

MOCCA: Modeling and Optimizing Cone-joints for Complex Assemblies

ZIQI WANG, EPFL, Switzerland

PENG SONG, Singapore University of Technology and Design, Singapore

MARK PAULY, EPFL, Switzerland

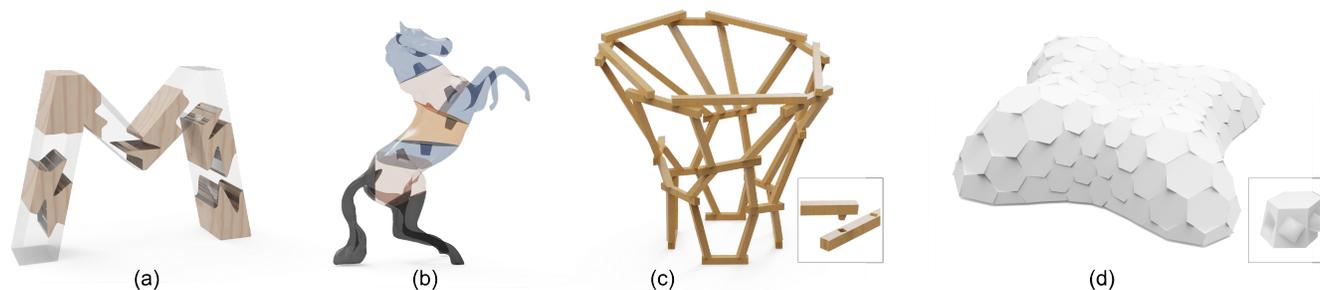


Fig. 1. Our computational framework optimizes cone joints for designing assembleable and stable structures with a variety of geometric forms: (a) planar, (b) volumetric, (c) frame, and (d) shell structures.

We present a computational framework for modeling and optimizing complex assemblies using cone joints. Cone joints are integral joints that generalize traditional single-direction joints such as mortise and tenon joints to support a general cone of directions for assembly. This additional motion flexibility not just reduces the risk of deadlocking for complex joint arrangements, but also simplifies the assembly process, in particular for automatic assembly by robots. On the other hand, compared to planar contacts, cone joints restrict relative part movement for improved structural stability. Cone joints can be realized in the form of curved contacts between associated parts, which have demonstrated good mechanical properties such as reduced stress concentration. To find the best trade-off between assembleability and stability, we propose an optimization approach that first determines the optimal motion cone for each part contact and subsequently derives a geometric realization of each joint to match this motion cone. We demonstrate that our approach can optimize cone joints for assemblies with a variety of geometric forms, and highlight several application examples.

CCS Concepts: • **Computing methodologies** → *Shape modeling*; • **Applied computing** → *Computer-aided manufacturing*.

Additional Key Words and Phrases: 3D assembly, cone joint, structural stability, assembleability, computational design

ACM Reference Format:

Ziqi Wang, Peng Song, and Mark Pauly. 2021. MOCCA: Modeling and Optimizing Cone-joints for Complex Assemblies. *ACM Trans. Graph.* 40, 4, Article 181 (August 2021), 14 pages. <https://doi.org/10.1145/3450626.3459680>

Authors' addresses: Ziqi Wang, EPFL, Switzerland, ziqi.wang@epfl.ch; Peng Song, Singapore University of Technology and Design, Singapore, peng_song@sutd.edu.sg; Mark Pauly, EPFL, Switzerland, mark.pauly@epfl.ch.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2021 Copyright held by the owner/author(s).

0730-0301/2021/8-ART181

<https://doi.org/10.1145/3450626.3459680>

1 INTRODUCTION

An assembly is a collection of parts that are deliberately arranged to have a specific functionality and/or form. The majority of man-made objects are designed as assemblies to accomplish a certain task (machines, vehicles), to make fabricating large objects feasible or cheaper (buildings, furniture), or simply to entertain (puzzles, toys). A necessary condition for an assembly to be practically used is structural stability. To this end, adjacent parts in an assembly have to be properly joined such that no unwanted relative part motions will happen under external forces.

Parts in an assembly are typically joined by glue, nails, screws, or some standard connectors. However, these joining methods do not encourage disassembly and re-assembly, and sometimes harm the external appearance of the assembly. With the advance of digital fabrication techniques, integral joints are more and more widely used for designing and making assemblies with intricate geometry. Integral joints are implicitly defined as the portion of each individual part that is in contact with adjacent parts. These joints can simplify the assembly process significantly as a sequence of operations to insert individual parts, without the need of installing external connectors with tools [Fairham 2013].

Integral joints are typically designed in a way that two parts can be separated by translating one part along a single direction, e.g., mortise and tenon joints and dovetail joints. We call these joints *single-direction joints*; see Figure 2(a). Single-direction joints are widely used in furniture, timber structures, and 3D printed assemblies due to their strong capacity to strengthen structural stability. However, complex arrangements of single-direction joints could lead to deadlocking, making the assembly physically unrealizable. Moreover, these joints may complicate the assembly process as inserting a part precisely along a certain direction to fit the other could be a challenging task, especially in robotic assembly.

On the other end of the spectrum are integral joints with planar contacts (see Figure 2(b)), which are very common in unreinforced

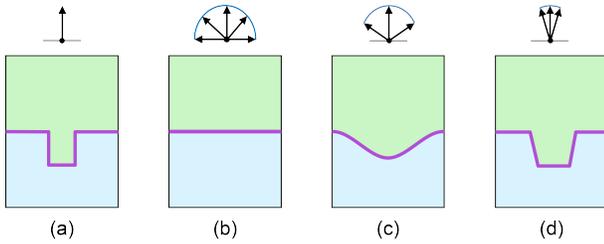


Fig. 2. Two parts joined in different ways using: (a) a single-direction joint, (a) a planar contact, and cone joints with (c) a curved or (d) a piecewise-planar contact (contacts are shown in purple). The four assemblies are all in equilibrium under gravity yet the translational motion space of taking out the green part varies (see cones on the top).

masonry structures [Panozzo et al. 2013; Whiting et al. 2009]. Advantages of *planar contacts* include simple geometry, ease of fabrication, avoidance of local stress concentration. Yet, these joints have the weakest capacity to restrict relative part motion. Incomplete assemblies with planar contacts usually require additional supports for being stable (i.e., in equilibrium) [Deuss et al. 2014].

In this paper we study integral joints that generalize single-direction joints in terms of restricting relative part motion. We focus on making use of these joints for designing structures that are assemblable and stable. We call these joints *cone joints* since they allow one part to be separated/inserted relative to the other by translation along any direction within a motion cone. A cone joint is a single-direction joint if its translational motion cone contains a single direction. And a cone joint degenerates into a planar contact if its translational motion cone becomes a half sphere. Cone joints are typically realized in the form of curved or piecewise-planar contacts between two parts (see Figure 2(c&d)), which have been demonstrated to have good mechanical properties such as reduced stress concentration in building structurally stable assemblies [Dyskin et al. 2003; Javan et al. 2016]. Parts with cone joints can be easily fabricated with 3D printing, CNC milling, and even hot-wire cutting for large-scale objects [Duenser et al. 2020].

Although cone joints have been successfully used in stable planar structures (e.g., brick-based flooring systems [Weizmann et al. 2017]), assembly-based 3D printing [Araújo et al. 2019], and space-filling blocks [Akleman et al. 2020], little is known about how the variation in geometry of cone joints affects a structure’s assemblability and stability, not to mention optimizing the geometry of cone joints for these two design goals. In this paper, assemblability has a two-fold meaning: 1) parts can be assembled into the final structure without collision; and 2) each part can be inserted by translating along any direction within a sufficiently large circular cone, aiming to simplify the assembly process. Stability means a structure is in equilibrium under known external forces such as gravity. To address the challenge of modeling and optimizing complex assemblies with cone joints, we make the following technical contributions:

- We establish a connection between the geometry of a joint and its motion space based on convexity theory. We show that the joint motion space is always convex and present a sampling-based approach to compute the motion space of curved-contact joints.

- We present a motion-based method for static analysis of assemblies with cone joints, which is dual to existing force-based methods. The strength of this new method is to quantify structural stability and assemblability coherently in motion space.
- We develop an optimization approach to construct cone joints for designing structures that are assemblable and stable, assuming the assembly sequence is given. Our framework iterates between a kinematic design stage that determines the required motion cone for each part contact and a geometric realization stage that finds the geometry of each joint to match this motion cone.

In this paper, we model the geometry of cone joints using a simple parametric model to demonstrate the core functionality of our computational framework. Other parametric models can be easily integrated into our framework since they affect only the geometric realization stage, but not the motion space computation, motion-based static analysis, and kinematic design stage.

2 RELATED WORK

Static analysis and structural optimization. Static analysis identifies whether an assembly is in a static equilibrium state under external forces. The equilibrium methods [Shin et al. 2016; Whiting et al. 2009; Yao et al. 2017b] are the current state of the art for static analysis of assemblies with rigid parts. An assembly is considered to be in equilibrium if a network of interaction forces between the parts can be found to balance external forces acting on each part. The equilibrium method has been integrated for designing assemblies that are stable under gravity, including masonry buildings [Block and Ochsendorf 2007; Whiting et al. 2009, 2012], LEGO sculptures [Luo et al. 2015], and furniture with decorative joints [Yao et al. 2017b], as well as for finding assembly plans that require fewer supports for constructing self-supporting structures [Deuss et al. 2014]. Recently, Wang et al. [2019] optimized the geometry of an assembly with planar contacts to achieve a static equilibrium under not just gravity but also lateral forces.

We show that the straightforward extension of the equilibrium method to cone joints leads to a significant increase of computational complexity. Inspired by static-kinematic duality, we propose an alternative motion-based equilibrium method for static analysis, enabling our two-stage approach for designing assemblies with cone joints; see Section 6.

Interlocking assemblies. In an interlocking assembly, parts are immobilized relative to one another by their geometric arrangement, optionally with the aid of friction. Single-direction joints are widely used for designing and making interlocking assemblies, due to their strong capacity to restrict relative part motion. Typical examples include 3D puzzles [Lo et al. 2009], laser-cut sculptures [Cignoni et al. 2014; Hildebrand et al. 2012; Schwartzburg and Pauly 2013], 3D-printable object assemblies [Luo et al. 2012], and woodworking furniture and architecture [Fairham 2013; Larsson et al. 2020].

A specific subclass is single-key interlocking assemblies, where all parts, except for a single key, are immobilized relative to one another purely by their geometric arrangement, without relying on friction [Song et al. 2012]. When an arrangement of initial parts is given, designing interlocking assemblies can be formulated as a *joint planning* problem such that the given parts can be interlocked

by integral joints planned among them. Different joint planning methods have been developed to design and fabricate various interlocking assemblies, including furniture [Fu et al. 2015; Song et al. 2017], laser-cut polyhedrons [Song et al. 2016], 3D-printable shell assemblies [Yao et al. 2017a], and spatial frame structures [Wang et al. 2018]. In addition, Zhang et al. [2021] studied voxel-like interlocking blocks and explored robotic assembly of 3D structures built with instances of these blocks.

In our work, we show that cone joints can be optimized for designing single-key interlocking assemblies. Compared with single-direction joints used in the above works, cone joints allow more flexibility (i.e., larger motion cone) to assemble parts.

Assembly-aware design. Assemblability means that individual parts can be combined into a final assembly without collision. To ensure assemblability, an assembly plan has to be found, which includes a sequence of operations to insert the parts and the corresponding motions that bring each part to its target pose. The problem of finding an assembly plan, known as *assembly planning*, has been shown to be NP-complete [Kavraki et al. 1993]. To make the problem tractable, researchers typically assume a linear assembly sequence (i.e., assemble one part at a time) and simple assembly motion (e.g., translational motion); readers are referred to the survey [Ghandi and Masehian 2015] for more details. *Assembly-aware design* aims to ensure assemblability and to simplify the physical assembly process by searching desirable assembly plans when designing assemblies. For example, Kao et al. [2017] designed masonry shell structures that require significantly fewer supports for construction while Desai et al. [2018] designed electromechanical devices that require only translational motion for parts assembly.

Our work shows that cone joints have great potential for assembly-aware design since they allow a flexible motion cone to assemble each part. In particular, we have optimized cone joints to design 2D equilibrium puzzles that can be assembled without using any support; see Figures 20 and 21.

Assemblies with curved contacts. When 3D printing objects as assemblies, it has been shown that curved contacts are good for respecting object geometric features [Hao et al. 2011] and user-specified surface segmentation [Araújo et al. 2019]. Blocks with curved contacts have been designed as 3D space-filling tiles, such as Delaunay lofts [Subramanian et al. 2019], generalized Abeille tiles [Akleman et al. 2020], and bi-axial woven tiles [Krishnamurthy et al. 2020]. All these works make use of curved contacts as a way of joining parts, without studying how these contacts restrict relative part motion and affect an assembly's structural stability.

Another typical application of blocks with curved contacts is to make structurally stable planar assemblies such as pavements and walls, using a concept called topological interlocking (TI) [Dyskin et al. 2019]. In these TI assemblies, the whole structure is held together by a global peripheral constraint, while locally the blocks are kept in place by kinematic constraints imposed by the curved contacts. Dyskin et al. [2003] introduced osteomorphic blocks for making planar TI assemblies, where the concavo-convex contact surface profiles are defined by using a set of simple rules and mathematical functions. Javan et al. [2016] proposed a new osteomorphic block, which has a symmetrical geometry with four curved side

surfaces. Weizmann et al. [2017] developed a computational tool for designing building floors made as TI assemblies with curved contacts. Recently, researchers in architecture [Fallacara et al. 2019] and civil engineering [Xu et al. 2020] designed 3D TI assemblies with curved contacts by adapting the planar TI assemblies to a given 3D free-form surface. In all these works, structural stability of the TI assemblies with curved contacts was not analyzed at the design stage but just verified in physical experiments and/or simulations. In contrast, we analyze stability of structures with curved contacts using a computational method, and optimize curved-contact cone joints to design structures that are assemblable and stable.

For a recent survey on computational methods for the design of assemblies with rigid parts, we refer to [Wang et al. 2021].

3 MOTION SPACE ANALYSIS OF CONE JOINTS

Giving two parts P_i and P_j , a cone joint $J_{i,j}$ connects them as a curved or piecewise-planar contact between the two parts; see Figure 2(c&d) and 3(a). For simplicity, we discuss only the more general case of curved contacts represented as parametric surfaces. We can always convert such a joint to a piecewise-planar representation (which might offer advantages for fabrication) by discretizing the curved contact surface appropriately. Assuming part P_i is fixed, our

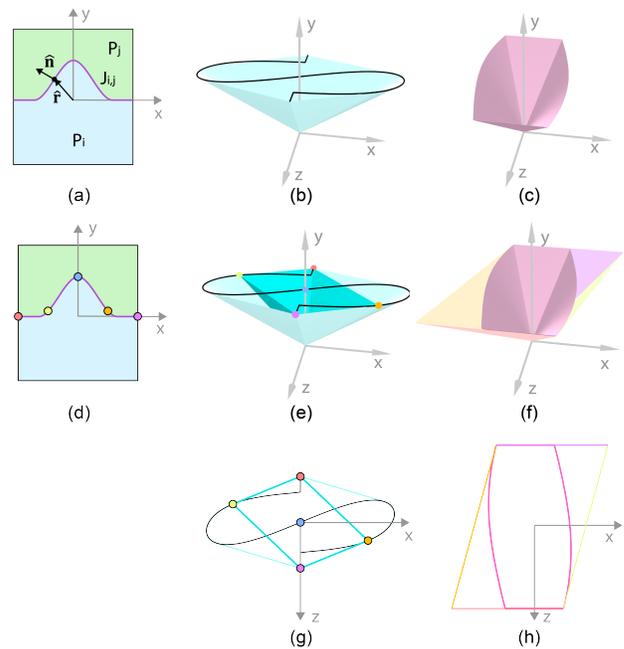


Fig. 3. Motion space analysis of (a) a joint $J_{i,j}$ in a local coordinate frame. (b) Generalized normal curve N (in black) and its minimum convex cone envelope (in light cyan). (c) Motion cone V . (d) Sampling the curved contact with five points. (e) Minimum convex cone envelope (dark cyan) of the sampled generalized normals is contained in the original minimum cone envelope. (f) Motion cone allowed by the sampled points (pyramid with four colored faces) contains the original motion cone, where the color of each face matches that of the corresponding sampled contact point and generalized normal. (g) Conic sections of the two cone envelopes in (e) by cut plane $y = 1$. (h) Conic sections of the two motion cones in (f) by cut plane $y = 1$. We refer to the accompanying video for interactive visualizations.

motion space analysis of joint $J_{i,j}$ considers all possible infinitesimal rigid motion to take out P_j from P_i without collision. This motion space of part P_j , denoted as \mathbf{V} , abstracts the ability of joint $J_{i,j}$ to restrict relative part motion. We will show later how this abstraction is crucial for computational analysis and design of assemblies with cone joints.

We express the infinitesimal rigid motion of part P_j using the generalized velocity \mathbf{v} , which is composed of both translation and rotation. Denote a point on the curved contact of $J_{i,j}$ as $\hat{\mathbf{r}}$ and its normal as $\hat{\mathbf{n}}$; see Figure 3(a). The motion space \mathbf{V} can be obtained by solving a linear inequality system that represents non-collision constraints for every point on the curved contact of $J_{i,j}$ [Wang et al. 2019; Wilson and Matsui 1992]:

$$\mathbf{V} = \{\mathbf{v} \mid \mathbf{n} \cdot \mathbf{v} \geq 0, \forall \mathbf{n} \in \mathbf{N}\} \quad (1)$$

$$\mathbf{n} = \begin{bmatrix} \hat{\mathbf{n}} \\ \hat{\mathbf{r}} \times \hat{\mathbf{n}} \end{bmatrix}$$

where \mathbf{n} is the generalized normal of point $\hat{\mathbf{r}}$ with normal $\hat{\mathbf{n}}$, and \mathbf{N} is the generalized normal space of joint $J_{i,j}$.

For a smooth 2D joint, the generalized normal space \mathbf{N} is a curve in 3D space (2 dimensions for normal $\hat{\mathbf{n}}$ and 1 dimension for $\hat{\mathbf{r}} \times \hat{\mathbf{n}}$); and the motion space \mathbf{V} is a cone in 3D space (2 dimensions for translation and 1 dimension for rotation); see Figure 3(b&c). We make a connection between the generalized normal space \mathbf{N} of joint $J_{i,j}$ and the motion space \mathbf{V} of part P_j based on the dual cone concept in convexity theory, and formulate the following theorems and lemma; please refer to Section 2.6.1 in [Boyd and Vandenberghe 2004] for a derivation.

THEOREM 3.1 (DUAL CONE). *The dual cone \mathbf{C}^* of a set \mathbf{C} in R^n , defined as*

$$\mathbf{C}^* = \{\mathbf{y} \in R^n : \mathbf{y} \cdot \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbf{C}\},$$

is always convex, even when the original set \mathbf{C} is not.

According to Theorem 3.1 and Equation 1, the motion space \mathbf{V} can be viewed as the dual cone of the generalized normal space \mathbf{N} , i.e., $\mathbf{V} = \mathbf{N}^*$. Hence, the motion space \mathbf{V} must be convex no matter whether the generalized normal space \mathbf{N} is convex or not; see Figure 3(c). Replacing the generalized normal space \mathbf{N} with its minimum convex cone envelope (see the cyan cone in Figure 3(b)) will not affect its dual cone. This minimum convex cone envelope of \mathbf{N} is called the generalized normal cone, or simply $\text{cone}(\mathbf{N})$.

A motion cone can be exactly defined by a cut plane and a cross section from that cut (called the conic section); see Figure 3(h). For a symmetric 2D joint like Figure 3(a), the cut plane is simply chosen as $y = 1$. However, for general motion cones, it is critical to choose a proper cut plane such that the conic section is finite; please refer to the supplementary material for details. Similarly, the generalized normal cone $\text{cone}(\mathbf{N})$ also can be represented and visualized by a cut plane and the corresponding conic section; see Figure 3(b&g).

THEOREM 3.2 (SAMPLING). *If $\tilde{\mathbf{N}}$ is a subset of space \mathbf{N} , the corresponding dual cone $\tilde{\mathbf{V}}$ of $\tilde{\mathbf{N}}$ should contain \mathbf{N}^* or equivalently \mathbf{V} :*

$$\tilde{\mathbf{N}} \subseteq \mathbf{N} \Rightarrow \mathbf{V} \subseteq \tilde{\mathbf{V}}$$

Theorem 3.2 enables us to compute an approximation $\tilde{\mathbf{V}}$ of the motion space \mathbf{V} numerically. First, we sample the generalized normal space \mathbf{N} to obtain a finite subset $\tilde{\mathbf{N}}$; see the five colored points in Figure 3(d&e). Next, an approximated motion cone $\tilde{\mathbf{V}}$ is computed by intersecting a finite number of half-spaces described by Equation 1; see the pyramid with four colored planes in Figure 3(f). Note that the half-space corresponding to the dark blue point is not shown since it does not contribute to the approximated motion cone $\tilde{\mathbf{V}}$. Theorem 3.2 guarantees that the approximated motion cone $\tilde{\mathbf{V}}$ must cover all possible motions in the original motion cone \mathbf{V} ; see Figure 3(f). In our experiments, we find that 50 (200) sample points per 2D (3D) joint provide a good approximation of the motion cone \mathbf{V} .

LEMMA 3.3. *If a motion cone \mathbf{V} is contained in a given space \mathbf{V}_0 , it is equivalent to require that $\text{cone}(\mathbf{N})$ includes the dual cone of \mathbf{V}_0 :*

$$\mathbf{V} \subseteq \mathbf{V}_0 \Leftrightarrow \mathbf{V}_0^* \subseteq \text{cone}(\mathbf{N})$$

Lemma 3.3 can be used to generate joint geometry that have a bounded motion cone \mathbf{V}_0 . For instance, if the motion cone \mathbf{V}_0 is chosen to be a circular cone with an angle α (i.e., $\{(\mathbf{x}, y) : \|\mathbf{x}\| \leq \tan(\alpha)y\}$), then its dual cone \mathbf{V}_0^* is simply $\{(\mathbf{x}, y) : \|\mathbf{x}\| \leq \frac{y}{\tan \alpha}\}$, which is also a circular cone with an angle $\frac{\pi}{2} - \alpha$. Classic geometric processing algorithms can be applied to find joint geometry whose generalized normal cone contains the dual circular cone \mathbf{V}_0^* . We will discuss in Section 6.3 how Lemma 3.3 guides geometric realization of our cone joints.

4 STATIC ANALYSIS FOR ASSEMBLIES WITH CONE JOINTS

Static analysis determines whether a given assembly is in equilibrium under known external forces. Existing equilibrium methods work in force space, and focus on assemblies with planar contacts [Whiting et al. 2009] or single-direction joints [Yao et al. 2017b]. This section shows that the force-based equilibrium method can be extended for static analysis of assemblies with curved contacts, albeit with a significant increase of computational cost (Section 4.1). As an alternative, we propose a new motion-based equilibrium method (Section 4.2). As we discuss in detail below, this approach offers drastic performance improvements for our optimization-based design of assemblies with cone joints.

4.1 Force-based Equilibrium Method

Test of equilibrium. The goal of the force-based equilibrium method [Whiting et al. 2009] is to find a network of interaction forces between the parts that can balance the external forces and torques acting on each part. In assemblies with planar contacts, the interaction forces are discretized as a finite number of forces at the vertices of each contact interface (usually a 3D polygon). Since we assume rigid parts and ignore friction, interaction forces only include compression forces along the contact normal direction; see Figure 4(a).

Combining equilibrium constraints for each part gives a linear system of equations:

$$\mathbf{A}_{\text{eq}} \cdot \mathbf{f} = -\mathbf{w} \quad \text{s.t. } \mathbf{f} \geq \mathbf{0} \quad (2)$$

where \mathbf{A}_{eq} is the matrix of coefficients for the equilibrium equations, \mathbf{f} represents the unknown interaction forces, and \mathbf{w} represents the

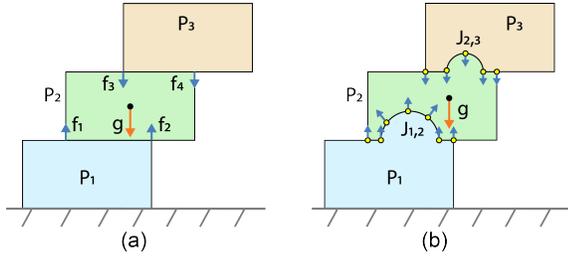


Fig. 4. Force-based equilibrium method for static analysis of assemblies with (a) planar contacts and (b) curved contacts. Taking the green part P_2 as an example, its external force due to gravity is colored in orange, and its contact forces f_i are colored in blue.

generalized external forces acting on the assembly; see [Whiting et al. 2009] for details. In particular, each element of \mathbf{f} is the magnitude of contact force at each vertex of each contact interface, and the dimension of \mathbf{f} is $M \times 1$, where M is the total number of vertices for all planar contacts in the assembly. For an assembly to be in equilibrium, a force solution \mathbf{f} that satisfies the linear system in Equation 2 must exist.

Measure of infeasibility. When an assembly is not in equilibrium, Whiting et al. [2009] proposed a method to measure its distance to a feasible solution (i.e., an equilibrium state) by introducing tension forces that act as “glue” at part contact interfaces to hold the assembly together and penalizing these tension forces:

$$\begin{aligned} \min_{\mathbf{f}^+, \mathbf{f}^-} \quad & \frac{1}{2} \mathbf{f}^- \cdot \mathbf{f}^- \\ \text{s.t.} \quad & \mathbf{A}_{\text{eq}} \cdot (\mathbf{f}^+ - \mathbf{f}^-) = -\mathbf{w}, \\ & \mathbf{f}^+, \mathbf{f}^- \geq 0 \end{aligned} \quad (3)$$

where \mathbf{f}^+ is the positive parts (i.e., compression forces) of the contact forces \mathbf{f} , and \mathbf{f}^- is the negative parts (i.e., tension forces) of \mathbf{f} . The quadratic programming in Equation 3 enables to measure infeasibility and to test equilibrium in a unified way, since an assembly is in equilibrium if the infeasibility measure equals zero. Existing algorithms to solve the quadratic programming with M variables typically have a polynomial complexity with respect to M .

Extension for assemblies with curved contacts. To make the force-based equilibrium method applicable for assemblies with curved contacts, we need to approximate each curved contact as a piecewise-planar surface. In practice, we simply sample points uniformly on the contact and stack interaction forces for each sampled point to form the vector \mathbf{f} in Equation 2; see the inset. If there exists a solution \mathbf{f} for Equation 2, the assembly is considered as in equilibrium. However, non-existence of such a solution does not mean that the assembly is not in equilibrium since our interaction forces are only samples of the actual forces. To obtain a good approximation, we typically require 50 (200) sample points per 2D (3D) joint, which is similar to approximating a motion cone in Section 3. Note that the number of sample points per curved contact is much higher than the number of vertices (typically 4) per planar contact, increasing the computational cost significantly to solve Equation 3.

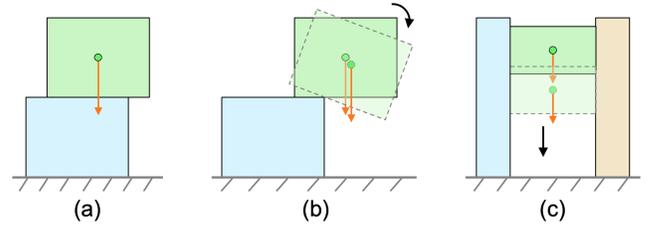
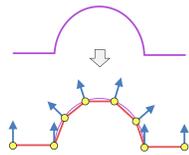


Fig. 5. Motion-based equilibrium method for static analysis. In these examples, external forces are gravity (in orange) and ground supporting forces only, and friction is ignored. (a) The assembly is in equilibrium as we cannot find any parts motion that satisfies Equations 5 and 6. (b&c) The two assemblies are not in equilibrium, where a parts motion solution that satisfies Equations 5 and 6 is shown as a dashed boundary and a black arrow.

4.2 Motion-based Equilibrium Method

As an alternative to the force-based approach, we propose a motion-based equilibrium method, inspired by static-kinematic duality.

Test of equilibrium. Given an assembly, we assume each part P_i can move freely in 3D space and model its infinitesimal rigid motion as a generalized velocity \mathbf{v}_i . Each pair of adjacent parts, say P_i and P_j , should not collide with each other during the movement:

$$\mathbf{n}_k \cdot (\mathbf{v}_j - \mathbf{v}_i) \geq 0 \quad (4)$$

where \mathbf{n}_k is a generalized normal sampled at the curved contact between P_i and P_j . Stacking all these non-penetration constraints forms a linear inequality system:

$$\mathbf{B}_{\text{in}} \cdot \mathbf{v} \geq 0 \quad (5)$$

where \mathbf{B}_{in} is the matrix of coefficients for the non-penetration constraints among the parts in the assembly, and \mathbf{v} is a vector of generalized velocities for all the parts. The parts that touch the ground are fixed by setting their velocities to be zero. Note that the relation between Equation 5 and Equation 2 has been established by Wang et al. [2019]; i.e., $\mathbf{B}_{\text{in}} = \mathbf{A}_{\text{eq}}^T$.

When the inequality system in Equation 5 does not have any non-zero solution, the assembly is considered as deadlocking, meaning no part can move in the assembly. Otherwise, parts are movable. If the part movement is driven by the given external forces (e.g., gravity), the assembly should not be in equilibrium; see Figure 5(b&c) for two examples. Inspired by this observation, we add an additional constraint for the equilibrium test:

$$\mathbf{w}^T \cdot \mathbf{v} > 0 \quad (6)$$

where \mathbf{w} is the generalized external forces defined in Equation 2, and $\mathbf{w}^T \cdot \mathbf{v}$ can be understood as the total power created by the external forces \mathbf{w} for a given motion configuration \mathbf{v} .

For an assembly to be in equilibrium, there should not exist any solution \mathbf{v} that satisfies the two linear constraints described in Equations 5 and 6; see Figure 5(a) for an example. This statement can be proved by showing that it is actually equivalent to the force-based equilibrium test (Equations 2) using Farkas’ lemma [Farkas 1902].

Measure of infeasibility. Similar to our test of equilibrium, we propose a measure of infeasibility based on the parts motion \mathbf{v} :

$$\begin{aligned} \max_{\mathbf{v}} \quad & \mathbf{w}^T \mathbf{v} - \frac{1}{2} \mathbf{v}^T \mathbf{v} \\ \text{s.t.} \quad & \mathbf{B}_{\text{int}} \cdot \mathbf{v} \geq 0 \end{aligned} \quad (7)$$

where the second term in the objective function is a regularization to prevent the energy from becoming infinity. The assembly should be in equilibrium when the infeasibility measure equals zero.

To understand the relation between our measure (Equation 7) and that of [Whiting et al. 2009] (Equation 3), we apply the strong duality theorem [Boyd and Vandenberghe 2004] to our measure, and obtain the following formulation:

$$\begin{aligned} \min_{\mathbf{f}, \mathbf{s}} \quad & \frac{1}{2} \mathbf{s}^T \mathbf{s} \\ \text{s.t.} \quad & \mathbf{A}_{\text{eq}} \cdot \mathbf{f} + \mathbf{s} = -\mathbf{w}, \\ & \mathbf{f} \geq 0 \end{aligned} \quad (8)$$

where \mathbf{s} is additional forces/torques required to make each part in balance; see the supplementary material for a proof. This dual formulation of our measure can be understood as an alternative of the measure in [Whiting et al. 2009].

Lastly, the non-penetration constraints in Equation 7 can be reformulated based on the motion cone $\mathbf{V}_{i,j}$ of each curve-contact joint $J_{i,j}$ (see again Section 3):

$$\mathbf{v}_j - \mathbf{v}_i \in \mathbf{V}_{i,j} \quad (9)$$

Hence, the infeasibility measure (Equation 7) can be reformulated as:

$$\begin{aligned} E(\mathbf{w}, \{\mathbf{V}_{i,j}\}) = \max_{\mathbf{v}_i} \quad & \mathbf{w}^T \mathbf{v} - \frac{1}{2} \mathbf{v}^T \mathbf{v} \\ \text{s.t.} \quad & \mathbf{v}_j - \mathbf{v}_i \in \mathbf{V}_{i,j}, \text{ for each } J_{i,j} \end{aligned} \quad (10)$$

Remark. In terms of static analysis for assemblies with curved contacts, our motion-based method has the same computational complexity as the force-based method since they are convex quadratic programs with dual constraints (see Equations 3 and 7), which are typically solved by primal-dual algorithms [Monteiro and Adler 1989]. The strength of our method is for efficient design of structurally stable assemblies with cone joints. As we pointed out earlier, static analysis of assemblies with cone joints is computationally expensive since each motion cone $\mathbf{V}_{i,j}$ in Equation 10 typically has thousands of faces corresponding to the sampled generalized normals on the curved contact. Instead, based on our motion-based method, we can satisfy the equilibrium condition efficiently by finding approximated motion cones with a simple representation (e.g., a cone with 4 faces), thanks to the *convexity* of these cones guaranteed by Theorem 3.1. The parts/joints geometry can then be found later to realize each of these approximated motion cones; see Section 6 for details about optimizing cone joints based on this idea.

5 MODELING ASSEMBLIES WITH CONE JOINTS

In this section, we first present a parametric model to represent the geometry of cone joints with curved contacts. Next, we introduce a method to model assemblies with cone joints.

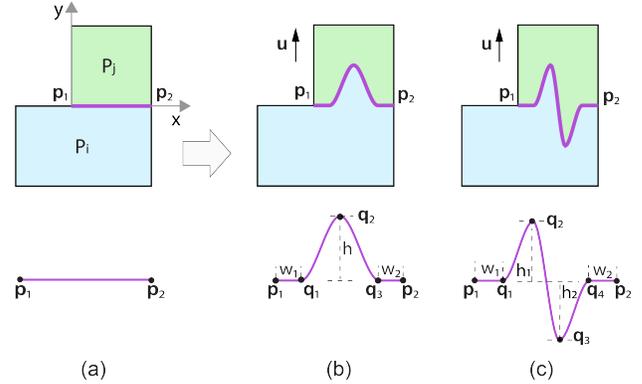


Fig. 6. Model cone joints, where the principal direction $\mathbf{u} = +y$. (a) An initial linear contact between two parts P_i and P_j ; (b) an n-type cone joint; and (c) a z-type cone joint. The curved contact of each cone joint is modeled as a cubic spline defined by a few parameters (e.g., w_1 , h , and w_2 in (b)).

5.1 Modeling Geometry of Cone Joints

We define a contact as a curve segment (in 2D) or surface patch (in 3D) that lies exactly on two adjacent parts in an assembly. Each contact is initialized as a linear segment (in 2D) or a planar surface (in 3D). Starting from each initial contact, a cone joint can be modeled by modifying the geometry of the contact (i.e., making it curved). We propose a parametric model to represent the geometry of these cone joints. To facilitate understanding, we illustrate our parametric model mainly on 2D examples, and briefly explain how it can be extended to 3D.

One joint between two parts. Given two 2D parts, P_i and P_j , with an initial linear contact defined by two endpoints \mathbf{p}_1 and \mathbf{p}_2 , a local coordinate system is defined in a way that the origin is at \mathbf{p}_1 and the $+x$ -axis is along vector $\mathbf{p}_2 - \mathbf{p}_1$; see Figure 6(a). We model the geometry of the cone joint between the two parts as a continuous curved contact. We parametrize the curved contact by using a cubic spline that interpolates the two endpoints \mathbf{p}_1 and \mathbf{p}_2 (see Figure 6(b&c)). According to the shape of the curved contact, joints can be classified into two classes:

- *n-type joint* involves a concavity on one part and an extrusion on the other part, similar to the mortise-and-tenon joint. The joint shape is adjustable by three control points \mathbf{q}_1 , \mathbf{q}_2 , and \mathbf{q}_3 inserted between the two endpoints \mathbf{p}_1 and \mathbf{p}_2 ; see Figure 6(b).
- *z-type joint* can be considered a combination of two n-type joints with opposite orientations. The joint shape is adjustable by four control points \mathbf{q}_1 , \mathbf{q}_2 , \mathbf{q}_3 , and \mathbf{q}_4 inserted between the two endpoints \mathbf{p}_1 and \mathbf{p}_2 ; see Figure 6(c).

Compared with n-type joints, z-type joints generally have stronger capacity to restrict relative part motion, at the cost of more complex joint geometry and higher chance of stress concentration. In our experiments, we set bounds for the joint parameters to preserve their appearance and structural soundness.

To ensure that P_j can translate along a principal direction \mathbf{u} while P_i is fixed, the contact curve has to be a height field along \mathbf{u} , taking the initial contact $\mathbf{p}_1\mathbf{p}_2$ as the base. Figure 6(b&c) shows example

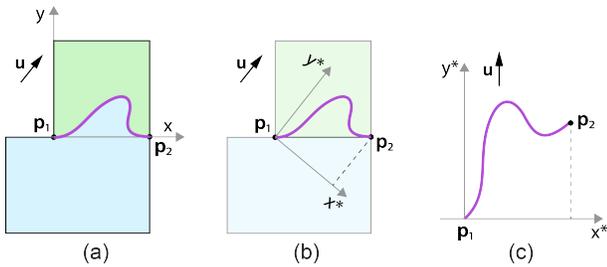


Fig. 7. Model cone joints when the principal direction \mathbf{u} deviates from the y -axis. (a&b) Define a new local coordinate system such that $+y^* = \mathbf{u}$. (c) Model the joint geometry in the new local coordinate system following the same method as $\mathbf{u} = +y$.

cone joints where P_j is removable along $\mathbf{u} = +y$. When the principal direction \mathbf{u} deviates from the y -axis, we need to define a new local coordinate system where the origin is still at \mathbf{p}_1 yet $+y^* = \mathbf{u}$; see Figure 7(a&b). Next, we transform the endpoint \mathbf{p}_1 and \mathbf{p}_2 into this new coordinate system, and follow the same method as $\mathbf{u} = +y$ to model the joint geometry as a cubic spline; see Figure 7(c).

Multiple joints between two parts. Two initial parts may have more than one linear contact when the parts have non-convex shape (see Figure 8(a-c)). Ideally, we construct a cone joint for each linear contact. To this end, we assign a principal direction \mathbf{u}^k to each contact C^k , and follow the above method to model the joint. To ensure that the two parts always can be separated, all assigned \mathbf{u}^k 's should have the same value; see Figure 8(d&e). Sometimes, some initial contact could be unsuitable for modeling a joint, e.g., when the contact is too small. In this case, we simply skip the joint modeling for that contact; see Figure 8(c&f) for an example.

Cone joints in 3D. The parametric model for cone joints in 2D can be easily extended to 3D. In the following, we take a 3D n -type joint as an example to illustrate the extension, for which the initial contact is a 3D polygon and the principal direction is $\mathbf{u} = +z$. We model the 3D joint geometry as a parametric surface based on bivariate

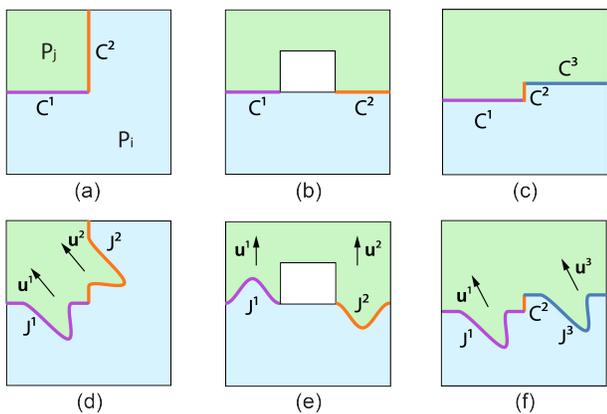


Fig. 8. Model cone joints when (a) P_i , (b) P_j , or (c) both P_i and P_j have non-convex shape, resulting multiple contacts between the two parts. (d&e) The principal directions of the two contacts should be the same to ensure P_1 and P_2 can be separated. (f) It is possible to skip joint modeling for some initial contacts that are too small, e.g., the orange one.

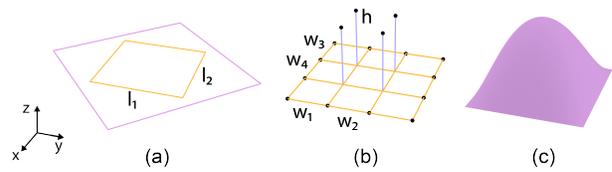


Fig. 9. Model n -type cone joints in 3D. (a) Identify a large rectangular region inside an initial 3D planar contact (i.e., the dashed polygon). (b) Compute sixteen control points, where h is the height of the four points in the middle. (c) Joint geometry modeled as a bicubic surface.

polynomials of degree 3 defined on a 3D rectangular region ($l_1 \times l_2$) within the initial contact. Hence, we first find a large rectangular region inside the initial contact (Figure 9(a)). Next, we divide the rectangular region into 9 rectangles with 16 interpolation points $\{\mathbf{p}_{ij}\}$ in total (Figure 9(b)). We assign the middle four points a height value h and the remaining points zero height, and apply bicubic interpolation to find the joint contact surface passing through these control points (Figure 9(c)). We show in Section 6 how our simple parametric model is well suited for optimization.

5.2 Modeling Assemblies with Cone Joints

In an assembly with $n \geq 2$ parts, we denote the parts as $\{P_i\}$, $1 \leq i \leq n$. We assume a user-specified assembly sequence, and name each part according to its assembly order (i.e., P_i is the i th assembled part). In the initial assembly with planar contacts, we model each part as a polyhedron. Faces in each part can be classified as *contact* and *non-contact* faces, according to whether the part is in contact with its neighbors through that face. Non-contact faces are allowed to be augmented with geometric features to enrich the assembly's appearance.

The initial planar contacts between each pair of adjacent parts, say P_i and P_j , are denoted as $\{C_{i,j}^k\}$. Each $\{C_{i,j}^k\}$ typically contains one contact, especially when initial parts are convex; see Figure 10(a)

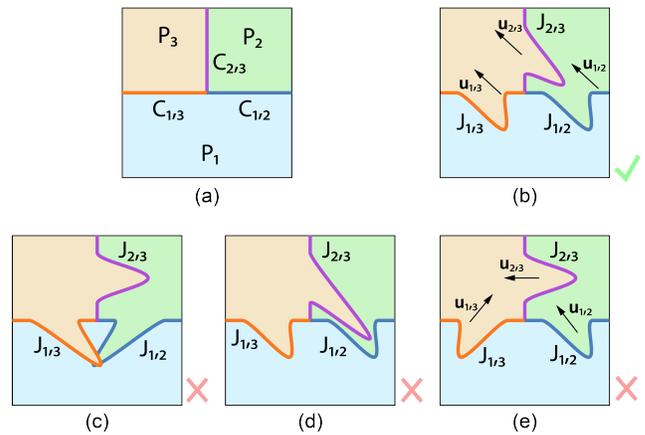


Fig. 10. Model cone joints in an assembly. (a) An initial assembly with three parts. (b) A modeled assembly that is disassemblable by taking out P_3 first. Examples of three cases that should be avoided: (c) cone joints intersect with one another; (d) cone joints are too close to one another, leading to structurally weak parts; and (e) motion cones of the joints are not well planned, leading to a deadlocking assembly.

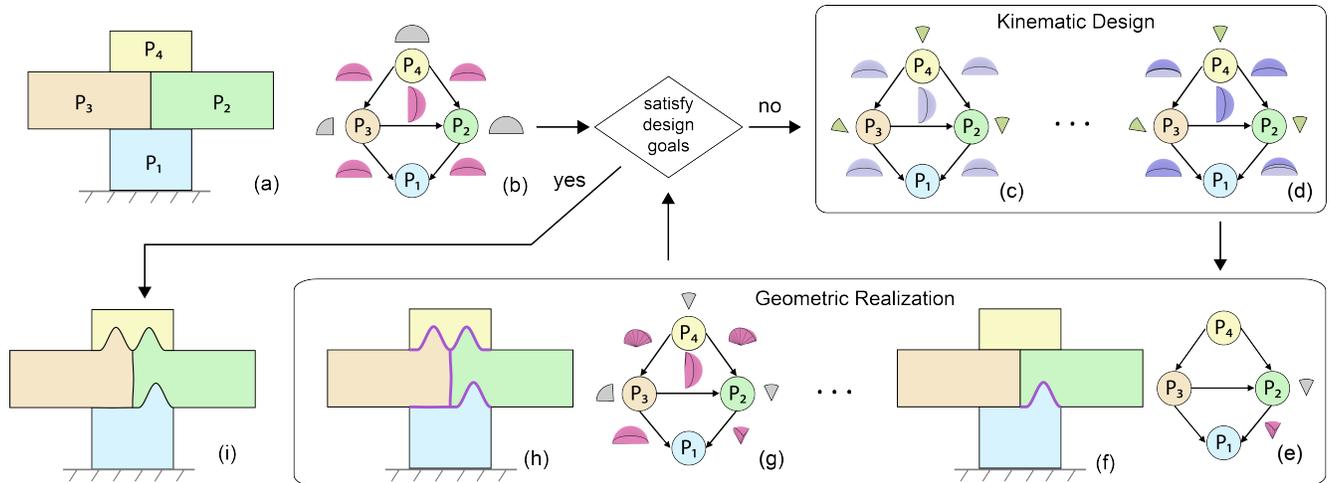


Fig. 11. Our computational design framework. (a) Our input is an initial assembly with planar contacts. (b) Motion-based representation of the initial assembly, where the joint motion cones (3D) and part motion cones (2D) are colored in magenta and gray respectively. (c&d) Kinematic design satisfies the two design goals by iteratively finding the required joint motion cones (colored in blue). (e-h) Geometric realization finds joints geometry independently to satisfy the corresponding joint motion cone. (i) The resulting assembly.

for an example. The cone joints modeled from the initial contacts $\{C_{i,j}^k\}$ are denoted as $\{J_{i,j}^k\}$. The geometry of each joint $J_{i,j}^k$ is parameterized by a small set of parameters $\Phi_{i,j}^k$, which include the joint $J_{i,j}^k$'s principal direction $\mathbf{u}_{i,j}^k$ (see Figure 10(b)) and the geometric parameters illustrated in Figure 6(b&c).

To ensure that parts (with cone joints) are fabricable and assemblable, several cases of joint configurations should be avoided by a careful selection of the joint parameters $\{\Phi_{i,j}^k\}$. First, intersection among the cone joints should be strictly avoided to ensure that the assembly is physically realizable. For example, the extrusion portions of $J_{1,2}$ and $J_{1,3}$ in P_1 should not intersect with each other like in Figure 10(c). Second, when a part has multiple cone joints, those cone joints should maintain a certain distance to one another to ensure structural soundness of the part. For example, part P_2 in Figure 10(d) is structurally weak since joints $J_{2,3}$ and $J_{1,2}$ are too close. Lastly, deadlocking should be avoided to ensure that parts can be physically assembled; compare Figure 10(b&e).

6 DESIGNING ASSEMBLIES WITH CONE JOINTS

Taking an assembly with planar contacts as an input, our goal is to make it structurally stable and assemblable by constructing cone joints among the parts; see Figure 11(a&i). Here, structural stability means that the assembly is in equilibrium under known external forces $\{\mathbf{w}_i\}$ (e.g., gravity of each part) while assemblability means that each individual part can be inserted without colliding with assembled parts, by translation along any direction within a sufficiently large cone. We model this required assemblability cone for each part P_i as a circular cone $K(\mathbf{d}_i, \alpha)$ for simplicity, where \mathbf{d}_i is the cone's axis and α is the cone angle.

To obtain a structurally stable assembly, we prefer that each constructed cone joint can restrict the relative motion between the associated parts as much as possible. In the limit, each joint becomes a single-direction joint. However, to satisfy the goal of

assemblability, single-direction joints should be strictly avoided and each constructed joint should have enough tolerance in the insertion direction of associated parts. Hence, our challenge is to find the geometry of cone joints that satisfy these two conflicting goals (i.e., stability and assemblability) simultaneously.

6.1 Overview of our approach

To address this problem, one straightforward approach is to directly search parameters that define the geometry of cone joints, e.g., by using a gradient-based method similar to [Whiting et al. 2012]. However, this approach has several limitations. First, the test of equilibrium could be very slow due to the dense sampling of cone joints; see Section 4.1. Second, the approach heavily relies on the initial geometry of the joints. Changing joint geometry significantly or even joint type requires restarting the whole search process.

Inspired by our motion-based equilibrium method in Section 4.2, we propose a new computational approach that is able to search the joint parameters efficiently and flexibly; see Figure 11. The key idea is to separate the search process into two stages: *kinematic design* and *geometric realization*, by introducing an intermediate motion-based representation of the assembly. In the kinematic design stage, we aim to satisfy the two design goals kinematically, by searching the motion cone required for each cone joint; see Figure 11(c&d). In the geometric realization stage, we compute for each joint suitable geometric parameters to satisfy the required motion cone; see Figure 11(e-h). The strength of our approach is that the kinematic design stage converts the two high-level design goals into a set of local requirements on the geometry of each individual joint. This allows us not just to search geometric parameters independently for each joint, but also to try as many initial joint parameters/types as possible to avoid local minima. Moreover, the kinematic design stage itself is very efficient since it purely works in motion space and focuses on the required motion cones with very few faces.

6.2 Kinematic Design

To facilitate discussion, we assume a single planar contact $C_{i,j}$ between each pair of adjacent parts P_i and P_j in an assembly, from which our approach will construct a cone joint $J_{i,j}$ by optimizing its parameters $\Phi_{i,j}$ defined in Section 5.2. The kinematic design stage aims at finding a required motion cone $\bar{\mathbf{V}}_{i,j}$ for each cone joint $J_{i,j}$ such that the two design goals can be satisfied kinematically. In the following, we first present a motion-based representation of a given assembly and then describe an optimization to search $\{\bar{\mathbf{V}}_{i,j}\}$ based on this representation.

Motion-based representation. An assembly $\{P_i\}$ can be represented by a parts-graph, where each node represents a part and each edge represents the contact/joint between the two associated parts. The known part assembly order can be easily encoded in the parts-graph by adding a direction for each edge. More precisely, each directed edge from P_j to P_i indicates that P_i will be assembled before P_j (i.e., $i < j$ according to our notation); see Figure 11(b).

Our motion-based representation augments this directed parts-graph with two pieces of information. We associate to each edge from P_j to P_i a motion cone $\mathbf{V}_{i,j}$ allowed by the contact $C_{i,j}$ or joint $J_{i,j}$. The geometry of the motion cone $\mathbf{V}_{i,j}$ is defined by all possible infinitesimal rigid motions of P_j to separate it from a fixed P_i ; see again Figure 3(a&c). For each node P_j , we define a motion cone \mathbf{V}_j of all possible infinitesimal translational motion to take out P_j from the partial assembly $\{P_1, \dots, P_{j-1}\}$. Denote the indices of P_j 's adjacent parts in the given assembly as $A(j)$. The motion cone \mathbf{V}_j for taking out P_j can be represented as:

$$\mathbf{V}_j = \bigcap_{i < j, i \in A(j)} T(\mathbf{V}_{i,j}) \quad (11)$$

where $T(\cdot)$ is an operator that converts a rigid motion cone into a translational motion cone by ignoring the rotational component. This is because we assume translational motion for assembling the parts due to its simplicity of execution. \mathbf{V}_j is called the part motion cone and used for checking (dis)assemblability while $\mathbf{V}_{i,j}$ is called the joint motion cone and used for static analysis.

Given the motion-based representation, we are able to easily test whether the two design goals are satisfied. For equilibrium, we run the optimization in Equation 10, in which each joint motion cone $\mathbf{V}_{i,j}$ is defined in a global coordinate system associated with the assembly rather than the joint local coordinate system. To verify assemblability, we search whether each part motion cone \mathbf{V}_j contains a circular cone $K(-\mathbf{d}_j, \alpha)$, where the cone angle α is a constant and the cone direction \mathbf{d}_j is unknown. Note that the circular cone $K(-\mathbf{d}_j, \alpha)$ is a disassemblability cone, which is actually a reflection of the assemblability cone $K(\mathbf{d}_j, \alpha)$ in the motion space.

Search for required joint motion cones $\{\bar{\mathbf{V}}_{i,j}\}$. We model each required joint motion cone $\bar{\mathbf{V}}_{i,j}$ as a polyhedral cone with a small number of faces to speed up the static analysis process. In particular, the conic section of each required joint motion cone $\bar{\mathbf{V}}_{i,j}$ is a rectangle (4 faces) for 2D joints and 5D cuboid (10 faces) for 3D joints, defined in the joint local coordinate system; see Figure 12(top). When transformed to the global coordinate system, these conic sections will not be rectangle/cuboid any more; see Figure 12(bottom).

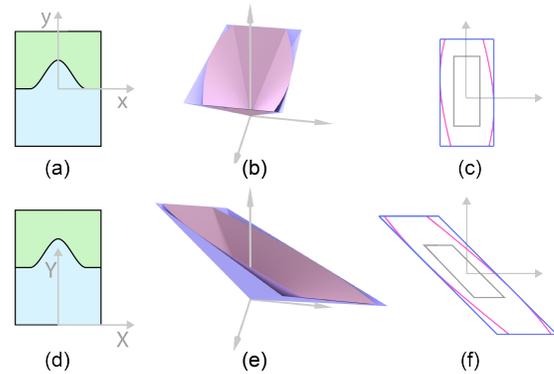


Fig. 12. (a&d) Given a joint, (b&e) its actual and required joint motion cones as well as (c&f) their conic sections are colored in magenta and blue, respectively. When changing from (top) the joint local coordinate system to (bottom) the assembly global coordinate system, the motion cones change accordingly. An example bound of the required joint motion cone is colored in grey in (c&f), for the joint's principal direction $\mathbf{u} = +y$.

We parameterize the geometry of each motion cone $\bar{\mathbf{V}}_{i,j}$ in 2D (3D) with a few variables denoted as $\Psi_{i,j} = \{\psi_{i,j}\}$. Based on the required joint motion cones $\{\bar{\mathbf{V}}_{i,j}\}$, we compute the corresponding part motion cones $\{\bar{\mathbf{V}}_j\}$ according to Equation 11.

Given the required joint motion cones $\{\bar{\mathbf{V}}_{i,j}\}$ as well as the known external forces $\{\mathbf{w}_i\}$, we compute the assembly's infeasibility measure for equilibrium by solving Equation 10. Based on our modeling, each required joint motion cone $\bar{\mathbf{V}}_{i,j}$ can be converted back to 4 (10) linear constraints in our infeasibility energy optimization for each 2D (3D) joint. With very few constraints, the optimization can be performed efficiently. We also satisfy the goal of assemblability by requiring the following constraint for each part motion cone $\bar{\mathbf{V}}_j$:

$$K(-\mathbf{d}_j, \alpha) \subseteq \bar{\mathbf{V}}_j \quad (12)$$

To this end, the kinematic design can be formulated as a problem to search for the required joint motion cones $\{\bar{\mathbf{V}}_{i,j}\}$ by solving the following optimization:

$$\begin{aligned} \min_{\Psi_{i,j}, \mathbf{d}_j} \quad & E(\mathbf{w}, \{\bar{\mathbf{V}}_{i,j}\}) \\ \text{s.t.} \quad & K(-\mathbf{d}_j, \alpha) \subseteq \bar{\mathbf{V}}_j, \end{aligned} \quad (13)$$

where E is the motion-based infeasibility measure in Equation 10. In practice, we avoid too small motion cones for the cone joints such that the motion cones can be realized by joint geometry later. Hence, we set a bound for the parameters $\Psi_{i,j}$ of each required joint motion cone $\bar{\mathbf{V}}_{i,j}$; see Figure 12(c&f). The above optimization can be solved by using an off-the-shelf interior-point method. Please refer to the supplementary material for details.

Once we have obtained the required joint motion cones $\{\bar{\mathbf{V}}_{i,j}\}$, we should require $\mathbf{V}_{i,j} \subseteq \bar{\mathbf{V}}_{i,j}$, where $\mathbf{V}_{i,j}$ is the actual joint motion cone, to ensure the resulting assembly is in equilibrium. To satisfy the goal of assemblability, we should require $\hat{K}(-\mathbf{d}_j, \alpha) \subseteq \mathbf{V}_j \subseteq \bar{\mathbf{V}}_j$, where $\hat{K}(-\mathbf{d}_j, \alpha) = K(-\mathbf{d}_j, \alpha) \times \{\mathbf{0}\}^{2m-3}$, $m = 2, 3$ is the circular cone $K(\mathbf{d}_j, \alpha)$ of a m D joint represented in a higher dimensional space (i.e., represent translational motion cone in the rigid motion space). In summary, the kinematic design stage converts the two

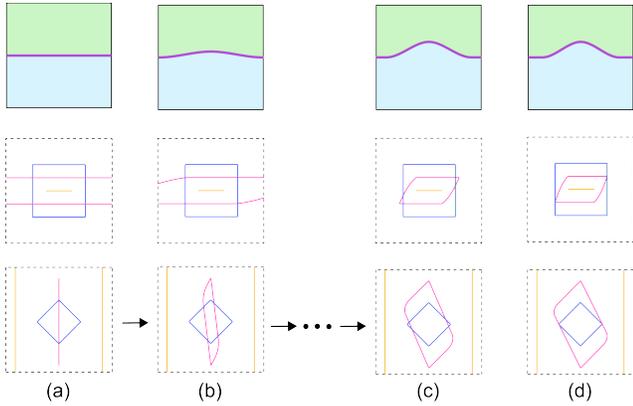


Fig. 13. Geometric realization process for a single joint. From left to right, initial planar contact, intermediate joints, and resulting joint. From top to bottom, joints represented in geometric space, motion space, and generalized normal space, respectively. The conic sections of the required joint motion cones are shown in blue, the required disassemblability cones in orange, and the actual joint motion cones in magenta.

design goals into a set of local requirements for each cone joint $J_{i,j}$ to be constructed at the geometric realization stage:

$$\hat{K}(-\mathbf{d}_j, \alpha) \subseteq \mathbf{V}_{i,j} \subseteq \bar{\mathbf{V}}_{i,j}, \quad \text{for each } J_{i,j} \quad (14)$$

6.3 Geometric Realization

Geometric realization aims to construct the geometry of each cone joint to satisfy the constraint on the joint motion cone $\mathbf{V}_{i,j}$ in Equation 14. According to Theorem 3.2, $\mathbf{V}_{i,j}$ can be approximated by sampling the joint's generalized normal space $\tilde{\mathbf{N}}_{i,j} = \{\mathbf{n}_l(\Phi_{i,j})\}$ and computing the dual cone, i.e., $\mathbf{V}_{i,j} \approx \tilde{\mathbf{N}}_{i,j}^*$.

To achieve our goal, we could optimize the joint shape (i.e., search the joint design parameters $\Phi_{i,j}$) such that the joint motion cone $\mathbf{V}_{i,j}$ satisfies the constraint in Equation 14. However, computing the joint motion cone is already very time-consuming, let alone doing shape optimization for it. A better way is to perform the shape optimization in the generalized normal space $\tilde{\mathbf{N}}_{i,j}$, which is directly controlled by the joint parameters $\Phi_{i,j}$. Lemma 3.3 provides a strategy to transform the constraints from the motion space (see Equation 14) to the generalized normal space:

$$\bar{\mathbf{V}}_{i,j}^* \subseteq \text{cone}(\mathbf{N}_{i,j}) \subseteq \hat{K}(-\mathbf{d}_j, \alpha)^* \quad (15)$$

in which the direction of the subset symbol " \subseteq " is reversed due to the dual operator.

In particular, the dual of the required joint motion cone $\bar{\mathbf{V}}_{i,j}^*$, a polyhedral cone, is just the minimum convex cone envelope of the polyhedral cone's face normals $\{\mathbf{f}_k(\Psi_{i,j})\}$:

$$\bar{\mathbf{V}}_{i,j}^* = \text{cone}(\{\mathbf{f}_k(\Psi_{i,j})\}) \quad (16)$$

And the dual of the circular cone $\hat{K}(-\mathbf{d}_j, \alpha)$ is still a circular cone, just with a cone angle $\frac{\pi}{2} - \alpha$:

$$\begin{aligned} \hat{K}(-\mathbf{d}_j, \alpha)^* &= (K(-\mathbf{d}_j, \alpha) \times \{\mathbf{0}\}^{2m-3})^* \\ &= K(-\mathbf{d}_j, \frac{\pi}{2} - \alpha) \times \mathbb{R}^{2m-3} \end{aligned} \quad (17)$$

We illustrate the relation between the dual cones using their conic sections for 2D assemblies in Figure 13. The required disassemblability cone is an infinitely long strip, and the required joint motion cone is the minimum convex cone envelope of face normals $\{\mathbf{f}_k\}$. The minimum convex cone envelope of $\mathbf{N}_{i,j}$ has to stay in between the conic sections of these two required cones. To this end, we formulate our joint shape optimization as an energy minimization problem:

$$\begin{aligned} \min_{\Phi_{i,j}} \quad & \sum_{l,k} \text{dist}(\text{cone}(\{\mathbf{n}_l\}), \mathbf{f}_k) \\ \text{s.t.} \quad & \mathbf{n}_l \in K(-\mathbf{d}_j, \frac{\pi}{2} - \alpha) \times \mathbb{R}^{2m-3} \end{aligned} \quad (18)$$

where $\text{dist}(\text{cone}(\{\mathbf{n}_l\}), \mathbf{f}_k)$ is the distance between the minimum convex cone envelope $\text{cone}(\{\mathbf{n}_l\})$ and the face normal \mathbf{f}_k . In particular, $\text{dist}(\text{cone}(\{\mathbf{n}_l\}), \mathbf{f}_k) = 0$ when the face normal \mathbf{f}_k is inside the $\text{cone}(\{\mathbf{n}_l\})$. We compute the distance using quadratic programming:

$$\text{dist}(\text{cone}(\{\mathbf{n}_l\}), \mathbf{f}_k) = \min_{\lambda_l \geq 0} \|\mathbf{f}_k - \sum_l \lambda_l \mathbf{n}_l\|^2 \quad (19)$$

One additional constraint is to maintain a certain distance between each joint $J_{i,j}$ and the boundary of parts P_i and P_j to ensure validity and structural soundness of these two parts; see again Figure 10(c&d). To this end, we combine the shape of parts P_i and P_j , and compute a signed distance function to the shape boundary; see Figure 14(b&d). The constraint is satisfied by requiring the distance value of each point on the joint $J_{i,j}$ to be larger than a threshold; see Figure 14.

The initial values of joint parameters $\Phi_{i,j}$ can profoundly influence the result. Fortunately, the geometric realization is performed independently among the joints (see again Figure 14), and each joint optimization problem only has a few variables. Hence, for each joint, we try as many initial values as possible to avoid local minima by uniformly sampling the variables. Similar to the optimization at the kinematic design stage, the joint optimization at the geometric realization stage can be solved by an off-the-shelf interior-point method.

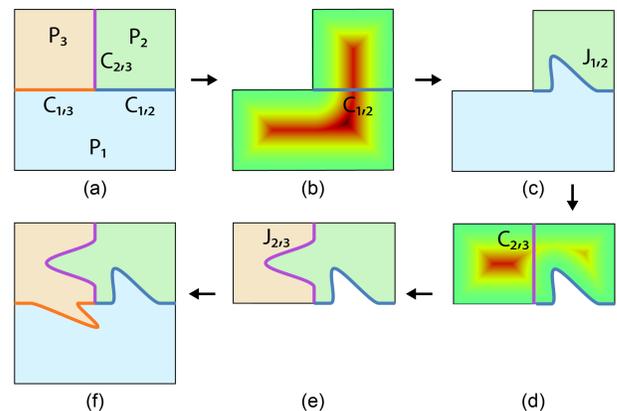


Fig. 14. Geometric realization process on a 3-part assembly. Starting from (a) an initial assembly, we (b) compute a distance function for the boundary of parts P_1 and P_2 to (c) construct joint $J_{1,2}$. Next, we (d) compute a distance function for the boundary of parts P_2 and P_3 to (e) construct joint $J_{2,3}$. (f) The resulting assembly. In (b&d) the distance functions, green and red indicate small and large distance values, respectively.

In practice, we need to iterate between the kinematic design and geometric realization stages several times. This is because each part's centroid and weight (i.e., volume) are assumed to be fixed at the kinematic design stage, yet are subject to change after joints are constructed in the geometric realization stage. To alleviate this issue, our approach prefers not to have dramatic changes on the joint geometry at each iteration. Occasionally, the geometric realization stage may not be able to find any joint geometry that satisfies the required motion cone computed at the kinematic design stage. To this end, updating the lower bound for the required joint motion cones according to the current joint geometry (i.e., the new bound is the same as the joint motion cone) would facilitate the search of joint geometry in the next iteration.

7 RESULTS

We implemented our tool in C++ and libigl [Jacobson et al. 2018], and employed Knitro [Artelys 2020] for solving our optimizations. We conducted all experiments on a Linux workstation with an AMD Ryzen Threadripper 3990X 64-Core Processor and 128GB of RAM. We show that our approach can handle assemblies with a variety of geometric forms, including planar, volumetric, frame, and shell structures; see Figure 1. Thanks to the flexibility offered by our cone joints, our approach can compute assemblies that are stable to different degrees, e.g., *equilibrium* under gravity, stable under *lateral* forces, and single-key *interlocking*. Our approach can also consider stability not just for the final assembly but also for intermediate stages of the assembly process. By this, we can generate *support-free* structures that can be assembled without using any support.

Statistics. Table 1 summarizes the statistics of all the results presented in the paper. The third to seventh columns show if the result is 2D or 3D, the total number of parts, the total number of joints, angle α of the required assemblability cone, one of the four features mentioned above, and optimization time, respectively. The angle α is typically set as 5 degrees, which is sufficient for inserting parts

Table 1. Statistics of the results shown in the paper.

Fig	Result	2D/3D	# Part	# Joint	α	Feature	Optim. Time (min)
1	M	2D	10	9	5	Equilibrium	1.00
	Horse	3D	13	9	5	Equilibrium	9.82
	Pavilion	3D	48	62	5	Equilibrium	2.19
	Lilium Tower	3D	139	325	5	Equilibrium	203.78
15(rig)	Scarecrow	2D	4	5	25	Equilibrium	0.08
16(c)	Sphere	3D	6	12	3	Interlocking	2.98
17	Stack	2D	10	10	-	Equilibrium	0.04
18	Tree	2D	7	9	10	Equilibrium	0.37
19	Leaning Tower (lef)	2D	14	28	5	Equilibrium	0.23
	Leaning Tower (mid)	2D	14	28	5	Equilibrium	0.45
	Leaning Tower (rig)	2D	13	27	5	Equilibrium	0.38
20	Leaning Tower	2D	14	28	5	Support-free	0.68
21	Deer	2D	14	23	5	Support-free	0.70
22	Igloo	3D	139	326	5	Laterally stable	61.11

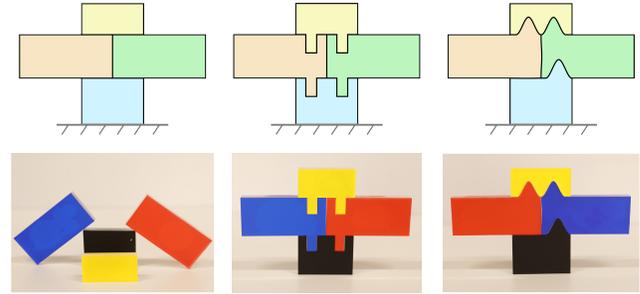


Fig. 15. Comparison of stability and assemblability of 4-part SCARECROWS with (left) planar contacts, (middle) single-direction joints, and (right) our optimized cone joints.

easily. We set a larger α for making the cone joints less sharp (Figures 15(right) and 18) and a smaller α for making the assembly more stable (Figure 16(c)). We can also not consider assemblability as a design goal by not setting any value for angle α (Figure 17). Our approach typically takes less than 1 minute to generate 2D results yet may take hours for 3D results due to the larger number of joints as well as more variables to define the geometry of each joint.

Evaluation of cone joints. We compare our optimized cone joints with planar contacts and single-direction joints by designing three 2D puzzles; see Figure 15. To verify stability and assemblability of these puzzles, we fabricate them with laser-cutting. Our experiment shows that the puzzle with planar contacts is not in equilibrium under gravity; see Figure 15(left). To assemble the puzzle with single-direction joints, we need to align each part with the partially completed puzzle carefully before it can be successfully inserted; see Figure 15(middle). Compared with these two puzzles, our puzzle with cone joints avoids the two issues by achieving a good balance between stability and assemblability; see Figure 15(right). Please watch the accompanying video for demos.

We compare our cone joints with planar contacts and standard mortise-and-tenon joints by designing four 6-part SPHERES. The sphere with planar contacts cannot be in equilibrium under gravity, and the sphere with standard mortise-and-tenon joints are deadlocking; see Figure 16(a&b). We show that our approach can make the sphere single-key interlocking by constructing cone joints; see Figure 16(c). This is achieved by applying an external force configuration on the assembly, where at each contact the external force tries to push the two associated parts and separate them. The assembly is single-key interlocking if it passes our equilibrium test under this external force configuration, assuming the key is held by other

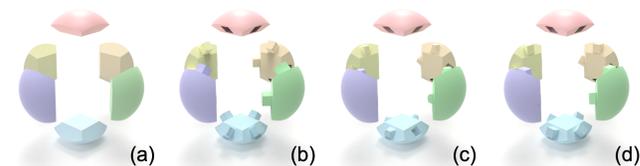


Fig. 16. Comparison of stability and assemblability of 6-part SPHERES with (a) planar contacts, (b) standard mortise-and-tenon joints, (c) our optimized cone joints, and (d) tilted mortise-and-tenon joints derived from our result. The assembly in (a) is not stable while the one in (b) is deadlocking.

means; see the supplementary material for a proof. Our design result can be easily modified to derive some variants, e.g., to facilitate fabrication. For example, we can replace each curved-contact joint with a tilted mortise-and-tenon joint based on the principal direction u of the joint; see Figures 16(d) and 7. The resulting assembly is guaranteed to be interlocking since a tilted mortise-and-tenon joint has a smaller motion cone than any curved-contact joint.

Evaluation of our optimization. To show the benefits of our design approach, we compare it with a baseline approach that directly searches parameters of cone joints using a gradient-based method, similar to [Whiting et al. 2012]. Both approaches take the same assembly with planar contacts as input (Figure 17(a)), and generate curved-contact joints to make the assembly be in equilibrium (Figure 17(b&c)). Figure 17(d) shows the computation time of the two approaches with respect to the number of parts. When the number of parts is small, the computation time of the two approaches are comparable. However, the computation time of the baseline increases dramatically when the number of parts is more than 16 since the computation cost becomes dominated by the equilibrium test, which is expensive for assemblies with many curved-contact joints. Thanks to the kinematic design stage, our approach avoids executing the expensive equilibrium test frequently whenever changing joints geometry. During the experiment, we find that the baseline approach sometimes can find better solutions than ours, especially when the number of parts is small. For example, when there are only two parts where the bottom one is fixed, the baseline creates a cone joint that makes the assembly in equilibrium yet our approach fails to do so; see Figure 17(e). This is because our kinematic design stage

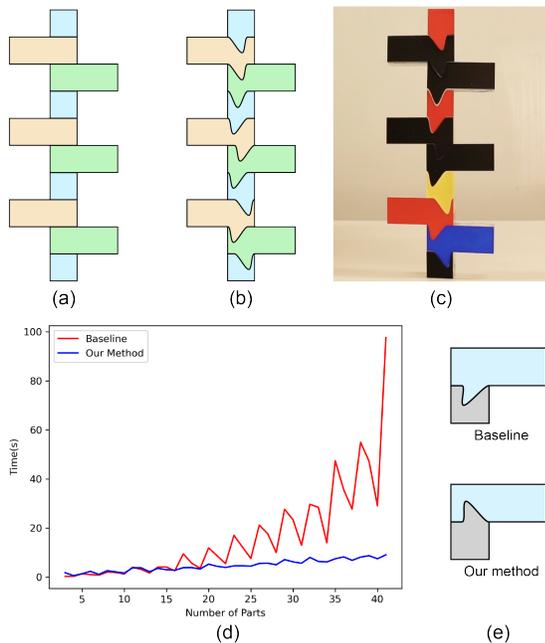


Fig. 17. Comparison of our design approach with a baseline approach: (a) an example input assembly; (b&c) a result generated by our approach; (d) computation time of the two approaches with respect to the number of parts; and (e) an example case where the baseline performs better.

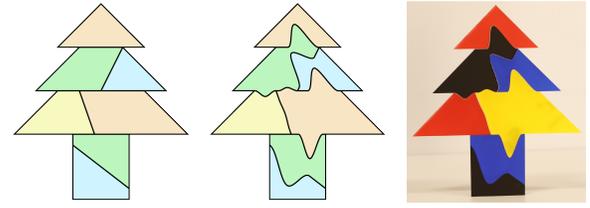


Fig. 18. Equilibrium puzzle TREE generated by our approach. From left to right: input assembly, our result, and laser-cut puzzle.

assumes the parts' centroids are fixed, which actually may change after joints have been constructed in the geometric realization stage.

Equilibrium puzzles. We have used our approach to generate 2D/3D equilibrium puzzles. Figure 1(a&b) shows two puzzles, M and HORSE, which cannot be in equilibrium if we simply use planar contacts. This is because planar contacts cannot prevent the motion (e.g., sliding) of some parts caused by gravity such as the two parts at the middle of M and the head part of HORSE. Figure 18 shows a puzzle TREE as well as the fabricated result. Our designed cone joints not only make the puzzle stable under gravity, but also provide a hint to find puzzle pieces that should match with one another.

Given input assemblies that are far from an equilibrium state, our approach can still make it stable. To demonstrate this, we create a sequence of input assemblies, where parts have the same shape (except bottom ones) but have been tilted for a certain angle; see Figure 19(left) for an example input. Due to the parts arrangement, the larger the tilt angle is, the further the assembly is from an equilibrium state. We construct cone joints for each input assembly to make it be in equilibrium using our approach; see Figure 19 for three example results. We observe that our optimized cone joints become more and more sharp when the tilt angle increases since a sharp joint corresponds to a small motion cone. A drawback of these sharp cone joints is that they may fail due to stress concentration. To alleviate this issue, additional constraints on the joint shape can be enforced during the geometric realization stage.

Our approach can be extended to design support-free puzzles; see Figures 20 and 21. These puzzles are in equilibrium for each intermediate assembly state, and thus can be assembled without using any support. To generate these puzzles, we assume a predefined assembly sequence, and take all the intermediate assemblies and the final assembly as input of our kinematic design. We obtain the required joint motion cones by summing the infeasibility energy in Equation 13 for all the input assemblies and solving the optimization. The geometric realization is performed without any change.

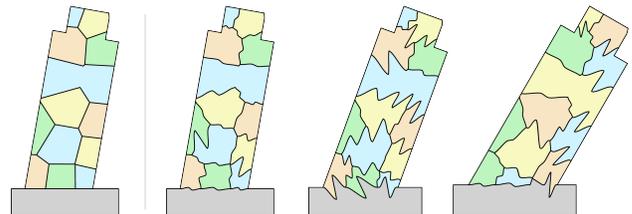


Fig. 19. Tilt experiment on (left) an input LEANING TOWER to verify the ability of our designed cone joints to make an assembly stable. From left to right: the tilt angles of the three results are 10, 20, and 30 degrees, respectively.

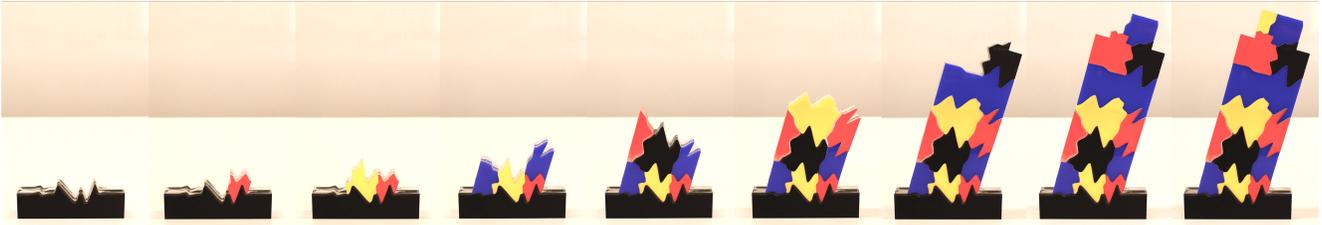


Fig. 20. Support-free equilibrium puzzle LEANING TOWER with 15 degrees tilt angle (see also Figure 19).

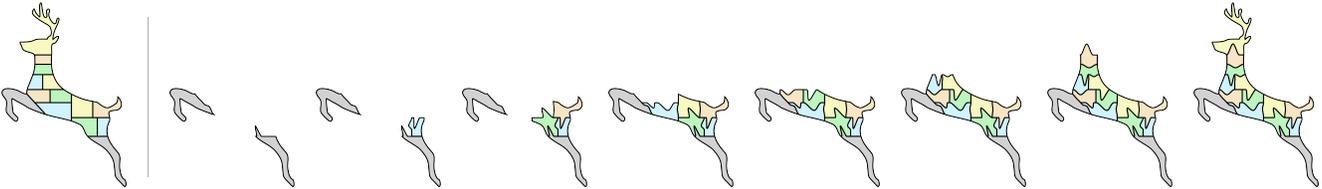


Fig. 21. Support-free equilibrium puzzle DEER. The input assembly is shown on the left and parts that are fixed during the assembly process are colored in gray.



Fig. 22. An IGLoo shell with lateral stability designed by our approach.

Shell structures. Shell structures with planar contacts can be used as masonries in architecture. These structures are *self-supporting* if they can be in equilibrium under gravity [Panozzo et al. 2013]. Shell structures with an inverted bump on the top usually cannot be self-supporting; see LILIUM TOWER in Figure 1(d). Taking such a structure as input, our approach can make it self-supporting by creating cone joints for some of the planar contacts. Our approach only modifies a small subset of the planar contacts since our gradient-based optimization changes contact geometry only if it can reduce the infeasibility energy in Equation 13.

In some applications (e.g., architecture), a shell structure should be in equilibrium under not just gravity but also lateral forces (e.g., from wind). This lateral stability can be evaluated by simulating a tilt analysis experiment, where the ground plane of the structure is tilted to apply a lateral force to the structure caused by gravity. The lateral stability is measured by the largest angle of tilting the ground plane without collapse of the structure [Wang et al. 2019]. Our approach can generate shell structures with lateral stability by a slight change on the kinematic design stage. In detail, we sum the infeasibility energy E in Equation 13 for multiple external force configurations (i.e., gravity, and lateral forces from different directions). The assembly is considered as laterally stable if it is in equilibrium under any of the external force configurations. Figure 22 shows a shell structure IGLoo that cannot be in equilibrium under gravity if only planar contacts are used. By using our cone joints, we show that the structure can be tilted for at most 35 degrees.

Frame structures. Frame structures composed of rod-like parts are widely used in architecture and furniture. In these structures, each part contacts others at its two ends and each contact area is usually quite small. Our approach can optimize cone joints to make these structures stable. The resulting cone joints look similar to single-direction joints; see PAVILION in Figure 1(c). This is perhaps because these structures with planar contacts are far away from an equilibrium state, and thus joints with a small motion cone are needed to make them stable.

8 CONCLUSION

Assemblability and stability are two necessary conditions for using assemblies in the physical world. However, they could be in conflict with each other when restricting relative part motion with joints. Finding a trade-off between these two conditions is a challenging task for conventional approaches based on planar contacts or single-direction joints. We propose cone joints to address this challenge, which interpolate between planar contacts and single-direction joints in terms of capacity to restrict relative part motion. We quantify this capacity as a motion cone, and present an approach to optimize cone joints for designing structures that are assemblable and stable. We found the separation into kinematic and geometric stages essential to make the optimization computationally tractable. We show versatility of our approach by designing a variety of 2D/3D assemblies that are in equilibrium under gravity, stable under lateral forces, interlocking, or support-free for the assembly process.

Limitations and Future Work. Our work has several limitations that open up interesting directions for future research. First, the geometry of 2D/3D cone joints that can be supported by our current parametric model is quite limited. Exploring more complex parametric models and studying their impact on the motion cones would be an interesting future work. Second, we plan to include friction in our motion-based equilibrium method as well as the design approach. Third, our geometric realization stage focuses on generating geometry of cone joints to match optimized motion cones. Taking other aspects such as appearance and structural soundness of joints into

consideration would make the designed cone joints more practical for use. Lastly, we approximate the required motion cones in the kinematic design stage as a pyramid with 4 (10) faces for 2D (3D) joints to speed up the static analysis. In the future, we may try more accurate approximations to see if they are helpful to improve the design performance.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments, Tian Chen for fabricating the physical models, Héloïse Dupont de Dinechin and Mingyang Li for implementing some experiment code. This research was supported by the NCCR Digital Fabrication, funded by the Swiss National Science Foundation (NCCR Digital Fabrication Agreement #51NF40-141853), the TL@SUTD Seed Research Project Grant (Number: RTDS S 1907 011), and the SUTD Start-up Research Grant (Number: SRG ISTD 2019 148).

REFERENCES

- Ergun Akleman, Vinayak R. Krishnamurthy, Chia-An Fu, Sai Ganesh Subramanian, Matthew Ebert, Matthew Eng, Courtney Starrett, and Haard Panchal. 2020. Generalized Abeille Tiles: Topologically Interlocked Space-filling Shapes Generated Based on Fabric Symmetries. *Comp. & Graph. (SMI)* 89 (2020), 156–166.
- Christiano Araújo, Daniela Cabiddu, Marco Attene, Marco Livesu, Nicholas Vining, and Alla Sheffer. 2019. Surface2Volume: Surface Segmentation Conforming Assemblable Volumetric Partition. *ACM Trans. on Graph. (SIGGRAPH)* 38, 4 (2019), 1:1–1:16.
- Artelys. 2020. Knitro software package. <https://www.artelys.com/solvers/knitro/>.
- Philippe Block and John Ochsendorf. 2007. 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.
- Stephen Boyd and Lieven Vandenbergh. 2004. *Convex Optimization*. Cambridge University Press.
- Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. 2014. Field-aligned Mesh Joinery. *ACM Trans. on Graph.* 33, 1 (2014), 11:1–11:12.
- Ruta Desai, James McCann, and Stelian Coros. 2018. Assembly-aware Design of Printable Electromechanical Devices. In *Proc. ACM UIST*. 457–472.
- Mario Deuss, Daniele Panozzo, Emily Whiting, Yang Liu, Philippe Block, Olga Sorkine-Hornung, and Mark Pauly. 2014. Assembling Self-Supporting Structures. *ACM Trans. on Graph. (SIGGRAPH Asia)* 33, 6 (2014), 214:1–214:10.
- Simon Duenster, Roi Poranne, Bernhard Thomaszewski, and Stelian Coros. 2020. Robo-Cut: Hot-wire Cutting with Robot-controlled Flexible Rods. *ACM Trans. on Graph. (SIGGRAPH)* 39, 4 (2020), 98:1–98:15.
- A. V. Dyskin, Yuri Estrin, and E. Pasternak. 2019. Topological Interlocking Materials. In *Architected Materials in Nature and Engineering*, Yuri Estrin, Yves Bréchet, John Dunlop, and Peter Fratzl (Eds.). Springer International Publishing, Chapter 2, 23–49.
- A. V. Dyskin, Y. Estrin, E. Pasternak, H. C. Khor, and A. J. Kanel-Belov. 2003. Fracture Resistant Structures Based on Topological Interlocking with Non-planar Contacts. *Advanced Engineering Materials* 5, 3 (2003), 116–119.
- William Fairham. 2013. *Woodwork Joints: How to Make and Where to Use Them*. Skyhorse.
- Giuseppe Fallacara, Maurizio Barberio, and Micaela Colella. 2019. Topological Interlocking Blocks for Architecture: From Flat to Curved Morphologies. In *Architected Materials in Nature and Engineering*, Yuri Estrin, Yves Bréchet, John Dunlop, and Peter Fratzl (Eds.). Springer International Publishing, Chapter 14, 423–445.
- Julius Farkas. 1902. Theorie der Einfachen Ungleichungen. *Journal für die Reine und Angewandte Mathematik* 124 (1902), 1–27.
- Chi-Wing Fu, Peng Song, Xiaoqi Yan, Lee Wei Yang, Pradeep Kumar Jayaraman, and Daniel Cohen-Or. 2015. Computational Interlocking Furniture Assembly. *ACM Trans. on Graph. (SIGGRAPH)* 34, 4 (2015), 91:1–91:11.
- Somayé Ghandi and Ellips Masehian. 2015. Review and Taxonomies of Assembly and Disassembly Path Planning Problems and Approaches. *Computer-Aided Design* 67–68 (2015), 58–86.
- Jingbin Hao, Liang Fang, and Robert E. Williams. 2011. An Efficient Curvature-based Partitioning of Large-scale STL Models. *Rapid Prototyping Journal* 17, 2 (2011), 116–127.
- Kristian Hildebrand, Bernd Bickel, and Marc Alexa. 2012. crdbd: Shape Fabrication by Sliding Planar Slices. *Comp. Graph. Forum (Eurographics)* 31, 2 (2012), 583–592.
- Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>.
- Anooshe Rezaee Javan, Hamed Seifi, Shanqing Xu, and Yi Min Xie. 2016. Design of A New Type of Interlocking Brick and Evaluation of Its Dynamic Performance. In *Proceedings of the International Association for Shell and Spatial Structures Annual Symposium*. 1–8.
- Gene T.C. Kao, Axel Körner, Daniel Sonntag, Long Nguyen, Achim Menges, and Jan Knippers. 2017. Assembly-aware Design of Masonry Shell Structures: A Computational Approach. In *Proceedings of the International Association for Shell and Spatial Structures Symposium*.
- Lydia Kavradi, Jean-Claude Latombe, and Randall H. Wilson. 1993. On the Complexity of Assembly Partitioning. *Inform. Process. Lett.* 48, 5 (1993), 229–235.
- Vinayak R. Krishnamurthy, Ergun Akleman, Sai Ganesh Subramanian, Katherine Boyd, Chia-An Fu, Matthew Ebert, Courtney Starrett, and Neeraj Yadav. 2020. Bi-Axial Woven Tiles: Interlocking Space-Filling Shapes Based on Symmetries of Bi-Axial Weaving Patterns. In *Proc. Graphics Interface*. 286–298.
- Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. 2020. Tsugite: Interactive Design and Fabrication of Wood Joints. In *Proc. ACM UIST*. 317–327.
- Kui-Yip Lo, Chi-Wing Fu, and Hongwei Li. 2009. 3D Polyomino Puzzle. *ACM Trans. on Graph. (SIGGRAPH Asia)* 28, 5 (2009), 157:1–157:8.
- Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. 2012. Chopper: Partitioning Models into 3D-printable Parts. *ACM Trans. on Graph. (SIGGRAPH Asia)* 31, 6 (2012), 129:1–129:9.
- Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. 2015. Legolization: Optimizing LEGO Designs. *ACM Trans. on Graph. (SIGGRAPH Asia)* 34, 6 (2015), 222:1–222:12.
- Renato D. C. Monteiro and Ilan Adler. 1989. Interior Path Following Primal-dual Algorithms. Part II: Convex Quadratic Programming. *Mathematical Programming* 44 (1989), 43–66.
- Daniele Panozzo, Philippe Block, and Olga Sorkine-Hornung. 2013. Designing Unreinforced Masonry Models. *ACM Trans. on Graph. (SIGGRAPH)* 32, 4 (2013), 91:1–91:11.
- Yuliy Schwartzburg and Mark Pauly. 2013. Fabrication-aware Design with Intersecting Planar Pieces. *Comp. Graph. Forum (Eurographics)* 32, 2 (2013), 317–326.
- Hijung V. Shin, Christopher F. Porst, Etienne Vouga, John Ochsendorf, and Frédo Durand. 2016. Reconciling Elastic and Equilibrium Methods for Static Analysis. *ACM Trans. on Graph.* 35, 2 (2016), 13:1–13:16.
- Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. 2016. CofFab: Coarse-to-Fine Fabrication of Large 3D Objects. *ACM Trans. on Graph. (SIGGRAPH)* 35, 4 (2016), 45:1–45:11.
- Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. 2012. Recursive Interlocking Puzzles. *ACM Trans. on Graph. (SIGGRAPH Asia)* 31, 6 (2012), 128:1–128:10.
- Peng Song, Chi-Wing Fu, Yueming Jin, Hongfei Xu, Ligang Liu, Pheng-Ann Heng, and Daniel Cohen-Or. 2017. Reconfigurable Interlocking Furniture. *ACM Trans. on Graph. (SIGGRAPH Asia)* 36, 6 (2017), 174:1–174:14.
- Sai Ganesh Subramanian, Mathew Eng, Vinayak R. Krishnamurthy, and Ergun Akleman. 2019. Delaunay Lofts: A Biologically Inspired Approach for Modeling Space Filling Modular Structures. *Comp. & Graph. (SMI)* 82 (2019), 73–83.
- Ziqi Wang, Peng Song, Florin Isvoranu, and Mark Pauly. 2019. Design and Structural Optimization of Topological Interlocking Assemblies. *ACM Trans. on Graph. (SIGGRAPH Asia)* 38, 6 (2019), 193:1–193:13.
- Ziqi Wang, Peng Song, and Mark Pauly. 2018. DESLA: A General Framework for Designing Interlocking Assemblies. *ACM Trans. on Graph. (SIGGRAPH Asia)* 37, 6 (2018), 191:1–191:14.
- Ziqi Wang, Peng Song, and Mark Pauly. 2021. State of the Art on Computational Design of Assemblies with Rigid Parts. *Comp. Graph. Forum (Eurographics)* 40, 2 (2021).
- Michael Weizmann, Oded Amir, and Yasha Jacob Grobman. 2017. 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.
- Emily Whiting, John Ochsendorf, and Frédo Durand. 2009. Procedural Modeling of Structurally-Sound Masonry Buildings. *ACM Trans. on Graph. (SIGGRAPH Asia)* 28, 5 (2009), 112:1–112:9.
- Emily Whiting, Hijung Shin, Robert Wang, John Ochsendorf, and Frédo Durand. 2012. Structural Optimization of 3D Masonry Buildings. *ACM Trans. on Graph. (SIGGRAPH Asia)* 31, 6 (2012), 159:1–159:11.
- Randall H. Wilson and Toshihiro Matsui. 1992. Partitioning an Assembly for Infinitesimal Motions in Translation and Rotation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. 1311–1318.
- Wenzheng Xu, Xiaoshan Lin, and Yi Min Xie. 2020. A Novel Non-planar Interlocking Element for Tubular Structures. *Tunnelling and Underground Space Technology* 103 (2020). Article No. 103503.
- Jiaxian Yao, Danny M. Kaufman, Yotam Gingold, and Maneesh Agrawala. 2017b. Interactive Design and Stability Analysis of Decorative Joinery for Furniture. *ACM Trans. on Graph.* 36, 2 (2017), 20:1–20:16.
- Miaojun Yao, Zhili Chen, Weiwei Xu, and Huamin Wang. 2017a. Modeling, Evaluation and Optimization of Interlocking Shell Pieces. *Comp. Graph. Forum (Pacific Graphics)* 36, 7 (2017), 1–13.
- Yinan Zhang, Yotto Koga, and Devin Balkcom. 2021. Interlocking Block Assembly With Robots. *IEEE Transactions on Automation Science and Engineering* (2021). online article.