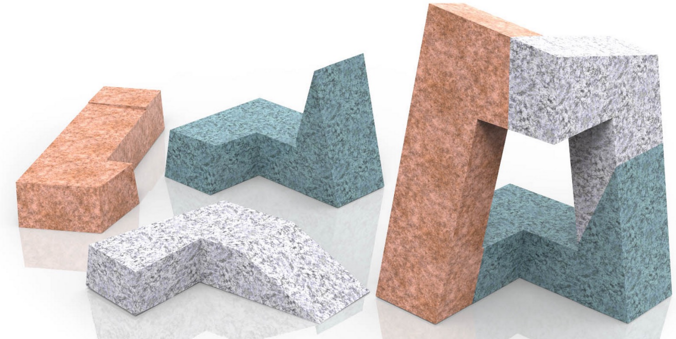
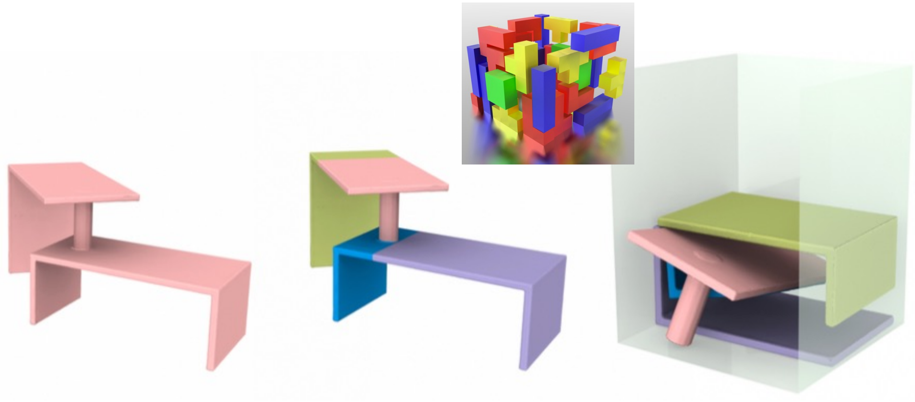
Pyramids, Tetris, and Spirals: New Geometry Problems for 3D Printing



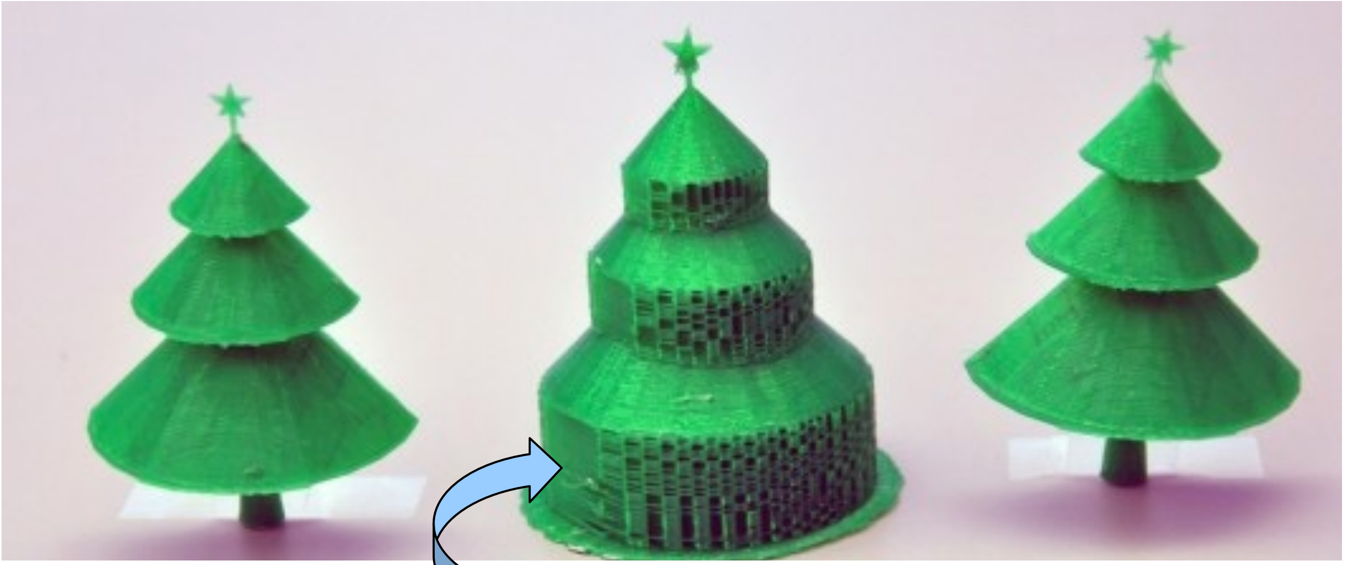


CMPT 464/764

Lecture 15

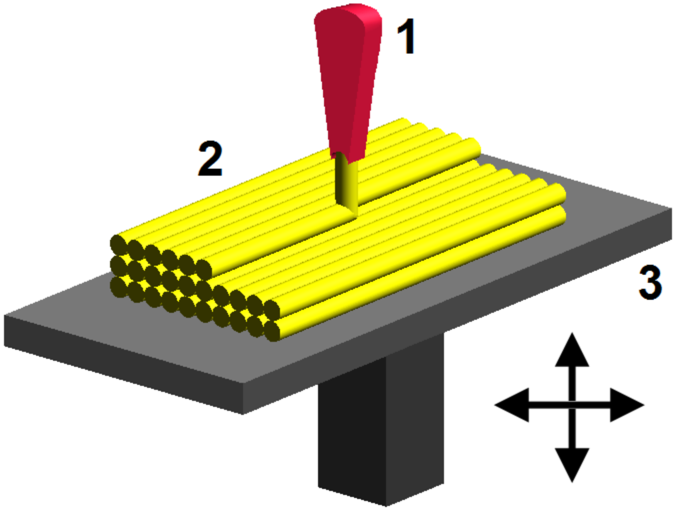
Important 3DP criteria we consider

- Cost saving: print time and material usage
  - Typically takes hours of time …



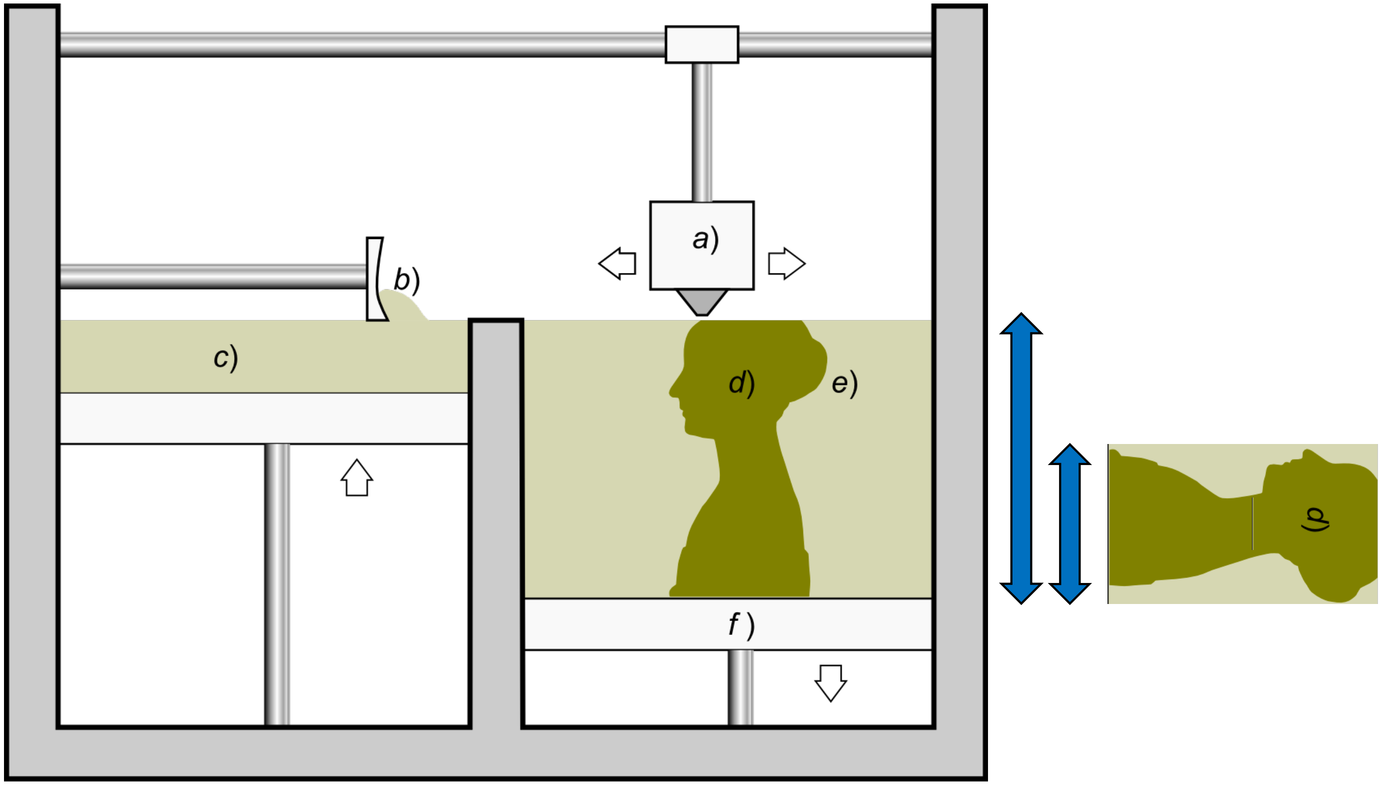
Support material waste in FDM

FDM: Fusion Deposition Modeling



Fuse deposition modeling (FDM) – minimizing total printed material

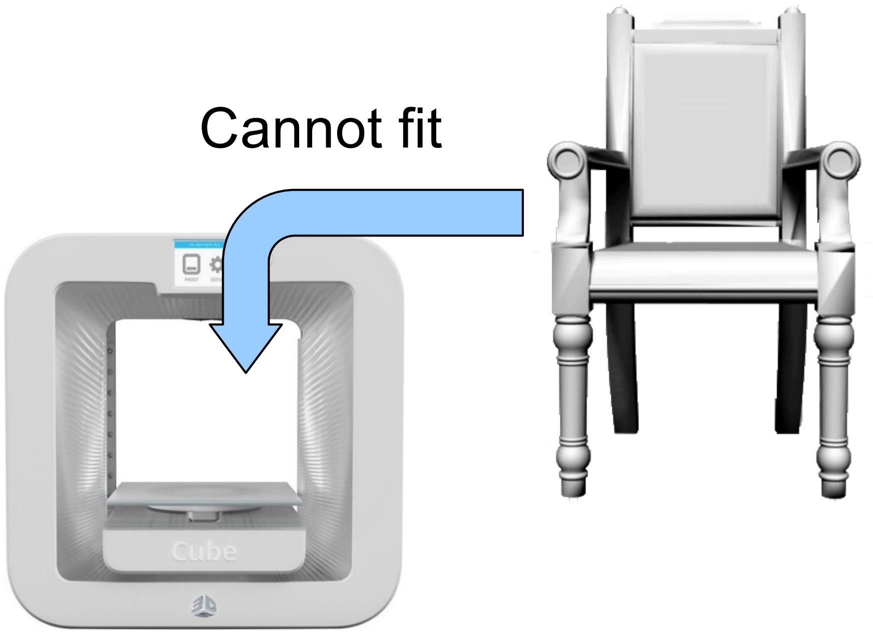
Powder-based printing



Powder-based 3D printing –
minimizing object height

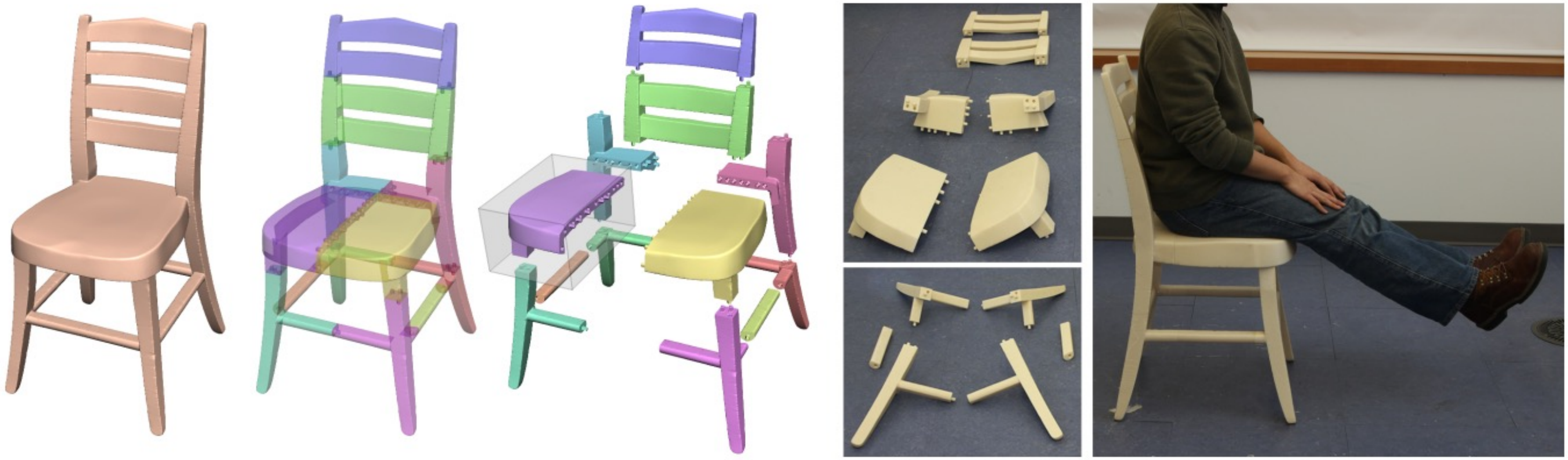
Important 3DP criteria we consider

- Cost saving: print time and material usage
- Best utilization of limited print volume



Chopper

- Decompose a large 3D object
  - Each part fits inside print volume



[Luo et al. SIG Asia 2012]

Decompose-and-pack (DAP) problem

- Decompose and pack a 3D object optimally
  - Combine packing with decomposition
  - Best utilization of limited print volume

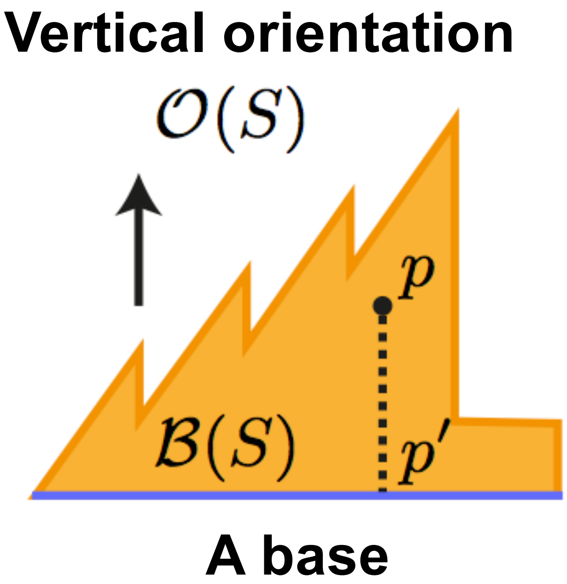
Decompose-and-pack (DAP) problem

- Decompose and pack a 3D object optimally
  - Combine packing with decomposition
  - Best utilization of limited print volume
- Important: do not decompose-and-then-pack
  - Two optimization problems must be strongly coupled
- Seems to be a very difficult problem

Re-thinking of an “easier” problem?

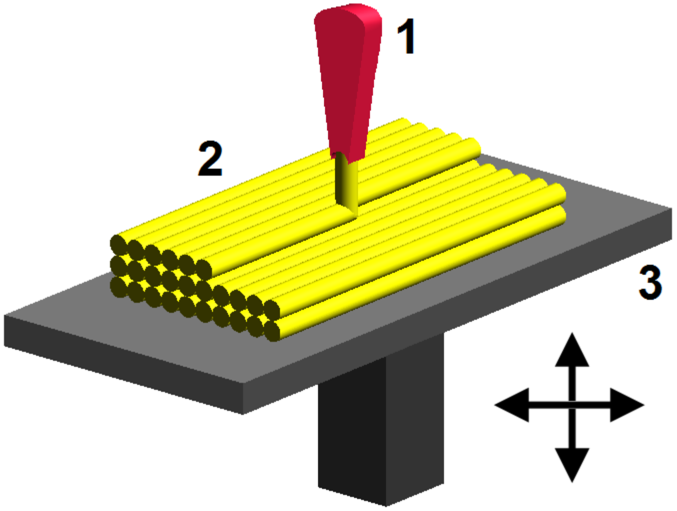
- Let us only decompose, no packing
- But beyond just fitting into print volume (Chopper)
- Decompose so each part is best for 3D printing
- So what geometric property would be best?

Pyramidal (terrain) shape



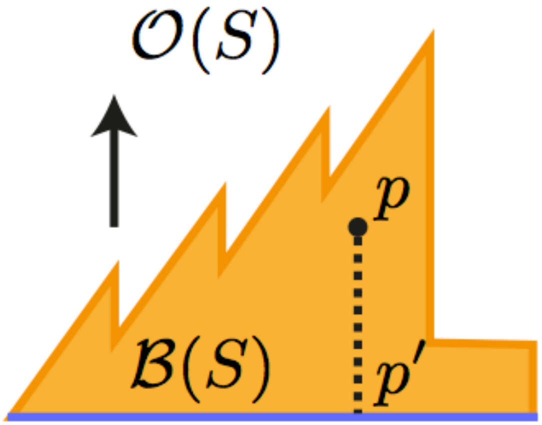
A pyramidal shape models a height function over the base.

No support (waste) material for layered (FDM) printing; also shorter print time.

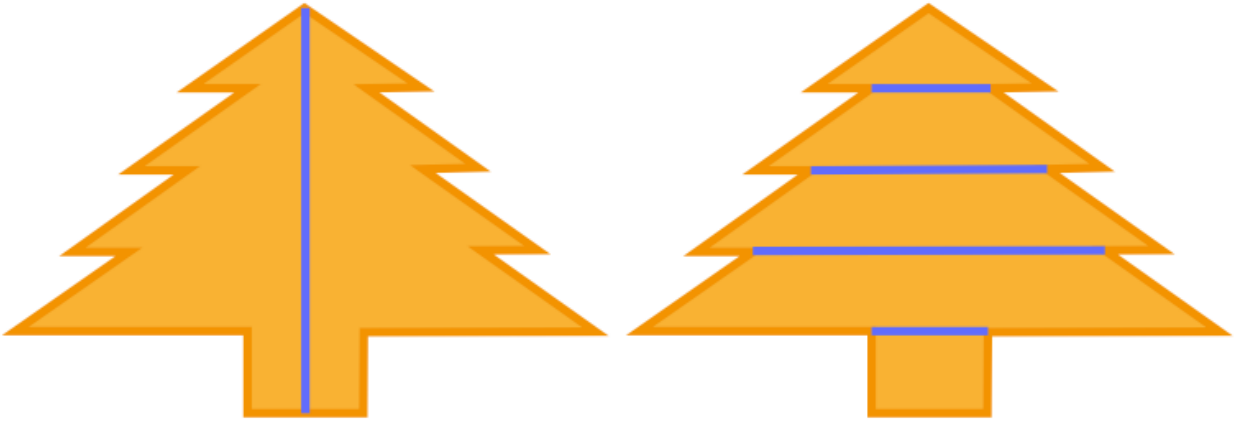


Pyramidal (terrain) shape

Vertical orientation



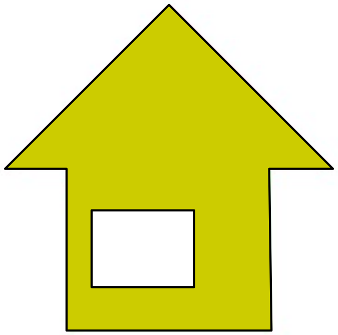
A base



Pyramidal vs. convex decompositions

2014: pyramidal decomposition

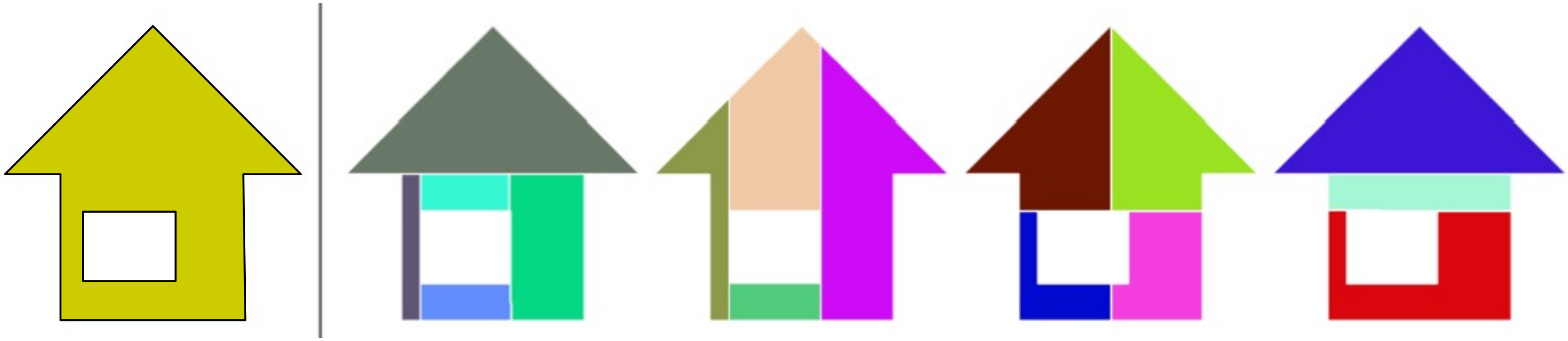
- Goal: decompose into min# of pyramidal parts
- How hard is this problem for humans?



What is the best you can do?

2014: pyramidal decomposition

- Goal: decompose into min# of pyramidal parts
- How hard is this problem for humans?

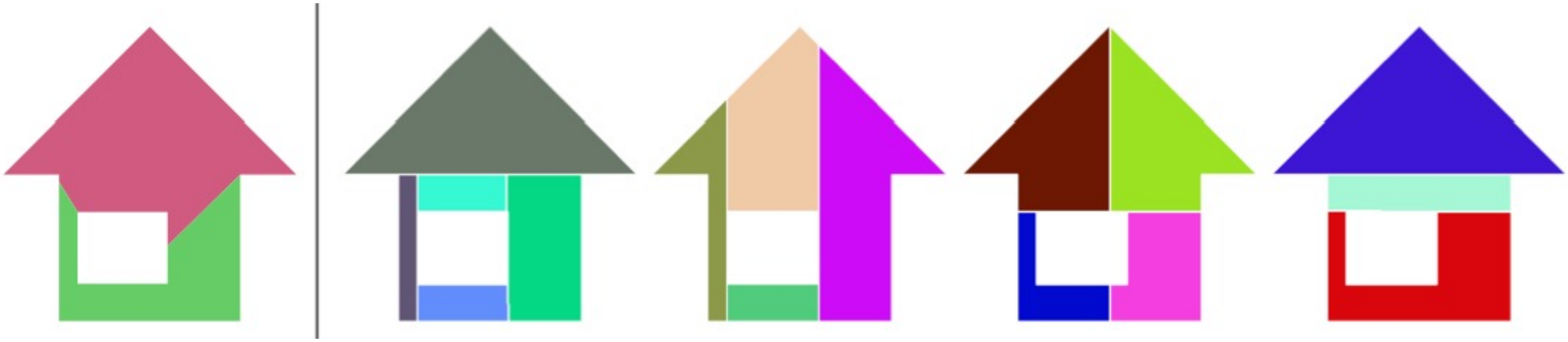


What is the best you can do?

Results obtained by human users

2014: pyramidal decomposition

- Goal: decompose into min# of pyramidal parts
- How hard is this problem for humans?



Our solution

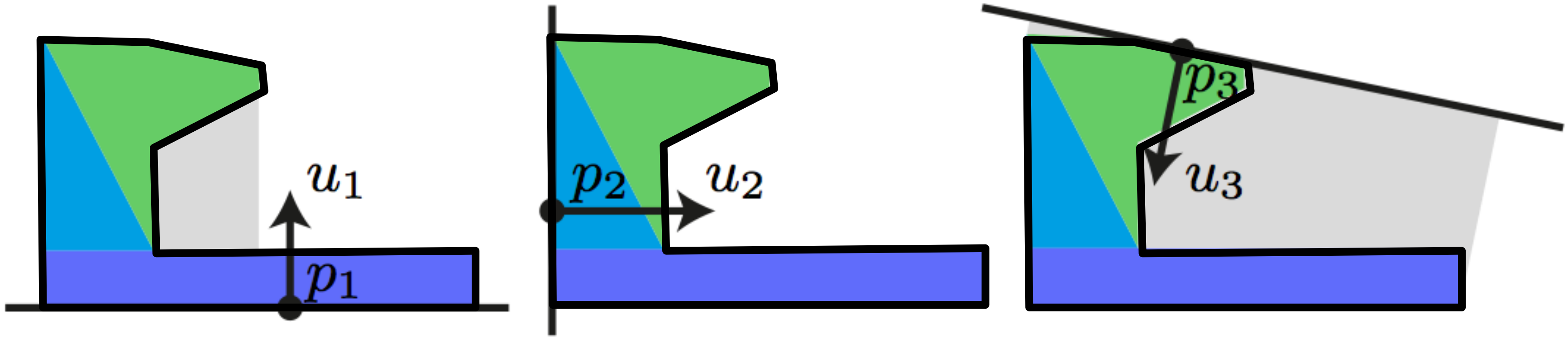
Results obtained by human users

[Hu et al. 2014]

Exact vs. approximation decomposition

- Exact pyramidal decomposition is NP-hard
  - Proved for 3D shapes and 2D polygons with holes [Fekete and Mitchell 2001]
- Exact decomposition may lead to too many parts
- New problem: approx pyramidal decomposition
  - APD: Each part is only approximately pyramidal
  - Still seeks as few parts as possible

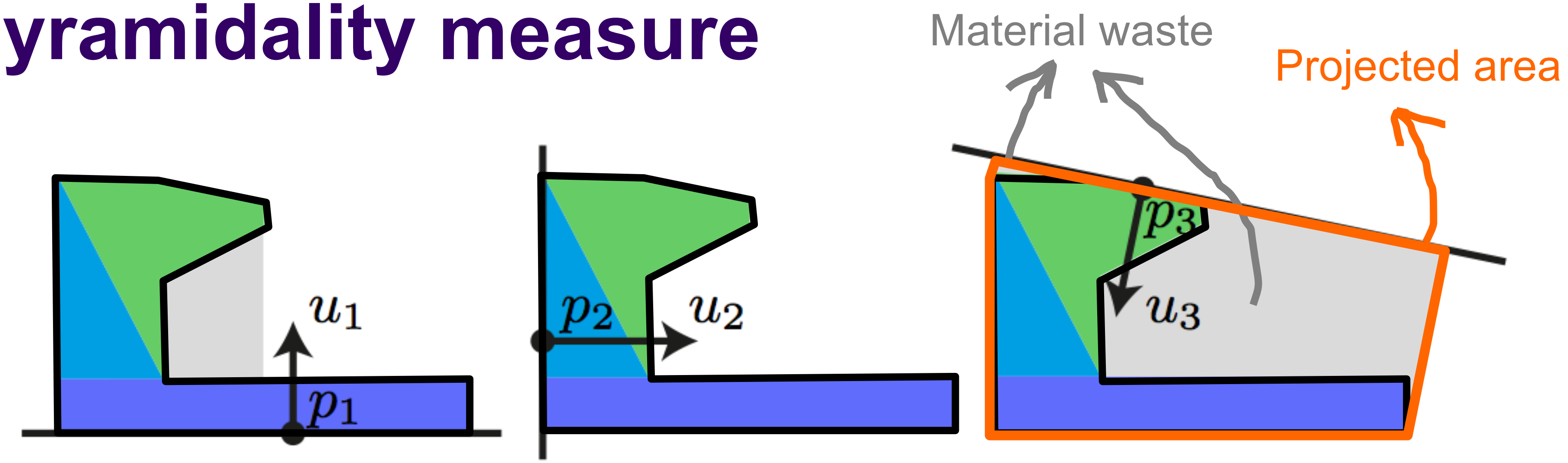
Pyramidality measure



Pyramidality estimated along three directions u_1, u_2 and u_3

- Pyramidality of part is estimated over all directions

Pyramidality measure

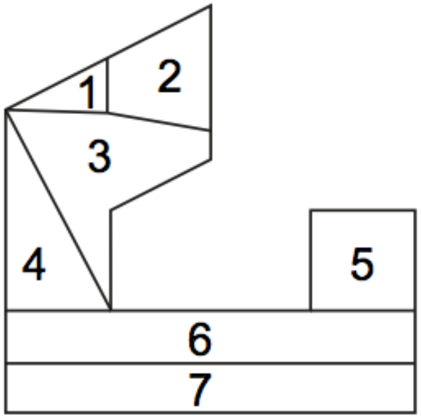


Pyramidality estimated along three directions u_1, u_2 and u_3

- Pyramidality of part is estimated over all directions
- Take direction with the least (estimated) material waste relative to projected area

Construction algorithm

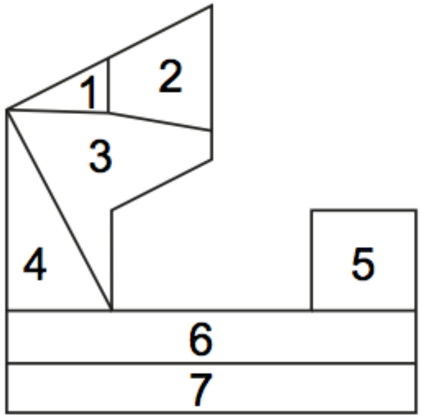
- Convert APD into an Exact Cover Problem (ECP)



A block partition

Construction algorithm

- Convert APD into an Exact Cover Problem (ECP)



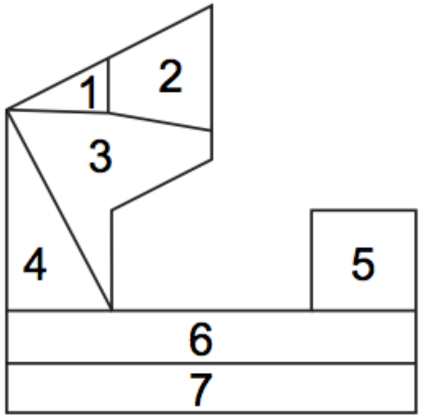
A block partition

{1} {2} {3} {4} {5} {6} {7} {1,2} {1,3} {3,4} {4,6} {5,6} {6,7}
{1,2,3} {3,4,6} {4,5,6} {4,6,7} {5,6,7} {3,4,6,7} {4,5,6,7}

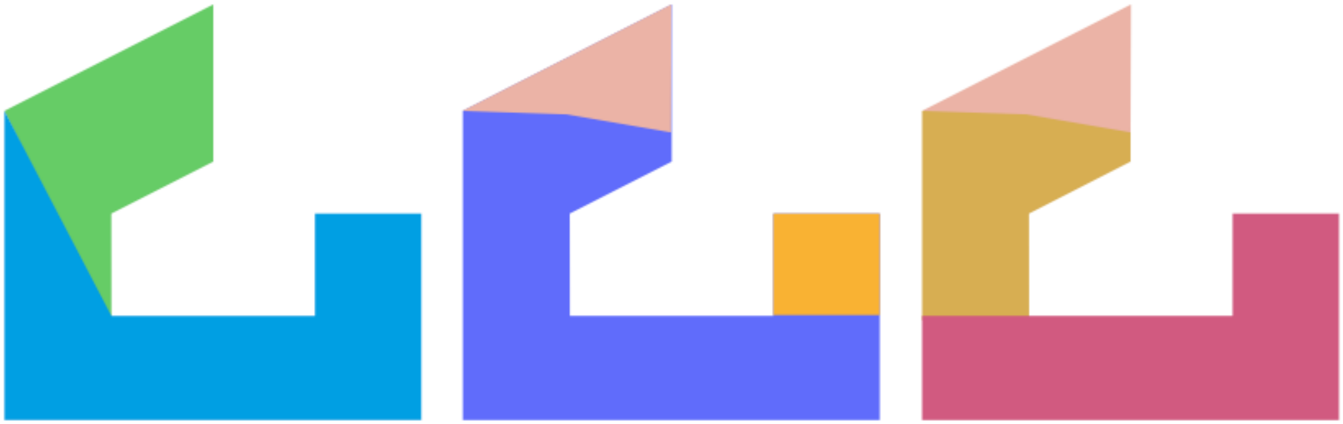
A cover consisting of a set of candidate pyramidal parts

Construction algorithm

- Convert into an Exact Cover Problem (ECP)



A block partition



Solutions to ECP by Algorithm X [Knuth 2000]

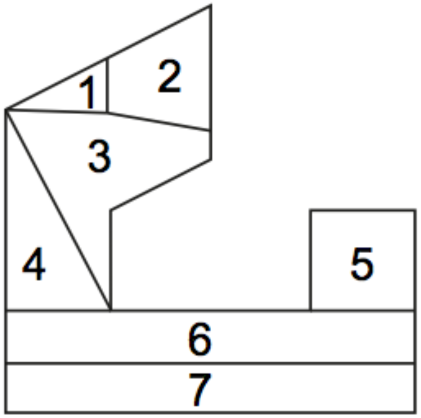
ECP is NP-complete.

Algorithm X efficiently enumerates all solutions to ECP.

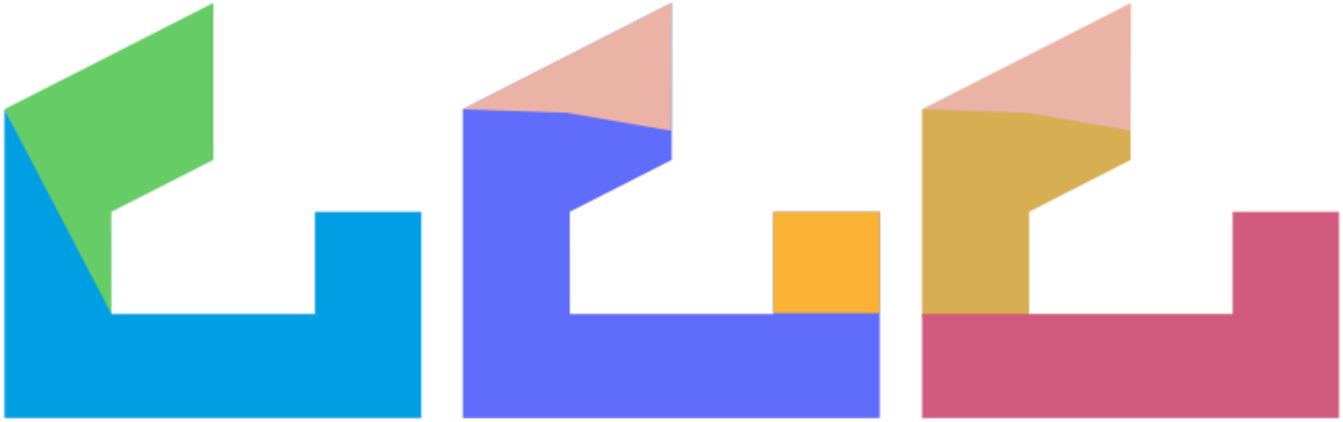
Any objective function can be employed to pick solutions.

Construction algorithm

- Convert into an Exact Cover Problem (ECP)



A block partition

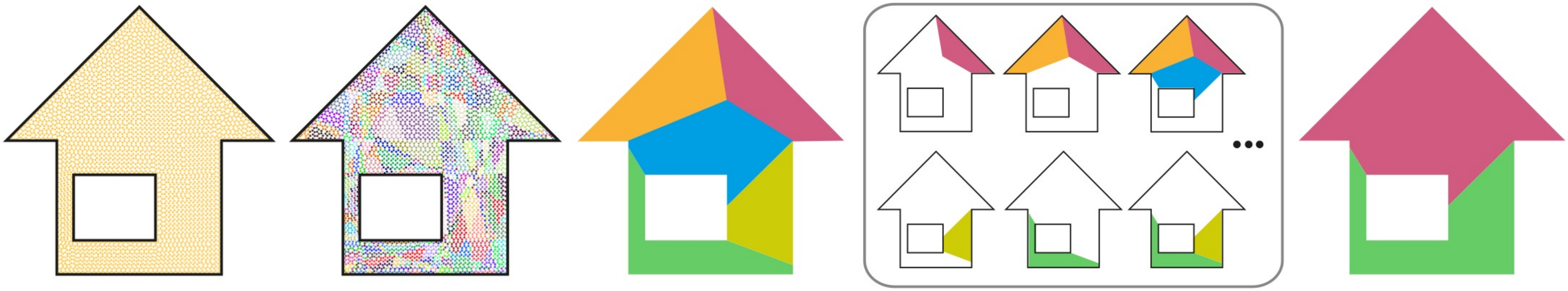


Solutions to ECP by Algorithm X [Knuth 2000]

How to get the set of candidate pyramidal parts?

A clustering approach

- Progressively build larger & larger building blocks

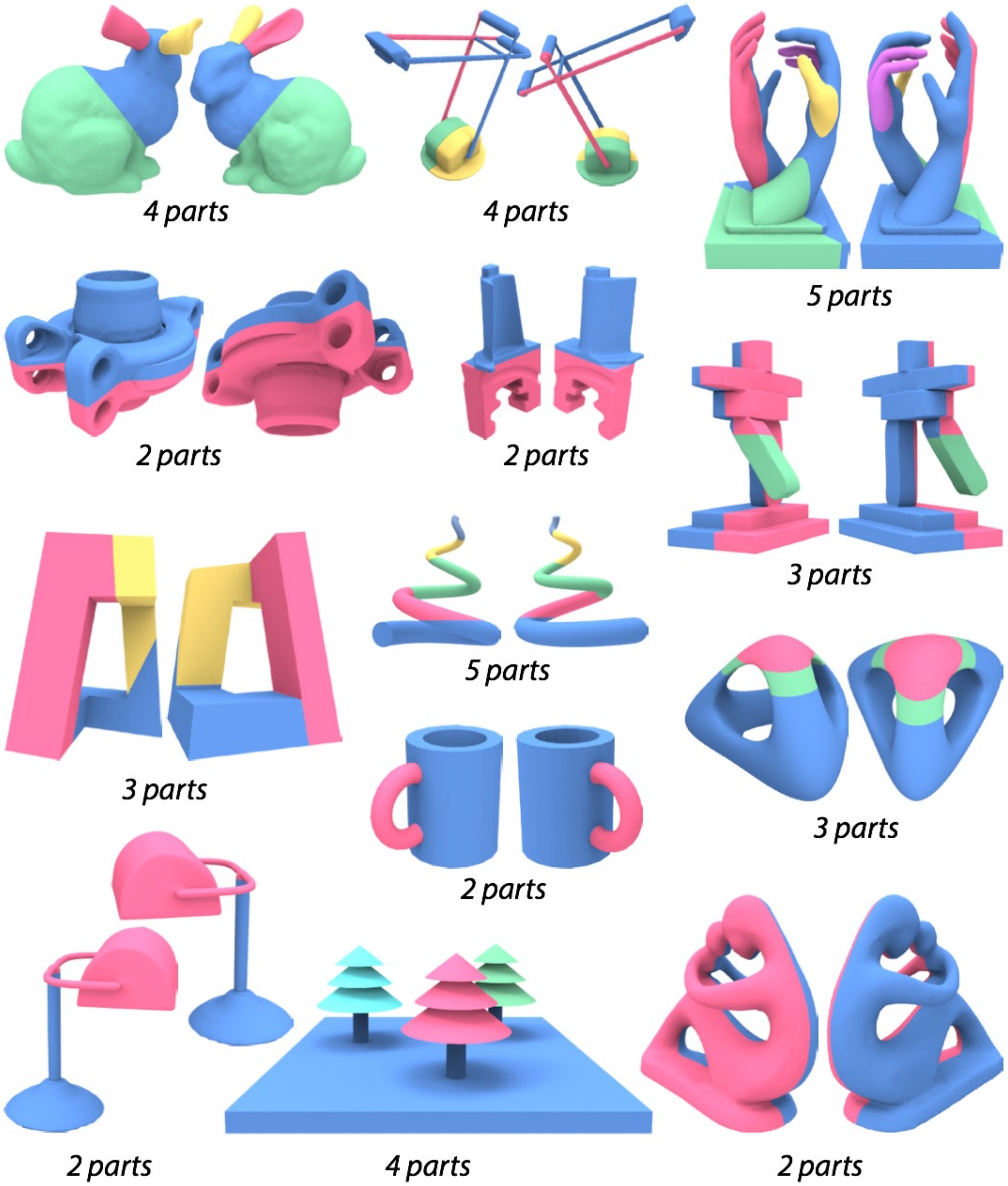


(a) Point samples (b) Cells (c) Blocks (d) Candidate pyramidal parts (e) Decomposition

- Key clustering criterion: group elements that are likely to belong to large pyramidal parts

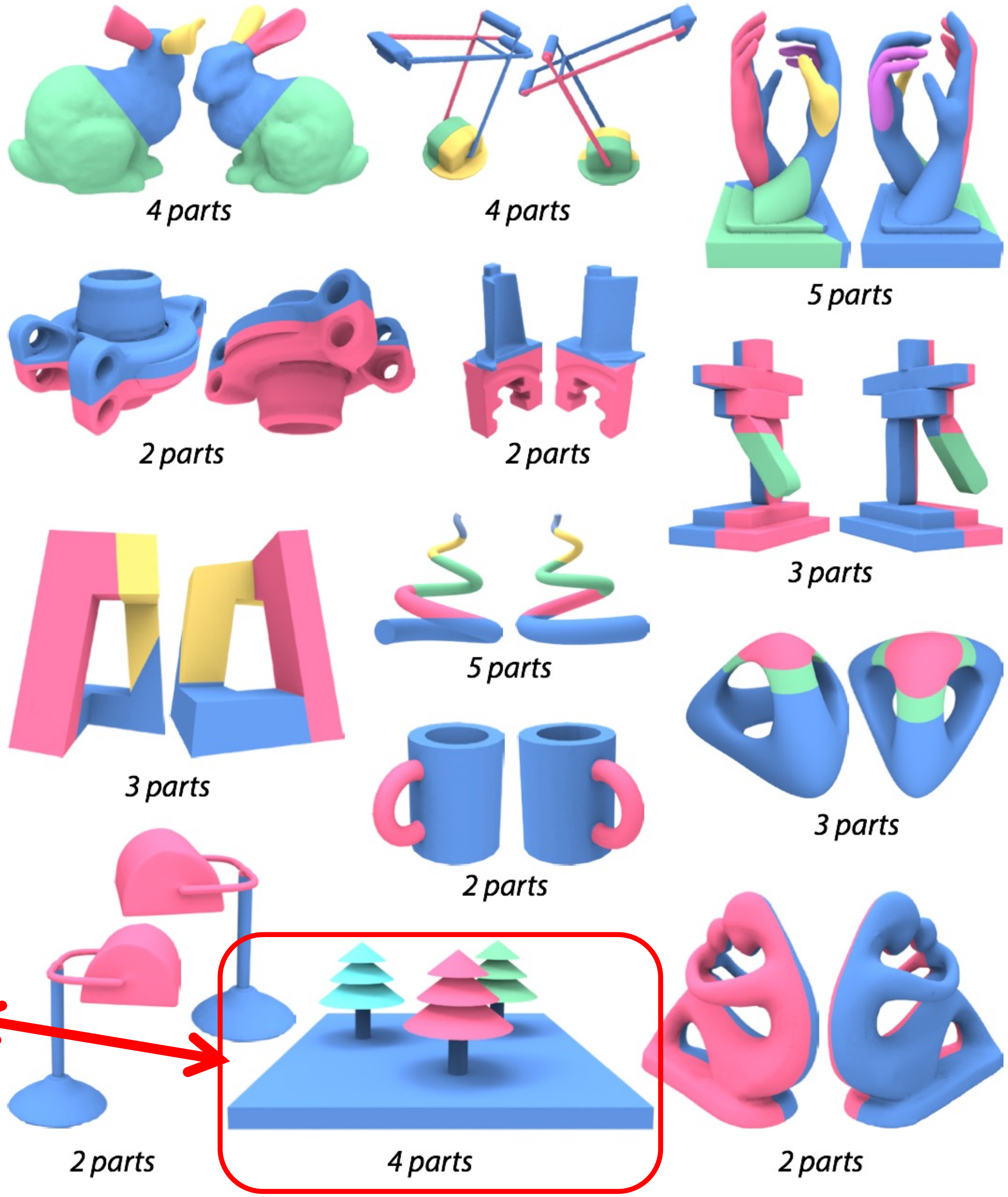
Results

Results combining **material saving** and **part count**

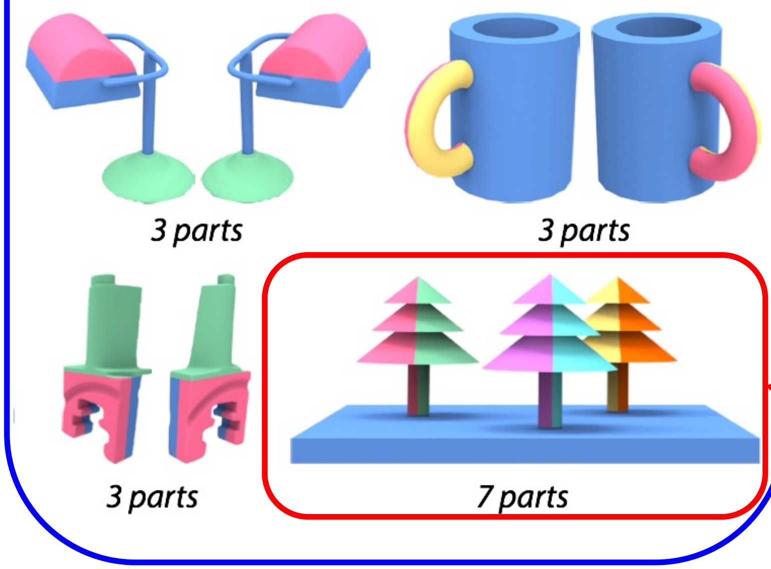


Results

Results combining **material saving** and **part count**



Best results in terms of only material saving



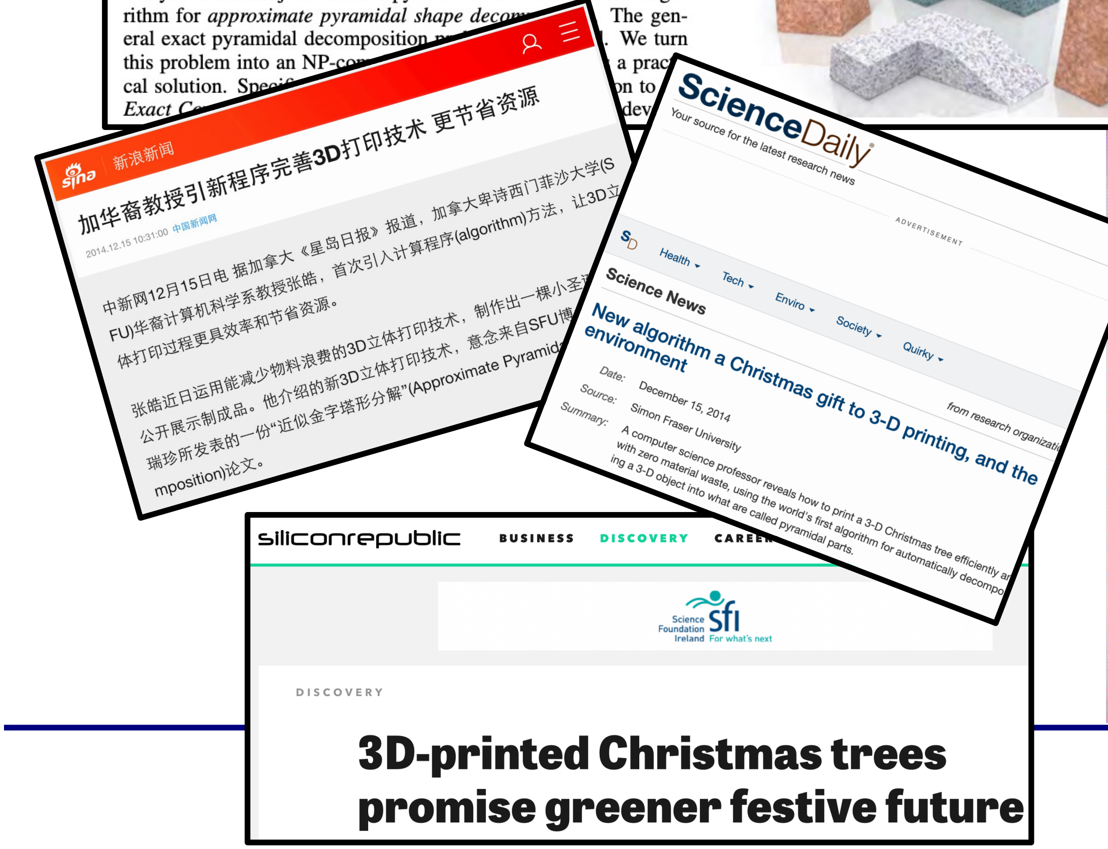
Paper and press coverage

Approximate Pyramidal Shape Decomposition

Ruizhen Hu[1,2*] Honghua Li[1] Hao Zhang[1] Daniel Cohen-Or[3]
[1]Simon Fraser University [2]Zhejiang University [3]Tel Aviv University

Abstract

A shape is pyramidal if it has a flat base with the remaining boundary forming a height function over the base. Pyramidal shapes are optimal for molding, casting, and layered 3D printing. However, many common objects are not pyramidal. We introduce an algorithm for *approximate pyramidal shape decom*
The general exact pyramidal decomposition pr
. We turn this problem into an NP-co
a practical solution. Speci
on to *Exact Co*
dev



加华裔教授引新程序完善3D打印技术 更节省资源

2014.12.15 10:31:00 中国新闻网

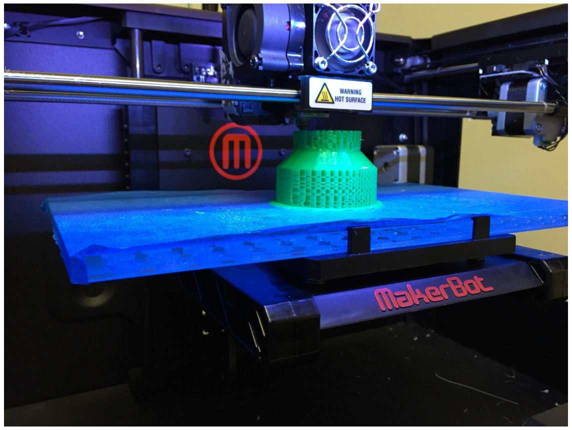
中新网12月15日电 据加拿大《星岛日报》报道，加拿大卑诗西门菲沙大学(S
FU)华裔计算机科学系教授张皓，首次引入计算程序(algorithm)方法，让3D立
体打印过程更具效率和节省资源。

张皓近日运用能减少物料浪费的3D立体打印技术，制作出一棵小圣
公开展示制成品。他介绍的新3D立体打印技术，意念来自SFU博
瑞珍所发表的一份“近似金字塔形分解”(Approximate Pyramida
mposition)论文。

New algorithm a Christmas gift to 3-D printing, and the environment

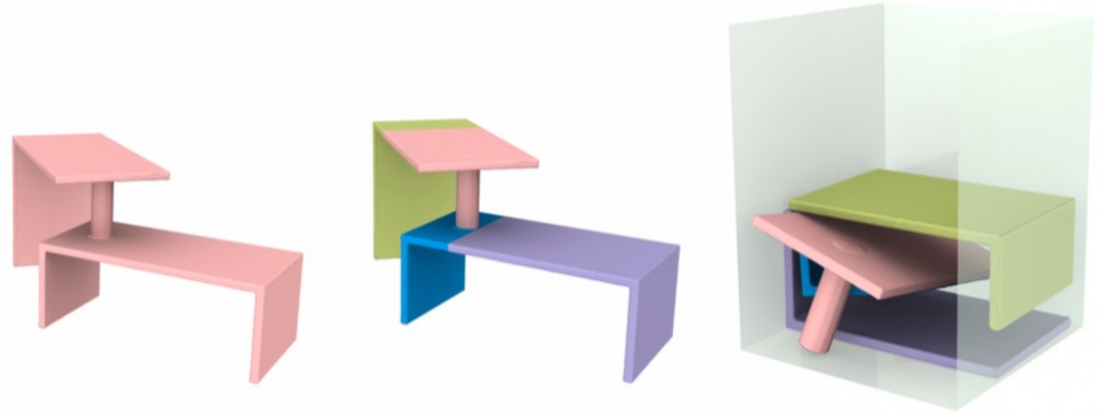
Date: December 15, 2014
Source: Simon Fraser University
Summary: A computer science professor reveals how to print a 3-D Christmas tree efficiently ar
with zero material waste, using the world's first algorithm for automatically decompo
ing a 3-D object into what are called pyramidal parts.

3D-printed Christmas trees promise greener festive future



Back to DAP

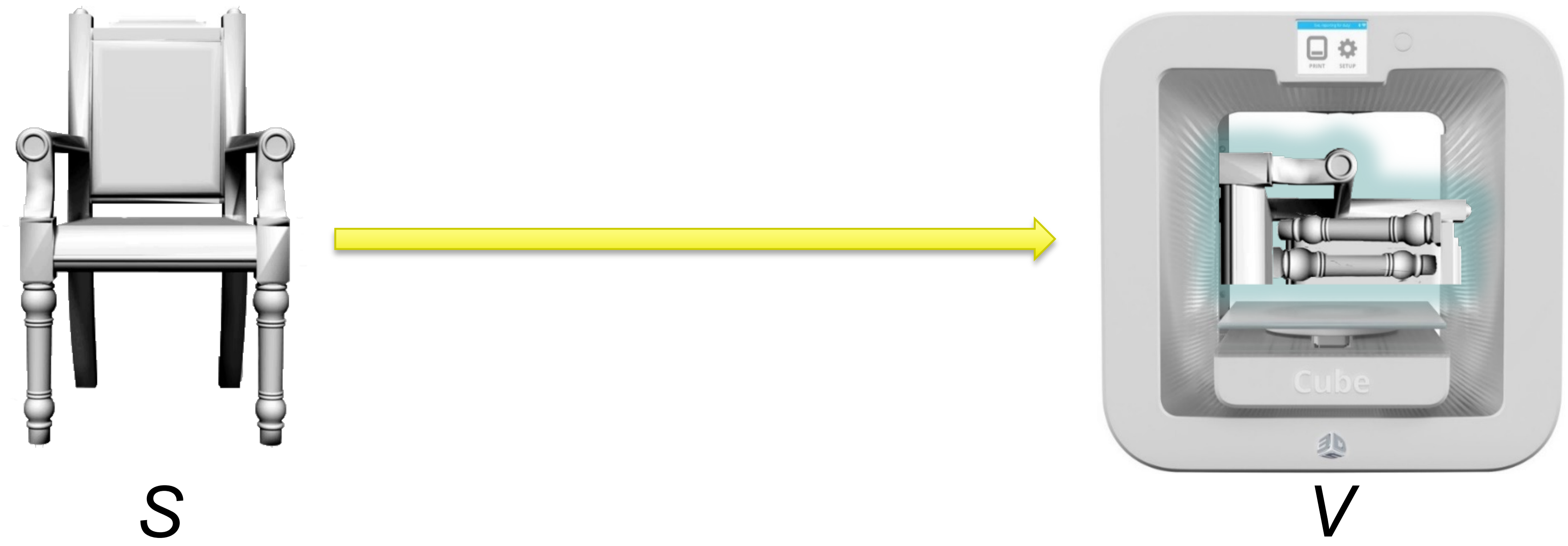
- Better utilize the print volume, material, and time
  - Dapper: Decompose AND Pack (DAP) a 3D object
  - Especially attractive for powder-based 3D printing



[Chen et al. 2015]

DAP problem

- Given a 3D shape *S* and print volume *V*, decompose *S* into a small number of parts to be packed compactly into *V*



DAP problem

- Geared towards efficient 3D printing
  - support material, build time, and assembly cost
- Adjustable for powder and FDM 3D printing
- Object function combines part count with printing criteria DM)

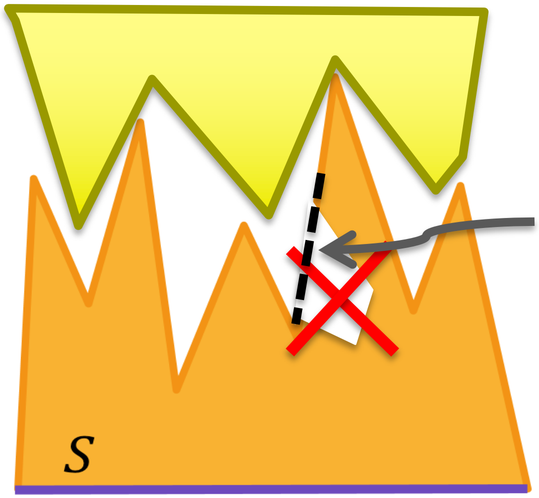
DAP: Must solve D AND P, not D-and-THEN-P

Making the problem tractable

- Restrict the geometric primitives for DAP
  - Search space too large for arbitrary primitives
  - Arbitrary primitives are also difficult to pack
- Restrict cut and packing directions
- Settling for heuristics and sub-optimality

Primitives: pyramidal parts

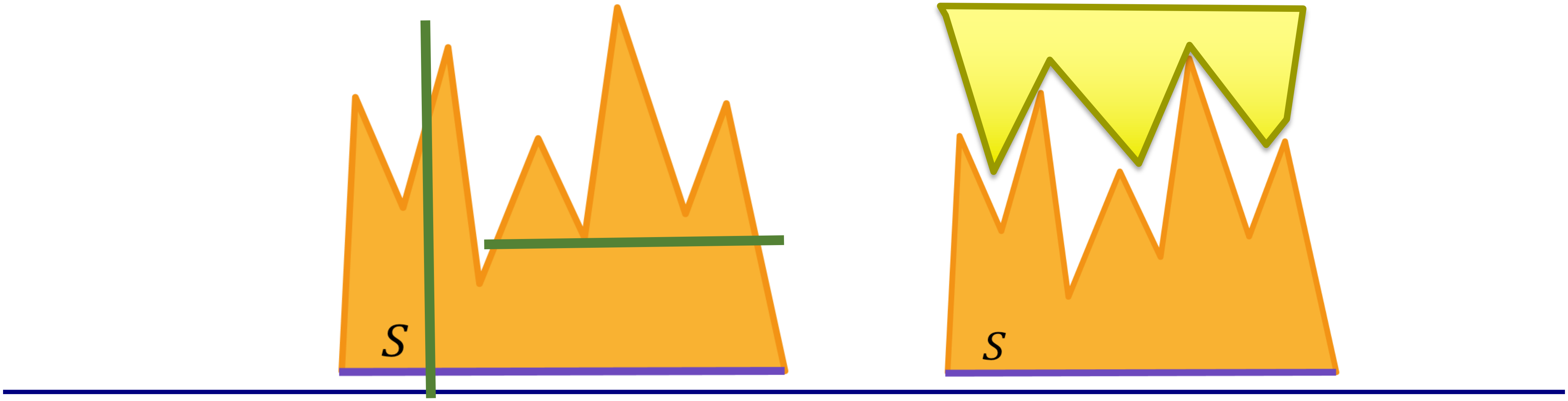
- Not only printing-friendly, also packing-friendly
  - No inner pockets to fill
  - Packing = matching of only one side, the "teeth" side



No inner pockets

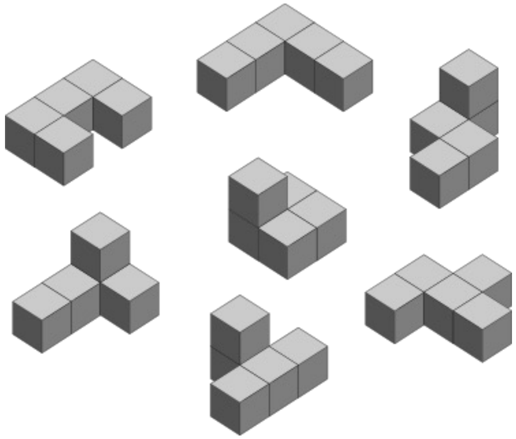
Primitives: pyramidal parts

- Not only printing-friendly, also packing-friendly
  - No inner pockets to fill
  - Packing = matching of only one side, the “teeth” side
- Decomposition: closure under axial cuts



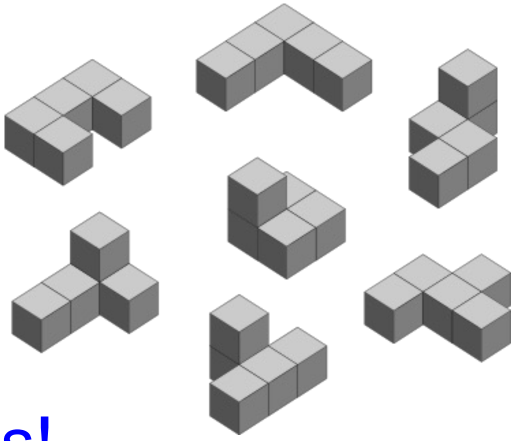
Further search reduction

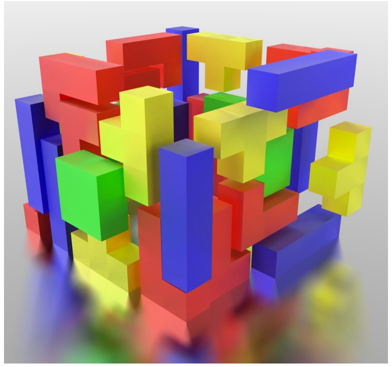
- Decompose into and pack only pyramidal polycubes
- Voxelize input shape and only axial cuts
  - Closure property with pyramidal primitives
- 90x degree rotations for packing



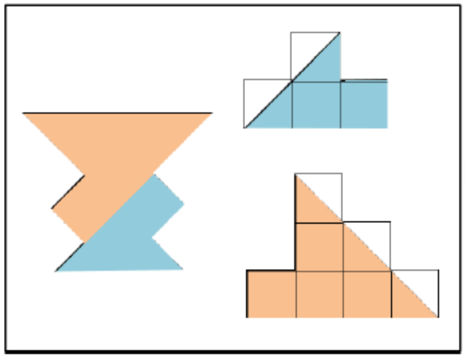
Further search reduction

- Decompose into and pack only pyramidal polycubes
- Voxelize input shape and only axial cuts
  - Closure property with pyramidal primitives
- 90x degree rotations for packing
- Problem is more fun: like playing 3D Tetris!





Algorithm overview

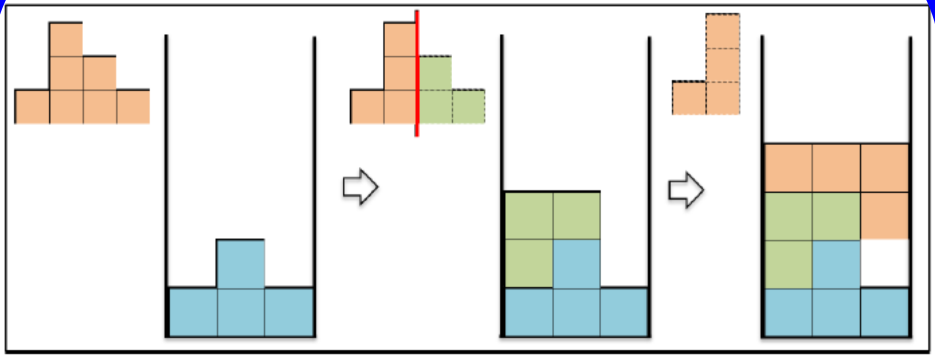


Initial decomposition

Pyramidal polycubes

Cut-n-Pack

Pack

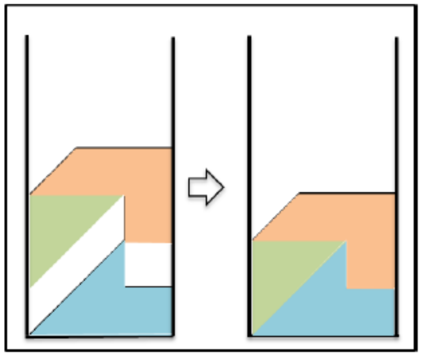


Global DAP optimization

Priority-driven & bounded beam search

Depth and breadth pruning via local and global DAP objectives

Optimal pile

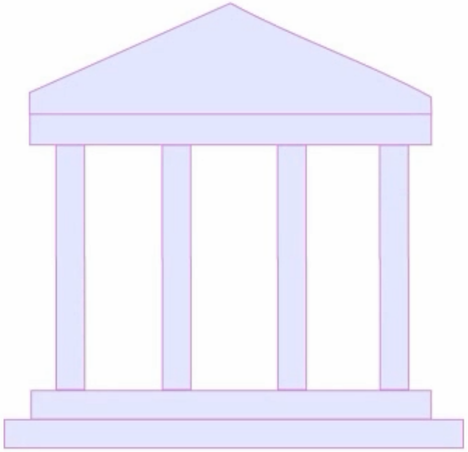


Local refinement

Compact pile

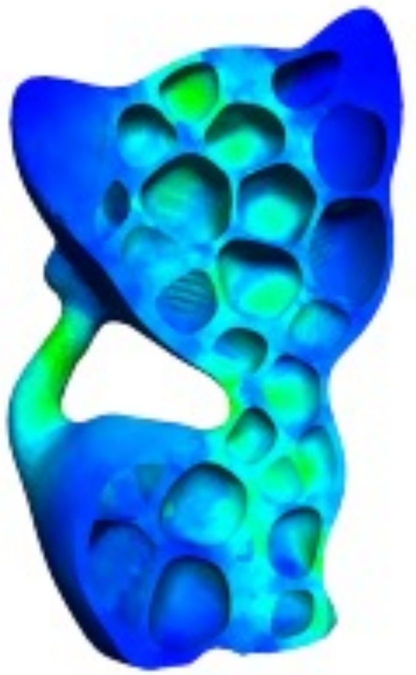
[Chen et al. SIG Asia 2015]

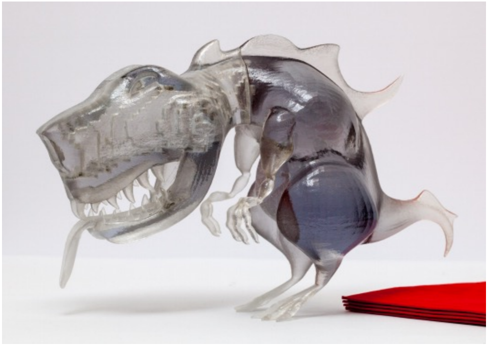
DAP like playing Tetris (video)



From “what” to “how” to print

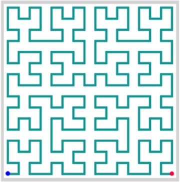
- Fabrication-aware (input) design: optimize the input 3D shape for fabrication = what to print





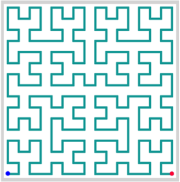
Tool path planning

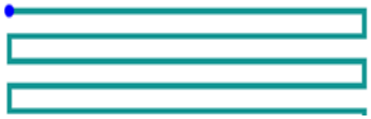
- Tool path fill = space-filling curve

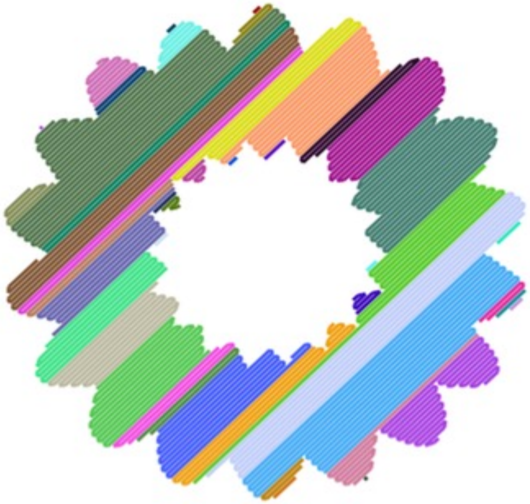


- Choice of tool path affects print time, inner fill + surface quality

Tool path planning

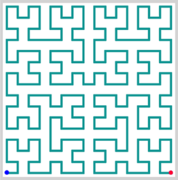
- Tool path fill = space-filling curve 
- Most popular tool path pattern: zigzag



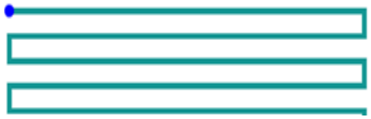


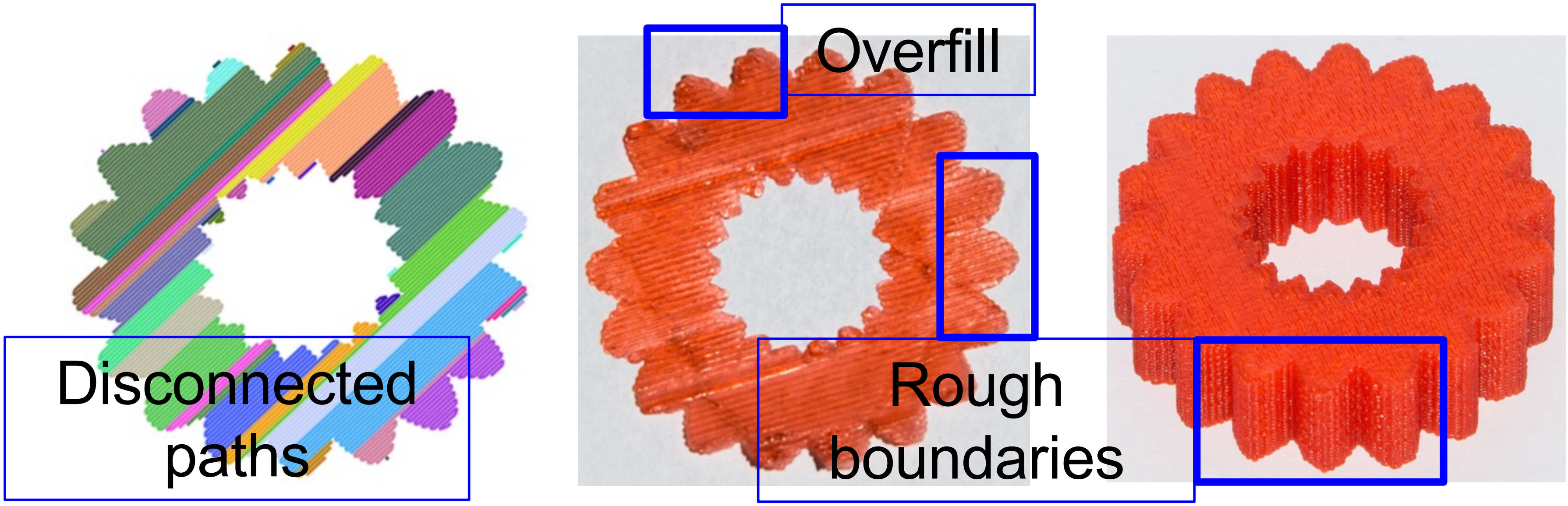
Tool path planning

- Tool path fill = space-filling curve



- Most popular tool path pattern: zigzag



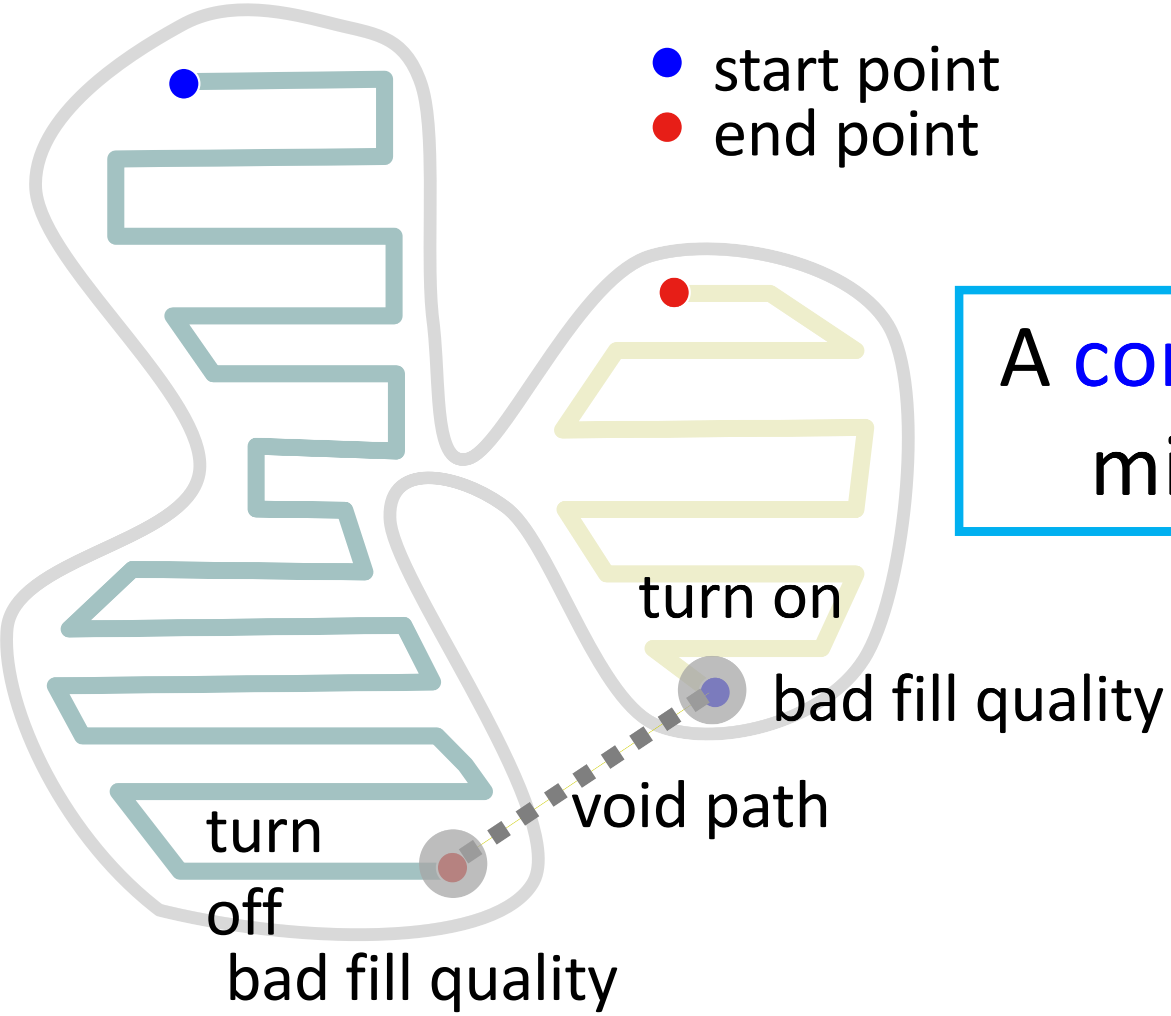


Overfill

Disconnected paths

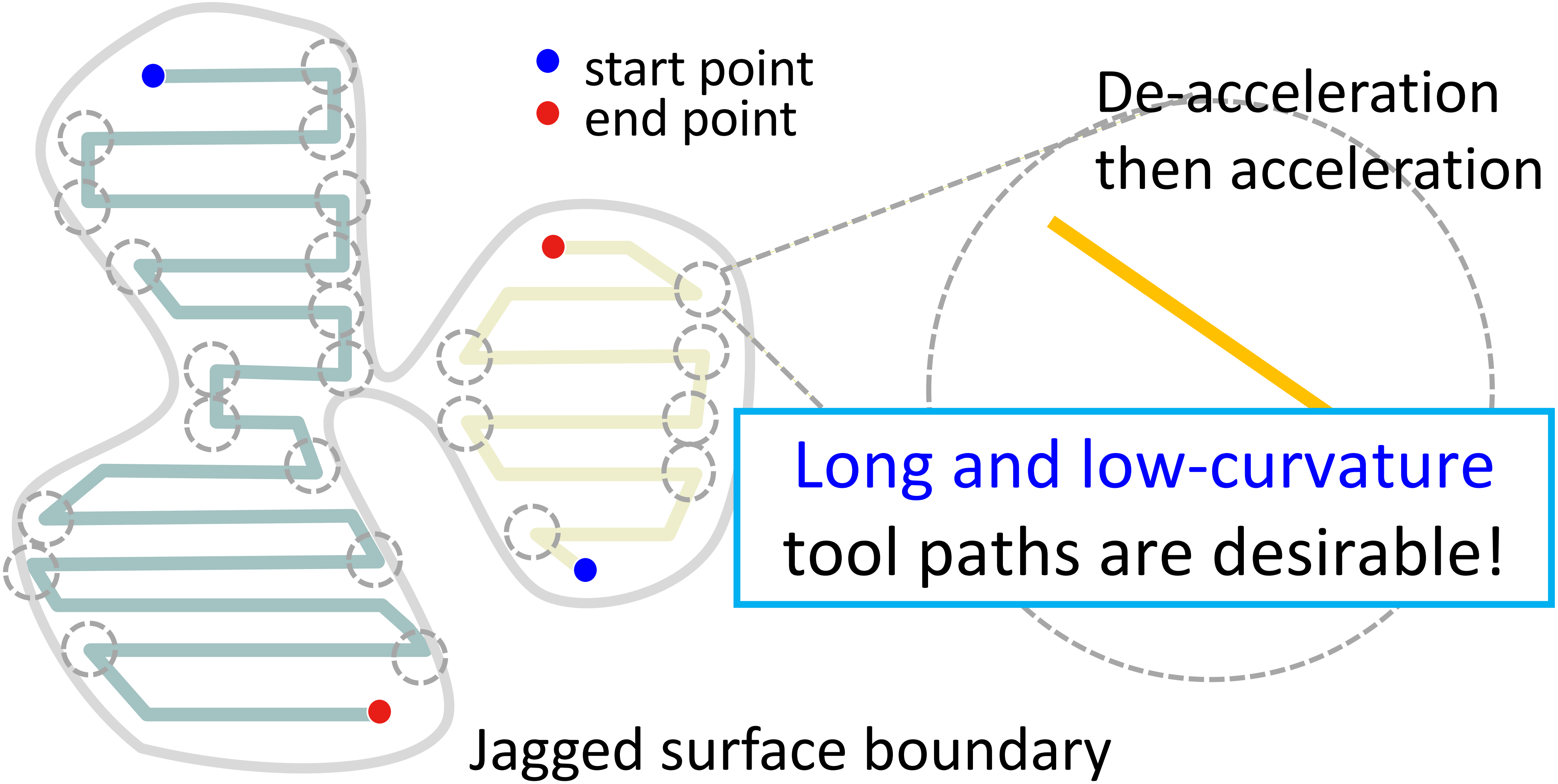
Rough boundaries

Zigzag fill: discontinuity



A continuous tool path minimizes on/offs

Zigzag: sharp turns



● start point

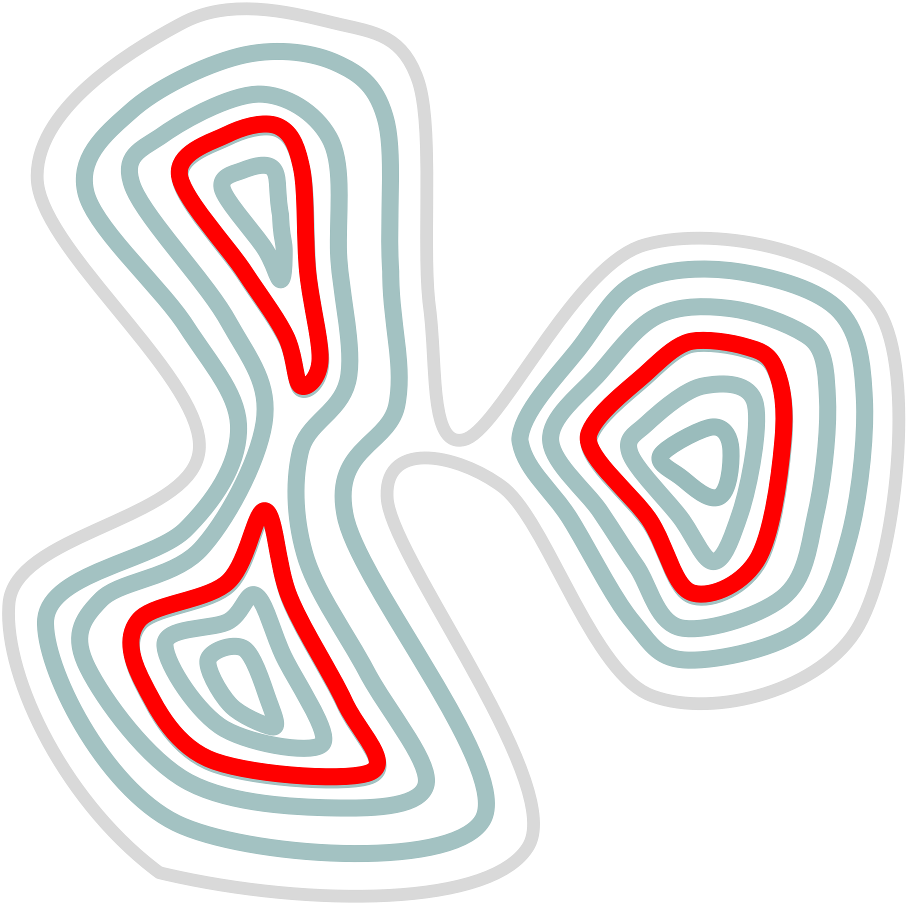
● end point

De-acceleration then acceleration

Long and low-curvature tool paths are desirable!

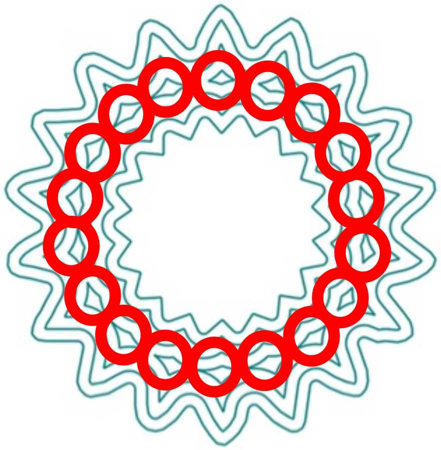
Jagged surface boundary

Contour-parallel paths (CPP): iso-contour



- Less sharp turns
- Conform to boundary

- Contours disconnected
- Disconnected “pockets”



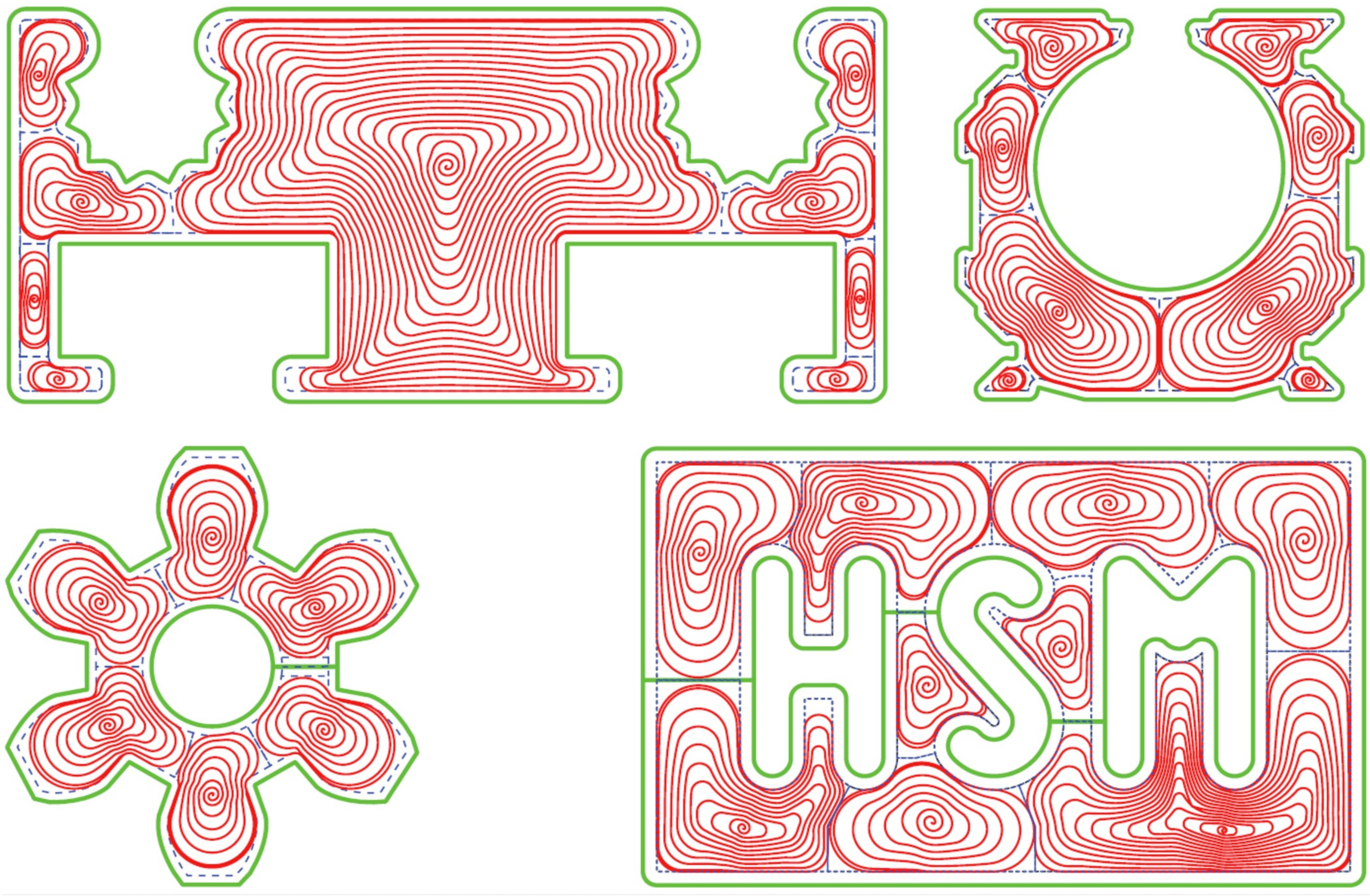
From CPP to conventional spirals

- Conformation to boundary
- Less sharp turns than zigzag

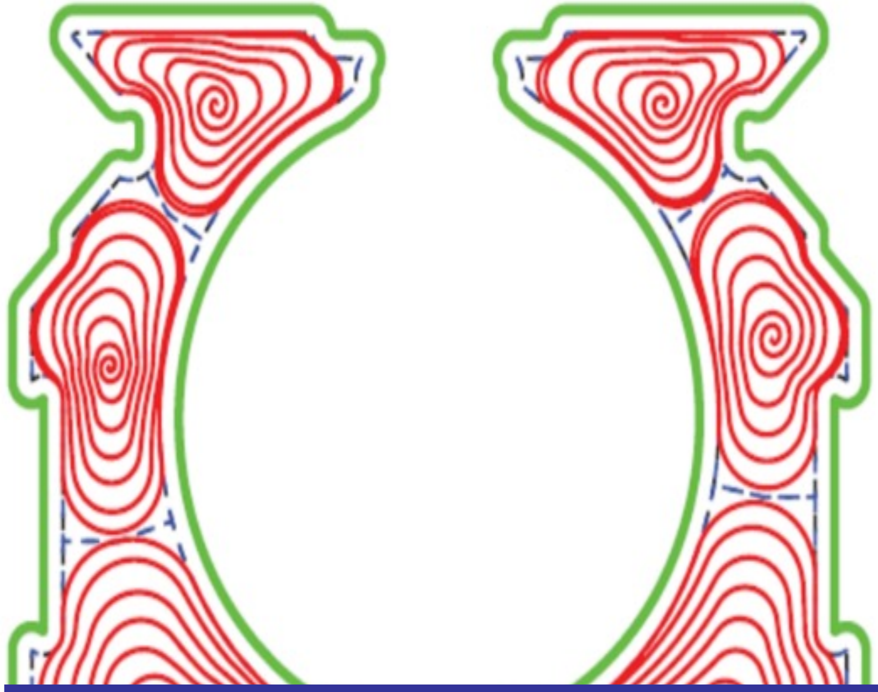
- Connect iso-contours by “offset”

Disconnected spirals

[Held et al. 2014]



Idea: connect the spirals?



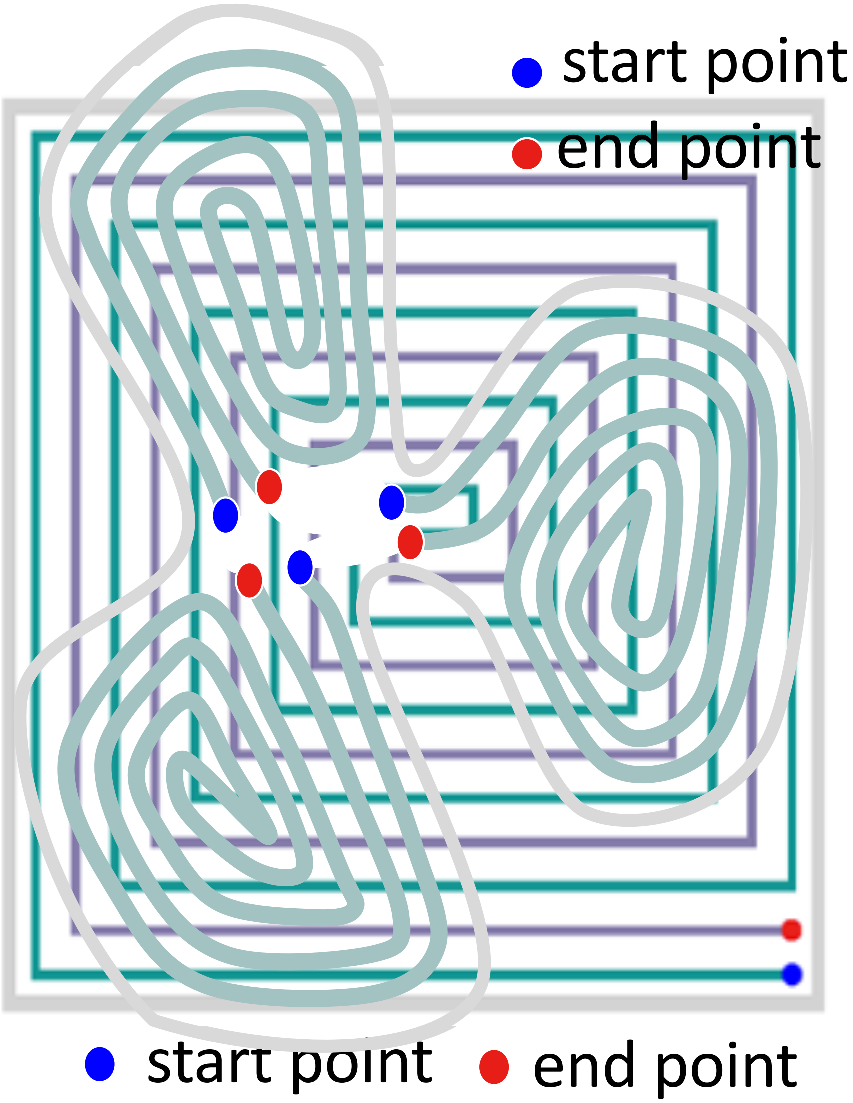
- Can connect two spirals:
  - inside-out & outside-in
  - Then stuck: both start and end points are enclosed
- Impossible to connect all

Is it always possible to fill a connected 2D region using a globally continuous path with low number of sharp turns?

Key idea: Fermat spirals!

Pierre de Fermat (1636)

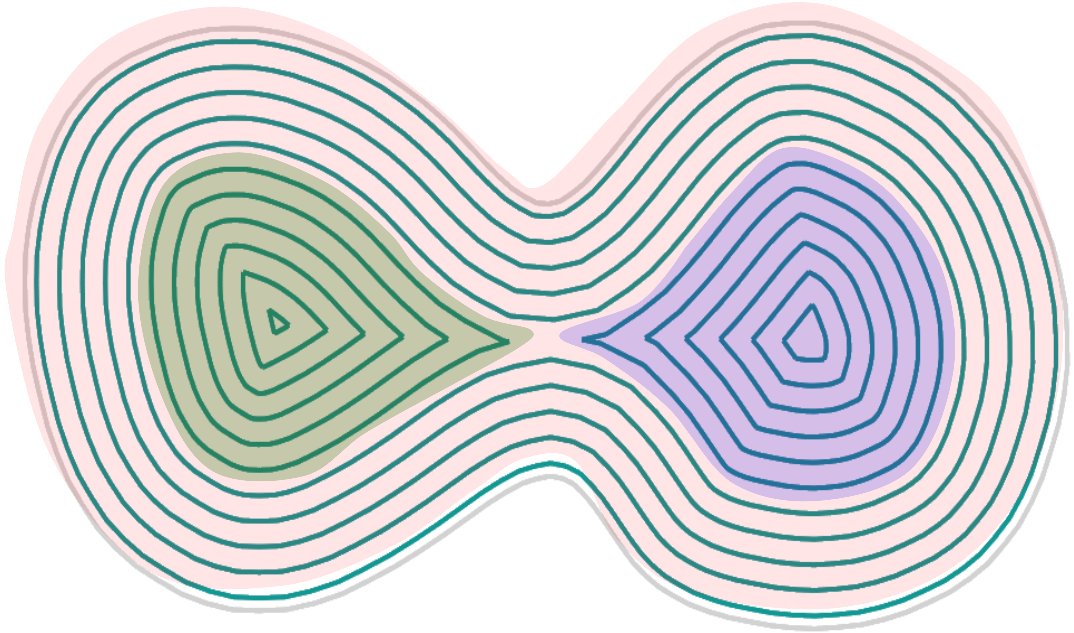
Fermat spiral: compelling properties



- Similarities to spiral and CPPs
  - Conform to surface boundaries
  - Less sharp turns than zigzag
  - Continuity for simple shapes
- New: start & end on boundary
- Key: can place start & end points freely along boundary
- Allows connection of all Fermat spirals for global continuity

Key steps

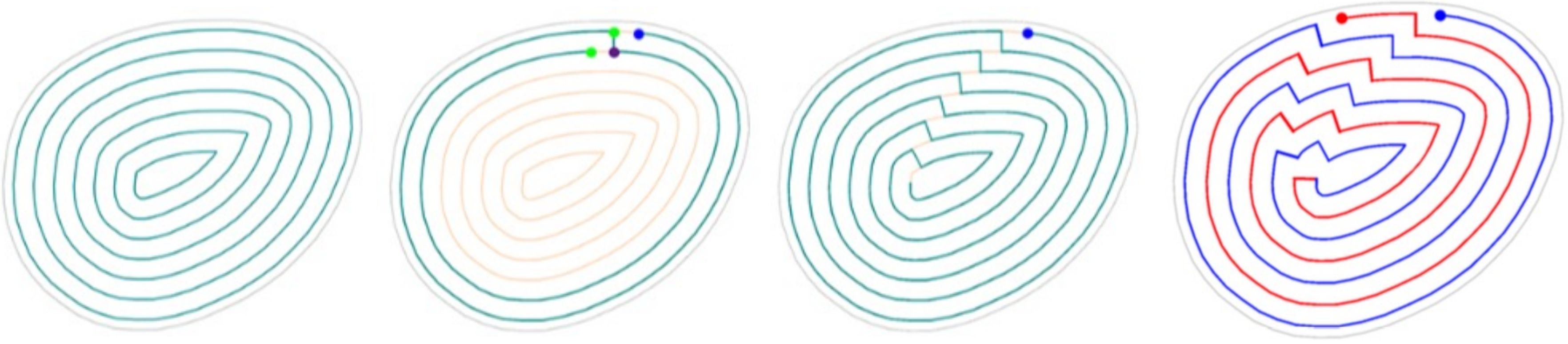
1. Apply Euclidean distance transform to input 2D layer to obtain iso-contours and set of pockets



Key steps

1. Apply Euclidean distance transform to input 2D layer to obtain iso-contours and set of pockets

2. For each pocket, covert contour parallel paths into a Fermat spiral with start and end points next to each other



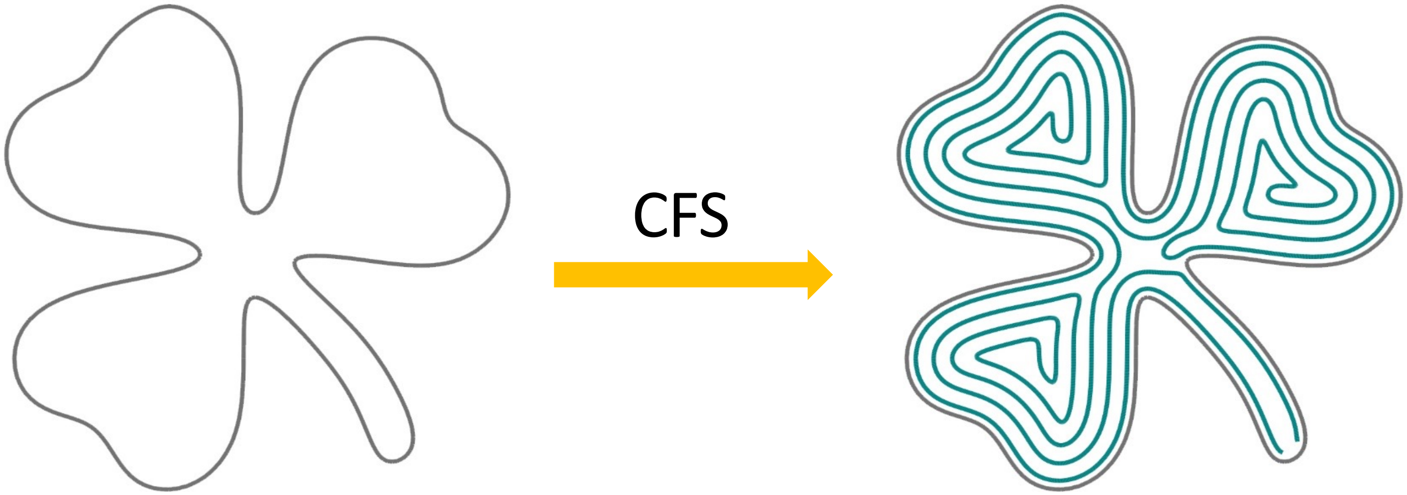
Parallel contours One re-routing To a spiral Re-route to Fermat spiral

Key steps

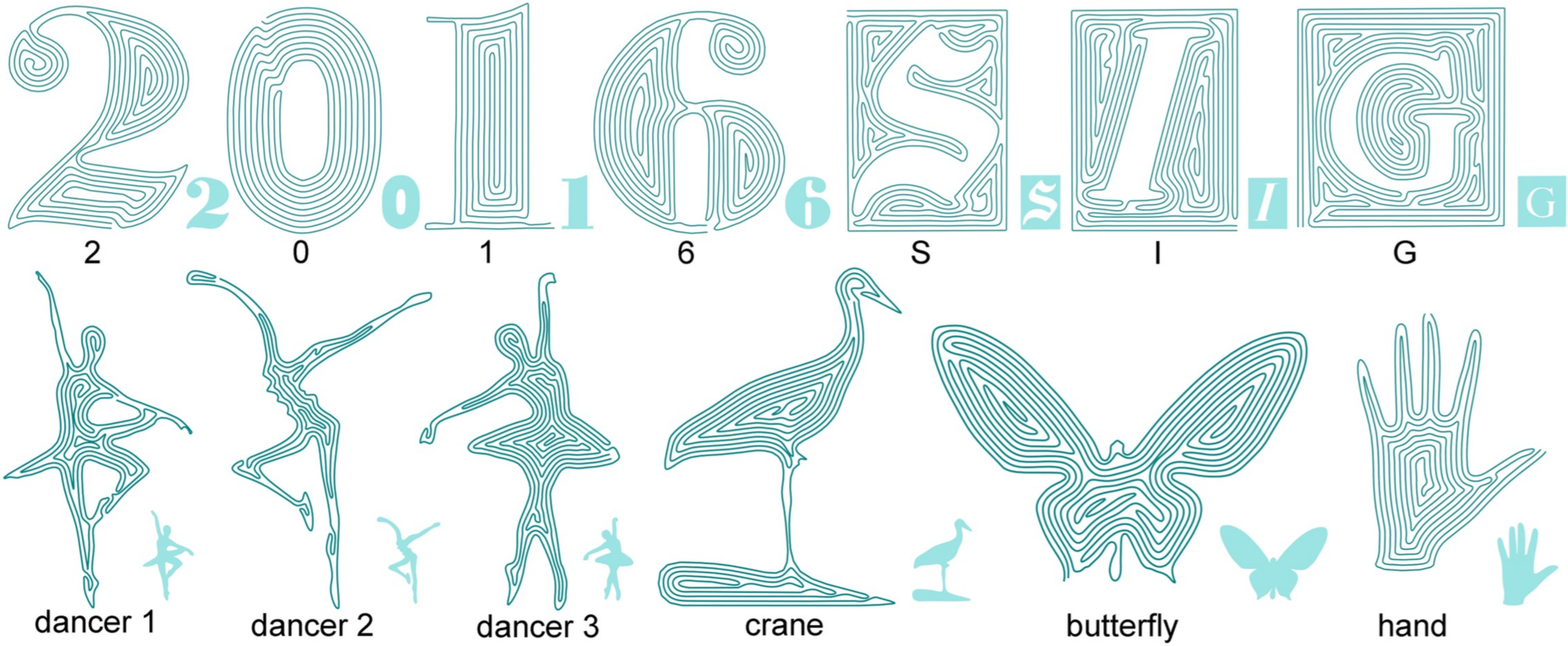
1. Apply Euclidean distance transform to input 2D layer to obtain iso-contours and set of pockets
2. For each pocket, covert contour parallel paths into a Fermat spiral with start and end points next to each other
3. Connect all Fermat spirals via a traversal and local re-routing
4. Localized post-smoothing of final curve

New kind of space-filling curves

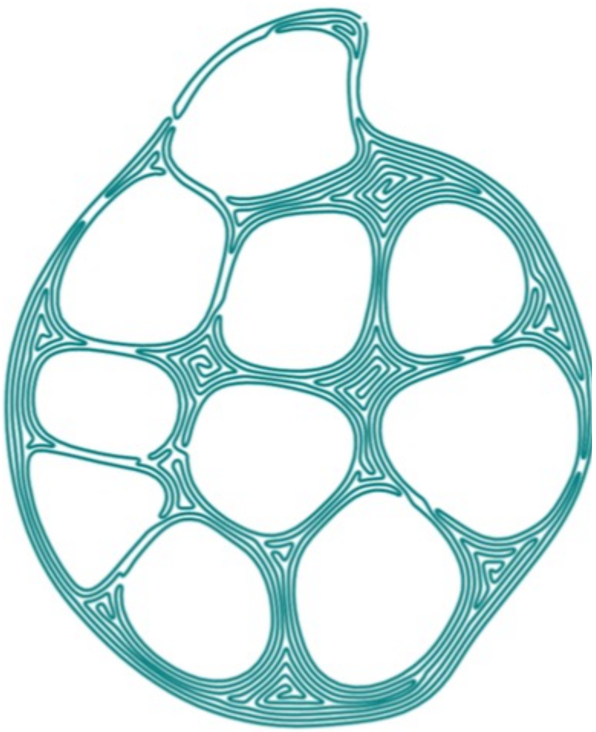
- Introducing Fermat spirals as a new kind of 2D fill pattern, contrasting Hilbert and Peano curves
- Tool path planning based on connected Fermat spirals (CFS) to continuously fill 2D region

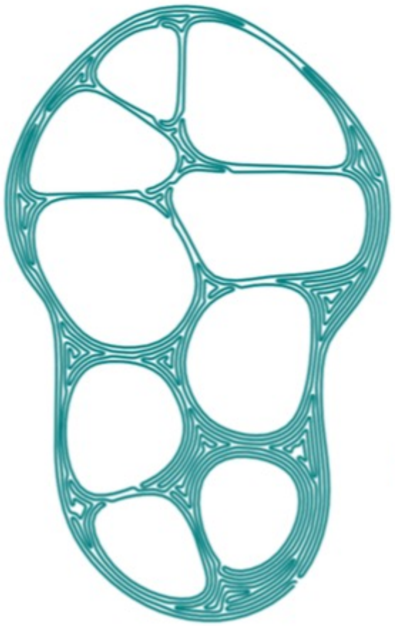


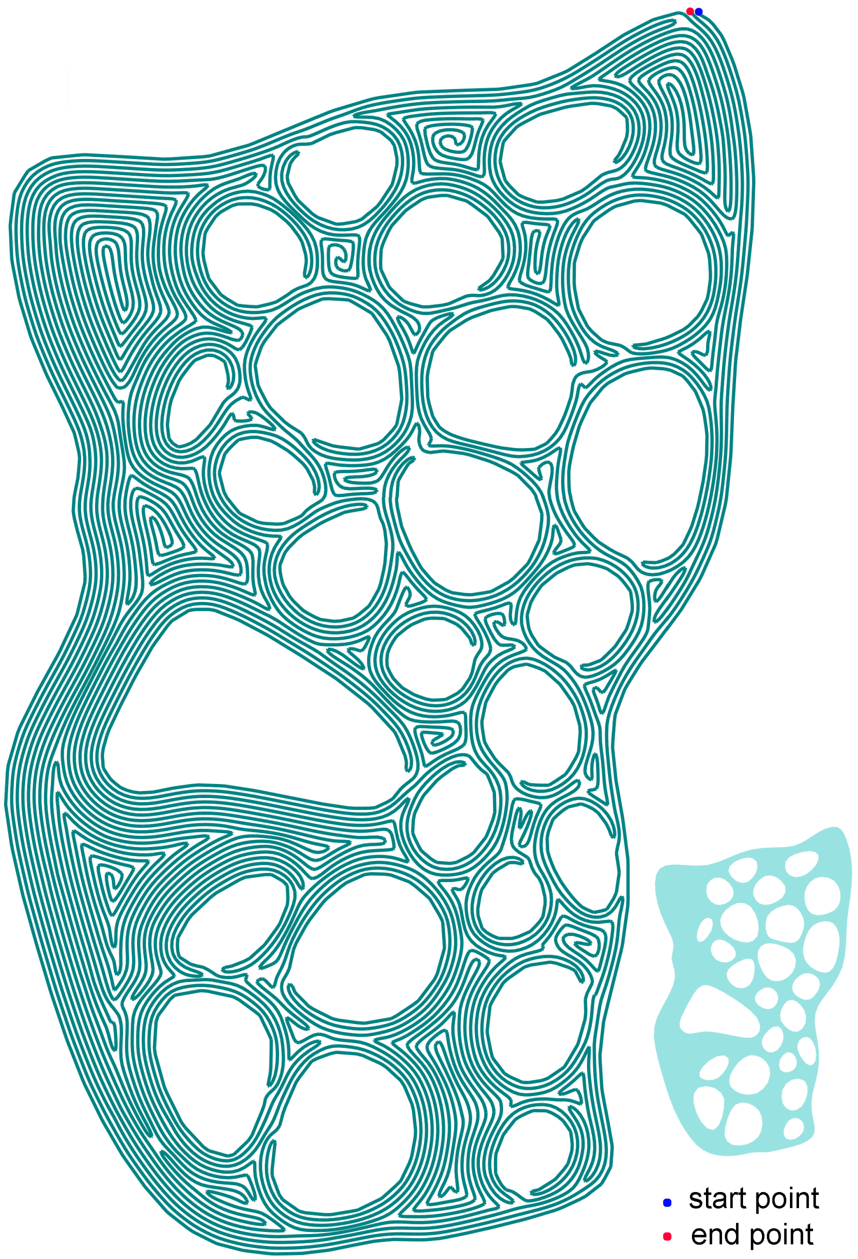
Some results



Some results

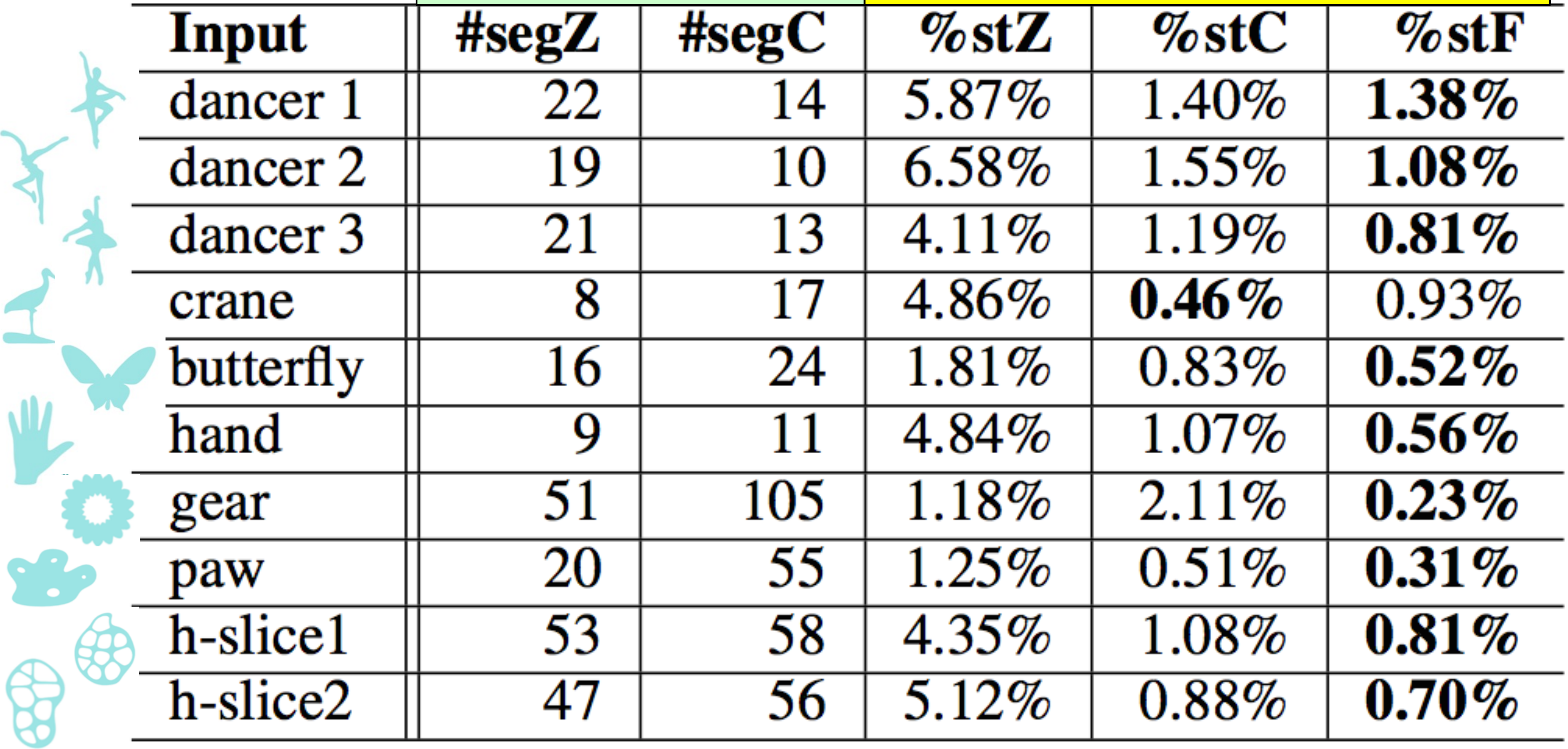






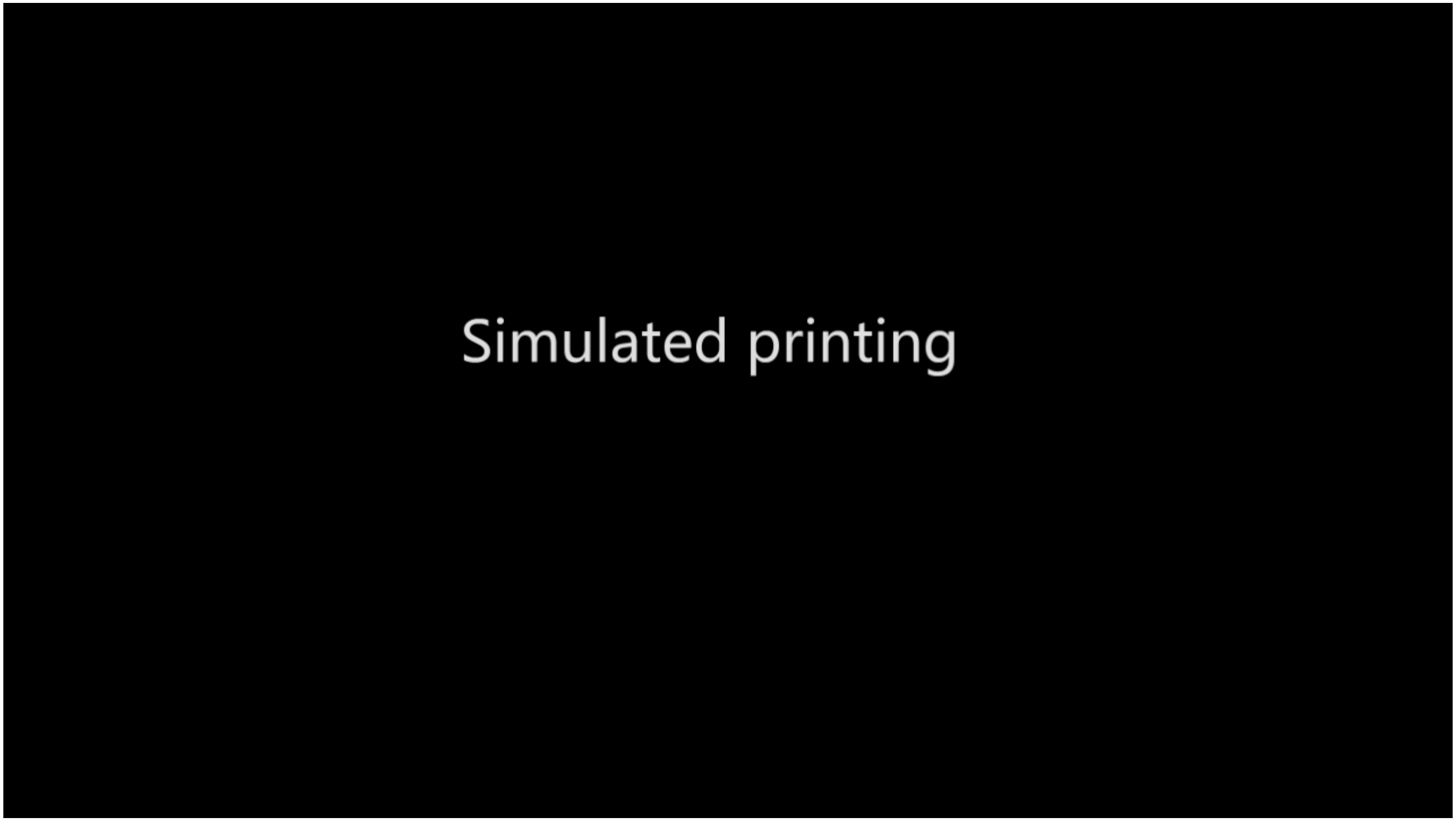
Z: zigzag paths
C: contour-parallel paths
F: connected Fermat spiral paths

Some stats



	Number of disconnected segments		Percentage of sharp turn (high-curvature) points		
Input	**#segZ**	**#segC**	**%stZ**	**%stC**	**%stF**
dancer 1	22	14	5.87%	1.40%	**1.38%**
dancer 2	19	10	6.58%	1.55%	**1.08%**
dancer 3	21	13	4.11%	1.19%	**0.81%**
crane	8	17	4.86%	**0.46%**	0.93%
butterfly	16	24	1.81%	0.83%	**0.52%**
hand	9	11	4.84%	1.07%	**0.56%**
gear	51	105	1.18%	2.11%	**0.23%**
paw	20	55	1.25%	0.51%	**0.31%**
h-slice1	53	58	4.35%	1.08%	**0.81%**
h-slice2	47	56	5.12%	0.88%	**0.70%**

Connected Fermat spirals in video



Simulated printing

Appearance on Two-Minute Papers

TWO+ MINUTE
PAPERS
WITH KÁROLY ZSOLNAI-FEHÉR (KZF)
FERMAT SPIRALS
FOR LAYERED 3D PRINTING
Disclaimer: I was not part of this research project, I am merely providing commentary on this work.