

# State of the Art on Computational Design of Assemblies with Rigid Parts

Ziqi Wang<sup>1</sup> Peng Song<sup>2</sup> Mark Pauly<sup>1</sup>

<sup>1</sup>GCM, EPFL, Switzerland

<sup>2</sup>Singapore University of Technology and Design

---

## Abstract

An assembly refers to a collection of parts joined together to achieve a specific form and/or functionality. Designing assemblies is a non-trivial task as a slight local modification on a part's geometry or its joining method could have a global impact on the structural and/or functional performance of the whole assembly. Assemblies can be classified as structures that transmit force to carry loads and mechanisms that transfer motion and force to perform mechanical work. In this state-of-the-art report, we focus on computational design of structures with rigid parts, which generally can be formulated as a geometric modeling and optimization problem. We broadly classify existing computational design approaches, mainly from the computer graphics community, according to high-level design objectives, including fabricability, structural stability, reconfigurability, and tileability. Computational analysis of various aspects of assemblies is an integral component in these design approaches. We review different classes of computational analysis and design methods, discuss their strengths and limitations, make connections among them, and propose possible directions for future research.

## CCS Concepts

- **Computer Graphics** → *Computational Geometry and Object Modelling*;
- 

## 1. Introduction

An assembly refers to a collection of parts joined together to achieve a specific form and/or functionality. Assemblies facilitate product fabrication, maintenance, and usage by allowing assembly, disassembly, and reassembly of the component parts. Due to this reason, assemblies are ubiquitous in our daily life, existing in a wide variety of scales and geometric forms; typical examples include consumer products, machines, and architectural structures. While assemblies have intriguing properties, designing assemblies is a non-trivial task, even for professionals. A slight modification on an individual part could have a global impact on the aesthetic, structural, and/or functional performance of the whole assembly. Ensuring that all parts can be put together and properly joined to form the final assembly is another challenge.

Modern digital fabrication technologies, such as additive manufacturing and computerized numerical control (CNC) machining, make it possible to easily manufacture shapes with geometric complexity that is unattainable by traditional techniques. The need to find creative and practical uses of these technologies has inspired researchers to investigate computational approaches and tools to design various assemblies for digital fabrication. These approaches generally take high-level user specifications as input, and compute the final detailed shape of each component part for fabrication and their assembly instructions for installation, aiming at lowering

the barrier for non-domain experts to design and manufacture customized products as assemblies.

### 1.1. Survey Scope and Classification

Assembly is a very broad concept. According to their functionality, assemblies can be classified as *structures* that transmit force to carry loads and *mechanisms* that transfer motion and force to perform mechanical work. This survey limits its scope to *structures with rigid parts*, and focuses on *computational design* of these assemblies, taking into account aspects of fabrication. By assuming that the fabrication material is rigid, design of assemblies generally can be formulated as a geometric modeling and optimization problem, with minimal consideration of material behavior (e.g., friction). In the remaining part of this report, we will not differentiate between assemblies and structures.

The objective of this state-of-the-art report is to give the reader a comprehensive review of computational methods that assist in or automate the design of complex assemblies. Analysis of assemblies is necessary and crucial, not only for evaluating the resulting designs, but also for guiding the whole design process. Therefore, this report is organized into two parts: Section 2 for computational analysis of assemblies and Section 3-6 for computational design of assemblies. In particular, Section 2 reviews computational approaches that analyse four fundamental aspects

of given assemblies: parts joining, assembly planning, structural stability, and packing efficiency. We propose a three-level scheme to classify research works on computational design of assemblies, namely, high-level design objectives, specific design problems, and corresponding design methodologies. Following this classification scheme, Section 3-6 are organized according to the high-level design objectives:

- *Assembly-based fabrication* that enables making target 3D shapes with a specific fabrication technique by representing or approximating the shape with a set of simple parts (Section 3).
- *Computational design of structurally stable assemblies* without relying on any additional connector (Section 4), guided by a formal stability analysis method (Section 2.3).
- *Computational design of reconfigurable assemblies* that have multiple forms for use in different situations (Section 5).
- *Building assemblies with tileable blocks* to approximate a given 2D/3D shape (Section 6).

We also categorize existing works based on their formulation of design problems, and present a table that labels existing works based on their design objective and problem formulation, which helps discover correlation among various design methods reviewed in this report (Section 7). Based on comprehensive taxonomies and discussions, we hope the reader can efficiently identify existing methods and algorithms that are reusable or extensible for his/her own problem of designing assemblies. We highlight challenges that have not been fully addressed in computational analysis and design of assemblies, and propose possible directions for future research at the end of this report (Section 8).

## 1.2. Related STARS and Surveys

In recent years, a few surveys in the computer graphics [BFR17, BCMP18] and human computer interaction [BM17] communities have broadly overviewed computational and interactive approaches that enable users to design a wide variety of customized products for digital fabrication, such as monolithic shapes, articulated objects, mechanical automata, deformable objects, inflatable shapes, and so on. Other relevant surveys restrict their scope by focusing on a specific design objective (e.g., functionality [eS-REPC16]), a certain fabrication technique (e.g., additive manufacturing [OZB18, ALL\*18]), or a concrete application (e.g., cultural heritage [SCP\*17] and architectural construction [GJL20]).

This report complements above surveys by focusing on a specific subject: assemblies with rigid parts, which is ubiquitous in our daily life yet challenging to design. In the last decade, tremendous research works, mainly from the graphics community, have been devoted to studying computational methods for designing these assemblies. This report, for the first time, reviews and classifies computational methods introduced in these works, discusses correlation inherent in these methods, and proposes possible directions for future research.

## 2. Computational Analysis of Assemblies

Computational analysis of assemblies evaluates different aspects of assemblies with given geometry, including joining parts of an

assembly (Section 2.1), planning of the assembly process (Section 2.2), structural stability of the whole assembly (Section 2.3), and efficiency of packing the parts (Section 2.4).

### 2.1. Joining Parts

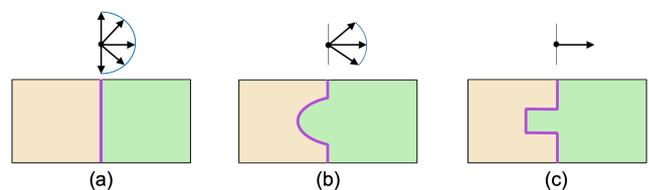
To form an assembly that can be used in practice, component parts need to be joined together to restrict relative movements among the parts. This (potentially additional) geometry or material used to connect parts defines the joining method, or simply the joint.

**Joint classification.** Joints can be classified as *permanent joints* and *non-permanent joints*. Typical permanent joints include adhesive material (e.g., glue, mortar) and permanent mechanical fasteners (e.g., rivets). Although assemblies connected with permanent joints can be structurally very stable, a significant drawback is that the assembly cannot be disassembled without potential damage to the parts. In contrast, non-permanent joints encourage parts disassembly and reassembly, facilitating storage, transportation, maintenance, and reconfiguration of assemblies.

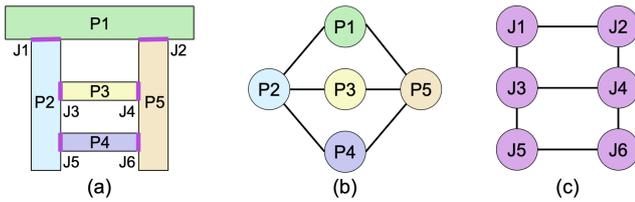
Non-permanent joints can be classified as *external joints* and *integral joints*, depending on whether the joint geometry is integrated on each individual part. Screws and pins are conventionally used external joints. To satisfy specific needs on parts joining in digital fabrication, customized external joints have been designed and used in practice [MMZ18]. These external joints are independent from the parts. Hence, they can be abstracted as “conceptual parts” in the analysis of assemblies, e.g., assembly planning.

Integral joints are implicitly defined as the portion of each individual part that is in contact with adjacent parts. The simplest integral joints would be planar contacts between neighboring parts [WOD09, WSIP19]. More complex integral joints include curved contacts between parts [KAS\*21] and conventional woodworking joints [Fai13]; see Figure 1 for examples. Integral joints can significantly simplify the assembly process by assisting parts alignment and reducing the total number of assembly steps (i.e., no external fasteners need to be applied). Integral joints can also add to an assembly’s structural durability (e.g., woodworking joints in architecture) and visual appeal (e.g., decorative joints in furniture [YKGA17]). Due to this reason, integral joints are more and more widely used in digital fabrication of assemblies [ZDB17, LYUI20].

**Joint analysis** identifies contacts between each pair of parts in an assembly by computing the minimum distance between them and checking if this minimum distance is less than a given threshold



**Figure 1:** Schematic of three kinds of integral joints with corresponding translational motion space of the green part illustrated on top. (a) Planar contact joint; (b) curved contact joint; and (c) mortise and tenon joint.



**Figure 2:** (a) A 5-part assembly, where part contacts (i.e., joints) are highlighted in purple; (b) parts-graph; and (c) joints-graph.

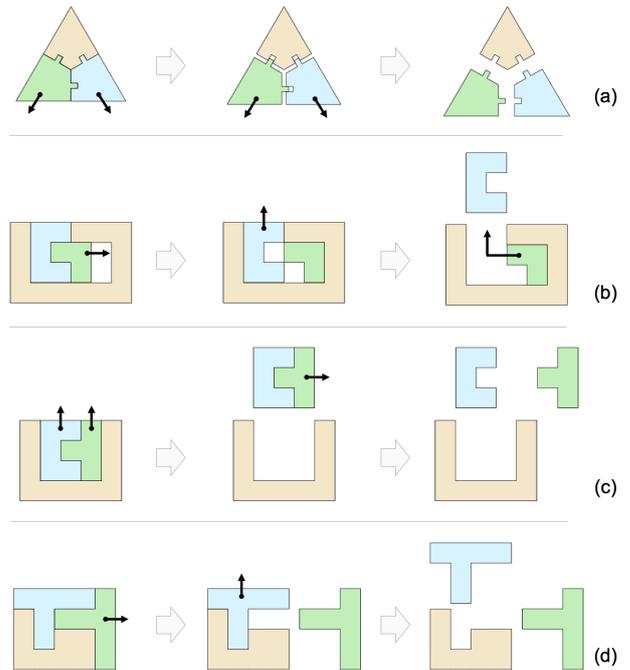
(very small positive number). Figure 1 highlights part contacts as purple lines or curves. Based on the identified part contacts, joint analysis can obtain the following information of an assembly:

- **Parts connectivity.** Two parts are connected if they have at least one contact. All contacts between the two parts define joints that connect the two parts. A *parts-graph* [FSY\*15] is typically used to represent parts connectivity in an assembly, where each node represents a part and each edge represents joints connecting the two associated parts. The dual of a parts-graph is a *joints-graph*. See Figure 2.
- **Parts mobility.** The contacts between two parts enforce constraints on their relative movement as collisions have to be avoided when moving one part relative to the other. These contact constraints typically can be formulated as a linear system [WM92], whose solution space corresponds to the infinitesimal motion space of one part relative to the other. See Figure 1 for examples, in which only translational motion is considered.
- **Joints strength.** Conceptually, arbitrarily small contacts or thin joints can constrain the relative movement between two rigid parts. However, in practice such joints should be avoided to reduce the risk of structural failure. To detect such issues, finite element methods can be used to analyze joint strength under external loads [YCXW17].

## 2.2. Assembly Planning

Widely used in automated manufacturing, robotics, and architecture, assembly planning is the process of creating detailed instructions to combine separate parts into the final structure. The goal of assembly planning is to find a sequence of operations to assemble the parts (*assembly sequencing* [Jim13]), determine the motions that bring each part to its target pose (*assembly path planning* [GM15]), and propose the utilization of additional resources such as supports and tools to assist the assembly process.

A closely related problem is *disassembly planning*, which creates a plan for disassembling component parts from an installed assembly. An important strategy of assembly planning is *assembly-by-disassembly*, where an assembly plan is obtained by disassembling an installed product into its component parts and then reversing the order and path of disassembly. This strategy is feasible as there is a bijection between assembly and disassembly sequences and paths when only geometric constraints are concerned and all parts are rigid [HLW00]. The advantage of this strategy is that it can drastically reduce the size of the solution space (i.e., valid assembly plans), since parts in an assembled state have far more precedence and motion constraints than in a disassembled state. However, when



**Figure 3:** Examples of disassembly plans, where the orange part is fixed as a reference. (a) A three-handed disassembly plan: the green and cyan parts translate along different directions simultaneously. (b) A non-monotone disassembly plan: intermediate placement of the green part is necessary. (c) A non-linear disassembly plan: the first disassembly operation is to translate the green and cyan parts together along the same direction. (d) A sequential, monotone, and linear disassembly plan.

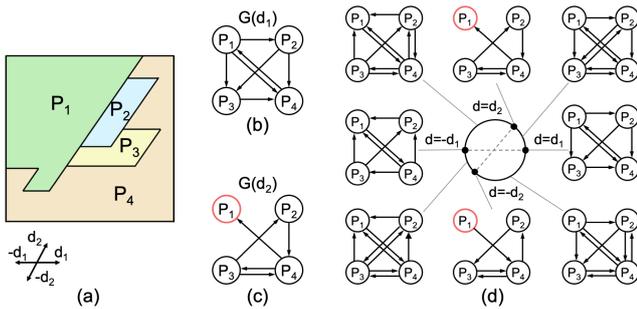
physical constraints are taken into consideration, e.g., supports of incomplete assemblies [DPW\*14], this strategy is not directly applicable to compute assembly plans.

The complexity of assembly planning is measured generally in terms of the number of parts and their shape complexity. However, this measure alone does not express how difficult it is to obtain a valid assembly plan. Other involved features are:

- **The number of hands:** the maximum number of moving sub-assemblies with respect to one another in any assembly operation.
- **Monotonicity:** whether or not operations of intermediate placement of sub-assemblies are required.
- **Linearity:** whether all assembly operations involve the insertion of a single part in the rest of the assembly.

Figure 3 shows disassembly plans to illustrate the above features. The simplest (dis)assembly plans are sequential (two-handed), monotone, and linear. Due to the simplicity, they are the most widely used (dis)assembly plans in computational design of assemblies.

Assembly planning problems can be broadly classified into two classes. The first aims to find a valid assembly plan to ensure assemblability of designed assemblies such as 3D puzzles. The second is to find a desired assembly plan to satisfy some objectives



**Figure 4:** Example DBGs and NDBG. (a) A 2D interlocking assembly, where the key  $P_1$  is movable along  $d_2$ ; (b&c) two base DBGs of the assembly; and (d) NDBG of the assembly. A part with zero out-degree or in-degree in a DBG is highlighted with a red circle.

on the assembly process such as reducing usage of additional resources (e.g., formwork for construction of architecture). This report reviews existing works in the graphics community to address these problems. We refer readers to [GM15] for a more comprehensive survey on (dis)assembly planning problems and approaches.

**Search for a valid assembly plan.** Given a 3D assembly, there could exist a number of valid plans to assemble the parts. Here, we consider only geometric constraints, i.e., an assembly plan is defined as valid if there is no collision when assembling each part. As mentioned above, assemblies can be naturally represented as graphs. Graph data structures can also guide us in finding valid disassembly plans by maintaining a dynamic graph corresponding to the remaining assembly as parts are successively removed.

**Parts-graph based approach.** In this approach, a valid assembly plan is computed by using the assembly-by-disassembly strategy. The idea is to identify removable parts guided by the parts-graph (see Figure 2(b)) since a part with fewer neighbors in the parts-graph has higher chance to be removable. First, we compute mobility for each part in the parts-graph, e.g., using the joint analysis approach in Section 2. Next, we choose one movable part (usually with few neighbors in the parts-graph), remove it from the assembly using the computed motion, and update the parts-graph accordingly. Since the first step only ensures collision-free infinitesimal rigid motion for the movable part, we still need to check collision with the remaining parts when taking out the movable part in the second step; e.g., the cyan part in the inset can be removed along the translational direction yet the green part has to translate one more step to avoid collision with the orange part. We iterate the above two steps until there is only one part remaining in the parts-graph.

The above approach assumes sequential, monotone, and linear disassembly plans. Thus, finding a valid disassembly plan is a sufficient but not necessary condition of assemblability. To support non-linear disassembly plans, the approach should check mobility not just for each individual part but also for each subassembly. However, this extension will increase complexity of the approach from linear to exponential in the number of parts.

**Blocking graph based approach.** This strategy assumes sequential and monotone disassembly plans, and can find valid non-linear dis-

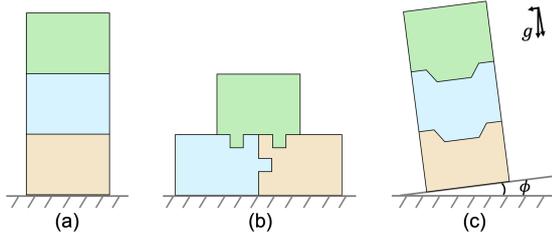
assembly plans with polynomial complexity. A directional blocking graph (DBG),  $G(d)$ , is a directed graph associated with a specific blocking direction  $d$ , where nodes represent parts and edges represent blocking relations between parts along the specific direction  $d$ ; see Figure 4(b&c). It was observed that the motion space (e.g., all possible translational directions) can be partitioned into a finite collection of subsets such that the blocking relations among the parts are constant over every subset. This partition of the motion space and associated DBGs of each motion subset form a non-directional blocking graph (NDBG) [Wil92]; see Figure 4(d). Motion types supported by NDBGs include infinitesimal translation, infinite translation, and infinitesimal rigid motion [WL94].

We describe an NDBG-based disassembly planning approach that considers infinitesimal translational motions. Thanks to the NDBG, the mobility test of each subassembly can be performed on a finite number of translational directions associated with the DBGs. In particular, in each DBG  $G(d)$ , a subassembly is locally free to translate in direction  $d$  ( $-d$ ), if and only if the out-degree (in-degree) of the subassembly in  $G(d)$  is zero; see the red node in Figure 4(c). Thus, if a DBG  $G(d)$  is strongly connected, i.e., if every node can be reached from every other node, no subassembly is movable along  $d$ ; see Figure 4(b). By this, identifying a movable subassembly and its moving direction is converted to identifying strongly connected components in the DBGs, which can be solved by polynomial algorithms [Tar72]. Similar to the parts-graph based approach, the NDBG is updated whenever a subassembly is removed, until there is only one part left in the NDBG.

**Search for a desired assembly plan.** Assembly planning can be formulated as an optimization to find a desired assembly plan. Typical optimization objectives include minimizing assembly complexity (e.g., short assembly path, simple assembly motion), minimizing usage of additional resources (e.g., supports to maintain stability of incomplete architectural structures [DPW\*14]), and maximizing parts visibility for creating visual assembly instructions [APH\*03, HPA\*04]. Please refer to [JW96] for an exhaustive list of objectives on searching assembly plans.

To find an optimal assembly plan, we need to enumerate and evaluate all possible assembly plans based on a selected objective. Although this is possible for assemblies with a small amount of parts, e.g., by using AND/OR tree data structure [dMS90], the complexity increases exponentially with the number of parts. Due to this reason, various practical algorithms were developed to find sub-optimal solutions, e.g., using a greedy algorithm [DPW\*14, MSY\*14], a heuristic search [APH\*03], or an adaptive sampling followed by user editing [KKSS15].

**Discussion.** The above existing works mainly focus on sequential and monotone assembly plans. Although these plans are relatively easy to execute, it is an open problem to study more complex assembly plans. One good example is a recent work [ZBKV20] that finds non-coherent assembly plans to solve two-part disentanglement puzzles, where a part that is inserted may not touch the other previously placed part. Other complex assembly plans include non-sequential plans (see Figure 3(a)) to stabilize parts in an assembly by making them harder to be taken out, and non-monotone plans [MG20] (see Figure 3(b)) to resolve cases where already-assembled parts impede the movements of subsequent parts.



**Figure 5:** Structurally stable assemblies: (a) an assembly in equilibrium; (b) an interlocking assembly, where the green part is the key; and (c) an assembly under tilt analysis, in which the assembly's stability is measured using the critical tilt angle  $\phi$ .

Another promising research direction is *assembly-aware design* that facilitates actual assembly process by not just searching assembly plans but also varying parts geometry and/or layout. This design methodology has been proposed for designing masonry shell structures that require fewer supports for the construction [KKS\*17] and electromechanical devices where each part can be inserted with multi-step translational motion [DMC18], and can be further studied for designing other kinds of assemblies where the assembly or construction process is a big concern.

### 2.3. Structural Stability

An assembly with rigid parts is structurally stable if it can preserve its form under external forces without collapse. Instability of assemblies can lead to catastrophic failure, e.g., in architecture, and thus must be analyzed and accounted for in the design process. Assemblies joined by permanent joints are usually very stable; e.g., certain glue is stronger than the part material. Such assemblies can then be analysed as a monolithic object using the finite element method. This section focuses on stability analysis of assemblies joined by non-permanent joints, which have intriguing property of encouraging disassembly.

To analyze stability of assemblies, two critical conditions, *static equilibrium* and *global interlocking*, are defined mathematically and identified computationally; see Figure 5(a&b). A connection between these two conditions has recently been formalized [WSIP19]; i.e., an interlocking assembly is an assembly that is in equilibrium under arbitrary external forces and torques. Based on this connection, a stability measure has been proposed to evaluate stability conditions that are more strict than equilibrium, but not as restrictive as global interlocking; see Figure 5(c). We review these two stability conditions and one stability measure below.

**Static analysis.** To identify whether an assembly is in equilibrium state under external forces or loads, there are two classes of static analysis methods: *linear elasticity analysis using finite element method (FEM)* and *rigid block equilibrium (RBE) method*. Shin et al. [SPV\*16] proved that a small modification to the linear elastic FEM makes it equivalent to the RBE method to get the same answer to the same static analysis problem. In the following, we describe the RBE method as it is more suitable to guide the design of assemblies in equilibrium.

Given the geometry of a 3D assembly, a static equilibrium state means that there exists a network of interaction forces between the

parts that can balance the external forces acting on each part; i.e., net force and net torque for each part equal zero [WOD09]. Combining equilibrium constraints for each part gives a linear system of equations:

$$\mathbf{A}_{\text{eq}} \cdot \mathbf{f} = -\mathbf{w} \quad (1)$$

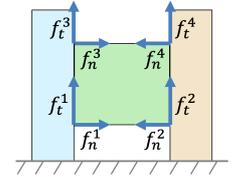
where  $\mathbf{w}$  is a vector containing the weights and external loads applied on each part,  $\mathbf{f}$  is the vector of interface forces, and  $\mathbf{A}_{\text{eq}}$  is the matrix of coefficients for the equilibrium equations. Since contacts or integral joints between rigid parts are typically assumed to have zero tensile strength and limited friction, we have the following two constraints:

$$f_n^i \geq 0, \quad \forall i \in \text{interface vertices} \quad (2)$$

$$|f_{t_1}^i|, |f_{t_2}^i| \leq \alpha f_n^i, \quad \forall i \in \text{interface vertices} \quad (3)$$

where  $f_n^i$  and  $\{f_{t_1}^i, f_{t_2}^i\}$  are respectively the axial and tangential components of the interface force  $f^i$ , and  $\alpha$  is the coefficient of static friction. An assembly is considered in static equilibrium if a solution  $\mathbf{f}$  in Equation 1 can be found, e.g., by using linear programming. Beyond this, an assembly's infeasibility to be in equilibrium can be measured by translating Equation 2 into a penalty form [WOD09].

One limitation of the above method [WOD09] is its inability to accurately predict when parts will slide against one another, i.e., sliding failures [YKGA17]; see the inset for a 2D example. A set of interface forces (see the dark blue arrows) can be found by [WOD09] to balance the gravity of the green part. However, the correct solution is that the green part should always fall under gravity with no resistance, no matter what coefficient of friction is used in this example. To address this limitation, Yao et al. [YKGA17] proposed a *variational static analysis* method that amends the above method [WOD09] with a pair of variational principles from classical mechanics to exclude physically unrealizable forces.



**Interlocking test.** In an interlocking assembly (with at least three parts), there is only one movable part, called the *key*, while all other parts as well as any subset of parts are immobilized relative to one another by their geometric arrangement [SFCO12]; see Figure 4(a) and 5(b) for 2D examples. Starting from the key, the assembly can be gradually disassembled into individual parts by following a specific order. An assembly is called *recursive interlocking* if it has a unique (dis)assembly order, meaning that the assembly remains interlocking after the sequential removal of parts [SFCO12]. An assembly is called *deadlocking* if there is no part that can be taken out from the assembly without collision (i.e., non-disassemblable).

The test for global interlocking essentially tries to identify if there exists a motion configuration that allows taking out any part(s) except the key from the assembly without collision. An assembly is considered to be interlocking if such a motion configuration does not exist. There are two state-of-the-art methods to test for global interlocking: 1) a DBG-based method that considers sequential translational motion only [WSP18]; 2) an inequality-based method that supports infinitesimal rigid motion [WSIP19]. Note

that when parts can be taken out only by non-sequential motion (see Figure 3(a)) and/or rotational motion, the DBG-based method will give a false-positive test result, which can be corrected by applying the inequality-based method.

**DBG-based method.** This approach relies on the DBGs that we have described in Section 2.2. The blocking relations in an NDBG have redundancy, and can be fully represented by using a small set of *base DBGs*; see Figure 4(b&c). The directions associated with the base DBGs are called *base directions*. The DBG-based method shows that an assembly is interlocking if every part (except the key) and every part group cannot translate along the base directions.

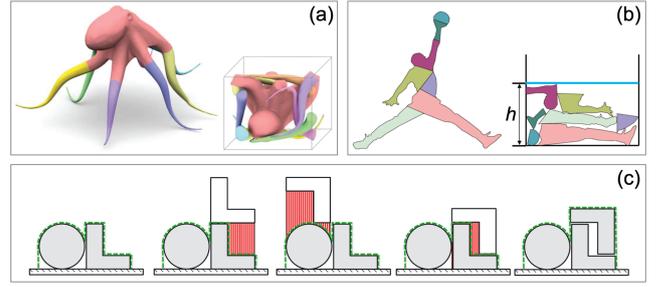
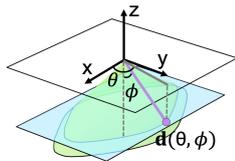
Similar to the assembly planning, we could test mobility of parts and part groups by identifying strongly connected components in the base DBGs. In detail, an assembly is interlocking, if all base DGBs are either: 1) strongly connected; or 2) have only two strongly connected components one of which has a single part that is identical across all base DGBs. Here, the strongly connected component with a single part is the key of the assembly. The direction associated with each DBG with two strongly connected components is the key's (reversed) movable direction according to the in-edge (out-edge) of the key in the DBG; see again Figure 4.

**Inequality-based method.** Wang et al. [WSIP19] proposed a more general method to test global interlocking based on solving the well-known non-penetration linear inequalities in a rigid body system [KSJP08]. Consider an assembly as a rigid body system  $\{P_i\}$ , where each part  $P_i$  can translate and rotate freely in 3D space. Denote the linear velocity of  $P_i$  as  $\mathbf{t}_i$ , the angular velocity of  $P_i$  as  $\omega_i$ , and the local motion of  $P_i$  as a 6D spatial vector  $\mathbf{Y}_i = [\mathbf{t}_i^T, \omega_i^T]^T$ . By stacking the non-penetration constraints for each contact in the assembly, we obtain a system of linear inequalities:

$$\mathbf{B}_{\text{in}} \cdot \mathbf{Y} \geq \mathbf{0} \quad \text{s.t. } \mathbf{Y} \neq \mathbf{0} \quad (4)$$

where  $\mathbf{Y}$  is the generalized velocity of the rigid body system  $\{P_i\}$ , and  $\mathbf{B}_{\text{in}}$  is the matrix of coefficients for the non-penetration constraints among the parts in the system. To avoid the case that the assembly moves as a whole, an arbitrary part (e.g.,  $P_r$ ) needs to be fixed by setting  $\mathbf{Y}_r = \mathbf{0}$ . The system can be solved by formulating a linear program. The assembly is considered as deadlocking if the linear program cannot find any non-zero solution. Interlocking is tested in a similar way, except using more effort to identify a single movable part (i.e., the key).

**Stability measure.** Above methods identify whether a given assembly is structurally stable (i.e., in equilibrium or globally interlocking). Yet, they cannot measure how stable the assembly is. Tilt analysis is a method to measure stability of masonry structures under lateral acceleration [Och02, Zes12]. In real experiments, the ground plane of a structure is rotated around a fixed axis to apply both horizontal and vertical acceleration to the structure based on the gravity. The critical tilt angle  $\phi$  gives the minimum value of lateral acceleration to cause the structure to collapse, and is used as a measure of the structure's lateral stability [SPV\*16, YKGA17]; see Figure 5(c). Wang et al. [WSIP19] generalized this measure by considering



**Figure 6:** Objectives of assembly packing. Minimize (a) the container's volume [YCL\*15], (b) the container's height [CZL\*15], and (c) the underlying free volume (e.g., in red). In (c), the configuration of the white part with minimal underlying free volume is selected on the right for packing it [VGB\*14b].

all possible rotation axes to tilt the ground plane, and proposed an algorithm to compute the measure based on the RBE method [WOD09]. The key idea is to compute a cone of gravity directions under which an assembly is in equilibrium; see the inset. The stability measure is defined as the minimum critical tilt angle  $\phi$  among all possible azimuthal tilt directions  $\theta$ .

## 2.4. Packing Efficiency

Packing is a classical problem in computational geometry and operational research, with the goal of placing as many objects as possible in a given container without overlapping. The packing problem is known to be NP-hard [LMM02]. Extensive efforts have been devoted to solving 2D packing problems (e.g., for texture atlas refinement [LFY\*19] and artistic primitive layout [RRS13]), packing similar objects on a 3D freeform surface (e.g., for architecture design [SHWP09] and personal fabrication [CML\*17]), and packing regular/irregular objects within a 3D container (usually for storage and shipping). For a comprehensive review on packing problems, we refer readers to [CKPT17, LTO\*20].

This report focuses on the problem of packing component parts of a given assembly within a container, called *assembly packing* problem. Since component parts usually have irregular shapes, assembly packing falls in the most challenging packing problems – *irregular packing* [LTO\*20]. In assembly packing, the set of objects (i.e., parts) to be packed are given, including the total number and the geometry, and the goal is to find a realizable packing solution with high packing efficiency. Assembly packing is useful not just in saving storage and transportation space, but also in improving manufacturing efficiency [CZL\*15] and minimizing material waste [KHLM17] by 3D printing or laser cutting parts together in a compactly packed state.

**Problem formulation.** For a given assembly  $\mathbf{A}$  with parts  $P_1, \dots, P_n$  of arbitrary shapes, assembly packing aims to put the parts within a (box) container  $C$  that is as small as possible by optimizing the layout of the parts. The search space of the problem is the pose of each part  $P_i$  in the packed state, represented as a rigid transform  $(R_i, T_i)$  on  $P_i$ 's default pose.

**Objectives.** The main objective of assembly packing is to maximize the packing efficiency, defined as  $E_{\text{pack}} = \sum_{i=1}^n V(P_i)/V(C)$  where

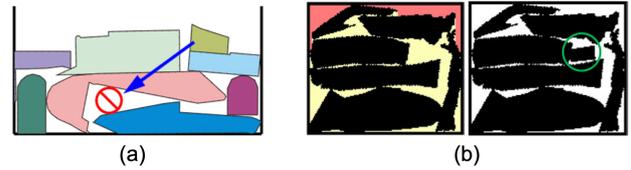
$V(\cdot)$  denotes the volume. Since  $\sum_{i=1}^n V(P_i) = V(\mathbf{A})$  in most cases, which is fixed for a given assembly  $\mathbf{A}$ , maximizing  $E_{\text{pack}}$  equals to minimizing  $V(C)$ ; see Figure 6(a). For powder-based 3D printing, the material cost is actually determined by the height of the packing configuration in a given 3D printing volume. In this case, the objective becomes minimizing the container's height  $h$  [CZL\*15]; see Figure 6(b). A secondary objective could be employed to facilitate finding good packing solutions, which is based on minimizing the underlying free volume [VGB\*14b, Att15]. The rationale is that the underlying free volume is potential wasted space, thus minimizing the free volume equals to maximizing the packing efficiency. In material deposition-based 3D printing, the underlying free volume is directly related to the amount of support material required for printing the packed parts; see Figure 6(c).

**Constraints.** There are three hard constraints for assembly packing. The first one is the bounding constraint, meaning that each part should be included in the container. The second one is overlap-free constraint, to avoid penetration among the parts in the packing configuration. The third one is assemblability constraint, meaning that there should exist a valid plan such that each individual part can be assembled into the packed state [YCL\*15]. This constraint can be satisfied by using the assembly planning methods in Section 2.2. Soft constraints are possible to be enforced in assembly packing. A typical example is preferred parts orientation for a more organized packing result [YCL\*15].

**Problem solver.** Existing approaches to solve the above assembly packing problems can be classified into two classes: *iterative* and *global* approaches.

*Iterative approaches* place the given set of parts one by one in a container, while minimizing the bounding volume  $V_{\text{bound}}$  and the underlying free volume  $V_{\text{free}}$  for the current subset of parts packed in the container [VGB\*14b, Att15, CZL\*15, KHL17]. Due to this iterative scheme, the search space becomes the order to place the parts, and the pose  $(R_i, T_i)$  to place each part  $P_i$  following the order. To solve the problem, the order is determined either based on an empirical criteria (e.g., parts are packed from largest to smallest [Att15, KHL17]) or by using a randomized [KHL17] or tabu search [VGB\*14b]. When placing each part  $P_i$ , its pose  $(R_i, T_i)$  is determined by a height-field based docking test; see Figure 6(c). In detail, for a fixed orientation  $\bar{R}_i$ , part  $P_i$  is docked on top of the packed parts in the container, where the optimal position  $T_i$  is the one that minimizes  $V_{\text{bound}}$  and  $V_{\text{free}}$ . The optimal orientation  $R_i$  is further determined by sampling all possible orientations, performing the docking test for each orientation, and selecting the one that minimizes  $V_{\text{bound}}$  and  $V_{\text{free}}$ . Due to the height-field based docking, there could be unnecessary holes (i.e., wasted space) in the packing configuration [CZL\*15] (see Figure 7(a)), which can be alleviated by additional operations of placing small parts into the holes [Att15] (see Figure 7(b)).

*Global approaches* seek to find poses of an assembly's component parts in a packing configuration by solving a global optimization problem. Yao et al. [YCL\*15] initialize the packed state by making the parts well separated in a large container. Next, they reduce the size of the container, and remove both part-part and part-container collisions by using a rigid body simulator based on shape matching. In their approach, multiple random initializations are re-



**Figure 7:** Iterative approach could have an issue of unnecessary holes in the packing configuration [CZL\*15], which can be alleviated by an additional operation of placing small parts (see the green circle) into the holes (in yellow) [Att15].

quired to ensure high packing efficiency of the final result. Ma et al. [MCHW18] proposed a hybrid optimization approach for packing irregular objects in a 3D container, which can be extended for assembly packing. In the continuous optimization, they initialize the placement of a subset of parts in the container, and use a growth-based strategy to iteratively adjust poses of the parts concurrently to achieve tight packing. In the combinatorial optimization, they perform swapping among the packed parts, as well as part replacement and hole filling operations with the rest unpacked parts, to improve the packing efficiency.

**Discussion.** In assembly packing, an assembly has two states: original state where parts are assembled and packed state where parts are packed in a container. Similar to the original state, a valid assembly plan is a hard constraint for the packed state. In many cases, we expect a simple assembly plan to facilitate the packing process. However, structural stability of the packed state is not a big concern as parts are held by a container and empty space among the parts are possible to be filled, e.g., using some soft material.

### 3. Assembly-based Fabrication

Starting from this section, we will review existing works on computational design of assemblies, some of which are guided by the analysis methods discussed in Section 2. One major reason for designing assemblies is to facilitate the use of specific fabrication techniques. Each fabrication technique has its limitations and constraints, making it impossible or undesirable to make some complex 3D shape as a single monolithic piece. *Assembly-based fabrication* addresses this challenge by representing or approximating the 3D shape with a set of simpler parts that satisfy the fabrication constraints. Additional benefits include reducing material waste, increasing manufacturing efficiency, and improving object surface quality.

This section reviews state-of-the-art methods that design assemblies for typical digital fabrication techniques: 3D printing (Section 3.1), CNC milling (Section 3.2), 2D laser cutting (Section 3.3), and mixed fabrication that employs multiple fabrication techniques (Section 3.4). For each fabrication technique, we briefly discuss its advantages and limitations, and review assembly-based fabrication methods that adapt to these specifics. In particular, assembly-based 3D printing or CNC milling is generally formulated as a *shape decomposition* problem, as these two techniques can fabricate 3D shapes with intricate geometric details. Assembly-based 2D laser cutting is typically formulated as a *shape approximation* or *ab-*

traction problem, where a 3D shape is approximated by a set of planar laser-cut slices.

### 3.1. 3D Printing

3D printing or additive manufacturing is a process of fabricating an object by successively adding material layer by layer. According to the way of adding each layer of material, 3D printing technologies can be categorized into two classes [LEM\*17]. The first class is *material deposition* that creates the next layer by locally depositing material on a previously printed layer; typical examples are Fused Deposition Modeling (FDM) and Material Jetting (MJ). The second class is *layer solidification* that solidifies the top (or bottom) layer of a non-solid material (powder, liquid), e.g., by using a laser; typical examples are Stereolithography (SLA), Digital Light Processing (DLP), and Selective Laser Sintering (SLS).

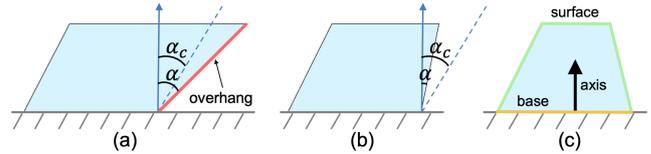
3D printing can fabricate complex objects with fine geometric details, yet also has some well-known limitations. First, the maximum size of an object that a 3D printer can produce as a whole is limited by the size of the printing volume. Second, material deposition-based 3D printing requires support structures to fabricate objects with overhangs; see Figure 8(a). Third, 3D printing process is slow and the printing material is expensive. Lastly, 3D printing objects with desired appearance (e.g., high surface quality) is a challenging task.

Assembly-based 3D printing addresses the above limitations by decomposing a 3D shape into multiple printable parts with simpler shape. The seminal work by Luo et al. [LBRM12] proposed a computational framework for assembly-based 3D printing, mainly addressing the first limitation (i.e., size restriction). After that, a number of techniques have been developed to address not just the size restriction but also the other limitations described above. Section 3.1.1 reviews shape decomposition methods for support-free 3D printing, aiming to generate parts that require no or minimal support for material deposition-based 3D printing. Section 3.1.2 reviews shape decomposition methods for packing efficiency that allow packing parts compactly in the printing volume to reduce printing time and required support material. Section 3.1.3 reviews shape decomposition methods for appearance of assemblies, such that the resulting assembly can have improved surface quality or desired surface color assignment.

#### 3.1.1. Shape Decomposition for Support-free 3D Printing

Material deposition-based 3D printing requires support material to be injected (and later removed) to produce overhangs in a fabricated shape. The angle between the shape surface and its printing orientation is defined as inclination angle denoted as  $\alpha$  [VGB14a]; see Figure 8(a). Due to the strong adhesion between successive layers, 3D printing material can be accumulated without extra support structures when the surface inclination angle  $\alpha$  is smaller than a threshold called the critical inclination angle  $\alpha_c$ , which is usually set to be  $45^\circ$  depending on stiffness of the material. A shape is support-free if there is no inclination angle  $\alpha$  of the shape surface that is larger than  $\alpha_c$ ; see Figure 8(b) for an example.

A support-free shape can be 3D printed without using any support material. This brings a number of advantages such as saving

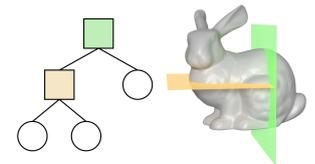


**Figure 8:** 3D printing shapes, where the dashed lines indicate the printing bed: (a) a shape with an overhang (in red) that requires supports for 3D printing; (b) a support-free shape; and (c) a height-field shape, where the height-field base, axis, and surface are colored in orange, black, and green respectively. In (a&b),  $\alpha$  is the inclination angle between the shape surface and the printing direction (the blue arrow) and  $\alpha_c$  is the critical inclination angle.

support material, reducing fabrication time, avoiding tedious work of removing the support, and enabling a smooth surface of the final print. A special case of support-free shape is height-field shape (also called pyramidal shape), which has a flat base with the remaining boundary forming a height function over the base along the fabrication direction; see Figure 8(c). Height-field shape is a necessary condition for fabrication with 3-axis CNC milling. We will review existing works that decompose a 3D shape into height-field parts in Section 3.2.

This section focuses on *support-free decomposition*, where a complex shape is decomposed into a set of (near) support-free parts for 3D printing. According to the strategy to decompose a 3D shape, existing approaches can be classified into: 1) top-down approaches that iteratively split the input shape, usually by using a 3D plane, resulting in planar contacts among the parts; and 2) bottom-up approaches in which primitives (e.g., tetrahedrons) are clustered and merged to form each individual part, possibly resulting in non-planar contacts among the parts. These two classes of approaches are discussed in more detail below.

**Top-down approaches** try to find a set of 3D planes that clip the input shape into a set of parts, aiming to minimize the total support required for 3D printing the parts. These 3D planes are typically organized as a Binary Space Partitioning (BSP) tree [LBRM12], where each intermediate node represents a cut by a 3D plane and each leaf node represents a partition; see the inset for an example. Yu et al. [YYT\*17] proposed an evolutionary approach



to search the clipping planes. In this approach, each candidate solution (referred to as an individual) is encoded as an array that concatenates parameters of each 3D plane in a BSP tree, starting from the root. The fitness function of each individual is evaluated based on the total overhanging area of the corresponding decomposed parts.

To assist the search of clipping planes, some researchers resorted to heuristics derived from shape analysis. Wei et al. [WQZ\*18] addressed the problem of decomposing a 3D shell model with locally tubular shapes, which can be well described by skeletons, into the least number of support-free parts. They reformulated the problem as decomposing a skeleton graph of the model into a minimal set of subgraphs that satisfy the support-free constraints, and developed a

semi-greedy stochastic approach to solve the graph partition problem. Finally, the input model is naturally partitioned into parts by performing planar cuts at each junction of the subgraphs. Karasik et al. [KFW19] addressed the problem of partitioning a general solid object into a small number of support-free parts. The key idea is to identify common non-printable geometric patterns in the object and to use a corresponding strategy to split them into printable or simpler parts. A stochastic search is proposed to apply the splitting strategies recursively over the remaining non-printable parts of the object, until the total number of cuts reaches a predefined limit.

All the above works assume fabrication with conventional 3D printers that have 3 degrees of freedom (DOFs), i.e., translation along X, Y, and Z axes. In contrast, some researchers attempted to decompose a 3D shape into (near) support-free parts for 5-DOF 3D printing such that each part can be 3D printed on top of the fabricated parts with a carefully chosen orientation (i.e., the additional 2 DOFs) [GWYN19]. To ensure printability, the extruder should not collide with the printed parts or the printing bed during the fabrication process. An approach combining the genetic algorithm with simulated annealing was developed to find clipping planes that ensure (near) support-free and collision-free printing process.

**Bottom-up approaches** perform support-free decomposition via clustering according to the likelihood that two primitives interior to the input shape belong to the same, large support-free part. Hu et al. [HLZCO14] addressed the problem of decomposing an object into as few as possible parts that are approximately pyramidal; see Figure 8(c) for an exact pyramidal shape. The algorithm progressively builds larger and larger interior elements of the input shape, from sample points to cells, blocks, and candidate pyramidal parts. The final decomposition, which is a selection of the candidate pyramidal parts, is obtained by solving the exact cover problem. As a post-processing, the contact surface between each pair of neighboring parts is refined to be piecewise planar such that the 3D printed parts can be easily glued together.

### 3.1.2. Shape Decomposition for Packing Efficiency

As mentioned in Section 2.4, parts that can be tightly packed into a 3D printing volume can save not just printing time but also support material. In practice, an over-segmented 3D shape certainly can improve the packing efficiency yet may also complicate the packing and assembly processes with too many parts. Hence, the goal of such decomposition is to achieve a high packing efficiency with a small number of parts. According to whether the decomposition and packing are performed sequentially or concurrently, existing solutions can be classified into *sequential* approaches and *decompose-and-pack* approaches.

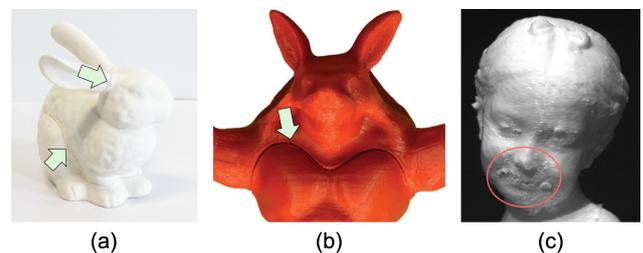
**Sequential approaches** split the problem into two sequential processes: *shape decomposition* followed by *parts packing*, in which heuristics could be applied at the decomposition stage to encourage generating parts that are easier to pack. Vanek et al. [VGB\*14b] developed a computational framework that decomposes a 3D shell model into a set of parts that can be tightly packed into a 3D printing volume. At the decomposition stage, the shell model is divided into a larger number of small segments. These segments are then pre-merged into parts that have sufficient volume and cross-sectional area. At the packing stage, the parts are packed using a

height-field based approach that has been detailed in Section 2.4. The above merging and packing processes can be iterated until further merging is impossible; e.g., one merged part exceeds the available printing volume. Attene [Att15] proposed an algorithm to decompose a 3D solid model into parts that can be efficiently packed within a box. Compared with the approach in [VGB\*14b], this algorithm produces box-like parts using hierarchical segmentation at the decomposition stage to facilitate the later packing process, as boxes are believed to be easier to achieve a high packing efficiency.

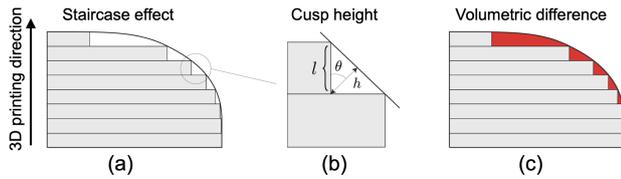
**Decompose-and-pack approaches** address the two sub-problems, shape decomposition and parts packing, concurrently in a unified framework. The goal is to maximize packing efficiency by adjusting not just the parts' poses but also their geometry. Yao et al. [YCL\*15] developed a multi-phase level set framework to decompose-and-pack 3D models for 3D printing; see Figure 6(a) for an example result. Taking a set of pre-segmented parts as input, the framework represents each part as a level set function and optimizes the partitioning and packing to improve packing efficiency as well as other metrics such as aesthetics. In the packing optimization, packing efficiency is maximized by iteratively performing two operations: 1) adjusting poses of the parts; and 2) modifying level set functions (i.e., geometry) of the parts. Chen et al. [CZL\*15] addressed the decompose-and-pack problem by using voxelization as an intermediate geometry representation and exploring the search space via a prioritized and bounded beam search; see Figure 6(b) for an example result. Starting with a coarse decomposition of the input shape, the initial parts are progressively packed into a pile in the printing volume by iteratively performing two possible operations: 1) packing a part onto the pile; and 2) cutting a part and packing one of the resulting sub-parts onto the pile. A key feature of this framework is that it works with pyramidal primitives, which are packing and printing-friendly.

### 3.1.3. Shape Decomposition for Assembly Appearance

When 3D printing an assembly, the final assembly's appearance will be affected by at least four factors: cutting seams, removing of supports, staircase effect, and material/color assignment. Below, we review existing approaches that address each of these factors, aiming to preserve visual appearance of the final assembly. Some research works address multiple factors, e.g., cutting seams and removing of supports in [FAG\*20], staircase effect and removing of



**Figure 9:** Preserve visual appearance of assemblies by (a) putting cutting seams (green arrows) on self-occluded surface areas [FAG\*20] or (b) aligning cutting seams with surface details where discontinuities are expected [YCL\*15]. Unsightly surface artifacts (red circle) due to removing of supports [ZLP\*15].



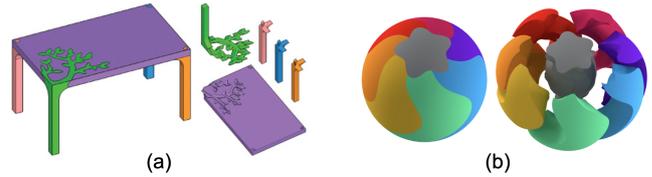
**Figure 10:** (a) Approximating a curved surface with a stack of layers piled along the 3D printing direction introduces the staircase effect and reduces fidelity. Two most widely used methods to measure the staircase effect: (b) cusp height ( $h$  in the figure); and (c) volumetric difference (red region) [LEM\*17].

supports in [WZK16]. For these works, we discuss how they handle each factor respectively.

**Cutting seams** are visible gaps between parts in the final assembly, which cannot be completely avoided due to manufacturing tolerances. To make cutting seams less obtrusive, they are preferred to run through areas of the object surface where they are least likely to be visible or distracting. One approach is based on the observation that cutting seams are less obtrusive in self-occluded areas. The seam unobtrusiveness is measured based on ambient occlusion and cuts are optimized to maximize this measure [LBRM12, FAG\*20]; see Figure 9(a). An alternative approach is based on the observation that cutting seams aligned with surface details, where people expect to see discontinuities, are less noticeable. Hence, non-planar cuts can be optimized to maximize alignment with these surface details [YCL\*15]; see Figure 9(b).

**Removing of supports.** As mentioned before, supports have to be 3D printed for objects that have overhangs (see again Figure 8(a)), and later removed in a post-processing step. Unfortunately, this may cause visual defects on the surface at regions contacted by supports, and even damage small geometric features; see Figure 9(c). These artifacts can be avoided (alleviated) by decomposing an object into (near) support-free parts; see Section 3.1.1. Besides these works, some researchers [FAG\*20, WZK16] aimed to reduce the artifacts of removing supports by decomposing an object into parts that require small supported areas, possibly in occluded surface regions. In these approaches, cuts to decompose the object, together with a printing direction of each part, are found by using a scheme of over-segmentation followed by clustering.

**Staircase effect of 3D printing.** Most 3D printing technologies work by layering, i.e., slicing the shape and then generating each slice independently. This introduces an anisotropy into the process, often as different accuracies in the tangential and normal directions. Due to this reason, regions of the object surface that do not align with the 3D printing direction expose a *staircase effect* and thus reduce fidelity; see Figure 10(a). There are two typical ways to measure the staircase effect [LEM\*17]: 1) the average weighted cusp height; see Figure 10(b); and 2) the sum of volumetric difference; see Figure 10(c). To minimize the staircase effect, some approaches [HBA13, WZK16] decompose an object into multiple parts and find an optimal 3D printing direction for each individual part guided by the two measures, while others [WCT\*15] alleviate the staircase effect by choosing the printing layer height adapting to a saliency map of an input model, i.e., allowing more slicing layers for salient model regions.



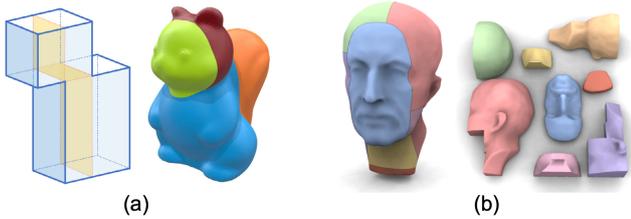
**Figure 11:** Multi-color assemblies. (a) Furniture with decorative joints (right) that conforms to visible surface regions (left) drawn by a user [YKGA17]. (b) A multi-color sphere assembly (right) that conforms to an input multi-color surface segmentation (left) [ACA\*19].

**Material/color assignment.** Making an object whose outer surface consists of regions with different attributes (in particular, material and color) enriches the object appearance significantly. Manufacturing such objects as a single solid requires the use of multi-material 3D printers, which are expensive and not common in domestic environments. One promising alternative is to decompose the object into single-attribute parts that can be 3D printed separately (with different materials/colors) and then assembled to form the target object. This leads to an interesting problem: decomposing a 3D object into a set of parts such that the resulting assembly's appearance conforms to a user-specified multi-color object surface segmentation; see Figure 11. The challenge of this problem is to determine topology and geometry of interface between each pair of neighboring parts while satisfying the requirements of appearance conformity and disassemblability. Yao et al. [YKGA17] addressed this challenge by using a two-step approach: first, find a disassembly sequence such that each part can be disassembled with a single collision-free translational motion; second, construct 3D geometry of each part based on extrusion of the surface segmentation along the part translational direction; see Figure 11(a). Araujo et al. [ACA\*19] developed an approach that supports more complex topology and geometry of interfaces between parts. They formulated the decomposition as an optimization that labels tetrahedrons in a tetrahedral mesh to minimize the energy of appearance conformity and disassemblability, and solved the optimization by sequentially computing three types of unknowns: assembly trajectories, interface topology, and interface location; see Figure 11(b).

### 3.2. CNC Milling

Different from 3D printing (additive manufacturing), CNC milling is a subtractive manufacturing technique that keeps removing material from a starting block using computer-controlled rotary cutters, until only the desired shape is left. Compared with 3D printing, CNC milling supports a larger variety of materials, including non-layerable materials such as stone, operates across a much wider range of scales, and provides higher manufacturing accuracy. Despite these advantages, one significant limitation of CNC milling is that some shapes are impossible to be fabricated when the tool cannot access all the surfaces of the shape.

Among all CNC milling techniques, 3-axis CNC milling is the most inexpensive and easiest to use, in which milling machines can translate the tool along the three axes of the Cartesian system. However, single-pass 3-axis CNC milling is limited to fabricating height-field blocks, which are solids bounded by a flat base and



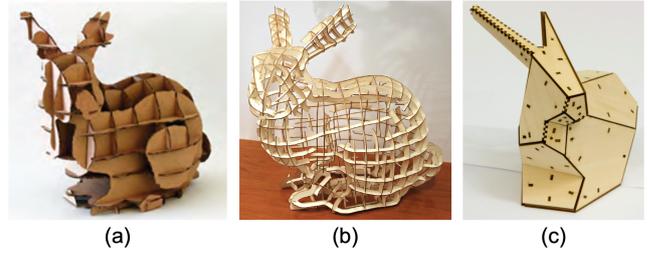
**Figure 12:** Height-field decomposition for CNC milling. (a) Two-step approach based on polycube mapping [FCM\*18], and (b) optimization approach to decompose solid shapes [MLS\*18]

a height-field surface defined along a direction orthogonal to, and located strictly above, the base. This is because the tool of 3-axis CNC milling machines has to be able to reach every point on the object surface by translational motion only; see Figure 8(c) for a 2D example. In practice, the realizable geometry is highly dependent on the available hardware of CNC machines such as the shape and dimension of the milling bit.

To address the above limitation, assembly-based (3-axis) CNC milling aims to decompose a given 3D shape into a set of overlap-free height-field blocks that cover the outer surface of the input shape. This *height-field decomposition* problem is known to be NP-hard [FM01], where the search space includes the number of blocks, and the base, axis, and height-field surface of each block. Compared with the support-free decomposition in Section 3.1.1, the height-field block is a hard constraint that has to be satisfied for every decomposed part. Below, we classify and review existing approaches that address this challenging problem.

**Two-step approaches** explore the search space with two steps: 1) define the number of blocks, and the base and axis of each block; and 2) compute the height-field surface of each block. Alemanno et al. [ACP\*14] approximated an input object shape with an internal base mesh by using a 3D modeling tool, and took each of the base mesh's face (normal) as a high-field base (axis). The height-field surface of each block is computed by decomposing the object surface based on the base mesh. Fanni et al. [FCM\*18] computed the base and axis of each height-field block by using a polycube representation of the input shape and generated the geometry of blocks by performing a polycube decomposition on the shape; see Figure 12(a). As the resulting blocks are not guaranteed to have a height-field surface, the blocks need to go through a height-field checking process.

**Optimization approaches** explore the search space using optimizations. Herholz et al. [HMA15] addressed the problem of decomposing a 3D mesh model into a small set of height-field patches, and formulated it as a discrete labeling problem over the mesh triangles. To solve the problem, they uniformly sample a large number of potential height-field directions, and find connected components of mesh triangles carrying the same label (i.e., the triangles are height-field with respect to the same direction) using a multi-label graph cut algorithm. These connected components (i.e., patches) are thickened to form shell blocks, resulting in possible collisions among the blocks that need to be removed by machining the blocks' backsides manually. Muntoni et al. [MLS\*18] proposed a fully automatic approach to decompose a 3D solid shape into a



**Figure 13:** Assembly-based approaches to make a Bunny model by laser-cutting: (a) abstraction with a spatial assembly [HBA12], (b) abstraction with a mesh-like assembly [CPMS14], and (c) approximation with a shell assembly [CSaLM13].

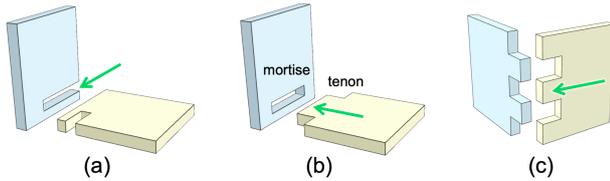
small set of overlap-free height-field blocks. To ensure that a valid decomposition can always be found, they focus on axis-aligned decomposition by constraining the height-field directions to the major axes. Given an input model, they first produce a compact set of large, possibly overlapping, height-field blocks that jointly cover the model surface by solving an optimization. Next, a minimal set cover algorithm is used to reduce the number of decomposed blocks, while allowing overlaps among them. Lastly, the overlaps are resolved by assigning the overlap portion to one of the associated blocks using a graph-based algorithm. Figure 12(b) shows an example decomposition.

### 3.3. Laser Cutting

Compared with 3D printers and CNC machines, laser cutters can only cut 2D polygonal shapes out of a flat plank by controlling the movement of a laser beam that is always perpendicular to the plank. Despite this limitation, laser cutting has many advantages in terms of fabrication like high precision, high speed, fewer constraints on the fabrication space, and supporting a wide range of materials, making it widely used in industrial production and research prototyping.

To fabricate a 3D shape, assembly-based laser cutting aims to abstract or approximate the 3D shape with a set of 2D planar parts made by laser cutting (called laser-cut parts). Assembly-based laser cutting methods can be classified into three classes according to the form of the resulting assembly: 1) a spatial assembly to abstract a 3D volumetric shape (see Section 3.3.1 and Figure 13(a)); 2) a mesh-like assembly to abstract a 3D surface (see Section 3.3.2 and Figure 13(b)); and 3) a shell assembly to approximate a 3D surface (see Section 3.3.3 and Figure 13(c)).

In these assemblies, joining 2D laser-cut parts to form a steady assembly is a challenging task due to the limited contact area among the planar parts. Although additional connectors [CSaLM13] or wires [RA15] can be used to join laser-cut parts, the most effective way is to create integral joints on the laser-cut parts (called laser-cut joints), which can not only stabilize the whole assembly but also simplify the assembly process; see again Section 2.1. Typical examples of laser-cut joints are halved joints, mortise-and-tenon joints, and finger joints; see Figure 14.



**Figure 14:** Three types of laser-cut joints: (a) halved joint, (b) mortise-and-tenon joint, and (c) finger joint.

### 3.3.1. Spatial Assembly

Laser-cut parts can mutually intersect with one another in the full 3D space to form a spatial assembly, which are typically used as shape abstractions (i.e., sculptures), furniture, or architecture.

*Shape abstractions.* A laser-cut spatial assembly can be designed to abstract a 3D shape, as a sculptural art form; see Figure 13(a). In these assemblies, planar parts have prefabricated halved joints (see Figure 14(a)) at their intersections and are assembled by sliding them together.

McCrae et al. [MSM11] generated shape abstractions by progressively selecting planes to cover geometric features of a given 3D shape, guided by principles inferred from user studies that dictate priorities of these features. This technique was later extended in a computational tool [MUS14] for interactive design and fabrication of user-desired sculptures with planar sections. In these two works, shape abstraction is the main focus, yet there is no guarantee that the computed spatial assemblies are physically constructible only by sliding parts. To resolve this issue, Hildebrand et al. [HBA12] represented such spatial assemblies as an extended BSP tree, and used it to quickly evaluate if a newly added planar part can be physically slid into the assembly. Shape abstractions are generated by iteratively adding new planar parts that are generated by a sampling strategy, following an assembly plan that is computed using a branch-and-bound approach.

*Furniture-like assemblies.* Schwartzburg and Pauly [SP13] proposed a computational approach to design spatial assemblies composed of planar parts that are nonorthogonally connected by prefabricated halved joints. An optimization algorithm is developed to find feasible configurations (i.e., orientation, position, and shape) of planar parts that satisfy the fabrication, rigidity, and assembly constraints. Koo et al. [KHLM17] investigated waste-minimizing furniture design, aiming to provide design variations that result in less wastage of materials. To generate such design variations, 2D packing efficiency of planar parts is maximized for laser cutting by jointly optimizing poses of the parts and furniture design parameters, while maintaining original design intent specified in the form of design constraints. Once obtaining the parts geometry, integral joints (see Figure 14) are further constructed on the parts for their connection.

### 3.3.2. Mesh-like Assembly

Mesh-like assemblies are designed to abstract a free-form 3D surface with a set of ribbon-shaped slices, where each slice is oriented guided by the surface tangent like a mesh edge; see Figure 13(b).

Given a 3D surface, there are two ways to generate such assem-

blies as shape abstractions: 1) use long, curved slices with prefabricated halved joints [CPMS14]; 2) use short, straight slices joined with additional connectors [RA15]. Cignoni et al. [CPMS14] approximated a 3D surface with a set of curved slices that are placed driven by a smooth cross-field defined on the surface, providing an appealing, uniform distribution of slices over the surface. Halved joints with enlarged slits are constructed on the slices to enable nonorthogonal connections among the slices in a structurally sound way. Richter et al. [RA15] represented a 3D surface with a set of straight slices distributed as torsion-free edges of the mesh that are jointed at their meeting point (i.e., vertex). The geometry of the slices are constructed in two steps: 1) generate a coarse, low valence mesh approximation using a variant of anisotropic centroidal Voronoi tessellation; and 2) create slices by modifying the mesh while incorporating constraints like torsion-free and minimal slice length using an iterative optimization.

### 3.3.3. Shell Assembly

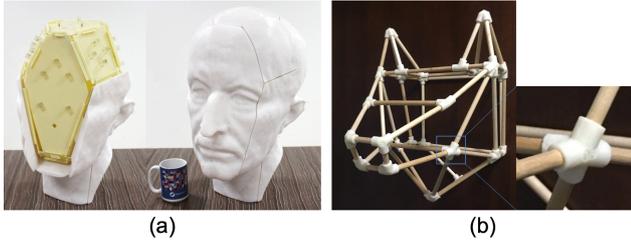
Shell assemblies are designed to approximate a closed 3D surface for quick and low-fidelity fabrication with laser cutting. Different from spatial and mesh-like assemblies, each part in shell assemblies are oriented following the surface normal like a mesh face; see Figure 13(c).

Chen et al. [CSaLM13] approximated a 3D mesh model with a small number of polygons that form a closed surface, by iteratively clustering mesh faces, estimating planes for each cluster, and slowly deforming the faces toward each plane. Once the polygons are obtained, they are thickened to form planar parts. Additional planar connectors are generated to internally join neighboring parts. Chen et al. [CS15] extracted planar surfaces of a given mesh model to form planar parts, and generated finger joints (see Figure 14(c)) sequentially at the intersection of every pair of neighboring parts for their connection. Baudisch et al. [BSK\*19] presented an interactive system for designing sturdy laser-cut 3D objects based on closed box structures connected with finger joints. Compared with [CSaLM13] that handles free-form shapes, the resulting shell assemblies in [CS15, BSK\*19] have more regular shape.

## 3.4. Mixed Fabrication

Mixed fabrication refers to utilizing multiple manufacturing techniques in combination to assemble a single product, aiming at combining their strengths and/or compensating their drawbacks. This technique, combined with interactive design, has been extensively studied in the human computer interaction community, e.g., for low-fidelity fabrication [BGM\*15, KSW\*17]. We refer readers to [BM17] for a comprehensive review on this topic. In this report, we focus on computational methods that automatically generate geometry of parts/joints for mixed fabrication, guided by some design objectives. Most of these methods combine the strengths of 3D printing and 2D laser cutting for fabrication; see Figure 15 for examples. More precisely, 3D printing is able to fabricate objects with fine geometric details, yet its drawbacks are high material cost and long printing time, which can be well compensated by 2D laser cutting. We classify these methods into two classes according to the design objective.

**Cost-effective large object fabrication.** This class of works aims

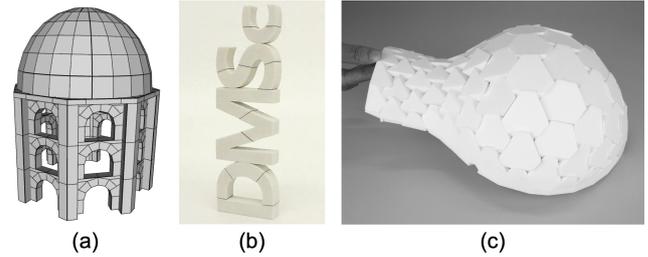


**Figure 15:** Mixed fabrication with 3D printing and 2D laser cutting: (a) cost-effective large object fabrication [SDW\*16]; and (b) frame structure with 3D printed joints and laser-cut rods [Jac19].

at fabricating large objects at lower cost and higher speed, by combining 3D printing with 2D laser cutting or universal building blocks. The key idea is to build a coarse internal base within the given 3D object, and then attach thin 3D printed parts, as an external shell, onto the base to recover the fine surface details. Gao et al. [GZN\*15] fabricated a functional object by 3D printing a partitioned object shell around a laser-cut cuboid, which is used as the internal base as well as a container to put electronic and sensory components. The total material cost is significantly reduced by maximizing volume of the laser-cut cuboid. Inspired by this work, Song et al. [SDW\*16] proposed to design the internal base as one or multiple convex polyhedrons, fabricated also with laser-cutting; see Figure 15(a). To minimize the total material cost, an optimization is developed to approximate a given 3D shape with convex polyhedrons of maximized volume, while requiring that each polyhedron is strictly within the shape. 3D printing also can be combined with reusable universal building blocks for cost-effective fabrication. Chen et al. [CLF\*18] built the internal base with universal building blocks, which have higher flexibility to fill up an input shape's interior. Due to this reason, this approach can achieve a better performance than [SDW\*16] in terms of material saving.

**Rod-joint structure fabrication.** This class of works aims at fabricating free-form frame structures by connecting laser-cut rods with 3D printed joints. The rationale behind is to delegate complex geometry to the joints fabricated by 3D printing while the intrinsic strength is retained by the rods that require only laser cuts. Tonelli et al. [TPCS16] designed a grid-shell architectural structure by optimizing a polygonal tessellation on a given 3D surface to maximize the static and aesthetic performance, and fabricated a mockup of the shell as a rod-joint structure. Chidambaram et al. [CZS\*19] developed a computational tool to design structurally sound rod-joint structures from a given 3D mesh model, where the key feature is a recommendation system that guides users' local editing to improve the structural and functional performance. Jacobson [Jac19] developed a computational tool to design, fabricate, and assemble general rod-joint structures. Once a structure is designed, the tool automatically generates geometry for 3D printable joints and cutting plans for rods, as well as assists users to find a feasible assembly plan; see Figure 15(b).

**Discussion.** Besides the four kinds of digital fabrication techniques described in this section, researchers are also exploring the potential of using assembly-based approaches for making objects with other fabrication techniques such as 5-axis CNC machining [LXG10], molding where the mold is an assembly while the



**Figure 16:** Designing assemblies in equilibrium: (a) masonry building [WOD09], (b) equilibrium puzzle [FMB15], and (c) free-form architectural assembly [WSIP19].

object is a single piece [NAI\*18], wire bending machines [LFZ18], and multi-tool fabrication devices [VTSOP20].

#### 4. Structurally Stable Assemblies

Section 3 represents or approximates a 3D shape with a set of simple parts for fabrication. Very few such works consider structural stability of the resulting assembly as they assume fabricated parts can be connected by a wide variety of joining methods such as glue; see again Section 2.1. Among them, some research works [CPMS14, SPI3] use integral joints and enforce local joining constraints at the design stage to facilitate stability based on some heuristics.

This section focuses on computational methods to design structurally stable assemblies, guided by a formal stability analysis that has been described in Section 2.3. We classify these methods into two classes according to the design objective, either to make the assembly *in equilibrium* or *globally interlocking*. To encourage disassembly, all these methods choose to use integral joints, or simply planar contacts, for connecting the parts. Assemblies designed in this section can have a wide variety of geometric forms and be used in many different applications; see Figure 16 and 17.

##### 4.1. Assemblies in Equilibrium

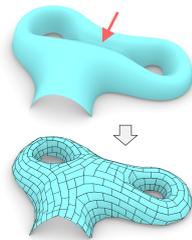
An assembly is in static equilibrium if interaction forces between the parts can balance external forces acting on the assembly, mainly the gravity. These assemblies are common in architecture, furniture, and puzzles; see Figure 16. However, designing them is a non-trivial task as an equilibrium state depends on not only the parts (with integral joints) geometry, but also their geometric arrangement as well as the material property (i.e., friction coefficient).

Section 2.3 has presented the RBE method to test whether an assembly is in equilibrium. Hence, a general way to design assemblies in equilibrium is using the RBE method to guide the search of a feasible configuration of parts geometry and arrangement. The other class of methods is more specific and focuses on designing free-form architectural assemblies like pavilion and dome, which relies on designing a *self-supporting surface*.

**Design guided by the RBE method.** The RBE method described in Section 2.3 can not only test if a given assembly is in equilibrium under known external forces, but also provide a measure of the assembly's infeasibility to be in equilibrium when it fails the test.

Whiting et al. [WOD09] integrated the RBE method with procedural modeling to design masonry structures that are in equilibrium under gravity, and they used a heuristic algorithm to search the parameter space such that the infeasibility measure can be decreased to zero; see Figure 16(a). Later, Whiting et al. [WSW\*12] extended this approach by using a gradient descent algorithm to explore the parameter space, in which a closed-form of the infeasibility measure's derivative with respect to the parts geometry variation is derived. Frick et al. [FMB15] developed an interactive tool to design assemblies in equilibrium by decomposing a given 3D shape into a set of parts with planar cuts; see Figure 16(b). The tool keeps visualizing required tension forces computed by the RBE method and allows users to edit the planar cuts interactively until all the tension forces are removed.

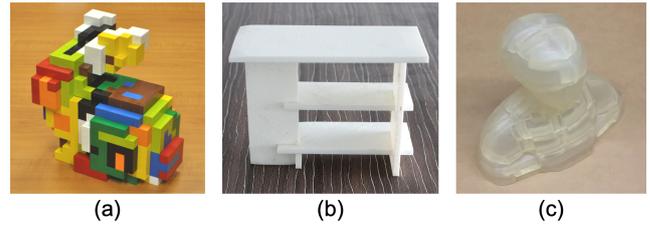
**Design based on self-supporting surfaces.** According to the safety theorem [Hey66], an assembly is self-supporting (i.e., equilibrium under gravity) if there exists a thrust surface contained within the structure that forms a compressive membrane resisting the load applied to the assembly. Once the thrust surface, also called *self-supporting surface*, is ready, a self-supporting assembly can be easily generated by thickening and partitioning the surface into parts. Thrust Network Analysis (TNA) developed by Block and Ochsendorf [BO07] is a well-known graphical approach for form exploration of self-supporting surfaces. Inspired by this work, the graphics community has proposed a number of geometry processing methods to approximate free-form surfaces with self-supporting ones; see [MHS\*19] for an overview of these methods. In particular, Panozzo et al. [PSSH13] not just generated self-supporting surfaces, but also fabricated corresponding self-supporting assemblies to verify the stability. The inset figure shows an example target surface and the resulting self-supporting assembly. Note that a local geometric feature in the target surface (highlighted by a red arrow) is deformed to make the surface self-supporting.



## 4.2. Interlocking Assemblies

An assembly is interlocking if it can be in equilibrium under arbitrary external forces when the key is held by other means [WSIP19]. According to the static-kinematic duality, the interlocking property of a given assembly can be tested by using kinematic methods described in Section 2.3. Compared with assemblies in equilibrium, interlocking assemblies are more structurally stable under unpredictable external forces yet enforce higher complexity on the parts geometry and their joining. The main challenge of designing interlocking assemblies is to ensure two conflicting properties simultaneously: interlocking and disassemblable. This is because interlocking requires strict joining to restrict relative movement among parts yet disassemblability demands at least one collision-free plan to separate the parts (i.e., not deadlocking).

A straightforward way to design interlocking assemblies is to exhaustive search all possible configurations and perform the interlocking test. This method has been tried by Cutler [Cut78] in the late 1970s to discover new six-piece interlocking configurations,



**Figure 17:** Designing interlocking assemblies: (a) interlocking puzzle [SFCO12], (b) furniture [FSY\*15], and (c) object shell assembly for 3D printing [YCXW17].

which took almost three years to search a cubical volume of less than  $4^3$  voxels due to the combinatorial complexity. Later, Xin et al. [XLF\*11] developed a retargeting approach to create 3D interlocking puzzles by replicating and connecting multiple instances of an existing six-piece interlocking burr structure within a given target shape. Until recently, a few computational methods have been developed to design new interlocking assemblies, making it possible to increase the number of parts and to enrich geometric forms of assemblies significantly; see Figure 17. The design problem in these works is formulated as, *shape decomposition* or *joint planning*, according to the given input.

**Shape decomposition.** When the input is a target shape, computational design of interlocking assemblies can be formulated as a *shape decomposition* problem. A typical approach to address this problem is to construct Local Interlocking Groups (LIGs), which are a subset of connected parts that are locked by a specific key in the group, and to enforce dependency among these LIGs. The advantage of this approach is that the resulting assemblies are guaranteed to be globally interlocking. Yet, the limitation is that the explored search space is restricted to a small subset of all possible interlocking configurations. Song et al. [SFCO12] first proposed this approach and used it to construct 3D interlocking puzzles. Given a voxelized 3D shape, their method iteratively extracts pieces while enforcing a local interlocking condition among every three consecutive pieces; see Figure 17(a). This method was later extended to handle smooth non-voxelized shapes for 3D printing [SFLF15] and to design 3D steady dissection puzzles [TSW\*19].

**Joint planning.** When the input is a set of initial parts without joints, designing interlocking assemblies can be formulated as a *joint planning* problem. The goal is to plan and construct a set of predefined joints (e.g., mortise and tenon joint in Figure 1(c)) on the initial parts to make them interlocking. Fu et al. [FSY\*15] focused on plate structures such as furniture that have been initially partitioned into parts, and computed an interlocking joint configuration following the LIG-based approach, where each LIG has only 3 or 4 parts and thus the joint configuration in each LIG can be searched exhaustively; see Figure 17(b). This method has been extended to interlock 2D laser-cut parts into a convex polyhedron [SDW\*16] and to design reconfigurable furniture with multi-key interlocking [SFJ\*17]. Yao et al. [YCXW17] designed interlocking shell assemblies for 3D printing by using a randomized search with pruning to generate candidate joint configurations and verifying their global interlocking by using physically based simulation; see Figure 17(c).

To address the above two classes of problems in a unified framework, Wang et al. [WSP18] represented parts blocking relations with a family of base DBGs (see again Figure 4) and leveraged efficient graph analysis tools to test and compute an interlocking arrangement of the parts; see also Section 2.2 and 2.3. Compared with the above approaches, the strength of this approach is two-fold: 1) allowing exploring the full search space of interlocking configurations; and 2) supporting a wide range of geometric forms of resulting assemblies. As mentioned in Section 2.3, the DBG-based interlocking test assumes that parts have to be taken out with sequential translational motion, which is not always the case (see Figure 3(a) for an example). Hence, this approach requires verify the generated designs by performing the inequality-based interlocking test method in Section 2.3, when parts are non-orthogonally connected.

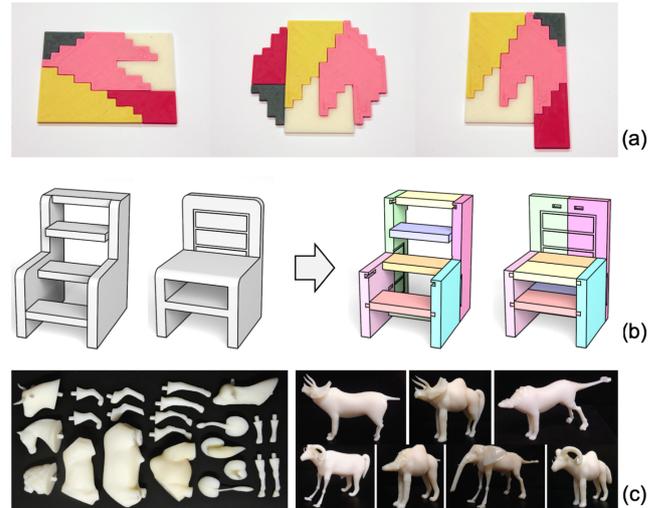
**Discussion.** In practice, equilibrium under gravity might be insufficient since an assembly could be exposed to different forces (e.g., live loads). This motivates the stability measure based on the tilt analysis described in Section 2.3, where an assembly can be in equilibrium for a cone of gravity directions, as well as the work of optimizing free-form architectural assemblies to maximize the stability measure [WSIP19]; see Figure 16(c). On the other hand, global interlocking might impose too strict constraints on the assembly's geometry, as real assemblies usually do not have to experience arbitrary external forces. Hence, some research works relax the constraint of global interlocking, e.g., by allowing multiple keys in the final assembly [SFJ\*17] or allowing parts to be immobilized by using geometric arrangement together with friction [TSW\*19].

## 5. Reconfigurable Assemblies

Different from the assemblies in Section 3 and 4 that typically have a single geometric form, reconfigurable assemblies consist of a common set of parts that can be rearranged into multiple forms for use in different situations [GJG16]. By reusing the component parts, reconfigurable assemblies have advantages like cost efficiency, multiple functionalities, and requiring less storage space, for example. In this report, we focus on reconfigurable assemblies that have clearly defined target geometric forms, usually specified by users as 2D or 3D shapes. We classify existing works on the design of reconfigurable assemblies into two classes, *free reconfigurable* assemblies and *hinged reconfigurable* assemblies, according to whether the component parts are separable or hinged during the reconfiguration process. Free reconfigurable assemblies can be disassembled into individual parts, and each of these parts can be rigidly transformed in full 6 DOFs before being re-assembled into another form; see Section 5.1. In contrast, in hinged reconfigurable assemblies, parts remain connected at certain joints when transforming across different forms; see Section 5.2.

### 5.1. Free Reconfiguration

The goal of designing free reconfigurable assemblies is to generate a common set of parts, as well as their geometric arrangement in each configuration, to reproduce a set of user-specified 2D or 3D shapes. This design problem can be formulated as (approximated) geometric dissection if only the target shapes are provided and as designing interchangeable parts if the target shapes have been compatibly segmented.



**Figure 18:** Free reconfigurable assemblies. (a) 2D dissection puzzle with three forms [ZW12]; (b) reconfigurable furniture (right) designed from two input 3D shapes (left) [SFJ\*17]; and (c) assemblies (right) built with interchangeable parts (left) [DYY16].

**Geometric dissection** is a problem that has been studied for a long time in recreational math and computational geometry [Fre97]. In its basic form, geometric dissection partitions a 2D shape into a finite number of pieces that can be rearranged into a new shape of equal area. According to the Wallace-Bolyai-Gerwien theorem, given any two 2D polygons with equal area, a dissection between them always exists. However, the proof does not restrict the number of parts, often leading to over-segmented parts that are physically implausible. Meanwhile, the task of finding an exact  $K$ -piece dissection between two 2D polygons has been proved to be NP-hard [BDD\*15].

Recently, several computational methods have been developed to address the dissection problem, e.g., for designing dissection puzzles or making reconfigurable furniture; see Figure 18(a&b). To facilitate the search of feasible dissection results with a small number of parts, these methods restrict the search on a discrete lattice space and/or allow small modifications on the input shapes. Zhou et al. [ZW12] proposed a hierarchical clustering method to create dissection puzzles on general 2D shapes represented as a discrete square or triangular lattice; see Figure 18(a). They also showed an extension to dissect a few simple 3D shapes that are represented as a cubic grid. Duncan et al. [DYYT17] relaxed the dissection problem by letting input shapes impose soft rather than hard constraints on the dissection process, and developed a tree search algorithm to create a dissection result that approximates a given pair of 2D naturalistic shapes.

Unlike 2D dissection puzzles that can be supported on a tabletop surface, 3D dissection puzzles are preferable to be stable under gravity and during manipulation. Tang et al. [TSW\*19] addressed this problem by co-decomposing two voxelized shapes and making the resulting assemblies stable based on a generalized interlocking model; see also Section 4.2. Their method allows slight modification on the input shapes, with preference on the interior to preserve

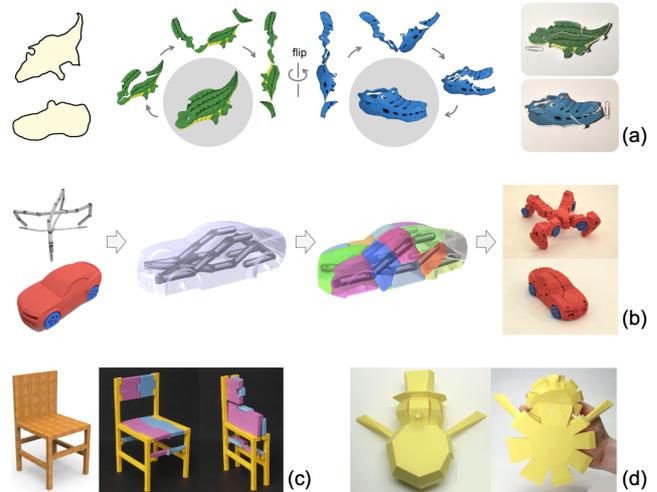
the puzzle's external appearance. Besides dissecting 3D voxelized shapes, researchers also attempted to dissect spatial shapes like furniture. Song et al. [SFJ\*17] formulated the design of reconfigurable furniture as a weakly-constrained dissection problem by relaxing the one-to-one correspondence of parts in different forms. They derived their solution by iteratively constructing parts with correspondence while maximizing parts reuse and reducing necessary modifications on the input shapes. They also proposed a half-joint graph that models joint connections between parts over multiple forms. This graph is used for joint planning to make the assembly interlocking in each form; see Figure 18(b).

**Designing interchangeable parts.** Given a collection of compatibly segmented 3D shapes, Duncan et al. [DYY16] partitioned them into parts and deformed connecting boundaries of parts with similar semantic meaning to make these connecting boundaries identical, while minimizing deviation from their original appearance. After adding unified mortise-and-tenon joints, parts with identical local boundaries are interchangeable for building novel objects with a coherent appearance; see Figure 18(c). This work was inspired by the concept of assembly-based modeling [FKS\*04], in which new shapes are constructed by connecting components from existing shapes, and can be considered as a physical realization of modeling with replaceable substructures [LVW\*15].

## 5.2. Hinged Reconfiguration

Designing hinged reconfigurable assemblies requires creating not just a common set of parts but also hinges among them. Compared with free reconfigurable assemblies, the motion of parts in hinged assemblies is more restricted due to the hinge connection. The focus of these hinged assemblies is to make use of their multiple forms, and hinges just provide a way to transform across these forms. This makes them distinct from mechanical assemblies with hinges such as linkages [NBA19, UTZ16], whose focus is to transfer motion and force by transforming the parts. Research efforts have been devoted to computational design of three classes of hinged reconfigurable assemblies: a special geometric dissection called hinged dissection, transformables that can shift across different shapes, and foldable assemblies for space saving.

**Hinged dissection** is a special class of geometric dissection that connects all the parts into a chain at "hinged" points and converts a shape to another by swinging the chain continuously. It has been proved that a hinged dissection always exists when two polygons have equal area [AAC\*12]. Li et al. [LMaH\*18] studied a special class of hinged dissection called reversible hinged dissection, where one 2D shape can be dissected in its interior and then inverted inside-out, with hinges placed at the shape boundary, to reproduce the other 2D shape; see Figure 19(a). They developed a quick filtering scheme to pick potential input shape pairs from a 2D shape collection, and proposed an automatic algorithm that constructs an approximate dissection between two input shapes by finding a pair of conjugate trunks, which are a pair of polygons with the same edges (yet in an inverse order) in the two shapes respectively. Zhou et al. [ZSMS14] addressed an approximate 3D hinged dissection problem to design an object assembly that can be folded into a box, possibly with visible gaps. Their algorithm has three steps: 1) partition a 3D shape into approximately cubic pieces



**Figure 19:** Hinged reconfigurable assemblies. (a) A 2D hinged dissection puzzle [LMaH\*18] that approximates a given pair of 2D shapes (left). (b) Designing a 3D transformer that has two forms (i.e., a quadruped robot and a car) [YZC18]. A foldable (c) chair [LHAZ15] and (d) snowman [MEKM17] for space saving.

based on voxelization; 2) assign a hinge type to each pair of neighboring voxels, aiming to obtain a low energy tree that spans all the voxels; and 3) find a collision-free sequence that folds the object into a cube based on physical simulation and user interactions.

**Transformables** are physical characters whose overall shape can be changed merely by rotating or translating their component parts. One typical example is transformers, which are robots that can change their shapes to perform different locomotions and/or tasks. To design transformables, researchers take a character skeleton and a target shape as inputs (see the left of Figure 19(b)), and formulate the design problem as a decomposition of the target shape guided by the skeleton. Huang et al. [HCLC16] addressed the problem by adjusting the skeleton, embedding it into the target shape, and finding an optimal decomposition of the shape into parts using simulated annealing. To animate the transformables, they proposed a two-level motion-planning process to find collision-free transforms between the two shapes. Yuan et al. [YZC18] addressed a similar problem, yet focused on designing transformables for fabrication; see Figure 19(b). Their algorithm is different from that of [HCLC16] in three aspects: 1) find an optimal pose of the skeleton inside the object mesh for a tight and collision-free embedding; 2) use a collision-aware multiple level set-based growth model to decompose the target shape, starting from the skeleton mesh; and 3) find a collision-free transform between the two shapes by using a greedy search together with post-processing (i.e., caving) on the parts to remove collision. Different from these two works, Yu et al. [YYL\*19] focused on transformables whose different shapes are formed via folding and twisting a chain structure with many identical parts hinged together. Given multiple input models represented as voxelizations, they designed such transformables by finding a simple path as the chain configuration that connects most voxels in these models.

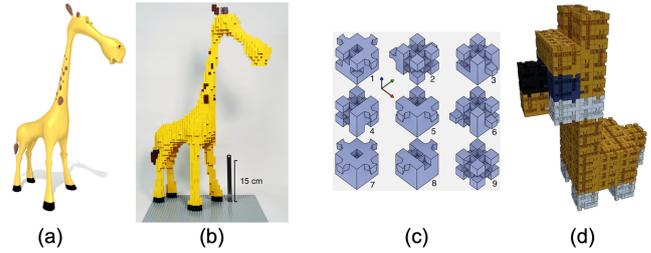
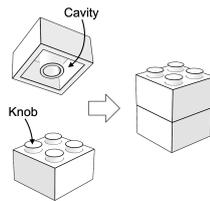
**Foldable assemblies** generally have two forms, one folded form for saving space and one unfolded form for usage. This concept has been successfully practiced in space-saving furniture, foldable shapes for 3D printing, and scissoring structures [ZWC\*16, ZSC16]. Li et al. [LHAZ15] addressed the problem of designing space-saving furniture. Given an input 3D object with a folding direction, their goal is to apply a minimum amount of modification to the object so that it can be folded to save space; see Figure 19(c). To achieve this goal, a two-level approach is proposed based on partitioning the input shape into a set of folding units. For each unit, they made it foldable with respect to the prescribed folding direction by hinge insertion and part shrinking. The order of folding the units is further found using a greedy scheme to minimize the total foldabilization costs. Ibrahim and Yan [IY18] addressed a similar problem, except that the folding direction is computed automatically rather than given. Taking a segmented furniture model as input, they jointly optimize for joint locations, a folding order, and folding angles for each part of the model, such that the input can be transformed to a compact form. Splitting of some parts is allowed to improve the compactness. Zhou and Chen [ZC18] addressed the problem of designing convertible furniture. Given a set of desired furniture units as input, they proposed an algorithm that assigns junctions (i.e., hinges or slides) to connect furniture units such that these units can collapse into a compact form. Rather than focusing on furniture, Miyamoto et al. [MEKM17] proposed a method for creating a flat-foldable model with rigid panels that approximates an input 3D mesh. Their method divides the input mesh into multiple components, converts them into convex shapes, and optimizes shapes and positions of panels of each convex component to make the whole model flat-foldable; see Figure 19(d).

## 6. Building Assemblies with Tileable Blocks

Tileable blocks are a set of geometrically unique parts whose instances can build up complex assemblies. These blocks are universal and reusable in many constructions, bringing advantages like high utilization and low cost for mass production. Probably the most well-known tileable blocks are LEGO bricks, which have been studied extensively. Researchers also designed and investigated customized tileable blocks to build planar and 3D structures. Once the tileable blocks are given, the design of assemblies can be formulated as a *layout optimization* problem, for which typical objectives include shape, structural stability, and visual appearance of the assembly.

### 6.1. LEGO Bricks

LEGO is a well-known assembly toy for creative construction. LEGO bricks are colorful plastic building blocks, among which the most commonly used ones are of cuboid shape with various lengths, widths and heights. A LEGO brick has knobs on its top side and cavities on its bottom side; see the inset. When the knobs and cavities are snapped together, there are normal forces exerted between them, leading to a non-zero maximum static friction load between a knob and a cavity. As a result, the bricks can fit firmly due to this friction



**Figure 20:** Solving the LEGO construction problem [LYH\*15]: (a) an input 3D model and (b) an output stable LEGO assembly. Building 3D assemblies with customized tileable blocks: (c) a set of nine interlocking voxels proposed by [ZB16] and (d) an assembly built with these voxels.

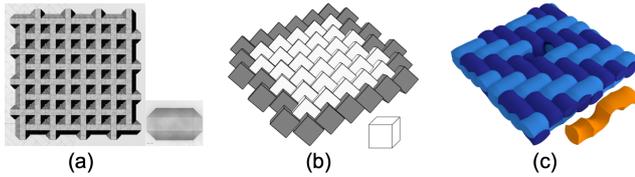
contact. Stacking bricks layer by layer will form 3D sculptures of various shapes.

The LEGO construction problem is defined as: ‘Given any 3D body, how can it be built from LEGO bricks?’ [GHP98]. One major challenge of this problem is to find layouts that are physically stable, especially for large-scale LEGO assemblies. Since the first work by Gower et al. [GHP98], a number of approaches have been proposed to address this challenge, typically using heuristics for stability estimation; see [KKL14] for a survey. In particular, Testuz et al. [TSP13] represented a LEGO assembly as a parts-graph and used a randomized greedy algorithm to search for a LEGO layout by performing merging and splitting operations on the LEGO bricks. To maximize the stability, their algorithm tries to find the optimal configuration with more edges and fewer weak articulation points in the graph representation. Yun et al. [YPYM17] extended the representation as a two-colored graph, where a red edge represents brick connectivity while a blue edge represents brick adjacency. They measured a LEGO assembly’s stability based on the red-edge distance for adjacent bricks connected by a blue edge. Different from the above heuristics, Luo et al. [LYH\*15] measured a LEGO assembly’s stability based on the difference between the maximum friction load and the actual friction forces computed using the RBE method described in Section 2.3. Guided by this measure, their approach iteratively applies local and stochastic search to obtain layouts that gradually and strictly improve the stability; see Figure 20(a&b).

The above works purely use cuboid LEGO bricks as the building blocks. Hence, an initial layout with  $1 \times 1$  LEGO bricks can be intuitively generated based on voxelization of the input 3D model. However, this strategy is infeasible when sloping or circular LEGO bricks are taken into consideration. Zhou et al. [ZCX19] addressed this challenge by identifying visual features in the input 3D model and representing them with best-fitting bricks, during which the input model could be deformed to adapt to the discrete nature of LEGO bricks.

### 6.2. Customized Tileable Blocks

Constructing structurally stable assemblies with tileable blocks is intriguing in engineering and architecture, and has recently attracted great interest in the graphics community. At the end of 17th



**Figure 21:** Building planar assemblies with tilable blocks: (a) the Abeille flat vault; (b) a topological interlocking assembly with cubes [DEKBP03], where the boundary is highlighted in black; and (c) an assembly with bi-axial weaving patterns built with a single tileable block (in orange) [KAS\*21]

century, Joseph Abeille discovered that identical tetrahedrons truncated at two opposite edges can be arranged to form a planar stable assembly, known as Abeille flat vaults; see Figure 21(a). Since then, several variants of these structures have been invented and studied under the name of *topological interlocking (TI) assemblies* [DEKBP03]. Physical experiments conducted on TI assemblies have shown that they possess interesting and unusual mechanical properties such as high strength and toughness [MZB18] and damage confinement [SBC\*16].

TI assemblies typically consist of a single tileable element that can be repeatedly arranged in such a way that the whole structure can be held together by a fixed boundary, while elements are kept in place by mutual blocking. Kanel-Belov et al. [KBDE\*10] proposed a constructive approach to generate TI assemblies based on a tiling of the middle plane, where the single tileable block could be one of the five platonic solids; see Figure 21(b) for an example. Weizmann et al. [WAG17] extended this approach and explored different 2D tessellations (regular, semi-regular and non-regular tessellations) to discover new TI blocks for building floors. Rather than relying on 2D tessellations, other researchers [KAS\*21,AKF\*20] made a connection between planar assemblies and bi-axial weaving patterns. They generated tileable blocks (with curved faces) by Voronoi partitioning of space using curve segments whose arrangement follows the weaving patterns; see Figure 21(c).

The above mentioned tileable blocks are designed for constructing planar assemblies. Although they can be adapted to 3D free-form surfaces, these blocks will not have identical shape any more as their geometry has to be modified to follow the curvature of the surface [FBC19,WSIP19]. This limitation motivates research to design tileable blocks for building 3D shapes. Zhang et al. [ZB16] designed a set of nine tileable blocks with integral joints, called interlocking voxels. By using a hierarchical and layer-by-layer assembly order, these blocks can build up voxelization-like assemblies that are globally interlocking; see Figure 20(c&d). Other examples include SL blocks [Shi16], which consist of an S-shaped and an L-shaped tetracubes attaching to each other along sides, for building voxelized shapes, and tileable blocks for building variations of a given 3D architectural shape [KWS16].

Following this line of work, computational design of tileable blocks that can form a wide variety of 3D shapes is still an ongoing research topic in the graphics community.

## 7. High-level Analysis of Computational Design Methods

In Sections 3 to 6, we have reviewed computational methods to design assemblies with four objectives: *fabricability*, *stability*, *reconfigurability*, and *tilability*. To achieve these four objectives, the problem of designing assemblies can be formulated and addressed in various ways. We categorize these formulations into six classes (see also the leftmost column of Table 1):

- *Shape decomposition.* An assembly is generated by decomposing a target shape into parts. The decomposition task is most commonly addressed by one of five different approaches: top-down cut with 3D planes (*planar cut*), solving the multi-label graph cut problem (*graph cut*), merging elements like voxels, tetrahedrons, or over-segmented primitives (*merge elements*), solving the set cover problem (*set cover*), and embedding an object in the shape to guide the decomposition (*embedding*).
- *Shape approximation.* An assembly is generated by approximating a given 3D shape and taking each geometric primitive (e.g., edge, face, or slice) of the approximation as an individual part. In most cases, the whole (or *exterior*) shape is approximated to fabricate a shape abstraction. Sometimes, the approximation is required to be exactly within the given shape (*interior*), e.g., for being used as an internal support base in mixed fabrication.
- *Parts optimization.* The input is an assembly with initial parts and their geometric arrangement. A design objective, e.g., improving the assembly's structural stability, is achieved by optimizing the parts geometry.
- *Joint planning.* Given an initial assembly without joints, a design objective such as making the assembly interlocking is achieved by planing and constructing (typically predefined) joints among the parts. In some methods, the parts geometry and/or poses have to be slightly modified. For example, when designing foldable assemblies by inserting hinges, the parts geometry may need to be adapted for collision-free folding.
- *Shape co-decomposition.* Given multiple (typically two) target shapes, the goal is to generate a common set of parts that can reproduce these shapes by co-decomposition. According to the parts generation scheme, the co-decomposition process can be classified as *iterative* or *hierarchical*.
- *Layout optimization.* Given the geometry of parts, a design objective such as creating a stable LEGO assembly is achieved by optimizing the layout of these parts or their instances. In some cases, the parts geometry is optimized together with the layout, e.g., for high packing efficiency.

Table 1 provides an overview and classification of over 50 research papers on computational design of assemblies that have been discussed in this report. Since each paper typically has a concrete design objective and an explicit problem formulation, we organize the table in a way that each column corresponds to a design objective and each row corresponds to the formulation of a design problem. In particular, the columns (from left to right) are ordered following Sections 3-6 in this report, while the rows (from top to down) are ordered in a way that most non-empty items are on the diagonal of the table. A few papers may have multiple design objectives. For this case, Table 1 colors the paper's main objective

**Table 1:** Overview and classification of research works about computational design of assemblies discussed in this report.

|                        |                           | Fabricability<br>(Section 3)   |                              |  |   | Stability<br>(Section 4)                            |   | Reconfigurability<br>(Section 5)  |  | Tilability<br>(Section 6)  |                              |
|------------------------|---------------------------|--|------------------------------|--|---|---|---|---|--|--|------------------------------|
|                        |                           | 3D printing  | CNC milling                  | Laser cutting  | Mixed fabrication   | Equilibrium   | Interlocking  | Free reconfiguration  | Hinged reconfiguration   | LEGO bricks  | Tileable blocks              |
| Shape decomposition    | Planar cut                | Chopper [LBRM12]<br>Support-free [YYT*17]<br>[WQZ*18]<br>[KFW19]<br>Polycube map [FCM*18]                            | Polycube map [FCM*18]        |  |   | Puzzle [FMB15]                                      |   |   |  |  |                              |
|                        | Graph cut                 | Surface2Volume [ACA*19]  | Height-field shell [HMA15]   |  |   |   |   |   |  |  |                              |
|                        | Merge elements            | PackMerger [VGB*14b]<br>Shapes In a Box [Att15]<br>Interlock object [SFLF15]<br>Visual artifacts [WZK16]<br>[FAG*20] |                              |  |   |   | Puzzle [SFCO12]<br>Interlock object [SFLF15]<br>DESIA [WSP18]                                   |   |  |  |                              |
|                        | Set cover                 | Pyramidal shape [HLZCO14]  | Height-field volume [MLS*18] |  |   |   |   |   |  |  |                              |
|                        | Embedding                 |  |                              |  |   |   | Burr puzzle [XLF*11]  |   | Transformable [HCLC16]<br>[YZC18]                              |  | Space-filling tiles [KAS*21] |
| Shape approximation    | Exterior                  |  |                              | Slices [MSM11]<br>crdbrd [HBA12]<br>Shell assembly [CSaLM13]<br>[CS15]<br>Mesh-like [CPMS14]<br>[RA15] |   |   |   |   | Flat-foldable model [MEKM17]                                   |  |                              |
|                        | Interior                  |  |                              |  | RevoMaker [GZN*15]<br>CofiFab [SDW*16]<br>Universal blocks [CLF*18] |   |   |   |  |  |                              |
| Parts optimization     | /                         |  |                              |  |   | Masonry [WOD09]<br>[WSW*12]<br>TI assembly [WSIP19] |   | Interchangeable [DYY16]   |  |  |                              |
| Joint planning         | Without part modification |  |                              |  |   |   | Furniture [FSY*15]<br>CofiFab [SDW*16]<br>Reconfig. furniture [SFJ*17]<br>Shell pieces [YCXW17] |   |  |  |                              |
|                        | With part modification    |  |                              | Furniture-like [SP13]  |   |   |   |   | Boxelization [ZSMS14]<br>Foldable furniture [LHAZ15]<br>[IY18] |  |                              |
| Shape co-decomposition | Iterative                 |  |                              |  |   |   |   | Approximate dissection [DYTT17]<br>Reconfig. furniture [SFJ*17]<br>3D dissection [TSW*19] |  |  |                              |
|                        | Hierarchical              |  |                              |  |   |   |   | 2D dissection [ZW12]  |  |  |                              |
| Layout optimization    | Without part modification |  |                              |  |   |   |   |   |  | Sculpture [TSP13]<br>Legolization [LYH*15]<br>Silhouette-fitted [YPYM17]<br>Architecture [ZCX19] | Interlock voxels [ZB16]      |
|                        | With part modification    | Decompose and pack [YCL*15]  |                              | Zero-Waste [KHLM17]  |   |   |   |   | Convertible furniture [ZC18]                                   |  |                              |

in black as usual while the secondary objective in gray. For example, Song et al. [SFJ\*17] addressed a problem of designing reconfigurable interlocking furniture. Its main objective is to design reconfigurable furniture while its secondary objective is to make the furniture interlocking. Hence, this work is put under the columns *reconfigurability* → *free reconfiguration* and *stability* → *interlocking*, and colored in black and gray respectively.

Table 1 shows a correlation between the design objectives (columns) and problem formulations (rows); see the non-empty items on the diagonal of the table. We think that this correlation is because of the nature of the design objectives. Realizing this correlation could be helpful when one wants to find a proper formulation for achieving a desired design objective. For example, *fabricability* with *3D printing* is generally formulated as a *shape decomposition* problem while *fabricability* with *laser cutting* is typically addressed as a *shape approximation* problem. This is because 3D printing is a high-fidelity fabrication technique that can produce 3D parts with fine geometric details while laser cutting is a low-fidelity fabrication technique that produces only 2D planar parts. The other example is that designing *free reconfigurable* assemblies is typically formulated as a *shape co-decomposition* problem, because of multiple forms of reconfigurable assemblies, which are usually specified by users as a set of target shapes.

One can easily notice that there are non-empty items beyond the diagonal of the table, which has a two-fold meaning. On the one hand, a design objective can be addressed by formulating the design problem in different ways, typically by variations in defining the given input. An example is the design objective of *stability* → *interlocking*. To achieve this objective, the design problem can be formulated either as a *shape decomposition* problem when the input is a 3D target shape or as a *joint planning* problem when a set of initial parts without joints is given. On the other hand, a design strategy can be used to achieve design objectives that look very different, as illustrated in the row of *shape decomposition* → *embedding*. This decomposition strategy has been used for three different objectives (from left to right in the table): 1) design of interlocking puzzles by embedding an existing Burr configuration in a target 3D shape; 2) design of transformables by embedding a character skeleton in a target 3D shape; and 3) design of tileable blocks by embedding curves with bi-axial weaving patterns in the 3D space.

Some of the empty boxes in Table 1 provide interesting opportunities for future work. This can be achieved by combining design objectives and problem formulations in novel and reasonable ways. Taking designing assemblies in *equilibrium* as an example, existing works mainly formulate a *parts optimization* problem for applications in architectural design. However, depending on the given input, alternative formulations such as a *joint planning* problem or a *layout optimization* problem are conceivable. One potential application could be designing self-supporting puzzles, taking user-specified or -sketched initial assemblies as input. Note that not every combination between design objectives and problem formulations will make sense. For example, designing *structurally stable* assemblies is not suitable to be formulated as a *shape co-decomposition* problem since these assemblies typically have a single geometric form.

**Limitations.** Fabricability, stability, reconfigurability, and tilabil-

ity are the four major objectives, but not all considerations, for designing assemblies. There exist research works that address a specific design problem, without focusing on any of the four design objectives. For example, Koyam et al. [KSS\*15] designed 3D-printable customized connectors for joining two physical objects together, while Sun et al. [SZ15] created complex twisty puzzles inspired by the Rubik's Cube mechanism. These works are typically devoted to a specific application such as puzzles [LFL09], work-like prototypes [KLY\*14], furniture [LOMI11, UIM12, SLR\*16], and architecture [SFG\*13]. This report does not discuss these works to maintain a coherent structure.

## 8. Discussion

The state of the art in computational analysis and design of assemblies has evolved rapidly over the last few years. Various computational methods and tools have been developed for designing different kinds of assemblies. In the future, more efforts are required to make these methods and tools accessible to casual and professional users in a way that they can design assemblies conveniently and even collaboratively, e.g., by using cloud-based design paradigm [WRWS15]. Motivated by the advance in technologies of digital fabrication, automatic construction, and material science, future research effort could be devoted to designing and fabricating advanced assemblies that are not accessible before such as architectural structures that can be constructed automatically by using robotic assembly [EGK17], modular robots that can be reconfigured autonomously or manually to form different assemblies [BRS\*17], and self-assembling systems based on shape memory material [SOL\*21]. To design these advanced assemblies, significant challenges remain in this field and should be addressed in future research:

**Analysis with high prediction accuracy.** To simplify computational analysis of assemblies, a common assumption is that the fabrication material is rigid and manufacturing precision is perfect. Yet in practice, materials are not infinitely rigid and fabrication devices have limited manufacturing precision. As a consequence, structural issues can arise when the relative motion of two parts is restricted by a small contact or a thin joint. Internal stress concentrations at the contact or joint can lead to material failure. Similarly, a geometrically stable design, e.g., a furniture assembly, could become instable when fabricated, due to gaps among the parts caused by machining tolerances. To improve prediction accuracy, one possible way is to take these imperfections into consideration in the analysis methods. For example, tolerance analysis [CJLL14] can provide insights into the structural performance of an assembly. Such analysis tasks are particularly challenging as imperfections of each individual part will propagate within the whole assembly.

Another challenge is functionality analysis of assemblies, which is critical for designing assemblies that function as expected by the user. Functionality analysis of given shapes is a relatively new topic in computer graphics [HSvK18], and developing methods for analyzing functionalities of assemblies could be a promising research direction.

**Design for multiple objectives.** This report has discussed a number of objectives for designing assemblies. To date, however, most

publications address and solve a specific objective and do not consider all aspects relevant for an assembly. For example, for large-scale assemblies in architecture, combining the objectives of parts fabricability, structural stability of the final and intermediate assemblies, and automated assembly process planning, would provide a more complete framework for design. Future research effort could address such design problems with multiple objectives, which might require a fundamentally new formulation of the problem, a combination of different representations of assemblies, and new strategies to search over a rather constrained solution space. Another challenge is to find a good trade-off among multiple objectives to satisfy users' high-level preference.

**Advanced design methods** can be studied by exploring several aspects of assemblies in a novel way, in particular, joints and assembly plans. First, existing design methods connect parts by using either integral joints that allow a single part removable direction or planar contacts that have limited capacity to restrict part motion. Studying integral joints in-between these two extremes, e.g., curved-contact joints [KAS\*21] (see also Figure 1(b)) or puzzle-like joints [LYUI20], as well as how they facilitate design of assemblies, would be an interesting topic for future exploration. Second, designing assemblies with (dis)assembly plans that are beyond sequential, monotone, and linear is another promising research topic, as these complex plans are possible to be precisely realized by robots nowadays. Such (dis)assembly plans could be useful in terms of entertaining people (e.g., disassembly puzzles) or enhancing structural stability (e.g., complex coordinated motion to take out parts).

**Machine learning for searching the design space.** Machine learning methods offer potential benefits for solving non-linear and non-convex problems. In recent years, tremendous research efforts have been devoted to solve traditional graphics problems such as shape synthesis and mesh segmentation using machine learning methods [MKG\*19]. However, developing learning methods to solve design and fabrication problems is challenging due to various hard constraints (e.g., fabricability) and/or global constraints (e.g., assemblability) involved in these problems. Another challenge is the lack of datasets to train the learning algorithms, as the design problem could be too specific or there is no existing way to generate desired designs. One possible way to address this challenge is to use reinforcement learning, which has been shown to be good at adaptively sampling the search space. We see a high potential for reinforcement learning to help solve challenging problems of designing assemblies.

## Biographies

**Ziqi Wang** is a Ph.D. candidate at the School of Computer and Communication Sciences at EPFL. He received his bachelor degree in Mathematics in 2017 from University of Science and Technology of China. His research interests focus on geometry processing, architecture geometry, and digital fabrication.

**Peng Song** is an Assistant Professor at the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design (SUTD), where he directs the Computer Graphics

Laboratory (CGL). Prior to joining SUTD, he was a research scientist at EPFL, Switzerland. He received his PhD from Nanyang Technological University, Singapore in 2013, his master and bachelor degrees both from Harbin Institute of Technology, China in 2010 and 2007 respectively. His research is in the area of computer graphics, with a focus on computational fabrication and geometry processing.

**Mark Pauly** is a Full Professor at the School of Computer and Communication Sciences at EPFL, where he directs the Geometric Computing Laboratory (GCM). He previously held academic positions at ETH Zurich and at Stanford University. His research interests include geometric computing, computational and architectural design, and digital fabrication. He received the ETH dissertation award in 2003, the Eurographics Young Researcher Award in 2006, an ERC grant in 2010, and the Eurographics Outstanding Technical Achievement Award in 2016. He has co-founded two EPFL spin-off companies, Faceshift AG in 2012 and Rayform AG in 2016.

**Acknowledgements.** This work was supported by the Swiss National Science Foundation (NCCR Digital Fabrication Agreement #51NF40-141853) and the SUTD Start-up Research Grant (Award Number: SRG ISTD 2019 148).

## References

- [AAC\*12] ABBOTT T. G., ABEL Z., CHARLTON D., DEMAINE E. D., DEMAINE M. L., KOMINERS S. D.: Hinged dissections exist. *Discrete & Comp. Geom.* 47, 1 (2012), 150–186. 16
- [ACA\*19] ARAÚJO C., CABIDDU D., ATTENE M., LIVESU M., VINING N., SHEFFER A.: Surface2Volume: Surface segmentation conforming assembleable volumetric partition. *ACM Trans. on Graph. (SIGGRAPH)* 38, 4 (2019), 1:1–1:16. 10, 19
- [ACP\*14] ALEMANNO G., CIGNONI P., PIETRONI N., PONCHIO F., SCOPIGNO R.: Pieces for printing tangible cultural heritage replicas. In *Proc. Eurographics Workshops on Graphics and Cultural Heritage* (2014), pp. 145–154. 11
- [AKF\*20] AKLEMAN E., KRISHNAMURTHY V. R., FU C.-A., SUBRAMANIAN S. G., EBERT M., ENG M., STARRETT C., PANCHAL H.: Generalized Abeille Tiles: Topologically interlocked space-filling shapes generated based on fabric symmetries. *Comp. & Graph. (SMI)* 89 (2020), 156–166. 18
- [ALL\*18] ATTENE M., LIVESU M., LEFEBVRE S., FUNKHOUSER T., RUSINKIEWICZ S., ELLERO S., MARTÍNEZ J., BERMANO A. H.: *Design, Representations, and Processing for Additive Manufacturing*. Morgan & Claypool Publishers, 2018. 2
- [APH\*03] AGRAWALA M., PHAN D., HEISER J., HAYMAKER J., KLINGNER J., HANRAHAN P., TVERSKY B.: Designing effective step-by-step assembly instructions. *ACM Trans. on Graph. (SIGGRAPH)* 22, 3 (2003), 828–837. 4
- [Att15] ATTENE M.: Shapes in a box: Disassembling 3D objects for efficient packing and fabrication. *Comp. Graph. Forum* 34, 8 (2015), 64–76. 7, 9, 19
- [BCMP18] BICKEL B., CIGNONI P., MALOMO L., PIETRONI N.: State of the art on stylized fabrication. *Comp. Graph. Forum* 37, 6 (2018), 325–342. 2
- [BDD\*15] BOSBOOM J., DEMAINE E. D., DEMAINE M. L., LYNCH J., MANURANGSI P., RUDROY M., YODPINYANEE A.: k-piece dissection is np-hard. In *Proc. 18th Japan Conference on Discrete and Computational Geometry and Graphs* (2015). 15

- [BFR17] BERMANO A. H., FUNKHOUSER T., RUSINKIEWICZ S.: State of the art in methods and representations for fabrication-aware design. *Comp. Graph. Forum (Eurographics STAR – State of The Art Report)* 36, 2 (2017), 509–535. 2
- [BGM\*15] BEYER D., GUREVICH S., MUELLER S., CHEN H.-T., BAUDISCH P.: Platener: Low-fidelity fabrication of 3D objects by substituting 3D print with laser-cut plates. In *Proc. ACM CHI* (2015), pp. 1799–1806. 12
- [BM17] BAUDISCH P., MUELLER S.: Personal fabrication. *Foundations and Trends® in Human-Computer Interaction* 10, 3-4 (2017), 165–293. 2, 12
- [BO07] BLOCK P., OCHSENDORF J.: Thrust Network Analysis: A new methodology for three-dimensional equilibrium. *Journal of the International Association for Shell and Spatial Structures* 48, 3 (2007), 167 – 173. 14
- [BR\*17] BRUNETE A., RANGANATH A., SEGOVIA S., DE FRUTOS J. P., HERNANDO M., GAMBAO E.: Current trends in reconfigurable modular robots design. *International Journal of Advanced Robotic Systems* (2017), 1–21. 20
- [BSK\*19] BAUDISCH P., SILBER A., KOMMANA Y., GRUNER M., WALL L., REUSS K., HEILMAN L., KOVACS R., RECHLITZ D., ROUMEN T.: Kyub: A 3D editor for modeling sturdy laser-cut objects. In *Proc. ACM CHI* (2019), pp. 566:1–566:12. 12
- [CJLL14] CHEN H., JIN S., LI Z., LAI X.: A comprehensive study of three dimensional tolerance analysis methods. *Computer-Aided Design* 53 (2014), 1–13. 20
- [CKPT17] CHRISTENSEN H. I., KHAN A., POKUTTA S., TETALI P.: Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review* 24 (2017), 63–79. 6
- [CLF\*18] CHEN X., LI H., FU C.-W., ZHANG H., COHEN-OR D., CHEN B.: 3D fabrication with universal building blocks and pyramidal shells. *ACM Trans. on Graph. (SIGGRAPH Asia)* 37, 6 (2018), 189:1–189:15. 13, 19
- [CML\*17] CHEN W., MA Y., LEFEBVRE S., XIN S., MARTÍNEZ J., WANG W.: Fabricable tile decors. *ACM Trans. on Graph. (SIGGRAPH Asia)* 36, 6 (2017), 175:1–175:15. 6
- [CPMS14] CIGNONI P., PIETRONI N., MALOMO L., SCOPIGNO R.: Field-aligned mesh joinery. *ACM Trans. on Graph.* 33, 1 (2014), 11:1–11:12. 11, 12, 13, 19
- [CS15] CHEN L., SASS L.: Fresh Press Modeler: A generative system for physically based low fidelity prototyping. *Comp. & Graph.* 54 (2015), 157–165. 12, 19
- [CSaLM13] CHEN D., SITTHI-AMORN P., LAN J. T., MATUSIK W.: Computing and fabricating multiplanar models. *Comp. Graph. Forum (Eurographics)* 32, 2 (2013), 305–315. 11, 12, 19
- [Cut78] CUTLER W. H.: The six-piece burt. *Journal of Recreational Mathematics* 10, 4 (1978), 241–250. 14
- [CZL\*15] CHEN X., ZHANG H., LIN J., HU R., LU L., HUANG Q., BENES B., COHEN-OR D., CHEN B.: Dapper: Decompose-and-pack for 3D printing. *ACM Trans. on Graph. (SIGGRAPH Asia)* 34, 6 (2015), 213:1–213:12. 6, 7, 9
- [CZS\*19] CHIDAMBARAM S., ZHANG Y., SUNDARARAJAN V., ELMQVIST N., RAMANI K.: Shape structuralizer: Design, fabrication, and user-driven iterative refinement of 3D mesh models. In *Proc. ACM CHI* (2019), pp. 663:1–663:12. 13
- [DEKBP03] DYSKIN A. V., ESTRIN Y., KANEL-BELOV A. J., PASTERNAK E.: Topological interlocking of platonic solids: A way to new materials and structures. *Philosophical Magazine Letters* 83, 3 (2003), 197–203. 18
- [DMC18] DESAI R., MCCANN J., COROS S.: Assembly-aware design of printable electromechanical devices. In *Proc. ACM UIST* (2018), pp. 457–472. 5
- [dMS90] DE MELLO L. S. H., SANDERSON A. C.: AND/OR graph representation of assembly plans. *IEEE Transactions on Robotics and Automation* 6, 2 (1990), 188–199. 4
- [DPW\*14] DEUSS M., PANOZZO D., WHITING E., LIU Y., BLOCK P., SORKINE-HORNUNG O., PAULY M.: Assembling self-supporting structures. *ACM Trans. on Graph. (SIGGRAPH Asia)* 33, 6 (2014), 214:1–214:10. 3, 4
- [DYY16] DUNCAN N., YU L.-F., YEUNG S.-K.: Interchangeable components for hands-on assembly based modelling. *ACM Trans. on Graph. (SIGGRAPH Asia)* 35, 6 (2016), 234:1–234:14. 15, 16, 19
- [DYTT17] DUNCAN N., YU L.-F., YEUNG S.-K., TERZOPOULOS D.: Approximate dissections. *ACM Trans. on Graph. (SIGGRAPH Asia)* 36, 6 (2017), 182:1–182:14. 15, 19
- [EGK17] EVERSMAANN P., GRAMAZIO F., KOHLER M.: Robotic pre-fabrication of timber structures: Towards automated large-scale spatial assembly. *Construction Robotics I* (2017), 49–60. 20
- [eSREPC16] E SÁ A. M., RODRIGUEZ-ECHAVARRIA K., PIETRONI N., CIGNONI P.: State of the art on functional fabrication. In *Eurographics Workshop on Graphics for Digital Fabrication* (2016), pp. 1–9. 2
- [FAG\*20] FILOSCIA I., ALDERIGHI T., GIORGI D., MALOMO L., CALLIERI M., CIGNONI P.: Optimizing object decomposition to reduce visual artifacts in 3D printing. *Comp. Graph. Forum (Eurographics)* 39, 2 (2020), 423–434. 9, 10, 19
- [Fai13] FAIRHAM W.: *Woodwork Joints: How to Make and Where to Use Them*. Skyhorse, 2013. 2
- [FBC19] FALLACARA G., BARBERIO M., COLELLA M.: Topological interlocking blocks for architecture: From flat to curved morphologies. In *Architected Materials in Nature and Engineering*, Estrin Y., Bréchet Y., Dunlop J., Fratzl P., (Eds.). Springer International Publishing, 2019, ch. 14, pp. 423–445. 18
- [FCM\*18] FANNI F. A., CHERCHI G., MUNTONI A., TOLA A., SCATENI R.: Fabrication oriented shape decomposition using polycube mapping. *Comp. & Graph.* 77 (2018), 183–193. 11, 19
- [FKS\*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM Trans. on Graph. (SIGGRAPH)* 23, 3 (2004), 652–663. 16
- [FM01] FEKETE S. P., MITCHELL J. S. B.: Terrain decomposition and layered manufacturing. *International Journal of Computational Geometry & Applications* 11, 6 (2001), 647–668. 11
- [FMB15] FRICK U., MELE T. V., BLOCK P.: Decomposing three-dimensional shapes into self-supporting, discrete-element assemblies. In *Proc. the Design Modelling Symposium* (2015), pp. 187–201. 13, 14, 19
- [Fre97] FREDERICKSON G. N.: *Dissections: Plane and Fancy*. Cambridge University Press, 1997. 15
- [FSY\*15] FU C.-W., SONG P., YAN X., YANG L. W., JAYARAMAN P. K., COHEN-OR D.: Computational interlocking furniture assembly. *ACM Trans. on Graph. (SIGGRAPH)* 34, 4 (2015), 91:1–91:11. 3, 14, 19
- [GHP98] GOWER R. A. H., HEYDTMANN A. E., PETERSEN H. G.: LEGO: Automated model construction. In *Proc. 32nd European Study Group with Industry* (1998), pp. 81–94. 17
- [GJG16] GARG A., JACOBSON A., GRINSPUN E.: Computational design of reconfigurables. *ACM Trans. on Graph. (SIGGRAPH)* 35, 4 (2016), 90:1–90:14. 15
- [GJL20] GAO S., JIN R., LU W.: Design for manufacture and assembly in construction: a review. *Building Research & Information* 48, 5 (2020), 538–550. 2
- [GM15] GHANDI S., MASEHIAN E.: Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design* 67-68 (2015), 58–86. 3, 4

- [GWYN19] GAO Y., WU L., YAN D.-M., NAN L.: Near support-free multi-directional 3D printing via global-optimal decomposition. *Graphical Models (CVM) 104* (2019), Article No. 101034. [9](#)
- [GZN\*15] GAO W., ZHANG Y., NAZZETTA D. C., RAMANI K., CIPRA R. J.: RevoMaker: Enabling multi-directional and functionally-embedded 3D printing using a rotational cuboidal platform. In *Proc. ACM UIST* (2015), pp. 437–446. [13](#), [19](#)
- [HBA12] HILDEBRAND K., BICKEL B., ALEXA M.: crdbrd: Shape fabrication by sliding planar slices. *Comp. Graph. Forum (Eurographics) 31*, 2 (2012), 583–592. [11](#), [12](#), [19](#)
- [HBA13] HILDEBRAND K., BICKEL B., ALEXA M.: Orthogonal slicing for additive manufacturing. *Comp. & Graph. (SMI) 37*, 6 (2013), 669–675. [10](#)
- [HCLC16] HUANG Y.-J., CHAN S.-Y., LIN W.-C., CHUANG S.-Y.: Making and animating transformable 3D models. *Comp. & Graph. (CAD/Graphics) 54* (2016), 127–134. [16](#), [19](#)
- [Hey66] HEYMAN J.: The stone skeleton. *International Journal of Solids and Structures* 2, 2 (1966), 249–279. [14](#)
- [HLW00] HALPERIN D., LATOMBE J.-C., WILSON R. H.: A general framework for assembly planning: The motion space approach. *Algorithmica* 26, 3–4 (2000), 577–601. [3](#)
- [HLZCO14] HU R., LI H., ZHANG H., COHEN-OR D.: Approximate pyramidal shape decomposition. *ACM Trans. on Graph. (SIGGRAPH Asia) 33*, 6 (2014), 213:1–213:12. [9](#), [19](#)
- [HMA15] HERHOLZ P., MATUSIK W., ALEXA M.: Approximating free-form geometry with height fields for manufacturing. *Comp. Graph. Forum (Eurographics) 34*, 2 (2015), 239–251. [11](#), [19](#)
- [HPA\*04] HEISER J., PHAN D., AGRAWALA M., TVERSKY B., HANRAHAN P.: Identification and validation of cognitive design principles for automated generation of assembly instructions. In *Proc. the Working Conference on Advanced Visual Interfaces* (2004), pp. 311–319. [4](#)
- [HSvK18] HU R., SAVVA M., VAN KAICK O.: Functionality representations and applications for shape analysis. *Comp. Graph. Forum (Eurographics STAR – State of The Art Report) 37*, 2 (2018), 603–624. [20](#)
- [IY18] IBRAHIM M., YAN D.: Fold and Fit: Space conserving shape editing. *Comp. & Graph. (CAD/Graphics) 70* (2018), 316–326. [17](#), [19](#)
- [Jac19] JACOBSON A.: RodSteward: A design-to-assembly system for fabrication using 3d-printed joints and precision-cut rods. *Comp. Graph. Forum (Pacific Graphics) 38*, 7 (2019), 765–774. [13](#)
- [Jim13] JIMÉNEZ P.: Survey on assembly sequencing: A combinatorial and geometrical perspective. *Journal of Intelligent Manufacturing* 24, 2 (2013), 235–250. [3](#)
- [JW96] JONES R. E., WILSON R. H.: A survey of constraints in automated assembly planning. In *Proc. IEEE Int. Conf. on Robotics and Automation* (1996), pp. 1525–1532. [4](#)
- [KAS\*21] KRISHNAMURTHY V. R., AKLEMAN E., SUBRAMANIAN S. G., EBERT M., CUI J., AN FU C., STARRETT C.: Geometrically interlocking space-filling tiling based on fabric weaves. *IEEE Trans. Vis. & Comp. Graphics* (2021). DOI: 10.1109/TVCG.2021.3065457. [2](#), [18](#), [21](#)
- [KBDE\*10] KANEL-BELOV A. J., DYSKIN A. V., ESTRIN Y., PASTERNAK E., IVANOV-POGODAEV I. A.: Interlocking of convex polyhedra: Towards a geometric theory of fragmented solids. *Moscow Mathematical Journal* 10, 2 (2010), 337–342. [18](#)
- [KFW19] KARASIK E., FATTAL R., WERMAN M.: Object partitioning for support-free 3D-printing. *Comp. Graph. Forum (Eurographics) 38*, 2 (2019), 305–316. [9](#), [19](#)
- [KHLM17] KOO B., HERGEL J., LEFEBVRE S., MITRA N. J.: Towards zero-waste furniture design. *IEEE Trans. Vis. & Comp. Graphics* 23, 12 (2017), 2627–2640. [6](#), [7](#), [12](#), [19](#)
- [KKL14] KIM J. W., KANG K. K., LEE J. H.: Survey on automated lego assembly construction. In *Proc. WSCG 2014 Conference on Computer Graphics, Visualization and Computer Vision* (2014), pp. 89–96. [17](#)
- [KKS\*17] KAO G. T., KÖRNER A., SONNTAG D., NGUYEN L., MENGES A., KNIPPERS J.: Assembly-aware design of masonry shell structures: A computational approach. In *Proceedings of the International Association for Shell and Spatial Structures Symposium* (2017). [5](#)
- [KKSS15] KERBL B., KALKOFEN D., STEINBERGER M., SCHMALSTIEG D.: Interactive disassembly planning for complex objects. *Comp. Graph. Forum (Eurographics) 34*, 2 (2015), 287–297. [4](#)
- [KLY\*14] KOO B., LI W., YAO J., AGRAWALA M., MITRA N. J.: Creating works-like prototypes of mechanical objects. *ACM Trans. on Graph. (SIGGRAPH Asia) 33*, 6 (2014), 217:1–217:9. [20](#)
- [KSJP08] KAUFMAN D. M., SUEDA S., JAMES D. L., PAI D. K.: Staggered projections for frictional contact in multibody systems. *ACM Trans. on Graph. (SIGGRAPH Asia) 27*, 5 (2008), 164:1–164:11. [6](#)
- [KSS\*15] KOYAMA Y., SUEDA S., STEINHARDT E., IGARASHI T., SHAMIR A., MATUSIK W.: AutoConnect: Computational design of 3D-printable connectors. *ACM Trans. on Graph. (SIGGRAPH Asia) 34*, 6 (2015), 231:1–231:11. [20](#)
- [KSW\*17] KOVACS R., SEUFERT A., WALL L., CHEN H.-T., MEINEL F., MÜLLER W., YOU S., BREHM M., STRIEBEL J., KOMMANA Y., POPIAK A., BLÄSIUS T., BAUDISCH P.: TrussFab: Fabricating sturdy large-scale structures on desktop 3D printers. In *Proc. ACM CHI* (2017), pp. 2606–2616. [12](#)
- [KWS16] KALOJANOV J., WAND M., SLUSALLEK P.: Building construction sets by tiling grammar simplification. *Comp. Graph. Forum (Eurographics) 35*, 2 (2016), 13–25. [18](#)
- [LBRM12] LUO L., BARAN I., RUSINKIEWICZ S., MATUSIK W.: Chopper: Partitioning models into 3D-printable parts. *ACM Trans. on Graph. (SIGGRAPH Asia) 31*, 6 (2012), 129:1–129:9. [8](#), [10](#), [19](#)
- [LEM\*17] LIVESU M., ELLERO S., MARTÍNEZ J., LEFEBVRE S., ATTENE M.: From 3D models to 3D prints: an overview of the processing pipeline. *Comp. Graph. Forum (Eurographics STAR – State of The Art Report) 36*, 2 (2017), 537–564. [8](#), [10](#)
- [LFL09] LO K.-Y., FU C.-W., LI H.: 3D polyomino puzzle. *ACM Trans. on Graph. (SIGGRAPH Asia) 28*, 5 (2009), 157:1–157:8. [20](#)
- [LFY\*19] LIU H.-Y., FU X.-M., YE C., CHAI S., LIU L.: Atlas refinement with bounded packing efficiency. *ACM Trans. on Graph. (SIGGRAPH) 38*, 4 (2019), 33:1–33:13. [6](#)
- [LFZ18] LIRA W., FU C.-W., ZHANG H.: Fabricable eulerian wires for 3D shape abstraction. *ACM Trans. on Graph. (SIGGRAPH Asia) 37*, 6 (2018), 240:1–240:13. [13](#)
- [LHAZ15] LI H., HU R., ALHASHIM I., ZHANG H.: Foldabilizing furniture. *ACM Trans. on Graph. (SIGGRAPH) 34*, 4 (2015), 90:1–90:12. [16](#), [17](#), [19](#)
- [LMaH\*18] LI S., MAHDAVI-AMIRI A., HU R., LIU H., ZOU C., KAICK O. V., LIU X., HUANG H., ZHANG H.: Construction and fabrication of reversible shape transforms. *ACM Trans. on Graph. (SIGGRAPH Asia) 37*, 6 (2018), 190:1–190:14. [16](#)
- [LMM02] LODI A., MARTELLO S., MONACI M.: Two-dimensional packing problems: A survey. *European Journal of Operational Research* 141, 2 (2002), 241–252. [6](#)
- [LOMI11] LAU M., OHGAWARA A., MITANI J., IGARASHI T.: Converting 3D furniture models to fabricatable parts and connectors. *ACM Trans. on Graph. (SIGGRAPH) 30*, 4 (2011), 85:1–85:6. [20](#)
- [LTO\*20] LEO A. A. S., TOLEDO F. M. B., OLIVEIRA J. F., CARAVILLA M. A., ALVAREZ-VALDÉS R.: Irregular packing problems: A review of mathematical models. *European Journal of Operational Research* 282, 3 (2020), 803–822. [6](#)
- [LVW\*15] LIU H., VIMONT U., WAND M., CANI M.-P., HAHMANN S., ROHMER D., MITRA N. J.: Replaceable substructures for efficient part-based modeling. *Comp. Graph. Forum (Eurographics) 34*, 2 (2015), 503–513. [16](#)

- [LXG10] LASEMI A., XUE D., GU P.: Recent development in CNC machining of freeform surfaces: A state-of-the-art review. *Computer-Aided Design* 42 (2010), 641–654. 13
- [LYH\*15] LUO S.-J., YUE Y., HUANG C.-K., CHUNG Y.-H., IMAI S., NISHITA T., CHEN B.-Y.: Legolization: Optimizing LEGO designs. *ACM Trans. on Graph. (SIGGRAPH Asia)* 34, 6 (2015), 222:1–222:12. 17, 19
- [LYUI20] LARSSON M., YOSHIDA H., UMETANI N., IGARASHI T.: Tsugite: Interactive design and fabrication of wood joints. In *Proc. ACM UIST* (2020), pp. 317–327. 2, 21
- [MCHW18] MA Y., CHEN Z., HU W., WANG W.: Packing irregular objects in 3D space via hybrid optimization. *Comp. Graph. Forum (SGP)* 37, 5 (2018), 49–59. 7
- [MEKM17] MIYAMOTO E., ENDO Y., KANAMORI Y., MITANI J.: Semi-automatic conversion of 3D shape into flat-foldable polygonal model. *Comp. Graph. Forum (Pacific Graphics)* 36, 7 (2017), 41–50. 16, 17, 19
- [MG20] MASEHIAN E., GHANDI S.: ASPPR: A new assembly sequence and path planner/replanner for monotone and nonmonotone assembly planning. *Computer-Aided Design* 123 (2020), 102828:1–102828:22. 4
- [MHS\*19] MA L., HE Y., SUN Q., ZHOU Y., ZHANG C., WANG W.: Constructing 3D self-supporting surfaces with isotropic stress using 4D minimal hypersurfaces of revolution. *ACM Trans. on Graph.* 38, 5 (2019), 144:1–144:13. 14
- [MKG\*19] MITRA N. J., KOKKINOS I., GUERRERO P., THUREY N., KIM V., GUIBAS L.: CreativeAI: Deep learning for graphics. In *SIGGRAPH Courses* (2019). 21
- [MLS\*18] MUNTONI A., LIVESU M., SCATENI R., SHEFFER A., PANOZZO D.: Axis-aligned height-field block decomposition of 3D shapes. *ACM Trans. on Graph.* 37, 5 (2018), 169:1–169:15. 11, 19
- [MMZ18] MAGRISIO S., MIZRAHI M., ZORAN A.: Digital joinery for hybrid carpentry. In *Proc. ACM CHI* (2018), pp. 167:1–167:11. 2
- [MSM11] MCCRAE J., SINGH K., MITRA N. J.: Slices: A shape-proxy based on planar sections. *ACM Trans. on Graph. (SIGGRAPH Asia)* 30, 6 (2011), 168:1–168:12. 12, 19
- [MSY\*14] MELLADO N., SONG P., YAN X., FU C.-W., MITRA N. J.: Computational design and construction of notch-free reciprocal frame structures. In *Proc. Advances in Architectural Geometry* (2014), pp. 181–197. 4
- [MUS14] MCCRAE J., UMETANI N., SINGH K.: FlatFitFab: Interactive modeling with planar sections. In *Proc. ACM UIST* (2014), pp. 13–22. 12
- [MZB18] MIRKHALAF M., ZHOU T., BARTHELAT F.: Simultaneous improvements of strength and toughness in topologically interlocked ceramics. *Proceedings of the National Academy of Sciences of the United States of America* 115, 37 (2018), 9128–9133. 18
- [NAI\*18] NAKASHIMA K., AUZINGER T., IARUSSI E., ZHANG R., IGARASHI T., BICKEL B.: CoreCavity: Interactive shell decomposition for fabrication with two-piece rigid molds. *ACM Trans. on Graph. (SIGGRAPH)* 37, 4 (2018), 135:1–135:13. 13
- [NBA19] NISHIDA G., BOUSSEAU A., ALIAGA D. G.: Multi-pose interactive linkage design. *Comp. Graph. Forum (Eurographics)* 38, 2 (2019), 277–289. 16
- [Och02] OCHSENDORF J. A.: *Collapse of Masonry Structures*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 6 2002. 6
- [OZB18] OH Y., ZHOU C., BEHDAD S.: Part decomposition and assembly-based (re) design for additive manufacturing: A review. *Additive Manufacturing* 22 (2018), 230–242. 2
- [PBSH13] PANOZZO D., BLOCK P., SORKINE-HORNUNG O.: Designing unreinforced masonry models. *ACM Trans. on Graph. (SIGGRAPH)* 32, 4 (2013), 91:1–91:11. 14
- [RA15] RICHTER R., ALEXA M.: Beam meshes. *Comp. & Graph.* 53 (2015), 28–36. 11, 12, 19
- [RRS13] REINERT B., RITSCHEL T., SEIDEL H.-P.: Interactive by-example design of artistic packing layouts. *ACM Trans. on Graph. (SIGGRAPH Asia)* 32, 6 (2013), 218:1–218:7. 6
- [SBC\*16] SIEGMUND T., BARTHELAT F., CIPRA R., HABTOUR E., RIDDICK J.: Manufacture and mechanics of topologically interlocked material assemblies. *Applied Mechanics Reviews* 68, 4 (2016). Article No. 040803. 18
- [SCP\*17] SCOPIGNO R., CIGNONI P., PIETRONI N., CALLIERI M., DELLEPIANE M.: Digital fabrication techniques for cultural heritage: A survey. *Comp. Graph. Forum* 36, 1 (2017), 6–21. 2
- [SDW\*16] SONG P., DENG B., WANG Z., DONG Z., LI W., FU C.-W., LIU L.: CofiFab: Coarse-to-fine fabrication of large 3D objects. *ACM Trans. on Graph. (SIGGRAPH)* 35, 4 (2016), 45:1–45:11. 13, 14, 19
- [SFCO12] SONG P., FU C.-W., COHEN-OR D.: Recursive interlocking puzzles. *ACM Trans. on Graph. (SIGGRAPH Asia)* 31, 6 (2012), 128:1–128:10. 5, 14, 19
- [SFG\*13] SONG P., FU C.-W., GOSWAMI P., ZHENG J., MITRA N. J., COHEN-OR D.: Reciprocal frame structures made easy. *ACM Trans. on Graph. (SIGGRAPH)* 32, 4 (2013), 94:1–94:10. 20
- [SFJ\*17] SONG P., FU C.-W., JIN Y., XU H., LIU L., HENG P.-A., COHEN-OR D.: Reconfigurable interlocking furniture. *ACM Trans. on Graph. (SIGGRAPH Asia)* 36, 6 (2017), 174:1–174:14. 14, 15, 16, 19, 20
- [SFLF15] SONG P., FU Z., LIU L., FU C.-W.: Printing 3D objects with interlocking parts. *Comp. Aided Geom. Des. (GMP)* 35-36 (2015), 137–148. 14, 19
- [Shi16] SHIH S.-G.: On the hierarchical construction of SL blocks. In *Proc. Advances in Architectural Geometry* (2016), pp. 124–136. 18
- [SHWP09] SCHIFTNER A., HÖBINGER M., WALLNER J., POTTMANN H.: Packing circles and spheres on surfaces. *ACM Trans. on Graph. (SIGGRAPH Asia)* 28, 5 (2009), 139:1–139:8. 6
- [SLR\*16] SHAO T., LI D., RONG Y., ZHENG C., ZHOU K.: Dynamic furniture modeling through assembly instructions. *ACM Trans. on Graph. (SIGGRAPH Asia)* 35, 6 (2016), 172:1–172:15. 20
- [SOL\*21] SUN Y., OUYANG W., LIU Z., NI N., SAVOYE Y., SONG P., LIU L.: Computational design of self-actuated deformable solids via shape memory material. *IEEE Trans. Vis. & Comp. Graphics* (2021). DOI: 10.1109/TVCG.2020.3039613. 20
- [SP13] SCHWARTZBURG Y., PAULY M.: Fabrication-aware design with intersecting planar pieces. *Comp. Graph. Forum (Eurographics)* 32, 2 (2013), 317–326. 12, 13, 19
- [SPV\*16] SHIN H. V., PORST C. F., VOUGA E., OCHSENDORF J., DURAND F.: Reconciling elastic and equilibrium methods for static analysis. *ACM Trans. on Graph.* 35, 2 (2016), 13:1–13:16. 5, 6
- [SZ15] SUN T., ZHENG C.: Computational design of twisty joints and puzzles. *ACM Trans. on Graph. (SIGGRAPH)* 34, 4 (2015), 101:1–101:11. 20
- [Tar72] TARJAN R.: Depth-first search and linear graph algorithms. *SIAM Journal on Computing* 1, 2 (1972), 146–160. 4
- [TPCS16] TONELLI D., PIETRONI N., CIGNONI P., SCOPIGNO R.: Design and fabrication of grid-shells mockups. In *STAG: Smart Tools and Apps in computer Graphics* (2016), pp. 21–27. 13
- [TSP13] TESTUZ R., SCHWARTZBURG Y., PAULY M.: Automatic generation of constructable brick sculptures. In *Eurographics* (2013). short paper. 17, 19
- [TSW\*19] TANG K., SONG P., WANG X., DENG B., FU C.-W., LIU L.: Computational design of steady 3D dissection puzzles. *Comp. Graph. Forum (Eurographics)* 38, 2 (2019), 291–303. 14, 15, 19
- [UIM12] UMETANI N., IGARASHI T., MITRA N. J.: Guided exploration of physically valid shapes for furniture design. *ACM Trans. on Graph. (SIGGRAPH)* 31, 4 (2012), 86:1–86:11. 20

- [UTZ16] URETA F. G., TYMMS C., ZORIN D.: Interactive modeling of mechanical objects. *Comp. Graph. Forum (SGP)* 35, 5 (2016), 145–155. 16
- [VGB14a] VANEK J., GALICIA J. A. G., BENES B.: Clever Support: Efficient support structure generation for digital fabrication. *Comp. Graph. Forum (SGP)* 33, 5 (2014), 117–125. 8
- [VGB\*14b] VANEK J., GALICIA J. A. G., BENES B., MĚCH R., CARR N., STAVA O., MILLER G. S.: PackMerger: A 3D print volume optimizer. *Comp. Graph. Forum* 33, 6 (2014), 322–332. 6, 7, 9, 19
- [VTSOP20] VASQUEZ J., TWIGG-SMITH H., O'LEARY J. T., PEEK N.: Jubilee: An extensible machine for multi-tool fabrication. In *Proc. ACM CHI (2020)*, pp. 298:1–298:13. 13
- [WAG17] WEIZMANN M., AMIR O., GROBMAN Y. J.: Topological interlocking in architecture: A new design method and computational tool for designing building floors. *International Journal of Architectural Computing* 15, 2 (2017), 107–118. 18
- [WCT\*15] WANG W., CHAO H., TONG J., YANG Z., TONG X., LI H., LIU X., LIU L.: Saliency-preserving slicing optimization for effective 3d printing. *Comp. Graph. Forum* 34, 6 (2015), 148–160. 10
- [Wil92] WILSON R. H.: *On Geometric Assembly Planning*. PhD thesis, Stanford University, 1992. 4
- [WL94] WILSON R. H., LATOMBE J.-C.: Geometric reasoning about mechanical assembly. *Artificial Intelligence* 71, 2 (1994), 371–396. 4
- [WM92] WILSON R. H., MATSUI T.: Partitioning an assembly for infinitesimal motions in translation and rotation. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems* (1992), pp. 1311–1318. 3
- [WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Trans. on Graph. (SIGGRAPH Asia)* 28, 5 (2009), 112:1–112:9. 2, 5, 6, 13, 14, 19
- [WQZ\*18] WEI X., QIU S., ZHU L., FENG R., TIAN Y., XI J., ZHENG Y.: Toward support-free 3D printing: A skeletal approach for partitioning models. *IEEE Trans. Vis. & Comp. Graphics* 24, 10 (2018), 2799–2812. 8, 19
- [WRWS15] WU D., ROSEN D. W., WANG L., SCHAEFER D.: Cloud-based design and manufacturing: A new paradigm in digital manufacturing and design innovation. *Computer-Aided Design* 59 (2015), 1–14. 20
- [WSIP19] WANG Z., SONG P., ISVORANU F., PAULY M.: Design and structural optimization of topological interlocking assemblies. *ACM Trans. on Graph. (SIGGRAPH Asia)* 38, 6 (2019), 193:1–193:13. 2, 5, 6, 13, 14, 15, 18, 19
- [WSP18] WANG Z., SONG P., PAULY M.: DESIA: A general framework for designing interlocking assemblies. *ACM Trans. on Graph. (SIGGRAPH Asia)* 37, 6 (2018), 191:1–191:14. 5, 15, 19
- [WSW\*12] WHITING E., SHIN H., WANG R., OCHSENDORF J., DURAND F.: Structural optimization of 3d masonry buildings. *ACM Trans. on Graph. (SIGGRAPH Asia)* 31, 6 (2012), 159:1–159:11. 14, 19
- [WZK16] WANG W. M., ZANNI C., KOBELT L.: Improved surface quality in 3d printing by optimizing the printing direction. *Comp. Graph. Forum (Eurographics)* 35, 2 (2016), 59–70. 10, 19
- [XLF\*11] XIN S.-Q., LAI C.-F., FU C.-W., WONG T.-T., HE Y., COHEN-OR D.: Making burr puzzles from 3D models. *ACM Trans. on Graph. (SIGGRAPH)* 30, 4 (2011), 97:1–97:8. 14, 19
- [YCL\*15] YAO M., CHEN Z., LUO L., WANG R., WANG H.: Level-set-based partitioning and packing optimization of a printable model. *ACM Trans. on Graph. (SIGGRAPH Asia)* 34, 6 (2015), 214:1–214:11. 6, 7, 9, 10, 19
- [YCXW17] YAO M., CHEN Z., XU W., WANG H.: Modeling, evaluation and optimization of interlocking shell pieces. *Comp. Graph. Forum (Pacific Graphics)* 36, 7 (2017), 1–13. 3, 14, 19
- [YKGA17] YAO J., KAUFMAN D. M., GINGOLD Y., AGRAWALA M.: Interactive design and stability analysis of decorative joinery for furniture. *ACM Trans. on Graph.* 36, 2 (2017), 20:1–20:16. 2, 5, 6, 10
- [YPYM17] YUN G., PARK C., YANG H., MIN K.: Legorization with multi-height bricks from silhouette-fitted voxelization. In *Proc. the Computer Graphics International Conference* (2017), pp. 40:1–40:6. 17, 19
- [YYL\*19] YU M., YE Z., LIU Y.-J., HE Y., WANG C. C. L.: LineUp: Computing chain-based physical transformation. *ACM Trans. on Graph.* 38, 1 (2019), 11:1–11:16. 16
- [YYT\*17] YU E. A., YEOM J., TUTUM C. C., VOUGA E., MIIKKU-LAINEN R.: Evolutionary decomposition for 3D printing. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2017), pp. 1272–1279. 8, 19
- [YZC18] YUAN Y., ZHENG C., COROS S.: Computational design of transformables. *Comp. Graph. Forum (SCA)* 37, 8 (2018), 103–113. 16, 19
- [ZB16] ZHANG Y., BALKCOM D.: Interlocking structure assembly with voxels. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems* (2016), pp. 2173–2180. 17, 18, 19
- [ZBKV20] ZHANG X., BELFER R., KRY P. G., VOUGA E.: C-space tunnel discovery for puzzle path planning. *ACM Trans. on Graph. (SIGGRAPH)* 39, 4 (2020), 104:1–104:14. 4
- [ZC18] ZHOU J., CHEN X.: Convertible furniture design. *Comp. & Graph. (CAD/Graphics)* 70 (2018), 165–175. 17, 19
- [ZCX19] ZHOU J., CHEN X., XU Y.: Automatic generation of vivid LEGO architectural sculptures. *Comp. Graph. Forum* 38, 6 (2019), 31–42. 17, 19
- [ZDB17] ZHENG C., DO E. Y.-L., BUDD J.: Joinery: Parametric joint generation for laser cut assemblies. In *Proc. ACM SIGCHI Conference on Creativity and Cognition* (2017), pp. 63–74. 2
- [Zes12] ZEISSIN J. F.: *Collapse Analysis of Unreinforced Masonry Domes and Curving Walls*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 2012. 6
- [ZLP\*15] ZHANG X., LE X., PANOTOPOULOU A., WHITING E., WANG C. C. L.: Perceptual models of preference in 3d printing direction. *ACM Trans. on Graph. (SIGGRAPH Asia)* 34, 6 (2015), 215:1–215:12. 9
- [ZSC16] ZHENG C., SUN T., CHEN X.: Deployable 3D linkages with collision avoidance. In *Proc. Eurographics/ ACM SIGGRAPH Symposium on Computer Animation* (2016), pp. 179–188. 17
- [ZSMS14] ZHOU Y., SUEDA S., MATUSIK W., SHAMIR A.: Boxelization: Folding 3D objects into boxes. *ACM Trans. on Graph. (SIGGRAPH)* 33, 4 (2014), 71:1–71:8. 16, 19
- [ZW12] ZHOU Y., WANG R.: An algorithm for creating geometric dissection puzzles. In *Proc. Bridges Towson: Mathematics, Music, Art, Architecture, Culture* (2012), pp. 49–56. 15, 19
- [ZWC\*16] ZHANG R., WANG S., CHEN X., DING C., JIANG L., ZHOU J., LIU L.: Designing planar deployable objects via scissor structures. *IEEE Trans. Vis. & Comp. Graphics* 22, 2 (2016), 1051–1062. 17