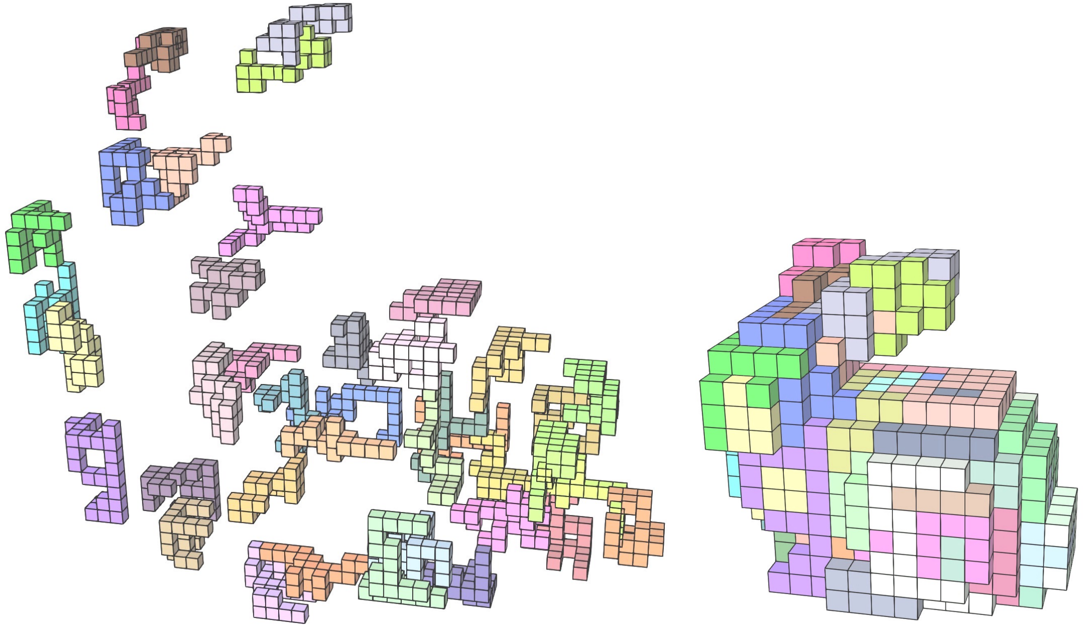
G(+y, A)



G(+z, A)

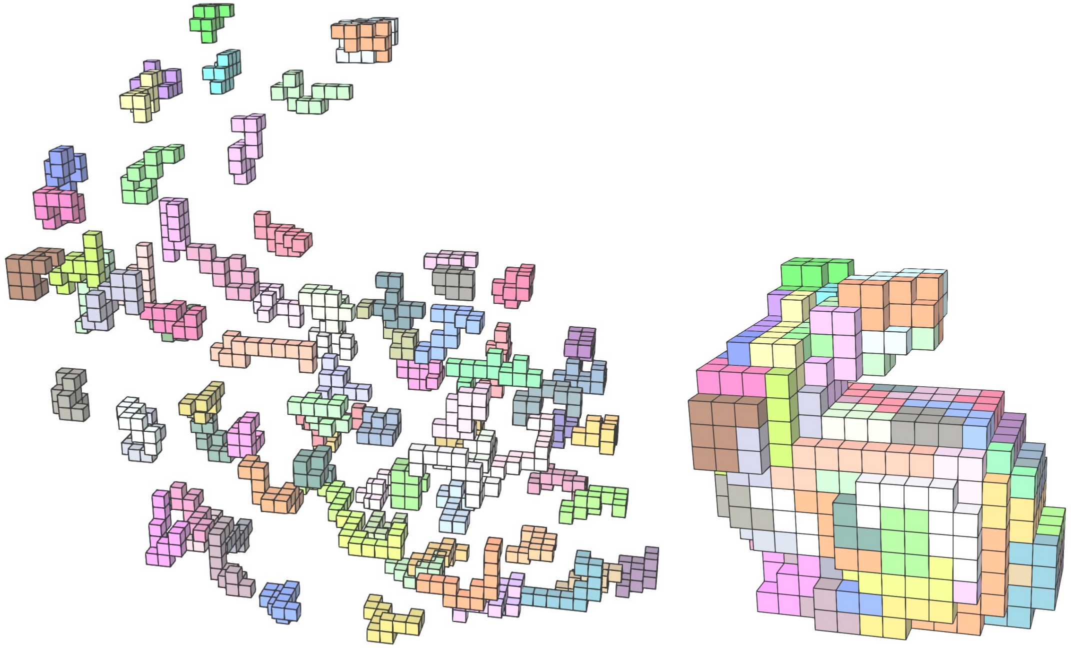
Results: Interlocking Bunny

Enumeration-based approach
[Song et al. 2012]



recursive interlocking, **40** parts

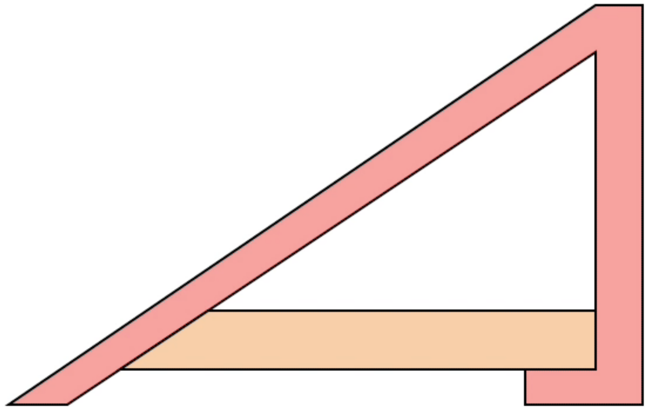
DBG-based approach
[Wang et al. 2018]



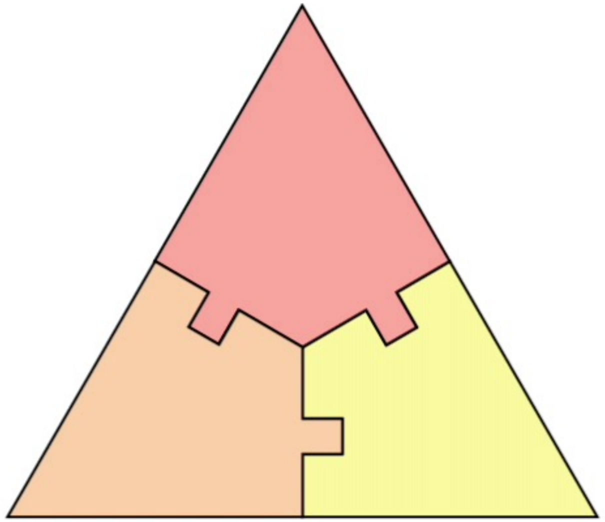
general interlocking, **80** parts

Limitation

- DBG-based interlocking test assume taking out each part or subassembly using translation motion following a certain order
  - No rotational motion
  - No coordinated motion



Failure example #1



Failure example #2

Three Research Projects

Enumeration-based Interlocking Test

Recursive Interlocking Puzzles

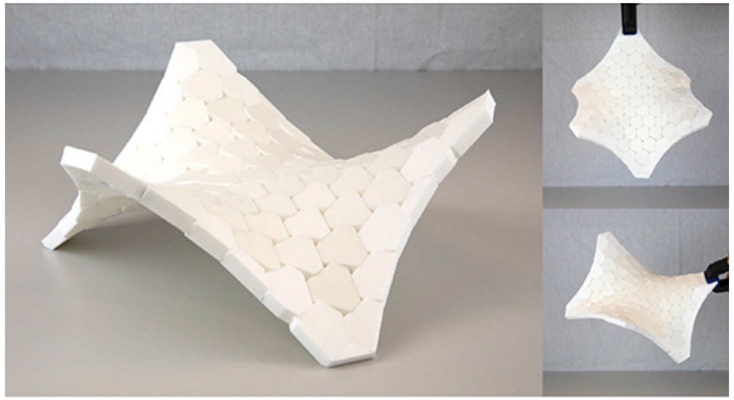
SIGGRAPH Asia 2012

DBG-based Interlocking Test

DESIA: A General Framework for Designing Interlocking Assemblies

SIGGRAPH Asia 2018

Inequality-based Interlocking Test

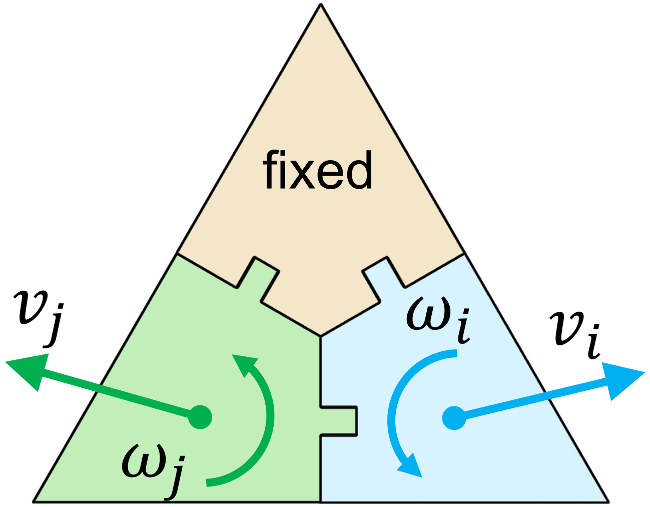


Design and Structural Optimization of Topological Interlocking Assemblies

SIGGRAPH Asia 2019

Inequality-based Interlocking Test

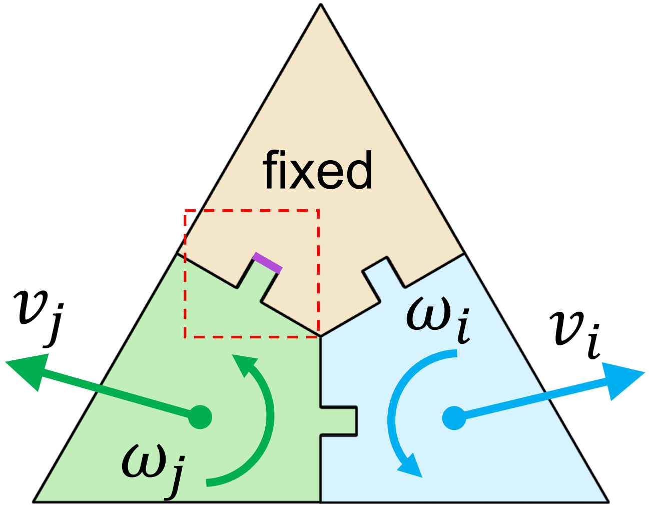
- **Search space**: each part moves freely in the 3D space, with velocity $y_i = [v_i, \omega_i]$

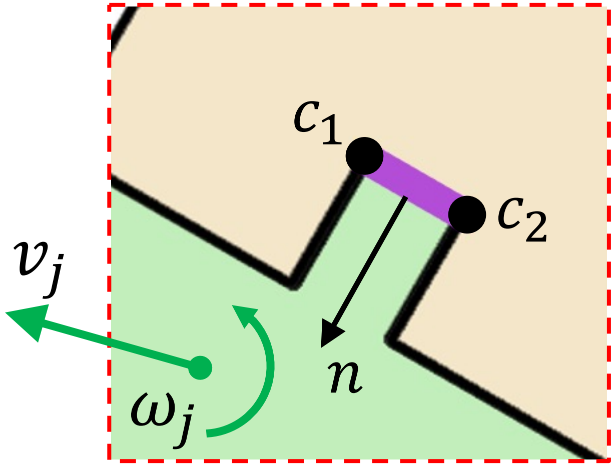


[Wang et al. 2019]

Inequality-based Interlocking Test

- **Search space**: each part moves freely in the 3D space, with velocity $y_i = [v_i, \omega_i]$
- **Constraint**: no collision between parts at each contact during parts movement





[Wang et al. 2019]

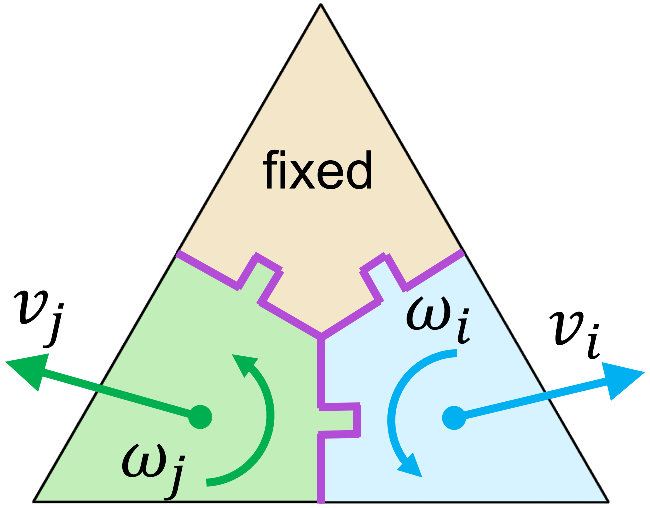
The contact constraints of a planar contact

$$v_{c1} \cdot n \geq 0$$

$$v_{c2} \cdot n \geq 0$$

Inequality-based Interlocking Test

- **Search space**: each part moves freely in the 3D space, with velocity $y_i = [v_i, \omega_i]$
- **Constraint**: no collision between parts at each contact during parts movement
- **Formulation**: a system of linear inequalities by stacking the constraints



$$B\,\boldsymbol{Y} \geq \mathbf{0}$$

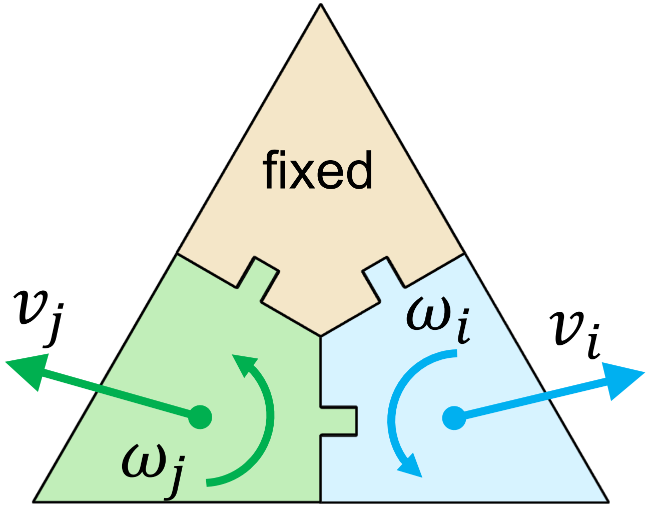
matrix of coefficients for collision-free constraints

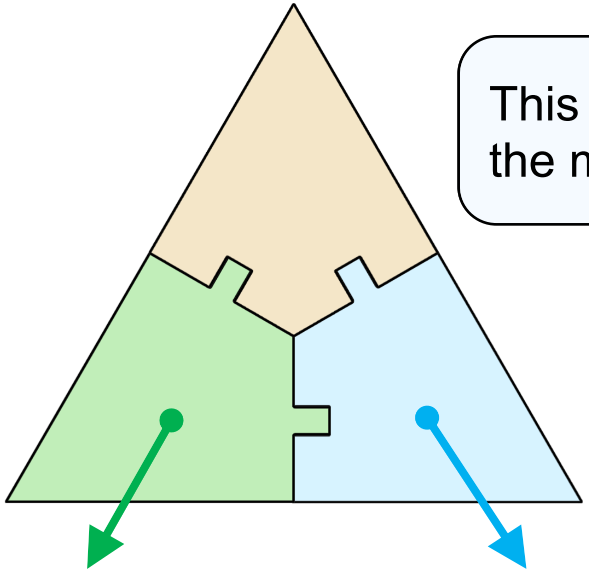
generalized velocity of the rigid body system

[Wang et al. 2019]

Inequality-based Interlocking Test

- **Search space**: each part moves freely in the 3D space, with velocity $y_i = [v_i, \omega_i]$
- **Constraint**: no collision between parts at each contact during parts movement
- **Formulation**: a system of linear inequalities by stacking the constraints $B\,\boldsymbol{Y} \geq \mathbf{0}$
- **Solve**: the assembly is interlocking if we cannot find such collision-free motion $\boldsymbol{Y}$

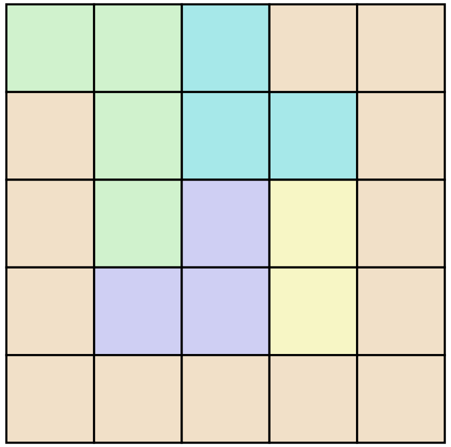




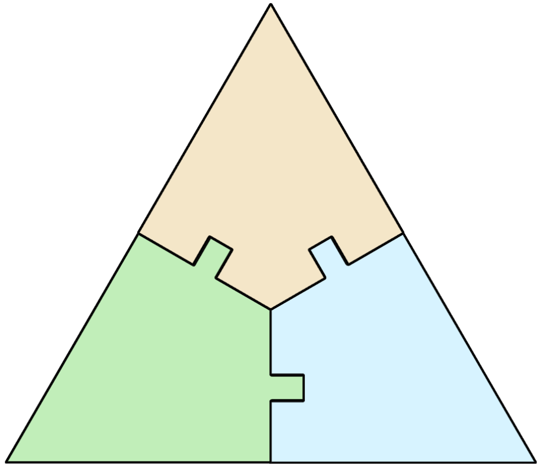
This assembly is not interlocking since the method can find collision-free motion

Inequality-based Interlocking Test

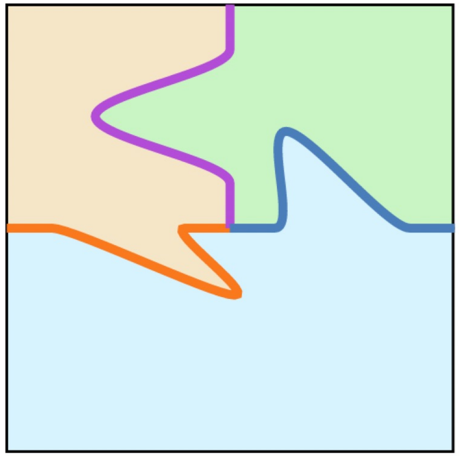
- Advantage: a general interlocking test method
  - parts can move along different directions simultaneously
  - each part can translate and rotate



planar contacts
(axis-aligned)

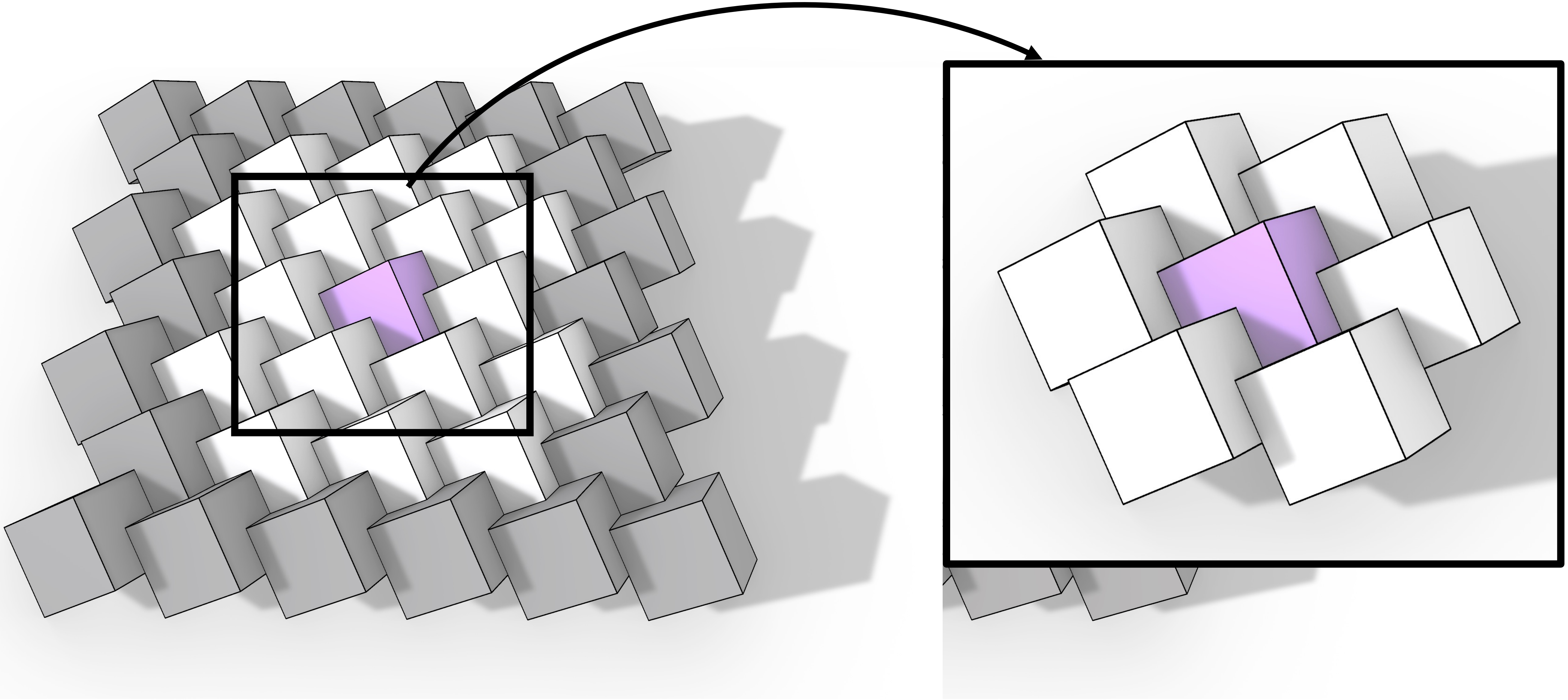


single-direction joints



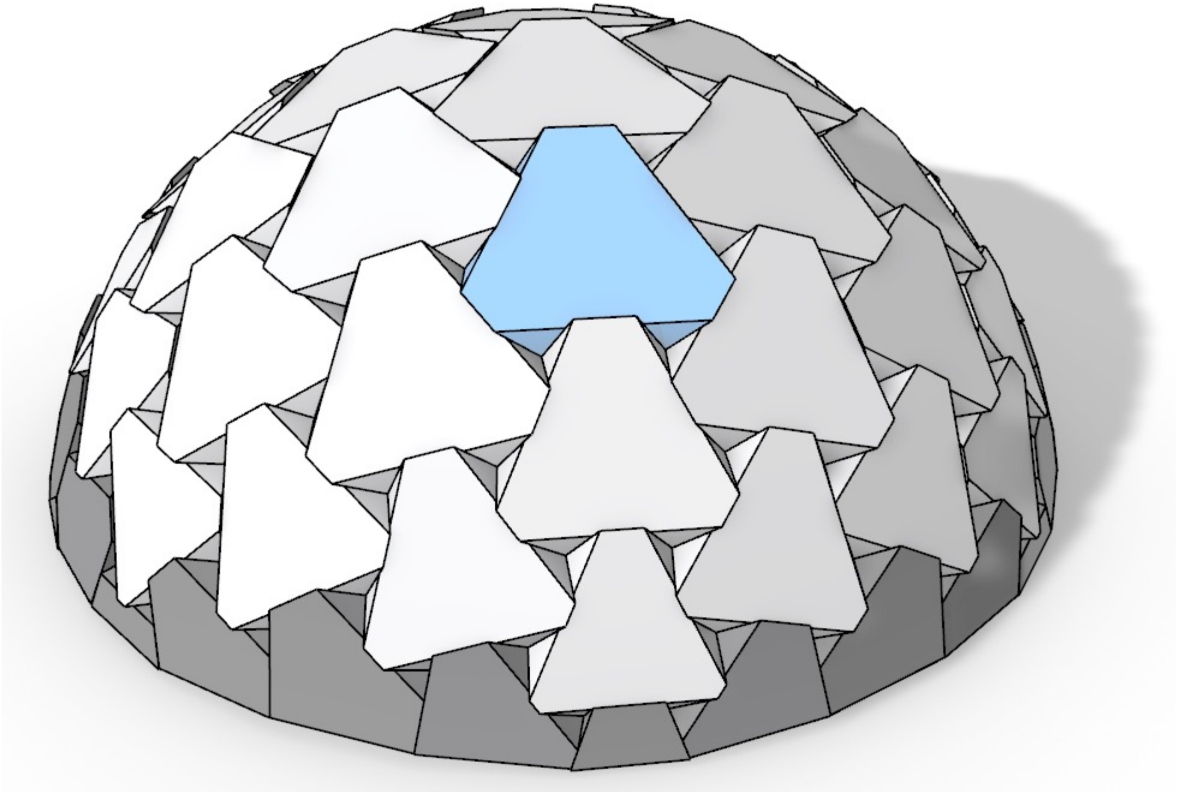
curved-contact joints

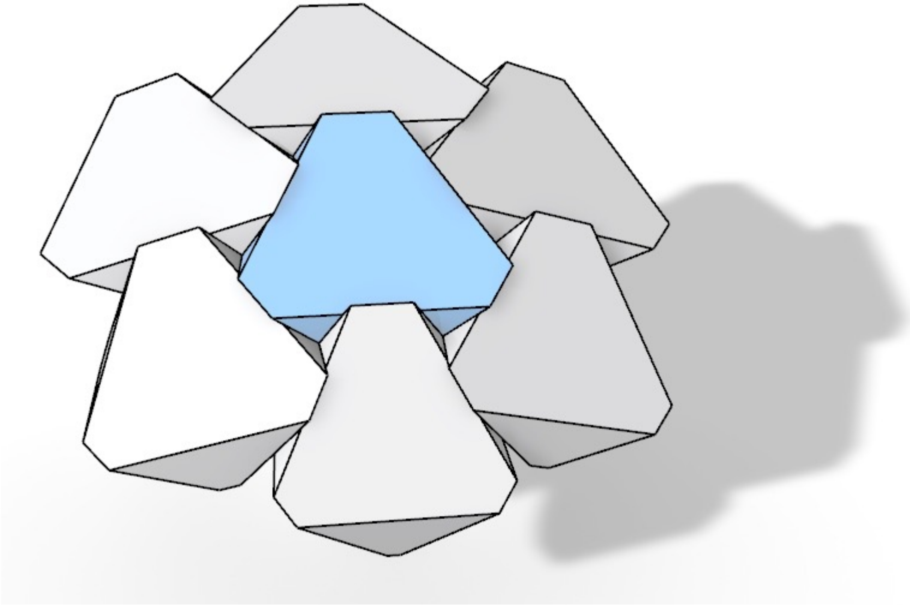
Topological Interlocking



3D Topological Interlocking

Each block is locally interlocked by its neighbors.

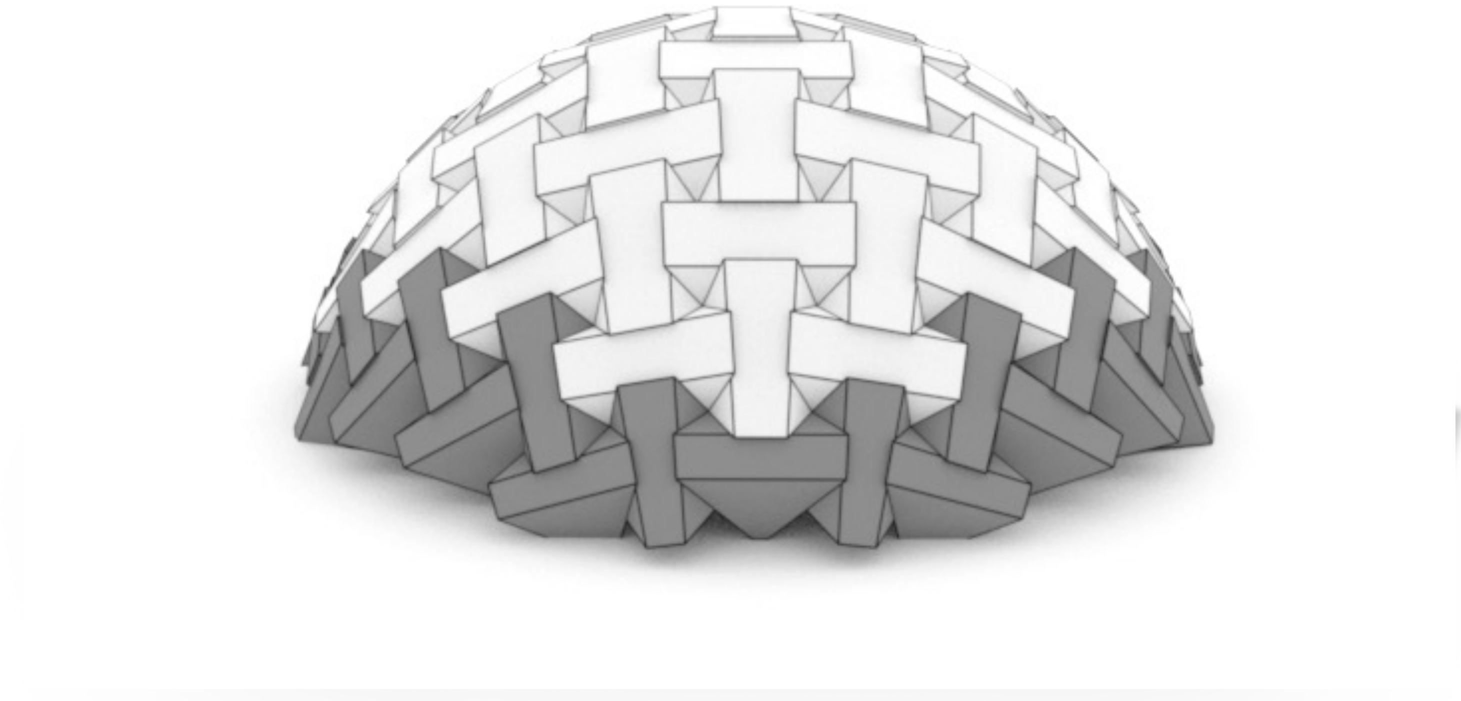




Locally Interlocking

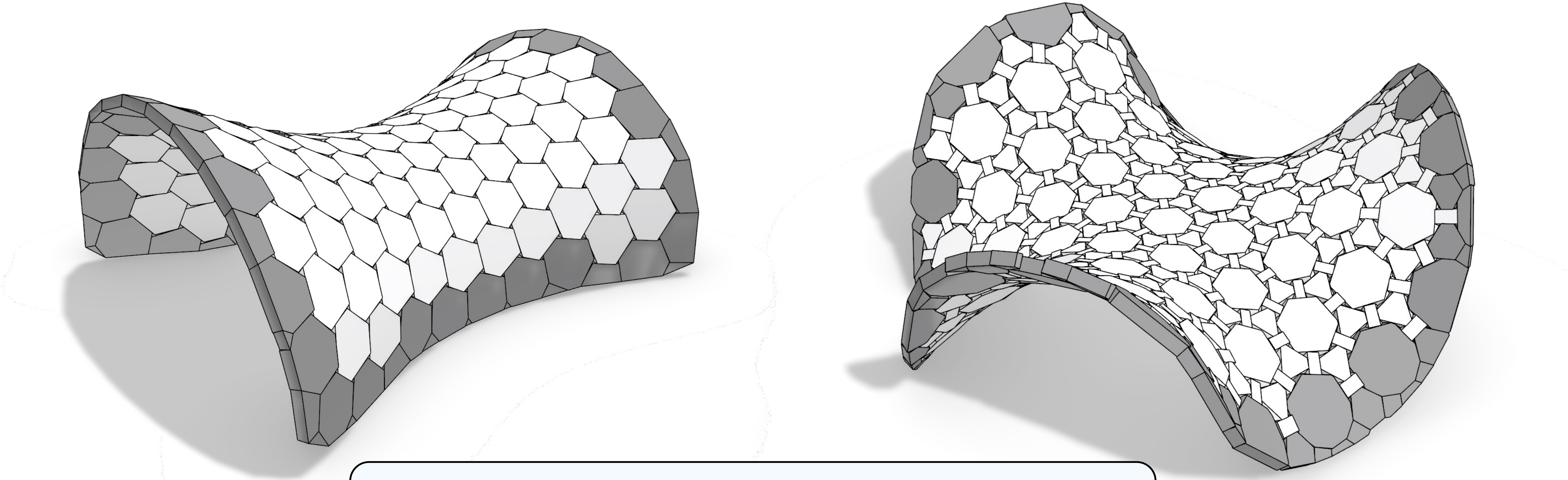
Inequality-based Interlocking Test

However, locally interlocking does not imply globally interlocking



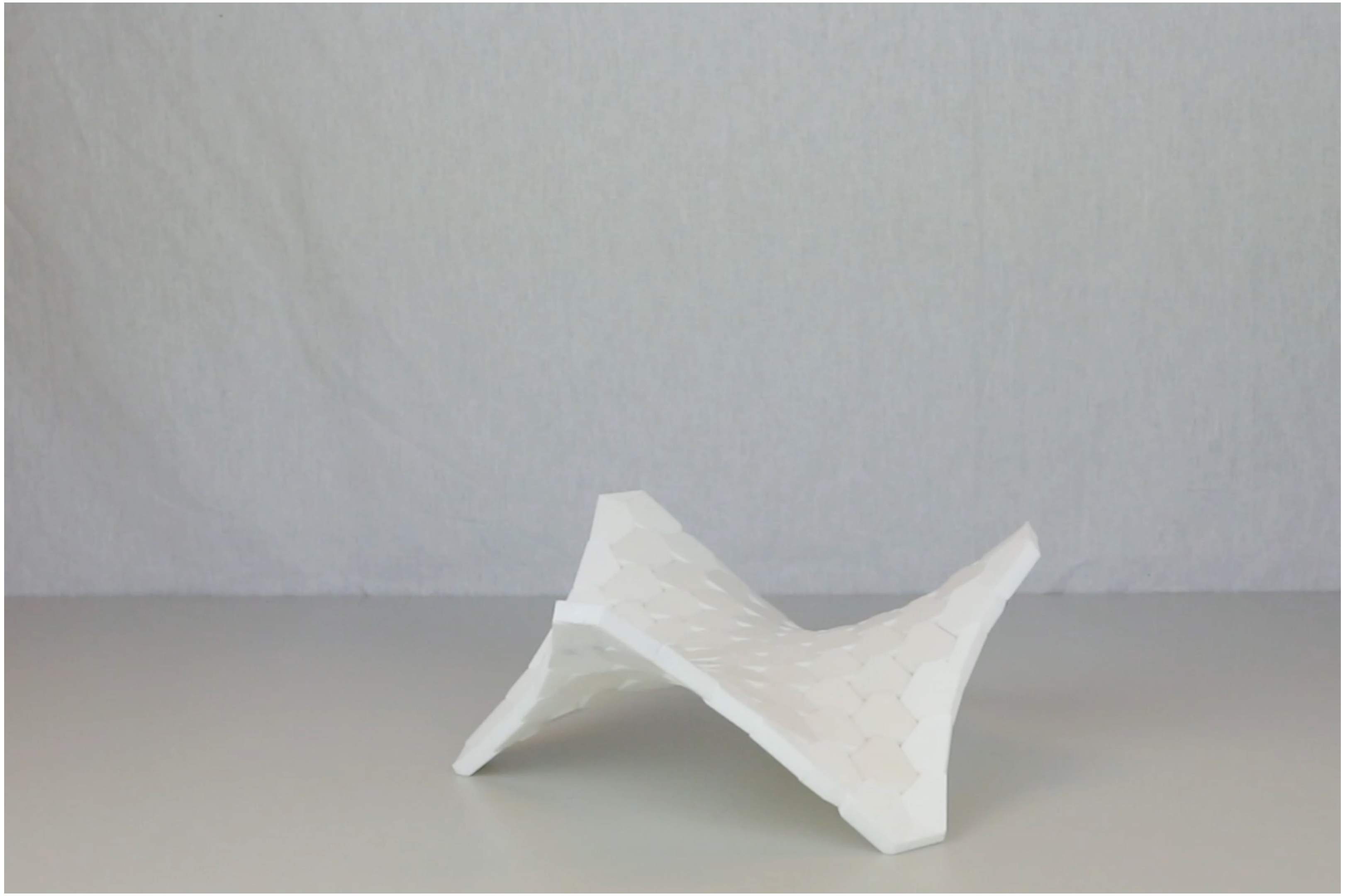
Inequality-based Interlocking Test

There exist 3D topological interlocking structures that are global interlocking

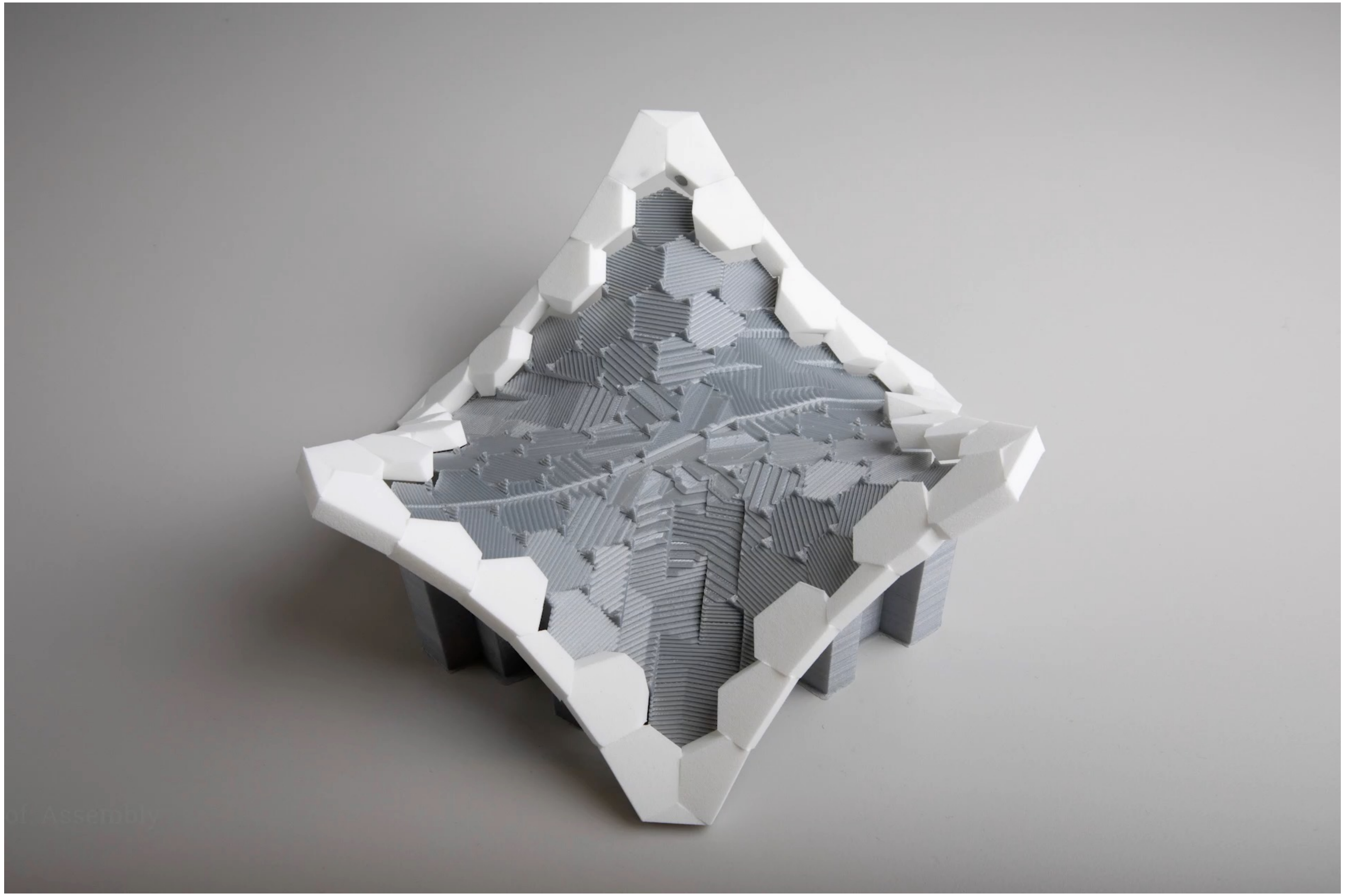


Each structure is generated from a minimal surface!

Fabrication Result



Fabrication Result



Summary

- Computational assemblies
  - Computational techniques for analyzing, designing, and fabricating assemblies
- Structurally stability
  - Static analysis
  - Tilt analysis
  - Interlocking test
- Computational design of interlocking assemblies
  - Enumeration-based method
  - DBG-based method
  - Inequality-based method

Thank You!

More information can be found at:
https://sutd-cgl.github.io/