Algorithms for 3D Printing and Other Manufacturing Processes

Movable separability I: Introduction

Dan Halperin

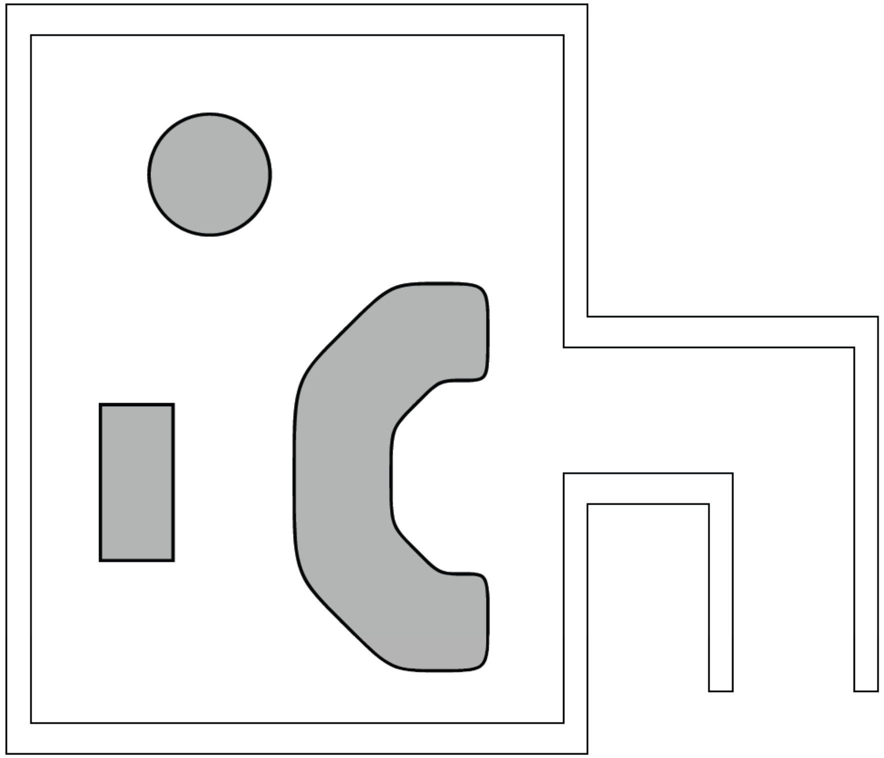
School of Computer Science

Tel Aviv University

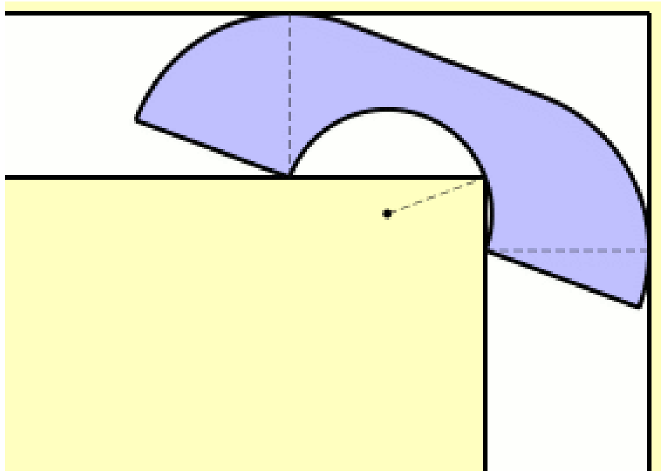
Spring 2017

Movable separability [Toussaint '85]

- Starts with the sofa problem: what is the maximum area object that can be moved through a corner between two corridors of unit width



The Hammersley sofa has area 2.2074 but is not the largest solution [Wikipedia: Moving sofa problem]



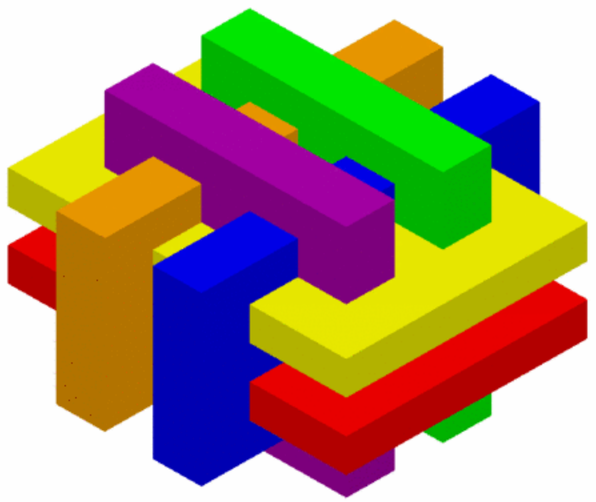
Outline

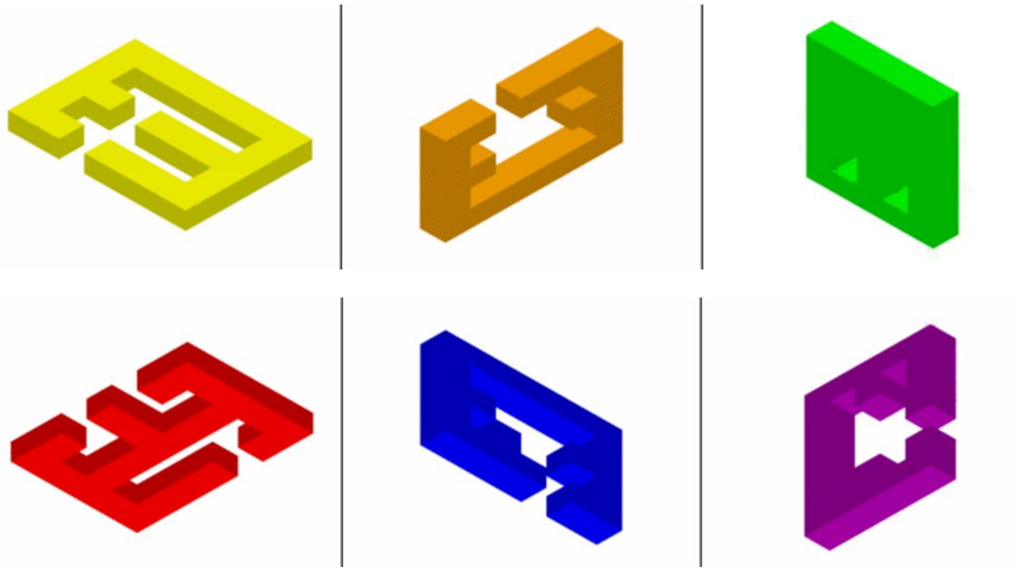
- Background
- Separation sequence in the plane (reminder)
- Some separation problems are hard
- Terminology for assembly planning

Background

Separating parts

At the end of a successful separation process the parts are sufficiently far away from one another

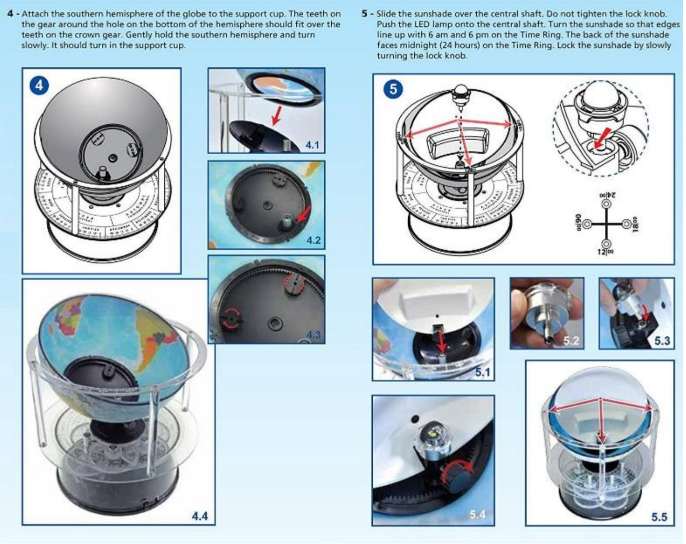




www.cs.tau.ac.il/~efif/applications/puzzles/knot/

Different flavors of separability

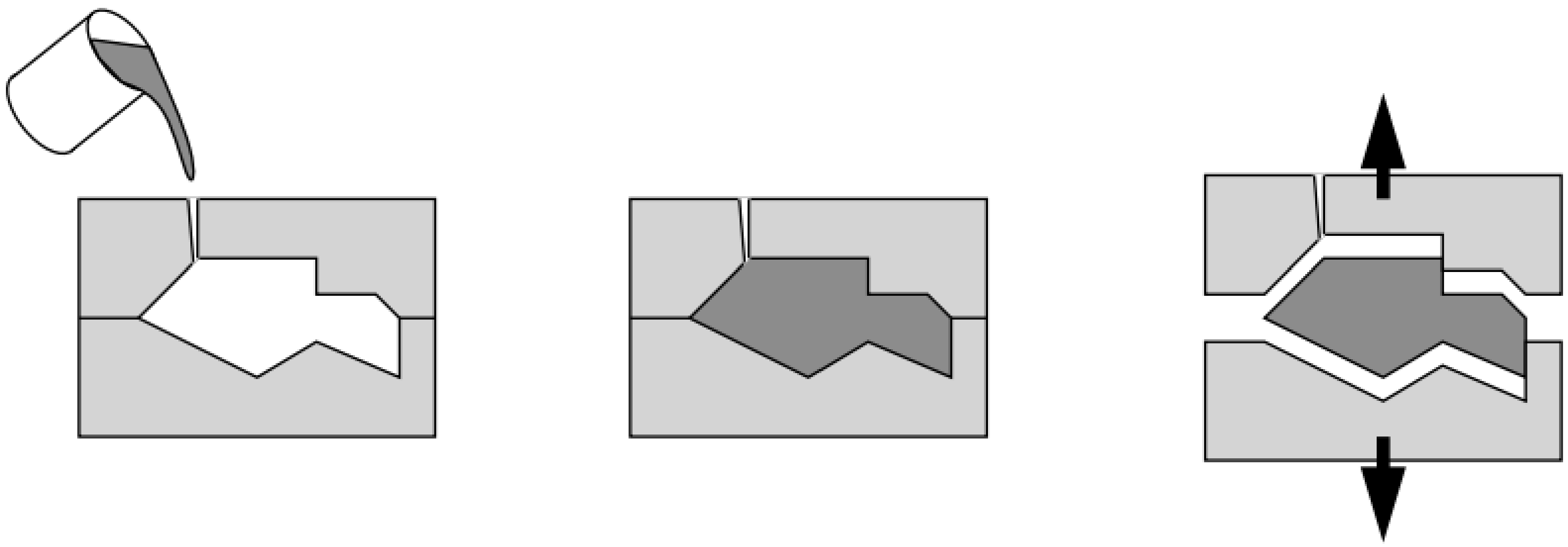
- Separability I: Unpacking the build assembly in 3DP
- Separability II: Removing a product from its cast
- Verifying a product assemblability or interlockedness
- Planning assembly sequences (in reverse)



Testing the separability of a pair of parts

- Theorem: Given two polyhedral parts in two- or three-dimensional space, then testing their separability under arbitrary rigid motions can be carried out in polynomial time in the complexity of the parts.
- Example: The separability of two polygons with m and n vertices respectively, can be tested in $O((mn)^{2+\varepsilon})$ time [H-Sharir '96]
- "Two parts" pertains also to the case of one moving part and a collection of static parts
- If the parts are sufficiently well spaced, motion planning methods from robotics could solve separability of two parts rather easily
- Our focus here is on the case of tightly packed parts

Separability in casting and molding



- The focus is on the design of a (possibly multi-part) mold, cores and inserts

Assembly/disassembly planning

Designing separation sequence (or, reversely, assembly) for a fixed set of parts, having more than two parts, and often tightly packed. We will focus on

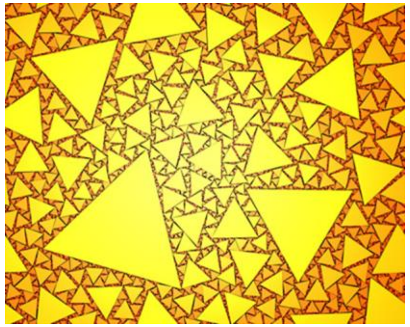
- Verifying a product assemblability or interlockedness
- Planning assembly sequences (in reverse)

Separation sequence in the plane

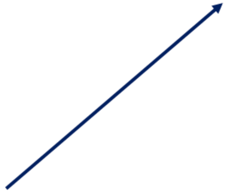
Finding a separation sequence for convex polygons, reminder

- Input: m pairwise disjoint convex polygons with a total of n vertices
- Output: an ordering of the parts, such that we can move each part in turn along the given direction, without colliding with the parts that have not yet been moved
- Find the segment connecting the topmost and bottommost points in each convex polygon
- Apply the segment-sequence algorithm
- Total running time $O(n + m \log m)$ time

[Guibas-Yao '83]



http://paulbourke.net/texture_colour/randomtile/



Unidirectional separability of convex polygons in the plane, summary

- The parts: convex polygons in the plane
- Translations only
- All translations are in the same (arbitrary) direction
- Moved once (a form of monotonicity, see later): each polygon is moved only once
- Only one polygon is moved at a time

Hard separability problems

Translational separability of polygons in the plane

- The parts: simple polygons
- Translations only
- Translations are in different directions
- Each polygon can be moved several times

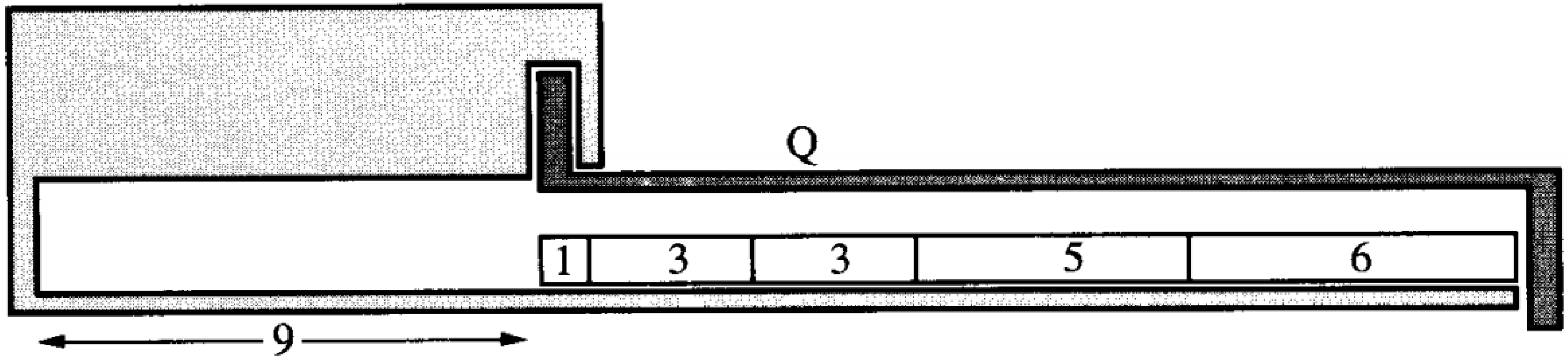
This problem is NP-hard

The Partition problem

- Given a collection of integers decide whether or not it may be partitioned into two parts whose sums are equal

- {1,2,4,4,6,9}? YES: 1+2+4+6=4+9
- {1,2,2,4,13}? NO

Partition -> Separability



[Chazelle et al '84, O'Rourke]

- Turn each number m in the input collection into a rectangle of height 1 and width m

Assembly partitioning

- Given a collection of non-overlapping polygons, decide if there is a proper subcollection of them that can be moved away as a rigid body without colliding with or disturbing the other parts of the assembly
- Assembly partitioning is NP-complete [Kavraki et al, '93]

Connected subassembly partitioning

- Given a collection of non-overlapping (but possibly touching) polygons in the plane, is there a proper *connected* subcollection of it that can be separated from its complement moving as a rigid body, without disturbing the other parts of the collection, and such that the complement is also connected?
- This decision problem is NP-complete [Kavraki et al, ‘95]
- This had been known to be true in three-dimensional space

Earlier approach

- In view of the typically hard problems, heuristics were developed

Assembly terminology

Taken mostly from [H-Latombe-Wilson 2000]

Basics

- A **workspace** is a subset of the two or three-dimensional physical space modeled by the Euclidean space R^k (k = 2 or 3)
- A **body** is a rigid physical object modeled as a compact manifold with boundary in R^k. We will only deal with polygons and polyhedral in the sequel
- We say that two bodies **overlap** if their interiors intersect
- We say that they **touch** each other if they intersect without overlapping

Assembly and subassemblies

- An **assembly** is a collection of bodies (also called **parts**) in some given relative placements in the workspace, such that no two bodies overlap
- A **subassembly** is a subset of the bodies composing an assembly *A* in their relative placements in *A*
- Remark*:* In most practical cases, each part of an assembly is in contact with a few other parts so that the union of all the parts forms a connected subset of the workspace. However, except when we indicate otherwise, we do not make such an assumption. The motion space approach, which we will study, applies to both cases.

Assembly operations and the number of hands

- Given an assembly A, an **assembly operation** is a motion that merges s ($s \geq 2$) pairwise separated subassemblies of A into a new subassembly of A. During this motion, each subassembly moves as a single body and no overlapping between bodies is allowed.
- The parameter s is called the **number of hands** of the operation
- We call the reverse of an assembly operation a **partitioning** operation
- An **assembly sequence** is a total ordering on assembly operations that merges the separated parts composing an assembly into this assembly

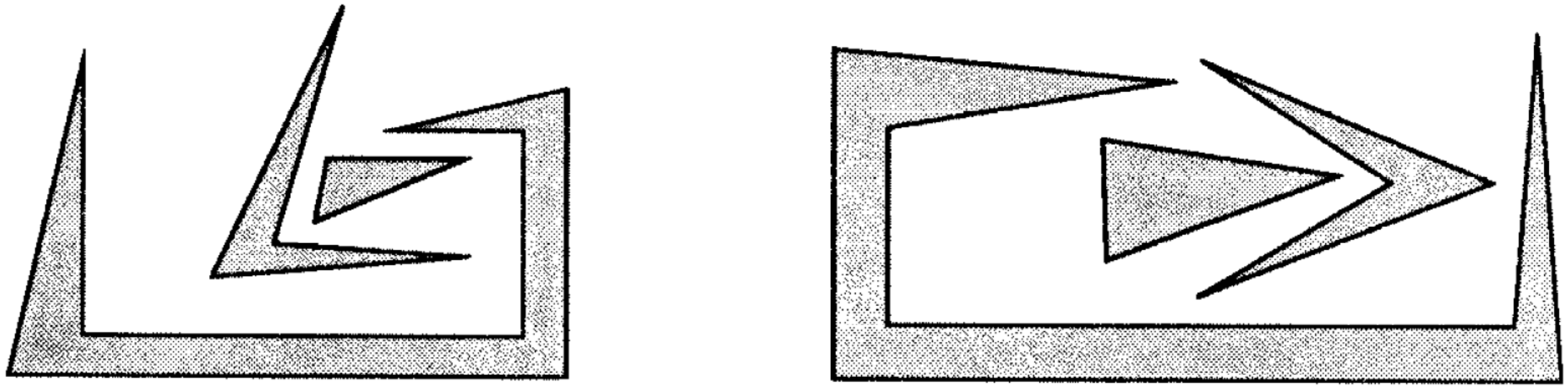
Two-handed assembly sequences

- The maximum, over all the operations in the sequence, of the number of hands (the parameter s) required by an operation is called the **number of hands of the sequence**
- We will only deal with **two-handed** assembly sequences
- (Planning k-handed assembly sequences, for k> 2, is a major challenge)

Monotone vs. non-monotone sequences

- With the above definitions, an assembly sequence only generates subassemblies of the final assembly. Such a sequence is said to be **monotone**
- A more general assembly sequence is one in which some operation brings a body to an intermediate placement (relative to other bodies), before another operation transfers it to its final placement. Such a sequence is called **non-monotone**
- We will only deal with **monotone** sequences

Exmaple



(a) (b)

Fig. 1. Both assemblies above admit two-handed sequences with translational motions only. While (a) accepts a monotone such sequence, (b) does not. To disassemble (a) one can first remove the subassembly consisting of the two “inner” parts, then break this subassembly into its two components. To disassemble (b) the triangle must be translated to an intermediate position, since the two inner parts cannot be removed as a subassembly. If general motions are accepted, there exists a monotone two-handed sequence for (b). A monotone three-handed sequence with translations only is also possible.

Assembly by disassembly

- Partitioning, first the input assembly *A* into subassemblies and then, recursively, the generated subassemblies that are not individual parts
- Implemented by two procedures:
- Partition: takes the description of an assembly *S* as input and generates two subassemblies S_1 and S_2, along with a path p such that moving S_1 along p separates it from S_2. Whenever such subassemblies and direction do not exist, the procedure returns failure
- Disassemble: applies partition to the given assembly *A* and, recursively, to the generated subassemblies

Categories of motion

- A motion step translates a body along a single direction t by some distance q, while rotating it at constant rate r about an axis a that is fixed relative to the body
- A **one-step motion** consists of a single motion step in which q is arbitrarily large
- If the rotation rate is null (i.e., the body has fixed orientation), the motion is a **one-step translation**
- A **multistep motion** is the concatenation of several motion steps, in which the last step has arbitrarily large q
- An **infinitesimal motion** consists of a single motion step in which q is arbitrarily small

Separability: Bibliography

For additional bibliography on separability, see the end of the set of slides *Separability II*

- Leonidas J. Guibas, F. Frances Yao: On Translating a Set of Rectangles. STOC 1980: 154-160
- Godfried T. Toussaint, Movable separability of sets, in Computational Geometry, G.T. Toussaint, Ed., North-Holland Publishing Co., 1985, pp. 335–375.
- Bernard Chazelle, Thomas Ottmann, Eljas Soisalon-Soininen, Derick Wood: The Complexity and Decidability of Separation. ICALP 1984:119-127
- Lydia E. Kavraki, Jean-Claude Latombe, Randall H. Wilson: On the Complexity of Assembly Partitioning. Inf. Process. Lett. 48(5):229-235 (1993)
- Lydia E. Kavraki, Mihail N. Kolountzakis: Partitioning a Planar Assembly Into Two Connected Parts is NP-Complete. Inf. Process. Lett. 55(3): 159-165 (1995)
- Randall H. Wilson, Jean-Claude Latombe: Geometric Reasoning About Mechanical Assembly. Artif. Intell. 71(2): 371-396 (1994)
- Dan Halperin, Jean-Claude Latombe, Randall H. Wilson: A General Framework for Assembly Planning: The Motion Space Approach. Algorithmica 26(3-4): 577-601 (2000)

THE END

[Gaither, ArtByAI, CGAL arrgs]