Mind the Gap: Flexibility Analysis of Interlocking Rigid Structures

Zezhou Sun[1][0009−0001−5883−8733], Megha Venkatesam[1], Devin Balkcom[2][0000−0002−6553−5650], and Emily Whiting[1][0000−0001−7997−1675]

[1] Boston University, Boston, MA 02215 USA
[2] Dartmouth College, Hanover, NH 03755 USA

Abstract. This work addresses the challenge of analyzing interlocking structures composed of discrete elements for applications such as manufacturing tolerances. While traditional computational methods assume perfect geometry and tight contact, real-world assemblies exhibit small gaps due to fabrication tolerances, leading to flexibility. Such kinematic freedom can compromise interlocking integrity not captured in idealized models. We develop a scalable pairwise formulation for approximating relative configuration spaces between elements. This formulation is integrated into a non-linear optimization framework to efficiently evaluate structural flexibility in large-scale interlocking assemblies.

Keywords: Fabrication· Optimization · Shape Analysis · Fabrication & Material Design

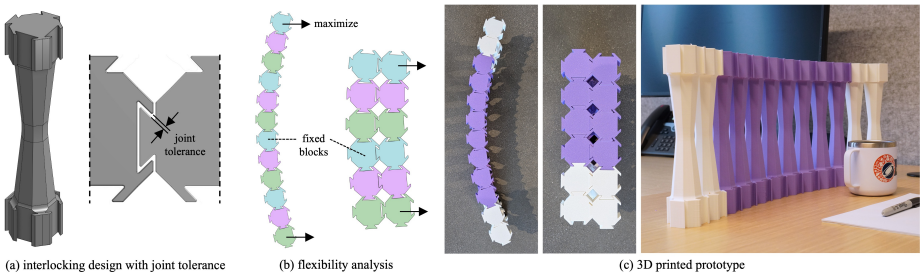


Fig. 1: Our method analyzes the flexibility of systems of interlocking rigid structures. In our approach we consider tolerances (gaps) in the joint design (a), and generate the configuration space to assess possible motion in the assembled structure (b). Here we show an application of a modular space divider (c), where our approach can predict flexibility for different arrangements of blocks (b). See Section 6.3 for discussion.

1 Introduction

The design of structures composed of discrete elements, such as blocks or polygons, is highly relevant across various fields, including architecture, mechanical engineering, robotics, and computational design. A core principle in such designs is *interlocking*, where the geometric arrangement of components prevents disassembly and ensures structural integrity.

Most computational methods for analyzing interlocking structures assume perfect geometry and tight contact [29,32,15]. This idealization often neglects the unavoidable presence of manufacturing tolerances and necessary clearances designed to accommodate fabrication errors and facilitate ease of assembly.

In practice, however, tolerances are common and introduce small gaps, allowing local motion through slight translations and rotations. Aggregated, such small movements can lead to global flexibility. Structures that appear rigid under idealized assumptions may fall apart when tolerances are considered [23,16]. This is particularly critical in mechanical design, where interlocking ensures bounded motion and doesn't rely on strength through attachments like screws or glue. Verifying robustness under these real-world conditions is essential but computationally heavy.

The core of our approach is a convex approximation of the configuration space for interlocking joints that is, connections whose geometry prevents two parts from being separated under any relative motion, constructed by stacking convex θ-slices. This representation allows block interactions to be encoded directly as constraints in an optimization framework to estimate overall structural displacement, while avoiding the repeated geometric computations required by signed distance field–based methods [20]. One important application of this work is to identify critical regions where precision has the greatest impact, helping designers balance fabrication cost and structural robustness. Our method scales efficiently while maintaining high precision, making it well-suited for analyzing large-scale assemblies of interlocking structures.

2 Related Work

Self-Locking Assemblies Structures composed of rigid components connected through joints constrained by geometry and gravity have been studied across various domains. Prior work has explored fabrication-aware assembly generation, tolerance analysis, and optimization for manufacturable structures [11,5,34]. Interlocking components have a long history of application in architecture, such as topological interlocking shell structures [29] and jointed timber networks [26,15,34]. Previous research has explored the design and generation of interlocking joints from 3D meshes, particularly for furniture or shapes created through interactive user input [32,15,13,1]. Additionally, some studies have extended the use of interlocking joints to plate-based structures [30,7,22].

Song et al. [25] introduced a more strict structure, where the last few pieces serve as the key for the whole interlocking assembly and all pieces stay in position with no relative movement between their neighbors without the help from

gravity. This idea was further developed using directed blocking graphs to represent interlocking puzzles, facilitating both disassembly planning and automated puzzle generation [30,6]. An infinitesimal motion-based analysis was introduced for interlock checking by Wang et al. [29].

Undesired gapping between parts can cause flexibility or failure in interlocking. This problem becomes particularly challenging for large-scale assemblies. The Puzzleflex system by Lensgraf et al. [16] proposed a half-plane method to linearize tolerances and formulated the problem as a linear program to handle large-scale systems. This approach enables flexibility analysis under interlocking conditions. However, because the method relies on a first-order approximation of rotation, it has higher approximation errors when global rotations are large.

However, tolerances are not always harmful. Some works [27,21] intentionally leverage them to create chainmail structures with nonlinear mechanical properties that are flexible under normal use, but rigid under extreme deformation.

Dynamics and Simulation Prior work has explored rigid-body simulation with collision handling [2,12,4,19] and penetration-free dynamics using incremental potential methods [17,10,14]. However, these approaches do not directly address interlocking structures, where disassembly paths and separation forces are often unknown, and simulation can be computationally expensive. Tian et al. [28] proposed a physics-based planner that partially addresses this problem.

Configuration Space and Minkowski Sum Free configuration space (c-space) represents all feasible states of a system. Lozano-Pérez [18] laid the foundational work for using c-space in motion planning by representing obstacles in a transformed space. Davis et al. [9] later explored the impact of kinematic tolerances on the topology of configuration space. Zhang et al. [33] applied machine learning techniques to identify connections in c-space for solving complex rigid-body disentanglement puzzles.

To compute free configuration space (c-space), the Minkowski sum is commonly used to identify and exclude invalid regions. Minarcik et al. [20] introduced a signed distance field representation based on the Minkowski sum of two polygons, which was applied to complex shape layout problems. This approach inspired our use of the Minkowski sum in the present work. Extending to 3D, Cui et al. [8] used a voxelized convolution-based approximation of the Minkowski sum for translational packing analysis. However, the method depends on voxel resolution and degrades for closely spaced parts, whereas our method accurately captures narrow gaps.

3 Background

3.1 Configuration Space

Free configuration space (c-space) is a core concept in robotics and motion planning, representing all possible system configurations. Each point in free c-space corresponds to a valid, collision-free state of the entire system. The dimension

of the c-space is determined by degrees of freedom, such as the number of independently controlled joint angles in a robotic arm.

In our investigation, we are considering a system of interlocked 2D polygonal rigid bodies. Each part can move freely in the plane except for a fixed base part, and valid configurations are states where no parts intersect. The c-space of a minimal two part system in our scenario has three degrees of freedom for translation and rotation (x, y, θ), which can be visualized as a 3D volume (see Fig. 2). For larger systems, such as the grid of blocks in Fig. 8a, the c-space will have dimension $3n$ where n is the number of freely moving blocks.

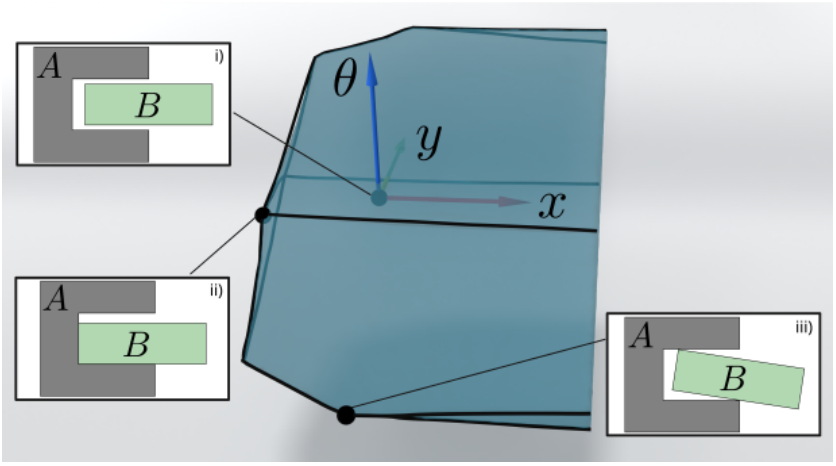


Fig. 2: Configuration space of block B relative to fixed block A. The 3D volume represents all valid intersection-free states for translations in (x, y) and rotations θ. Three sample configurations shown: i) origin: $(x, y, \theta) = (0, 0, 0)$. ii) Boundary of configuration space: maximum translation in $-x, -y$ direction before collision (for $\theta = 0$). iii) Boundary point involving rotation ($\theta \neq 0$).

3.2 Minkowski Sum

The Minkowski sum of two polygons A and B, denoted $A \oplus B$, is the set of all points $a + b$, where $a \in A$ and $b \in B$. Geometrically, it can be visualized by sweeping one shape around the boundary of the other (dashed polygons inside the gray area in Fig. 3).

In motion planning, the Minkowski sum $A \oplus (-B)$, where $-B$ is the reflection of B through the origin, defines the configuration space obstacle (C-obstacle) for B relative to a fixed object A, assuming translational motion only. The complement of this set is the free c-space of B. If B starts in a valid, non-overlapping configuration, the origin must lie within this free c-space. This space may include multiple disconnected regions, as shown in Fig. 3. No path from the

origin to the outside region means that B cannot be separated from A with pure translation motion.

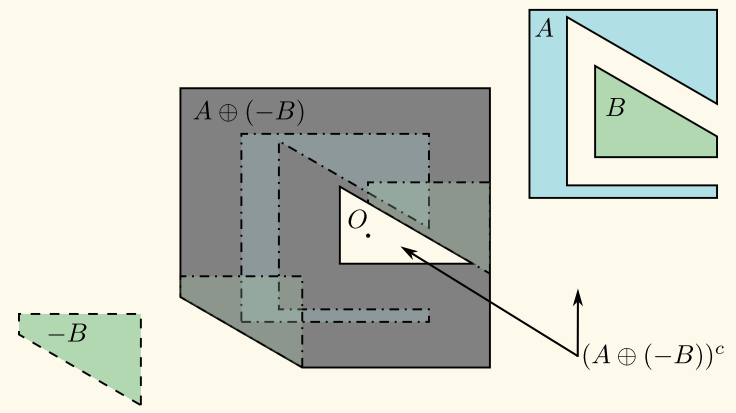


Fig. 3: The Minkowski sum between A and -B (dark grey) forms the space of *invalid* configurations for block B, where collision will occur. Note this considers only translations of B (no rotation). The complement $(A \oplus (-B))^c$ (light yellow) gives the free c-space for valid translations of B. The closed region containing the origin indicates the blocks are interlocked, B cannot escape.

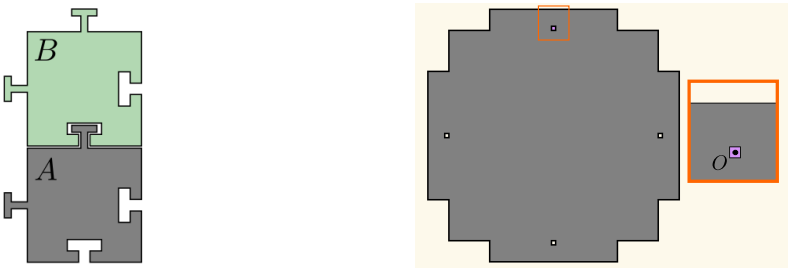
4 Constraints and Optimization

We show that the free c-space of interlocking joints with small tolerances can be approximated as convex, despite non-convex features in their geometry. We analyze convexity at fixed rotation angles, referred to as *θ-slices*, which are often individually convex under small tolerances. By stacking these slices, we approximate the free c-space and derive constraints for block interactions in optimization. A convex approximation of the free c-space allows us to encode block interactions as constraints, eliminating the need to compute Minkowski sums and query signed distance fields during each optimization iteration. Compared to signed distance field methods [20], our approach avoids repeated geometric computations, resulting in significantly higher efficiency.

4.1 C-space Slice Convexity

We begin with a simple two-block setup as shown in Fig. 4. Block A is fixed and block B translates in x, y (no rotation). We define the set $(A \oplus (-B))^c$, which contains valid configurations where A and B don't overlap.

This set covers more space than is needed to analyze the interlock. In $(A \oplus (-B))^c$, we are primarily concerned about the region near the origin, which



(a) (b)

Fig. 4: (a) T-Joint Blocks (b) Minkowski sum of A and $-B$, $A \oplus (-B)$ (grey), full free c-space (yellow & purple) and the shape that will be used to approximate the free c-space (purple) at this rotation slice.

reflects the configurations that can be reached from the current configuration of A and B.

For two polygons in close proximity, we first use a pair of line segments to establish a local supporting half-plane for the free C-space, then extend it to the polygonal setting to show that the configuration space of a polygon is convex under a bounded motion radius ϵ^*, derived below.

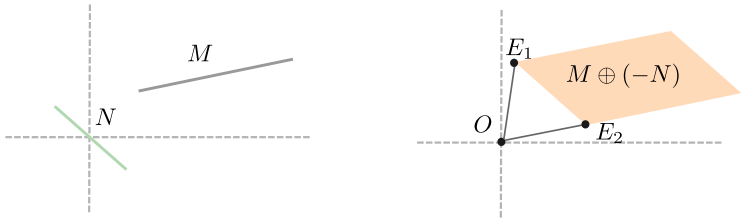


Fig. 5: Left: Two segments M and N, where M is fixed. Right: Configuration space Q(N) and Minkowski difference between M and N.

Lemma 1 (Local convexity of configuration space for a pair of line segments). *Consider two non-intersecting closed line segments, M and N, which can translate relative to one another. Without loss of generality, assume M is fixed in place, and let $Q(N)$ denote the configuration space of N with respect to M. In $Q(N)$, let E_1E_2 be the edge of the Minkowski sum boundary closest to the origin, with $\|OE_1\|$ and $\|OE_2\|$ denoting their respective distances to the origin O. Define*

$$\epsilon_{MN} = \min(\|OE_1\|, \|OE_2\|)\,.$$

Then M and N do not intersect for any translation $\mathbf{t}$ of N satisfying both:

1. $\|\mathbf{t}\| < \epsilon_{MN}$, *and*

2. $\mathbf{t}$ *lies in the open half-plane defined by the infinite line through* E_1E_2 *that contains the origin.*

Proof. The forbidden region in $Q(N)$ is the Minkowski sum $M \oplus (-N)$; a translation $\mathbf{t}$ is collision-free if and only if $\mathbf{t} \notin M \oplus (-N)$. The origin lies in the free region by assumption (since M and N are initially non-intersecting). The infinite line through E_1E_2 defines two open half-planes; the origin belongs to the free one. Any translation $\mathbf{t}$ remaining in this half-plane cannot cross E_1E_2 and therefore cannot enter the forbidden region through E_1E_2.

It remains to ensure that $\mathbf{t}$ does not reach the endpoints E_1 or E_2 and thereby enter the forbidden region through an adjacent boundary edge. Since $\|OE_1\|$ and $\|OE_2\|$ are the distances from the origin to these endpoints, any $\mathbf{t}$ with $\|\mathbf{t}\| < \epsilon_{MN} = \min(\|OE_1\|, \|OE_2\|)$ cannot reach either endpoint. Together, the two conditions guarantee that $\mathbf{t}$ stays within the free half-plane and never reaches E_1E_2 or its adjacent boundary, so M and N remain non-intersecting.

Next, we extend this result to the case of two polygons. Consider two non-overlapping polygons A and B and edge pairs (P_i, Q_j) where P_i is an edge of A and Q_j is an edge of B. Each such pair (P_i, Q_j) contributes one edge to the Minkowski sum boundary in the configuration space, and Lemma 1 applies to each pair independently, yielding a local ϵ_i.

The overall safe motion limit for the polygon pair is then $\epsilon^* = \min_{i=1,\ldots,k} \epsilon_i$. As long as the motion of B stays within an ϵ^*-ball around its initial configuration and within the intersection of the half-planes contributed by each edge pair, no edge pair can produce a collision, and the two polygons remain non-intersecting. Since both the ϵ^*-ball and the half-plane intersection are convex, their intersection is also convex. We then denote such an approximated space as our θ-slice, where θ is the current fixed rotation of B.

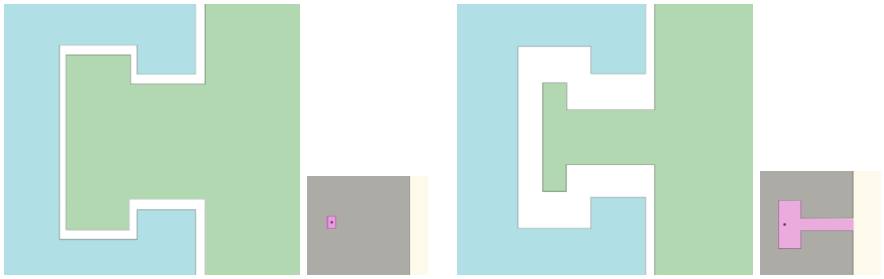


Fig. 6: (Left) When the distance between two polygons is small, the configuration space (the pink area that contains the origin) of the joint remains convex. (Right) If the distance between polygons is large, the configuration space is not necessarily convex.

With the θ-slice defined, we consider two cases that arise in practice. In the first case, the θ-slice near the origin forms a closed shape (as in Fig. 3 and

Fig. 4b), and the closed region indicates that A and B are interlocked under translational motion, and it is sufficient to serve as our convex approximation. In the second case, the region connects to open space (Fig. 7), meaning the two polygons can be fully separated by translation. In this case, we extend the nearby edges of the θ-slice boundary to close the region and form a bounded convex approximation, as shown in Fig. 7. However, if such an extension includes any forbidden C-space, the extension is discarded, and the motion of B is instead bounded by ϵ^*, ensuring the approximation remains collision-free. For example, Fig. 6(Right), the extension of edges will include invalid space in its convex hull, so this is not suitable for extension.

4.2 Free C-Space Convex Hull

We now incorporate rotation into the analysis. For blocks A and B with configurations $C_A = [T_A^T \;\; \theta_A]$ and $C_B = [T_B^T \;\; \theta_B]$, the motion of block B is modeled as an incremental configuration change $\Delta C_B = [\Delta T_B^T \;\; \Delta\theta_B]$. To approximate the local free c-space, we collect θ-slices and stack them around the current configuration. With A fixed, B is positioned so that its joint edges are approximately equal distance from the concavity of A, then rotated in place. The first positive and negative contact angles are recorded as pc and nc (Fig. 7). We stack the free c-space at $(nc, 0, pc)$ and take their convex hull as a polyhedral approximation of the free c-space. Although the true stacking of θ-slices is nonlinear, this convex hull provides a good local approximation for small rotations. The faces of the resulting polyhedron impose linear constraints on feasible motions,

$$N\Delta C_B + D \leq 0 \tag{1}$$

where each row of N is the normal of a convex hull plane and D is the corresponding offset. These constraints bound the admissible infinitesimal motions ΔC_B of block B while avoiding interpenetration with block A.

4.3 Free C-space Reference Frame

In previous sections, we approximated the free c-space of block B with block A fixed. We now extend this to the general case where both A and B are free to move by expressing C_B in the reference frame of A.

Because translations and rotations are coupled, the relative motion cannot be computed by directly subtracting configurations. Instead, we rotate the translational difference $T_B - T_A$ by the inverse rotation of A plus its rotation change $\Delta\theta_A$ to align it with A's local frame, thereby compensating for A's rotation.

Combining these terms, we define the relative configuration change between A and B as ΔG_{AB}:

$$\Delta G_{AB} = \Delta C_B - \Delta C_A - \left[(R^{-1}(\theta_A + \Delta\theta_A)(T_B - T_A))^T \quad 0\right] \tag{2}$$

This transformation effectively reduces the original 6D configuration space (c-space) constraint problem to a simplified 3D problem.

$$N \cdot \Delta G_{AB} + D \leq 0 \tag{3}$$

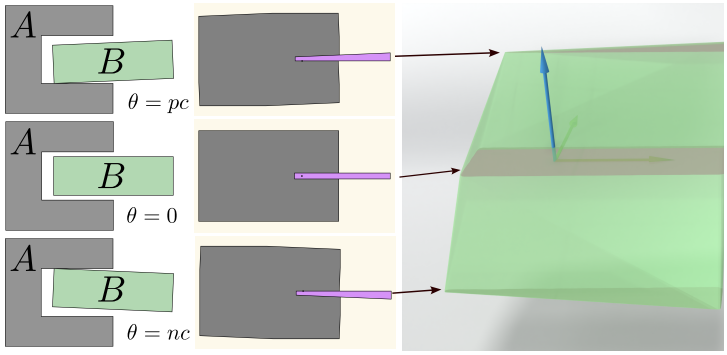


Fig. 7: Stacking θ-slices to approximate the free c-space. (Left) Blocks at different θ, where pc, nc are the first positive/negative θ where B collides with A. (Middle) The Minkowski sum of A and $-B$. The origin (dot) represents the configuration on the left, the purple area around the origin approximates the free c-space at this θ-slice. (Right) We use slices to approximate the free c-space (green area).

4.4 Optimization Formulation

We formulate flexibility analysis as a constrained optimization problem, where the goal is to find a configuration change ΔC that best satisfies a task-specific objective. A common choice is to maximize motion along a preferred direction W, i.e., max $\Delta C^T W$, which measures how freely parts can move in that direction. Other objectives can be used as needed, such as maximizing translational motion $\Delta T^T \Delta T$ or minimizing inter-part distances $\sum |(T_i + \Delta T_i) - (T_j + \Delta T_j)|^2$ to encourage parts to move closer together.

To adapt to general scenarios, we impose several types of constraints on the motion of the blocks:

- **Fixed blocks:** Let F be the set of blocks that should be grounded. For each $k \in F$, we constrain the motion to zero.
- **Collision constraints:** Let P be the set of block pairs whose convex hull overlap. For each overlapping pair $(i, j) \in P$, we apply relative free c-space constraints to prevent interpenetration.
- **Glue constraints:** Let G be the set of block pairs that are assigned to move together rigidly. For each $(i, j) \in G$, we enforce zero relative motion and rotation.

This gives us the optimization formulation as follows:

$$\begin{aligned} \max \quad & obj(\Delta C) \\ \text{s.t.} \quad & \Delta C_k = 0, && \forall k \in F \\ & N \cdot \Delta G_{ij} + D \leq 0, && \forall (i,j) \in P \\ & \Delta G_{ij} = 0, && \forall (i,j) \in G \end{aligned} \tag{4}$$

In section 6 we show how this framework can be applied to problems in validating interlock, finding disassembly trajectories, and analyzing flexibility of interlocked systems of blocks.

5 Evaluation

We evaluate our method in terms of computational performance and physical accuracy by comparing against ground-truth free c-space, measuring runtime, and validating results with physically fabricated interlocking blocks.

5.1 Approximation Accuracy

We evaluate the accuracy of our Minkowski free c-space approximation, , restricted to the pairwise configuration space between two blocks, against Monte Carlo ground truth (10^6 samples) for several block shapes (Table 1), classifying configurations as valid or colliding. Accuracy is measured by true positives and true negatives.

While individual free c-space slices are convex, rotational nonlinearity can introduce concavities in the full free c-space. These effects are negligible for small rotations but may exclude valid configurations under large rotations, leading to false negatives (e.g., the fourth example). Additional errors arise from approximations in computing (pc, nc), which may omit extreme rotations or include invalid configurations. Such errors can be reduced by refining the free c-space approximation near its boundary.

Bounds in Table 1 correspond to the maximum reachable translation and rotation from the current configuration. For non-interlocking cases, this allows full separation, while for arc-shaped joints, the offset rotation center limits rotation, requiring multiple iterations to separate the blocks.

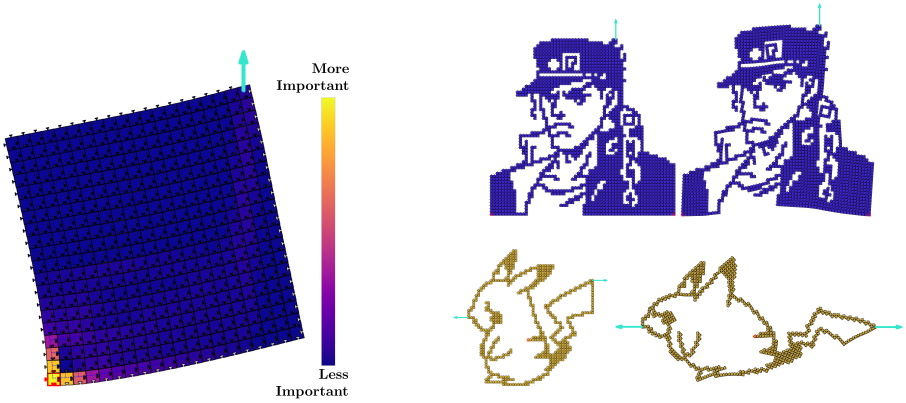
5.2 Performance

Minkowski sum computation is the main bottleneck in our free c-space analysis, particularly for complex geometries. We use a convolution-based reduced Minkowski method [3,31] and reuse computed free c-spaces across identical part pairs in repeating structures.

Performance results are summarized in Table 2. We compare our method to PuzzleFlex's half-plane approach [16] (Julia), run iteratively until convergence. Our implementation uses Python with IPOPT. Experiments use grids of repeated blocks (Figs. 4a, 8a) ranging from 30 (5×6) to 4096 (64×64) parts. With constant-time preprocessing under repeated joints, our approach scales efficiently and significantly outperforms PuzzleFlex on large systems. Unlike PuzzleFlex, our solutions are directly collision-free without stepwise motion. We further demonstrate scalability on large complex structured assemblies, including a Jotaro avatar that consists of 1669-block and a 532-block Pikachu model (Fig. 8b), showing algorithm robustness to irregular, non-grid arrangements.

Table 1: Accuracy analysis of four test cases with comparison to PuzzleFlex [16]. Confusion matrices compare ground truth (GT) configurations to approximated results from our Minkowski sum approach. True positives: Valid GT & Valid Approx; true negatives: Invalid GT & Invalid Approx. Our method shows high accuracy as the sum of true positive and true negative configurations detected.

Blocks	Confusion Matrix			Our Accuracy	PuzzleFlex Accuracy
	GT \ Approx.	Invalid	Valid		
	Invalid	55.89%	0.44%	94.66%	69.11%
	Valid	4.9%	38.77%		
		Invalid	Valid		
	Invalid	42.03%	0.38%	97.32%	61.25%
	Valid	2.29%	55.23%		
		Invalid	Valid		
	Invalid	31.89%	0.48%	99.02%	49.56%
	Valid	0.49%	67.14%		
		Invalid	Valid		
	Invalid	46.94%	1.35%	84.24%	75.61%
	Valid	14.41%	37.30%		



(a) (b)

Fig. 8: Large-scale examples. Red blocks are fixed, and arrows indicate objective directions. (a) Example of a dense grid of interlocking blocks. Showing tolerance importance analysis for flexibility. Importance is heavily dominated by blocks in the lower left corner. Performance results for grid assemblies are given in Table 2. (b) Jotaro avatar with 1,669 T-joint blocks (takes 17.76 s) with bottom left-most and right-most blocks fixed, motion maximized in $+y$. Pikachu with 532 T-joint blocks (takes 6.27s) with a central block fixed and pulled from both sides.

Table 2: Comparison of runtimes between our method and PuzzleFlex [16]. For our method, we report detailed preprocessing (free c-space approximation), constraint solving, and total runtime.

# of Parts (grid size)	Joint Type (number of edges)	Our method (s)			Puzzleflex (s)
		Preprocess	Solve	Total	Total
30 (6×5)	T-Joint(36)	1.20	0.01	1.21	0.38
182 (14×13)	T-Joint(36)	1.20	0.04	1.24	1.65
675 (25×27)	T-Joint(36)	1.20	0.16	1.36	6.03
1890 (45×42)	T-Joint(36)	1.20	20.1	21.3	381.0
4096 (64×64)	T-Joint(36)	1.20	127.0	128.2	1130.0
4096 (64×64)	Dovetail(14)	0.64	44.3	44.94	462.2
4096 (64×64)	Swallowtail(35)	1.40	64.7	66.1	1342.1
4096 (64×64)	Revolute-joint(74)	0.66	91.2	91.82	2356.4

5.3 Physical Evaluation

We applied our method to two interlocking block designs: one with a T-shaped joint and the other with a swallowtail-shaped joint. These designs were primarily evaluated in a linear arrangement, and the results were compared to those produced by the half-plane method, PuzzleFlex [16]. We chose the line structure because tolerance accumulation becomes more obvious in such a simple setup.

The results show that our method consistently outperforms the half-plane method and aligns more closely with observed physical behavior. However, because our free c-space approximation uses a limited set of rotational slices, it may miss some extreme configurations that occur under large rotations. This limitation accounts for the small gap between our results and the physical ground truth.

6 Results and Applications

This section details potential applications of our Minkowski configuration space algorithm, building upon the concepts and evaluations presented earlier.

6.1 Structure Interlocking Checking

Prior work on interlocking detection includes Directional Blocking Graph (DBG) methods [30] and infinitesimal motion–based approaches [29], which struggle to handle gaps between parts (Table 3). In contrast, our method robustly verifies interlocking in the presence of gaps.

Figure 10a shows an assembly that requires simultaneous motion of two blocks for disassembly, which DBG-based methods fail to capture due to their sequential assumption. Figure 10b shows a globally interlocking assembly from Drop Assembly [24], where our method correctly identifies full interlocking after the final two blocks are bonded.

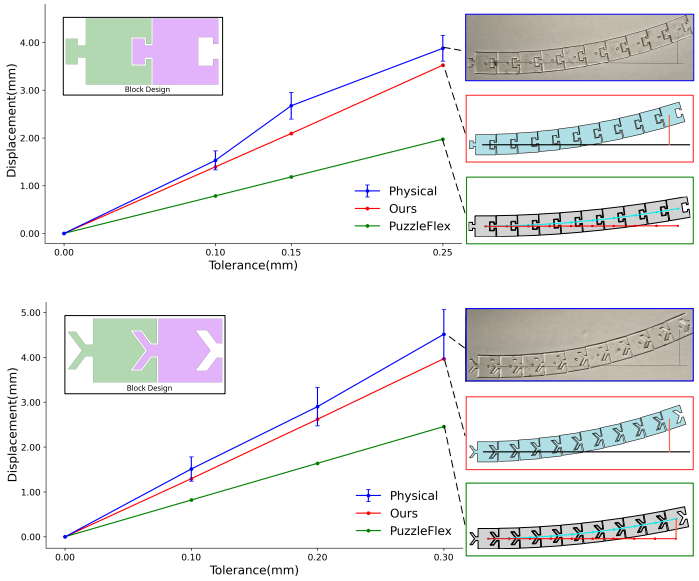


Fig. 9: Flexibility analysis of block chains with T-shaped (top) and swallow-tail (bottom) joints. Plots compare physical experiments (blue), PuzzleFlex [16] (green), and our method (red). The left block is fixed. The plotted distance is the maximum vertical displacement of the rightmost block across tolerance values.

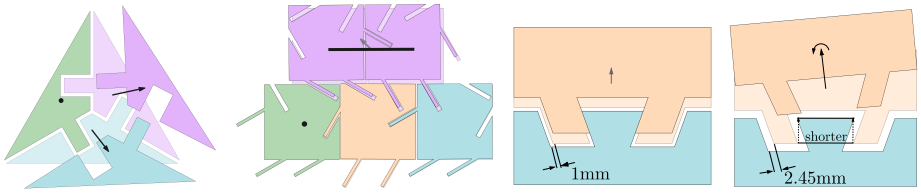
Our approach also detects interlocking failures caused by increased tolerances: while DBG-based methods may still report interlocking, our algorithm accounts for gaps and identifies cases where locking breaks down (Fig. 10c).

Table 3: Comparison with other interlock analysis algorithms

	DBG-Based	Infinitesimal Motion-Based	PuzzleFlex & Ours
Handles convex/concave shapes	✔	✔	✔
Direction independent complexity	✖	✔	✔
Supports simultaneous separation	✖	✔	✔
Supports tolerance	✖	✖	✔
Problem dimension	2D/3D	2D/3D	2D

6.2 Tolerance Analysis

A key application of this method is in detailed tolerance analysis, offering insights beyond traditional statistical approaches. Instead of just assessing overall



(a) (b) (c)

Fig. 10: Validating interlock in multi-block systems. Green blocks are fixed. (a) A three-block system that cannot be sequentially disassembled; our method finds a simultaneous disassembly motion (arrows). (b) Although pairwise joints allow translational disassembly, gluing the purple blocks (acting as a key) creates interlock, which our algorithm correctly verifies. The maximum displacement of the purple blocks is shown. (c) The structure remains interlocked under small tolerances but fails as tolerance increases. When the joint-tip distance is smaller than the wedge width, pure translation is impossible; our method still identifies a valid disassembly motion using rotation.

accumulation, this algorithm allows for identifying which specific part-to-part clearances or dimensional tolerances are most critical to the overall assembly's stability, interlocking integrity, or potential range of motion.

We apply our algorithm with a forward finite difference method to assess each block's influence by offsetting its boundary and measuring the resulting structural displacement. As shown in Fig. 11, some blocks have higher impact than others. Identifying these allows designers to focus on blocks where it matters most. High-impact blocks can be fabricated with tighter tolerances to reduce flexibility or instability, while lower-impact parts can use looser tolerances to cut costs. For instance, in Fig. 11(right), replacing 11 high-impact blocks (out of 56 total) lowered Block A's displacement by 53.8%, versus only 12.4% when the same number of blocks were randomly chosen.

6.3 Tolerance-Aided Design Applications

The computational efficiency of our algorithm enables interactive design exploration. Designers can rapidly evaluate maximum displacement or flexibility induced by manufacturing tolerances, supporting iterative refinement toward either controlled motion or increased rigidity.

3D Printed Space Divider We demonstrate this capability with an interlocking modular space divider (Fig. 1), where flexibility is functional. The design uses 0.9 mm joint tolerance; each unit is 25 cm tall with a 5.6 cm cross-section. Our simulation predicts maximal flexibility across configurations: a single-row arc remains flexible, while a two-row grid restricts motion. Both closely match experimental behavior. Unlike compliant designs, rigid components preserve strength and durability.

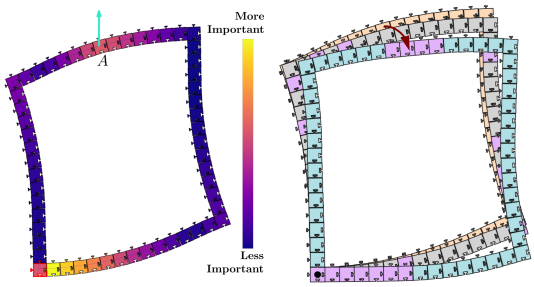


Fig. 11: Tolerance importance analysis. Frame of interlocking blocks with bottom-left corner block fixed. (Left) Displacement evaluated at extreme configuration when block A is moved vertically (blue arrow). Blocks colored by tolerance importance: impact on total displacement. (Right) High-impact blocks (purple ones in blue frame) are replaced with lower tolerances. This reduces overall deformation under the extreme configuration. The original deformed shape is shown in light orange for reference. The grey frame is a control case: an equal number of blocks were randomly selected (purple) and replaced with lower tolerances. Red arrow indicates change of Block A.

Lasercut Shelf Fig. 12 shows a shelf assembled from lasercut acrylic panels (2.8 mm thick). Slots have a minimum base width of 3.0 mm to allow assembly. Side slots include 0.6 mm tolerance, while shelf slots vary between 0.0 mm, 0.5 mm, and 1.5 mm. We analyze a 2D cross-section and apply glue constraints to group disconnected parts into single blocks. Higher tolerances increase flexibility, though predicted rotations are smaller than those observed physically.

7 Limitations and Conclusion

A key limitation of our approach is the free c-space approximation. While using the approximated convex hull collected from rotation slices is efficient, it cannot fully capture non-convexities when large relative rotations exist. This can lead to inaccuracies in predicting kinematic freedom or disassembly paths when significant rotation is involved, potentially requiring iterative re-approximation. Another limitation arises from the ϵ^* bound in our convexity analysis. Lemma 1 is not intended as a guarantee of convexity in general, but rather to provide intuition for when and why the convex approximation is reasonable with a conservative ϵ^* motion bound. A real weakness is that the lemma requires the relative motion to be bounded, and ϵ^* may be small, since the bound considers the worst case over all edge pairs, specifically the distances to the two endpoints of the nearest Minkowski boundary edge. For interlocked parts with a convex free c-space, or parts that can be separated by pure translational motion, this motion limit can be ignored. However, when the free c-space around the origin is non-convex, such as when parts can separate via a bent path, our convex hull

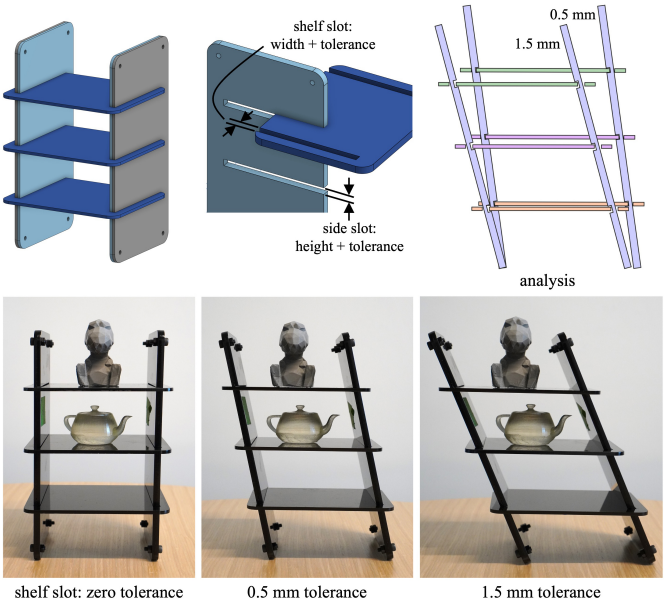


Fig. 12: Shelf design fabricated with lasercut acrylic sheets. Our analysis captures the increased flexibility for larger tolerances, although absolute rotation is higher in physical experiment.

approximation of the free c-space becomes inaccurate. Furthermore, our current slice collecting strategy is relatively naive and may miss valid rotations at extreme configurations.

In conclusion, we introduced a novel algorithm for tolerance analysis and interlocking checking that accounts for gaps between parts. Our method improves both performance and computational efficiency. Future work could focus on improving the approximation of configuration space to better handle non-convexities caused by rotation, possibly through adaptive sampling or more advanced geometric representations. Extending the approach to handle arbitrary 3D freeform shapes would also be a natural next step. Finally, there are promising applications to explore, such as optimizing dense packing or enabling tolerance-aware planning for robotic assembly.

8 Acknowledgments

The authors would like to thank Boston University RASTIC for use of their laser cutting facilities and materials. This work was partially supported by the National Science Foundation Award No.: 2047342.

References

1. Araújo, C., Cabiddu, D., Attene, M., Livesu, M., Vining, N., Sheffer, A.: Surface2Volume: surface segmentation conforming assemblable volumetric partition. ACM Transactions on Graphics **38**(4), 1–16 (Aug 2019). https://doi.org/10.1145/3306346.3323004, https://dl.acm.org/doi/10.1145/3306346.3323004
2. Baraff, D.: Fast contact force computation for nonpenetrating rigid bodies pp. 23–34 (Jul 1994). https://doi.org/10.1145/192161.192168, mAG ID: 2234650683
3. Behar, E., Lien, J.M.: Fast and robust 2D Minkowski sum using reduced convolution. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1573–1578 (Sep 2011). https://doi.org/10.1109/IROS.2011.6094482, https://ieeexplore.ieee.org/document/6094482/, iSSN: 2153-0866
4. Bender, J., Erleben, K., Trinkle, J.: Interactive Simulation of Rigid Body Dynamics in Computer Graphics **33**(1), 246–270 (Feb 2014). https://doi.org/10.1111/cgf.12272, mAG ID: 1917027178
5. Chase, K.W., Greenwood, W.H.: Design issues in mechanical tolerance analysis. Manufacturing Review (1988)
6. Chen, R., Wang, Z., Song, P., Bickel, B.: Computational design of high-level interlocking puzzles. ACM Transactions on Graphics **41**(4), 1–15 (Jul 2022). https://doi.org/10.1145/3528223.3530071, https://dl.acm.org/doi/10.1145/3528223.3530071
7. Cignoni, P., Pietroni, N., Malomo, L., Scopigno, R.: Field-aligned mesh joinery. ACM Transactions on Graphics **33**(1), 1–12 (Jan 2014). https://doi.org/10.1145/2537852, https://dl.acm.org/doi/10.1145/2537852
8. Cui, Q., Rong, V., Chen, D., Matusik, W.: Dense, Interlocking-Free and Scalable Spectral Packing of Generic 3D Objects. ACM Transactions on Graphics **42**(4), 1–14 (Aug 2023). https://doi.org/10.1145/3592126, https://dl.acm.org/doi/10.1145/3592126
9. Davis, E., St, M.: Kinematic Tolerance and the Topology of Configuration Space (2007)
10. Ferguson, Z., Li, M., Schneider, T., Gil-Ureta, F., Langlois, T.R., Jiang, C., Zorin, D., Kaufman, D.M., Panozzo, D.: Intersection-free rigid body dynamics. ACM Transactions on Graphics **40**(4), 1–16 (Jul 2021). https://doi.org/10.1145/3450626.3459802, mAG ID: 3184236041
11. Jones, B., Hildreth, D., Chen, D., Baran, I., Kim, V.G., Schulz, A.: AutoMate: a dataset and learning approach for automatic mating of CAD assemblies. ACM Transactions on Graphics **40**(6), 1–18 (Dec 2021). https://doi.org/10.1145/3478513.3480562, https://dl.acm.org/doi/10.1145/3478513.3480562
12. Kane, C., Marsden, J.E., Ortiz, M., West, M.: Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. International Journal for Numerical Methods in Engineering **49**(10), 1295–1325 (Dec 2000). https://doi.org/10.1002/1097-0207(20001210)49:10<1295::aid-nme993>3.0.co;2-w, mAG ID: 2124969394
13. Krishnamurthy, V.R., Akleman, E., Subramanian, S.G., Ebert, M., Cui, J., Fu, C.a., Starrett, C.: Geometrically Interlocking Space-Filling Tiling Based on Fabric Weaves. IEEE Transactions on Visualization and Computer Graphics **28**(10), 3391–3404 (Oct 2022). https://doi.org/10.1109/TVCG.2021.3065457, https://ieeexplore.ieee.org/document/9376630/
14. Lan, L., Kaufman, D.M., Li, M., Jiang, C., Yang, Y.: Affine body dynamics. ACM Transactions on Graphics **41**(4), 1–14 (Jul 2022). https://doi.org/10.1145/3528223.3530064, mAG ID: 4286611167

15. Larsson, M., Yoshida, H., Umetani, N., Igarashi, T.: Tsugite: Interactive Design and Fabrication of Wood Joints. In: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology. pp. 317–327. ACM, Virtual Event USA (Oct 2020). https://doi.org/10.1145/3379337.3415899, https://dl.acm.org/doi/10.1145/3379337.3415899
16. Lensgraf, S., Itani, K., Zhang, Y., Sun, Z., Wu, Y., Li, A.Q., Zhu, B., Whiting, E., Wang, W., Balkcom, D.: PuzzleFlex: kinematic motion of chains with loose joints. 2020 IEEE International Conference on Robotics and Automation (ICRA) pp. 6730–6737 (May 2020). https://doi.org/10.1109/ICRA40945.2020.9196854, http://arxiv.org/abs/1906.08708, arXiv: 1906.08708
17. Li, M., Ferguson, Z., Schneider, T., Langlois, T., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM Transactions on Graphics **39**(4) (Aug 2020). https://doi.org/10.1145/3386569.3392425, https://dl.acm.org/doi/10.1145/3386569.3392425
18. Lozano-Perez, T.: Spatial Planning: A Configuration Space Approach (Dec 1980), https://dspace.mit.edu/handle/1721.1/5684, accepted: 2004-10-01T20:30:59Z
19. Mazhar, H., Heyn, T., Negrut, D., Tasora, A.: Using Nesterov's Method to Accelerate Multibody Dynamics with Friction and Contact. ACM Transactions on Graphics **34**(3), 32 (May 2015). https://doi.org/10.1145/2735627, mAG ID: 2285831142
20. Minarčík, J., Estep, S., Ni, W., Crane, K.: Minkowski Penalties: Robust Differentiable Constraint Enforcement for Vector Graphics. In: Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24. pp. 1–12. ACM, Denver CO USA (Jul 2024). https://doi.org/10.1145/3641519.3657495, https://dl.acm.org/doi/10.1145/3641519.3657495
21. Montes Maestre, J.S., Du, Y., Hinchet, R., Coros, S., Thomaszewski, B.: FlexScale: Modeling and Characterization of Flexible Scaled Sheets. ACM Transactions on Graphics **43**(4), 1–14 (Jul 2024). https://doi.org/10.1145/3658175, https://dl.acm.org/doi/10.1145/3658175
22. Schwartzburg, Y., Pauly, M.: Fabrication-aware design with intersecting planar pieces. Computer Graphics Forum **32**(2pt3), 317–326 (2013). https://doi.org/https://doi.org/10.1111/cgf.12051, https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12051
23. Sniffen, A.: Design, Analysis, and Drop Assembly of Interlocking Rigid Bodies. Phd thesis, Dartmouth College, Hanover, NH (June 2024), available at https://digitalcommons.dartmouth.edu/dissertations/257
24. Sniffen, A., Sun, Z., Lensgraf, S., Whiting, E., Li, A., Balkcom, D.: Falling into place: Drop assembly of interlocking puzzles. Robotics: Science and Systems (July 2021). https://doi.org/10.15607/RSS.2021.XVII.055, https://par.nsf.gov/biblio/10316182
25. Song, P., Fu, C.W., Cohen-Or, D.: Recursive interlocking puzzles. ACM Transactions on Graphics **31**(6), 1–10 (Nov 2012). https://doi.org/10.1145/2366145.2366147, https://dl.acm.org/doi/10.1145/2366145.2366147
26. Tanadini, D., Boller, G., Leung, P.Y., Huang, Y., D'Acunto, P.: Robotic assembly of interweaving timber linear elements using bespoke interlocking timber-to-timber connections: the cantibox. In: ACADIA 2022: Hybrids & Haecceities (2022)
27. Tang, P., Coros, S., Thomaszewski, B.: Beyond Chainmail: Computational Modeling of Discrete Interlocking Materials. ACM Transactions on Graphics **42**(4), 1–12 (Aug 2023). https://doi.org/10.1145/3592112, https://dl.acm.org/doi/10.1145/3592112

28. Tian, Y., Xu, J., Li, Y., Luo, J., Sueda, S., Li, H., Willis, K.D.D., Matusik, W.: Assemble Them All: Physics-Based Planning for Generalizable Assembly by Disassembly. ACM Transactions on Graphics **41**(6), 1–11 (Dec 2022). https://doi.org/10.1145/3550454.3555525, https://dl.acm.org/doi/10.1145/3550454.3555525
29. Wang, Z., Song, P., Isvoranu, F., Pauly, M.: Design and structural optimization of topological interlocking assemblies. ACM Transactions on Graphics **38**(6), 1–13 (Nov 2019). https://doi.org/10.1145/3355089.3356489, https://dl.acm.org/doi/10.1145/3355089.3356489
30. Wang, Z., Song, P., Pauly, M.: DESIA: a general framework for designing interlocking assemblies. ACM Transactions on Graphics **37**(6), 1–14 (Dec 2018). https://doi.org/10.1145/3272127.3275034, http://dl.acm.org/citation.cfm?doid=3272127.3275034
31. Wein, R.: Exact and efficient construction of planar minkowski sums using the convolution method. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Azar, Y., Erlebach, T. (eds.) Algorithms – ESA 2006, vol. 4168, pp. 829–840. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11841036_73, http://link.springer.com/10.1007/11841036_73
32. Yao, J., Kaufman, D.M., Gingold, Y., Agrawala, M.: Interactive Design and Stability Analysis of Decorative Joinery for Furniture. ACM Transactions on Graphics **36**(2), 1–16 (Apr 2017). https://doi.org/10.1145/3054740, https://dl.acm.org/doi/10.1145/3054740
33. Zhang, X., Belfer, R., Kry, P.G., Vouga, E.: C-Space tunnel discovery for puzzle path planning. ACM Transactions on Graphics **39**(4) (Aug 2020). https://doi.org/10.1145/3386569.3392468, https://dl.acm.org/doi/10.1145/3386569.3392468
34. Zhao, H., Willsey, M., Zhu, A., Nandi, C., Tatlock, Z., Solomon, J., Schulz, A.: Co-Optimization of Design and Fabrication Plans for Carpentry. ACM Transactions on Graphics **41**(3), 1–13 (Jun 2022). https://doi.org/10.1145/3508499, https://dl.acm.org/doi/10.1145/3508499