StructCurves: Interlocking Block-Based Line Structures

Zezhou Sun
micou@bu.edu
Boston University
Boston, Massachusetts, USA

Devin Balkcom
devin.balkcom@dartmouth.com
Dartmouth College
Hanover, New Hampshire, USA

Emily Whiting
whiting@bu.edu
Boston University
Boston, Massachusetts, USA

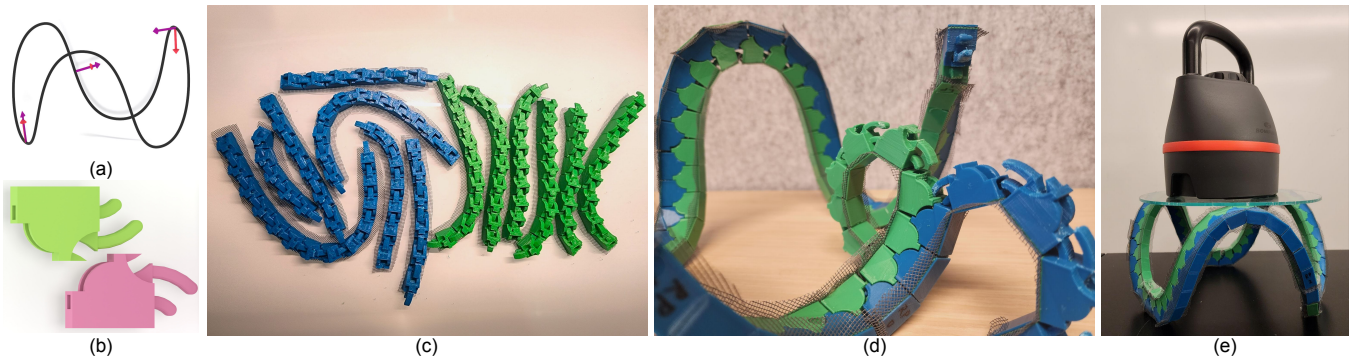


Figure 1: Provided with an oriented central spline (a), our pipeline uses template blocks (b) to generate curved line segments that are 3D printed on a flexible mesh backbone (c). The fabricated block-based structures can be assembled into the desired shape using a pivoting interlock mechanism (d). The assembled structures are rigid and capable of supporting substantial weight (e).

ABSTRACT

We present a new class of curved block-based line structures whose component chains are flexible when separated, and provably rigid when assembled together into an interlocking double chain. The joints are inspired by traditional zippers, where a binding fabric or mesh connects individual teeth. Unlike traditional zippers, the joint design produces a rigid interlock with programmable curvature. This allows fairly strong curved structures to be built out of easily stored flexible chains. In this paper, we introduce a pipeline for generating these curved structures using a novel block design template based on revolute joints. Mesh embedded in these structures maintains block spacing and assembly order. We evaluate the rigidity of the curved structures through mechanical performance testing and demonstrate several applications.

KEYWORDS

kinematics, interlocking structures, deployable structures

ACM Reference Format:
Zezhou Sun, Devin Balkcom, and Emily Whiting. 2024. StructCurves: Interlocking Block-Based Line Structures. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), October 13–16, 2024, Pittsburgh, PA, USA*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3654777.3676354

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA
© 2024 Copyright held by the owner/author(s).
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), October 13–16, 2024, Pittsburgh, PA, USA*, https://doi.org/10.1145/3654777.3676354.

1 INTRODUCTION

This paper presents the development of StructCurves: a new class of 3D-printed line structures based on chains of interlocking blocks. Inspired by traditional zippers, our structures consist of two parallel strips of fabric with interlocking rigid teeth, enabling a secure and reversible connection. Our approach designs individual chains that are flexible and can be tightly packed for transport and storage. The 3D-printed structure then deploys into a rigid structure when assembled, requiring only a constrained pivoting assembly motion. The technique allows for fabricating structures whose bounding boxes are much larger than the printer build volume. Furthermore, our rigid structure serves as a skeletal framework. This characteristic allows for its integration with various materials, for instance, incorporating the structures into fabric for garment design.

The primary contribution of this work is the expansion of the scope of mechanical structures that humans can design, mediated by computational design tools. The geometry of the blocks presents a novel interlocking approach. Complementary teeth sequentially form revolute joint mechanisms that enable a rigid interlock between the assembled chains. This design reduces stress on the flexible backbone material and increases the load-carrying capacity of the structure. By customizing the geometry of the teeth, we can achieve a variety of curved structures with distinct mechanical properties and aesthetic aspects. Practical fabrication is also an objective; we have introduced a flattening procedure that allows the chains to be printed flat on the 3D printer bed. The flattened chains are printed with an embedded mesh backbone that allows them to be bent into their final shape during assembly.

Our design incorporates a built-in assembly sequence within the mesh by alternately interlocking the teeth along the two strips

of fabric using a rotational motion (Figs. 1(d),3(b)). This feature streamlines the assembly process and mitigates the difficulty of selecting the correct components or following complex instructions.

We see three core benefits to designing with StructCurves. First, the ability to actuate between flexible and rigid states provides possibilities for dual modes. We demonstrate this in our garment design applications (Sec. 7.3) with a dress that can optionally take on a surprising silhouette, and a functional example of back support. Second, the line structure approach lends to creating light-weight skeletal frameworks. We show this in a hobbyist boat example (Sec. 7.2) where the final deployed volume is many times larger than the 3D printed segments and has a high load carrying capacity. Third, the flexibility of the disassembled chains enables efficient storage. We demonstrate this with a portable furniture scenario (Sec. 7.1) where the individual segments compactly align in a conventional container.

In summary, our contributions include:

- A novel revolute interlocking joint with its interlocking rigidity verified with physical strength tests
- A modeling pipeline for generating geometry of the block-based chains from user-specified curves.
- A set of physically prototyped application scenarios.

The paper describes the context and related work, describes the geometry of StructCurve designs, outlines the design pipeline, and finally discusses fabrication details and experimentally measured mechanical properties of assembled structures.

2 RELATED WORK

When constructing large-scale structures, assembling them from small pieces can simplify the fabrication process and make it easier for transportation. Various methodologies have been explored and two strategies have been popularly explored in the past: top-down decomposition and bottom-up construction. In this work, we follow the bottom-up strategy.

2.1 Top-down Decomposition

Top-down decomposition usually starts with a completed shape, where researchers employ different splitting strategies or rely on user inputs to break the shape down into smaller parts for fabrication. As discussed in Luo et al. [19], binary partitioning can help to decompose structures into smaller parts. Research by Larsson et al. [17], Yao et al. [46] underscores the versatility of user-driven customization in partitioning and connector design; Araújo et al. [5] discusses how to partition structure volumes guided by surface segmentation from users.

After decomposition, these parts can either be joined using external aids (e.g. glue or nails) or through their geometry. By designing the geometry at the joining area, parts can join/separate with only specific motions. If subsequently joined parts inhibit the motion of those assembled earlier, a more mechanically reliable structure can be achieved. A well-known representation of this principle is the burr puzzle. Parts assembled in such a manner are interlocked if they are immobilized relative to other parts, except for the part that serves as the key to the entire structure [35].

Addressing the challenges of part design and their sequential assembly, many works explore puzzles and structures of varied design and complexity [6, 11, 35, 38, 43, 44]. In [28] they are aiming for flexible 3D prints and showcased methods to embed textiles, a concept that inspired our chain's mesh backbone. Apart from complete structural decomposition, combining various fabrication methodologies for internal and external parts offers a fast-prototyping fabrication approach [34]. An inherent challenge, especially concerning transportation or storage, is that these parts typically lack uniform dimensions, necessitating packing designed specifically for each structure [12].

2.2 Bottom-up Construction

An alternative approach is to design simple universal building blocks that can be connected to form larger structures of various shapes.

2.2.1 Line and Frame Structures. For structures represented by a single, continuous chain, Tibbits et al. [39], [40] introduce techniques to attach components with embedded angles. These chains can be reconfigured to form curves that fill a shape, but are not designed to bear significant loads. Path planning strategies [31, 45] can be used to find shape-covering curves. 3D models can also be approximated using frameworks, as seen in innovations like [18] and [36].

As a type of line structure, zippered ribbons have been designed to constrain fabric for the construction of 3D shapes [31]. Zippers have also been used to form cylindrical robotic arms, competent at weight-bearing [8]. Wang et al. [41] fabricates line elements by mixing different materials; these elements are capable of morphing into 3D shapes when heated. He et al. [13] enable custom deformation behavior in 3D printable springs, including embedded joints for assembly.

As highlighted in [22], structures composed of multiple wires can create a self-sustaining form. Lira et al. [18] proposes a general method to generate machine fabricable wires. Other research includes Panetta et al. [25]'s structures realized through the elastic deformation of beams, Cignoni et al. [7]'s algorithm to generate ribbon-shaped pieces from cross-fields defined on surfaces, Ren et al. [27]'s methodology to build 3D surfaces using optimized ribbons, and Panetta et al. [24] that focuses on surface-based inflatable constructs. Innovations also span novel fabrication materials and methods like Protopiper, which employs adhesive tapes to craft tubes [3], and TrussFab which binds plastic bottles with 3D printed joints to produce structures capable of bearing substantial loads [16].

2.2.2 Plate Structures. Building 3D models from 2D plates is another efficient method. McCrae et al. [21], Zhang et al. [48] and Schwartzburg and Pauly [32] introduce algorithms enabling design of interlocking planar sections for 3D models. Roumen et al. [29, 30] focus on the generation of cutting plans for laser cutting facilitating the production of planar plates, which are then assembled into 3D models. Related works like Park et al. [26] mitigate the risk of incorrect piece picking in the assembly process and Abdullah et al. [1, 2] helps enhance model strength. Alternatively to planar construction, our line structures serve as a skeletal framework. This allows for integration with various materials, for instance, incorporating the structures into fabric.

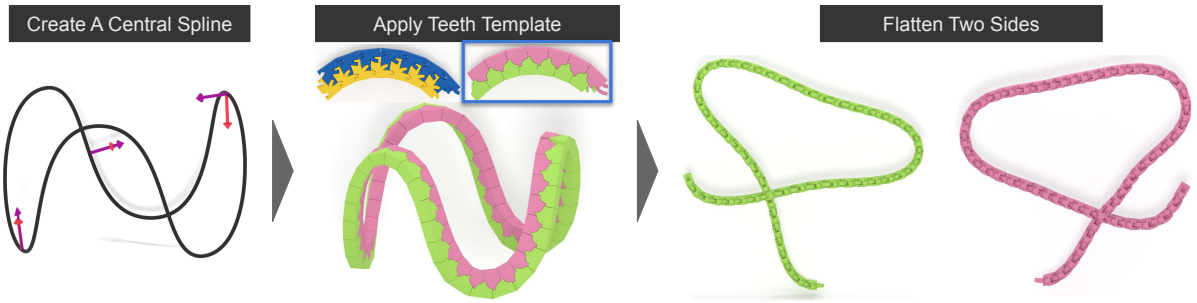


Figure 2: Overview of the modeling pipeline.

2.2.3 Block Structures. Assembling structures from modular parts has been an active area of fabrication research. The ubiquity of Lego blocks underscores their potential in shape approximation. For instance, Mueller et al. [23] uses Lego blocks for fast prototype creation, while the Dynablock system is a dynamic 3D printer enabling reconstructable shape formation [37]. In another notable study, Wang et al. [42] employed convex rigid blocks to build a 3D surface. The assembly process for these modular components, especially for large-scale structures, remains intricate. Works such as [9] shed light on assembly methods for masonry shell structures. Block designs in Sniffen et al. [33], Zhang and Balkcom [47] allow the construction of 3D interlocking structures by robot assembly. Falcone's study [10] introduced a novel block type, conducive to crafting beams with branching and merging capabilities, allowing surface construction. Compared to our presented approach, conventional interlocking designs often have complex assembly instructions making the process time-consuming and difficult to execute accurately.

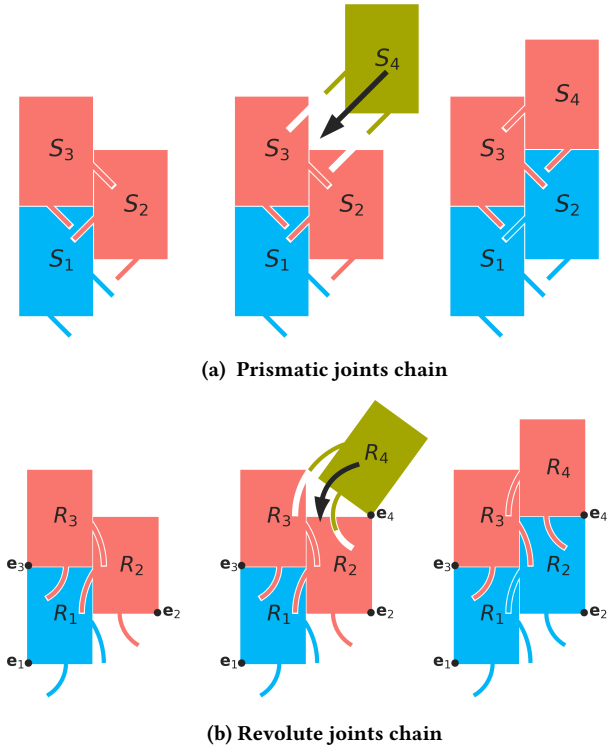
3 INTERLOCKING JOINT DESIGN

In this section, we present our novel revolute joint structure and show how to use this design to generate a curved solid line structure.

3.1 Interlocking Prismatic Joints

Interlocking structures composed of multiple bodies have been explored extensively in prior work. Sniffen et al. [33] introduced a system of blocks stacked following a zig-zag pattern to complete an interlocked 2D shape that can be constructed by pure translation of each block in sequence. Unlike Legos, these interlocked blocks rely on geometry rather than friction to maintain rigidity and connections.

In Figure 3 (a) we show a similar scenario that involves two columns of bodies with an interlocking mechanism. The bodies are assembled in the order: S_1, S_2, S_3, S_4 through translation motions. Each body S_i has two parallel joints connecting to S_{i-1} (adjacent) and S_{i-2} (underneath in the same column). S_4 is "glued" to S_3 by some external means. After gluing, the entire structure is interlocked and behaves as a single rigid body. The reasoning is as follows: S_3 and S_4 pierce S_2 in different directions, immobilizing it. Then S_2 and S_3 pierce S_1 along different directions, forbidding relative motion.



(a) Prismatic joints chain

(b) Revolute joints chain

Figure 3: Schematic of interlocking joint designs. (a) Prismatic joints assembled by straight-line translation inspired by Sniffen et al. [33]). (b) Our novel joint design where each block in the chain R_i assembles via rotation about its pivot point e_i. When the top two blocks in the chain (red) are bound together, the entire structure is immobilized. The yellow block is the next to be assembled.

3.2 Revolute Interlocking Joints

Blocks with prismatic joints have fundamental limitations with the assembly process. Each new block that is inserted must translate in a straight-line motion relative to previously assembled bodies. However, like traditional zippers, we desire a flexible 'backbone' that connects chains of blocks for easy assembly. This requires that adjacent blocks in the chain maintain a fixed distance at their

attachment points to the fabric. To solve this issue we introduce a new interlocking mechanism based on *revolute* joints. Revolute joints make it possible to assign a pivot point, such that no translation is needed for interlocking, and adjacent blocks may remain connected both pre- and post-assembly.

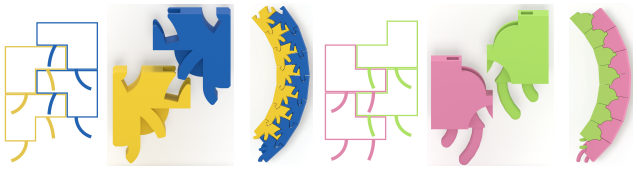
Our design consists of a chain of 3D blocks (teeth of zipper) arranged in two columns. The blocks interlock through revolute joints formed by the complementary tooth and hole in connecting blocks (see Fig. 3(b)). Each block R_i is assembled by rotation about its pivot point e_i.

Consider the base case for block R_1 in Fig. 3(b) left, where R_2 and R_3 are glued together. According to Chasles' Theorem [20], the motion of R_1 can be decomposed into a combination of translational and rotational motions. Furthermore, Euler's theorem [20], posits that a rotation axis must exist for any spatial rotation. However, R_1 is constrained by two revolute joints, each with a different rotation axis (e_2 and e_3). As a result, the only feasible motion for R_1 is either a null motion (immobilized) or one that moves together with both R_2 and R_3. Therefore, we can say that R_1 is interlocked.

Following the same reasoning as interlocking prismatic joints, the entire chain is 'locked' when the last two blocks in the chain are bound to each other, preventing any relative movement.

3.3 Tooth Template

We demonstrate two example tooth templates that incorporate our novel revolute joints (Sec. 3.2), noting that our algorithm's deformation and flattening stages generalize to different template designs. Figure 4 shows the 2D schematic design and its adaptation to a 3D-printable model. We add extra material to provide lateral rigidity, which prevents any relative motion perpendicular to the plane of the joints.



(a) Template A (b) Template B

Figure 4: Two example templates that implement our new revolute interlocking joint design, showing a 2D schematic and the 3D geometry. Our geometric modeling process generalizes to different template designs.

The first template (Fig. 4a) requires little to no support material to print but has exposed joints and may have thin sections prone to breaking during assembly. StructCurves created with this template need minimal post-processing and are nearly ready for use after printing (pictured in Fig. 8).

The second template (Fig. 4b) requires support material but exhibits smoother assembly on curved segments. Its joints are hidden inside the structure, resulting in increased stiffness (pictured in Fig. 1 and most results). However, StructCurves made with this template necessitate post-processing to remove support structures and careful polishing of the joints to ensure tight contact between them. In our experiments, we primarily used the second template.

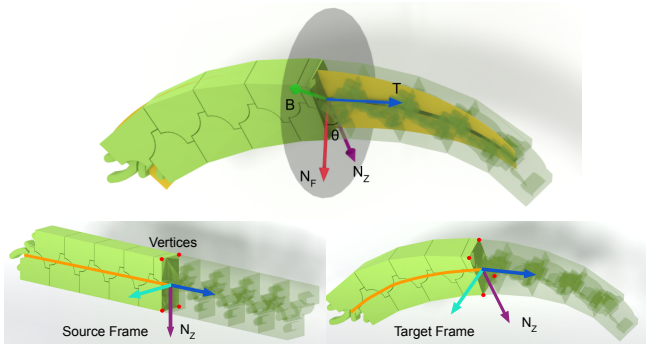


Figure 5: (Top) Our line structure's curve representation. Tangent T, normal N_F and binormal B define the local Frenet-Serret reference frame. The tooth template is aligned with vector N_Z. User-defined orientation, θ, is the angle between N_Z and N_F. (Bottom) We initialize the tooth template along a straight line, then deform the teeth to a curved spline by transforming each vertex (red points) from its source frame to its target frame.

4 CHAIN GEOMETRY GENERATION

In this section, we outline the geometric modeling process for our zipper-like structures using the interlocking tooth design described in section 3.

4.1 Central Spline Curve

To shape our structure, we use a cubic Bézier curve for the central spline for the ability to generate a variety of forms while maintaining a consistent mathematical framework. In addition to the curve that defines the global shape, we must also define how the teeth will be locally oriented along the curve. We define a local reference frame using the Frenet-Serret formula (Fig. 5): the normal vector N_F is the derivative of the tangent vector T with respect to the curve's arc length. We then introduce a user-specified orientation vector for the teeth, N_Z, defined by an angle θ rotation from the normal vector N_F. The user-defined θ parameter is assigned at the Bézier control points and is linearly interpolated along the curve. The θ values can be selected based on factors such as aesthetic considerations or mechanical performance objectives (e.g. resistance to deformation).

4.2 Deform Tooth Template to Spline

To generate the curved geometry of our interlocking chains, we first replicate the template along a straight line that matches the length of the central spline. Each vertex along this straight StructCurve corresponds to an associated source frame, with frame origin at the closest point on the line. The target spline has a continuously defined target frame along its length (Fig. 5), which is rotated from the Frenet-Serret frame by angle θ. The target frame's normal is N_Z while keeping the same tangent T. For a given source frame, the target frame is positioned at the same interpolated distance along the spline curve. We use the corresponding frames to transform

every vertex from the straight line initialization to the target spline (Fig. 5-bottom).

Note that in the deformation mapping, the joints are handled separately from the geometry of the tooth structures. To preserve the assembly behavior, the joints need to retain their arc shape and pivot positions. To address this, we construct a source frame at the center of the face where the joint's peg connects to the tooth body. After the tooth body deformation, we identify the contact area and construct a target frame at its center. The entire peg is then rigidly transformed from the source to the target frame after the teeth bodies undergo deformation.

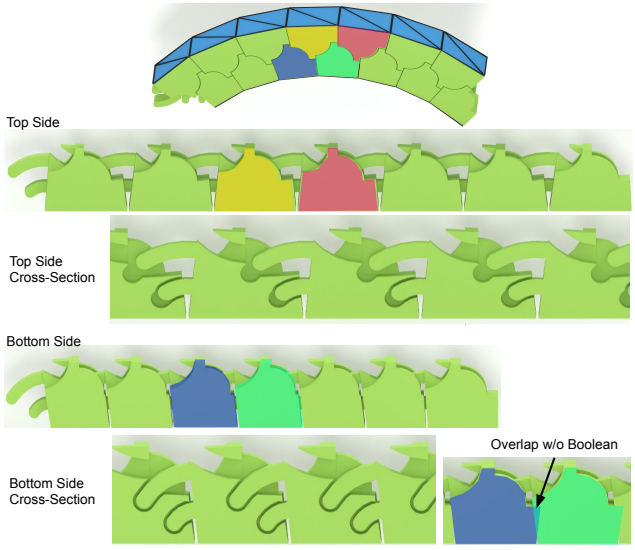


Figure 6: Flattening process. The teeth are aligned to the flattened bottom ribbon. Flattening can create intersections between adjacent teeth (e.g. between the highlighted blue and green regions) which would fuse together when printed. We perform boolean operations to remove these intersections.

4.3 Flattening

The StructCurve geometry is unrolled to a flattened state to enable integrating the mesh during fabrication (see Sec. 6). This process involves extracting the bottom faces of the teeth on both sides and forming two flattened ribbons without distortion using the method in [15].

In this process, each tooth is aligned and positioned on a 2D plane, guided by the frame attached to its bottom face. The source frame is defined by the curve tangent (T) and tooth orientation vector (N_Z). A target frame is constructed corresponding to each tooth's back face in the flattened 2D ribbon. For each tooth we apply a rigid transformation from the source frame to the target frame to finish the flattening process.

However, depending on the original curvature this flattening can lead to overlaps in some areas, risking teeth fusion during printing (as shown in Figure 6). To mitigate this, we offset the faces of each tooth along their face normals to create an "inflated" block. We then use as a tool in a boolean operation with its neighbors to eliminate potential overlaps. This procedure introduces a necessary tolerance to avoid the fusion of neighboring teeth due to fabrication errors.

To address the challenge of large-scale structures exceeding 3D printer capacity, we divide the chains into segments that conform to the size limits of the printer's capacity. Additionally, to address the problem of segments looping back and creating self-intersections (Fig. 2 (Right)), we repeatedly halve each segment until all self-intersections are resolved.

5 DESIGN INTERFACE

As pictured in Figure 7, our design tool is implemented in Python, and integrated into the Blender environment. We employ the Blender built-in tools and libigl [14] library to handle basic geometric and boolean operations. The design tool can conveniently generate StructCurves through a combination of interactive capabilities and automated geometric modeling features.

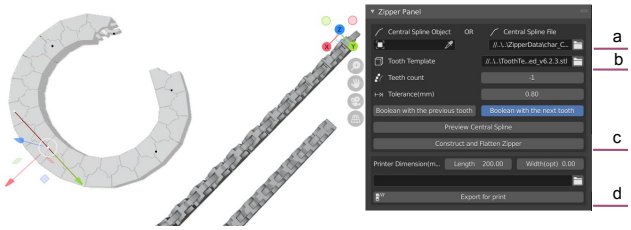


Figure 7: User interface of the StructCurve design tool integrated into Blender. (Left) 3D Viewport showing generated model and flattened chains. (a) Import central spline from Blender curve object or configuration file. Users can freely modify the spline using Blender tools. (b) Select interlocking tooth template. (c) Advanced settings for other parameters such as tolerance. (d) Export chains for printing with segments partitioned to fit on printbed.

Spline Input. Our tool is an add-on designed for seamless compatibility with Blender. It can accept either a Blender curve or a configuration file containing the central spline information as input. Our tool manages the conversion between the Blender curve and StructCurve spline, making it compatible with Blender's tools. The Frenet-Serret normal N_F is automatically computed along the spline.

Block Orientation. Users can specify orientation (as defined in Sec. 4.1) at each control point of the spline using the 'tilt' parameter in Blender. In general 0° orientations offer better mechanical performance and resistance to deformation, making them suitable for high-stress areas. 90° orientations allow for slight flexibility, and are recommended for applications requiring better fit or comfort, such as in wearable technologies. Varying the orientation parameter can accommodate other design objectives, for example, giving a larger contact surface for the tabletop in Fig. 1. We discuss different uses of the orientation parameter in Sec. 7.

Other Parameters. Various templates can be selected for the tooth geometry, including our two sample templates (Fig. 4). The design tool automatically determines the required number of teeth to cover

the whole spline. We also include an option for users to adjust the tolerance to better suit their specific printer and design requirements. The recommended tolerance range is between 0.6 mm to 1.2 mm. It is important to balance rigidity and assemblability when choosing the tolerance. StructCurves with larger tolerance are less rigid but easier to assemble, which is acceptable for short pieces. For longer StructCurves, a smaller tolerance is necessary to maintain rigidity (e.g., the boat example in Sec. 7.2).

Export for Fabrication. Our software flattens each side of the chain onto a 2D plane and splits them into segments that fit within the print bed. Users can specify their printer's dimensions, allowing our software to output segments optimized for their specific printing capacities.

6 FABRICATION WITH EMBEDDED MESH

In this section, we describe our 3D printing fabrication process to embed the mesh backbone. Our fabrication method is compatible with most FDM printers that allow users to pause the print so the mesh can be put in place. For our experiments, we used an Ultimaker S5 printer using Tough PLA filament material, and all models were generated with a uniform tolerance of 0.8mm.

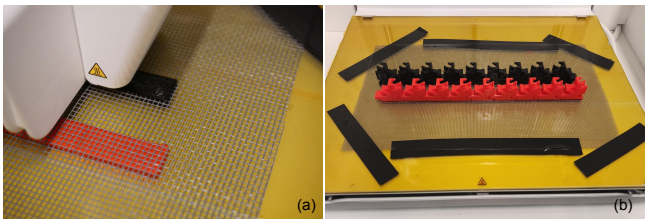


Figure 8: Fabrication of a chain segment. (a) The layer immediately above the mesh is extruded at 220% flow rate to improve adhesion. (b) Two segments after print completion.

A suitable mesh needs to have sufficient tensile strength and withstand repeated bending. In our experiments, we used fiberglass window screening, but other materials are possible.

While directly printing segments on the mesh is possible with FDM printers, our tests showed that neighboring teeth sometimes fused due to over-extruded material. Therefore, we reserve space for the mesh layer (see Fig. 8) using Cura slicing software and the "support blocker" feature. We partition the geometry into lower, placeholder, and upper parts with different printing settings:

- Lower part: Acts as a base for mesh embedding. Built three times the layer height with standard print settings.
- Placeholder part: Reserves space for the mesh with a height equal to the mesh thickness (0.3 mm in our experiments). Wall thickness is set to zero so the printer bypasses the part. A pause command is inserted into the G-code after the lower part. Printing is resumed after placing the mesh tightly in contact with the printed part.
- Upper part: We increase the extrusion flow rate for the first layer (excluding the outer wall) to ensure effective fusion with the lower part through the mesh. The increased flow rate is given by $(1 + \frac{\text{mesh thickness}}{\text{layer height}}) \times 110\%$ (220% in our experiments).

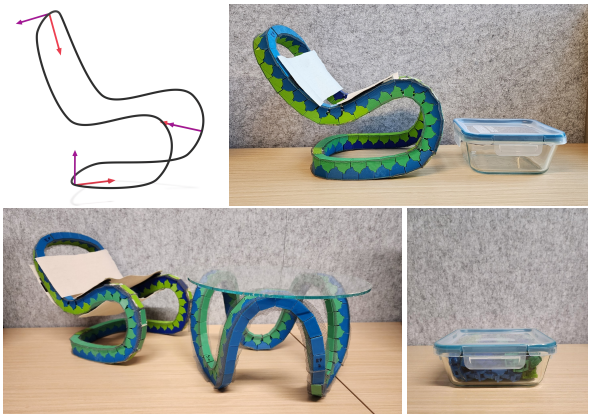


Figure 9: (Top-Left) Central spline of chair model. Purple vectors denote N_Z (tooth orientation) and red vectors represent N_F (curve normal). (Top-Right) Assembled chair model (white cloth is decorative) next to its storage container. (Bottom-Left) Matching chair and table. (Bottom-Right) Chair model segments packed up.

This technique ensures a strong bond between the fabric and the interlocking structure of the joints.

7 DESIGN APPLICATIONS

In this section, we showcase the practical applications of our design, highlighting its dual-state functionality that allows transitions between flexible and rigid states.

7.1 Portable Furniture

The disassembled flexibility of StructCurves makes them ideal for portability and storage. We constructed two prototype furniture pieces demonstrating this concept: a table (Fig. 1) and a chair (Fig. 9). Both can be disassembled for efficient packing, with the chair fitting into a 16.5 cm x 20 cm x 6 cm space, and the table into a 20 cm x 15 cm x 6 cm space. Upon assembly the chair occupies 25 cm x 20.5 cm x 27 cm and is capable of supporting up to 4 kg. The assembled table measures 32 cm x 28 cm x 17.5 cm and supports up to 9 kg. Both models are larger than our printer's capacity.

7.2 Structural Boat Design

We showcase our structure's practicality in a hobbyist boat design, featuring four arches as primary supports. The boat, measuring 150 cm (L) x 40 cm (W) x 40 cm (H), is over ten times larger than our printer's volume and weighs 1.96 kg. Its shell, wrapped in a water-resistant tarp, successfully carried 27 kg of bricks and simulated passengers in a buoyancy test. Notably, when disassembled, the boat's components compact into a 30 cm (L) x 20 cm (W) x 12 cm (H) space, which is 1/30 of the volume in its assembled state, demonstrating the structure's space efficiency.

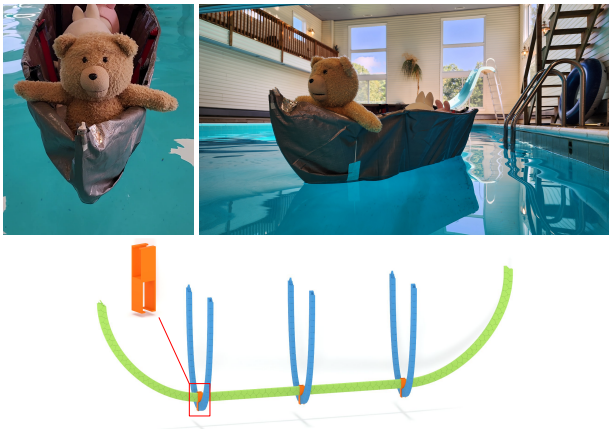


Figure 10: A boat design using four arc-shaped structures as the skeleton and tarp fabric for the shell. Crossing pieces are attached with a connector component (orange). The boat is loaded with 27 kg of weights.

7.3 Integration with Garments

Our structure's rigid and flexible states offer both functional and aesthetic advantages in garment design, enabling outfits to transform their appearance. As demonstrated in Fig. 11, inspired by Loewe[4]'s 2023 collection, the dress features sides that create a floating illusion, supported by two curved line segments anchored at the waist. This design allows the garment to alternate between a surprising silhouette and a natural draping.

Additionally, as shown in Fig. 12, we demonstrate a posture correction application where rigid StructCurves tailored to an individual's spinal shape can provide precise back support across the entire spine. The reversible nature of the StructCurve enables users to engage or disengage the back support as needed without the necessity of removing the garment.

7.4 Choice of Tooth Orientation Parameter

In selecting the appropriate θ tooth orientation, consideration can be given to the specific needs of each application. For example, in our table model (Figs. 1, 9), a θ of 90° was chosen for areas in contact with the glass tabletop to improve stability with increased contact area, whereas a θ of 0° was chosen to provide better strength for the remaining structure. In the boat model a consistent 0-degree orientation was selected across its structure to maximize its strength and load carrying ability (Fig. 10). Conversely in the case of the posture correction garment, a consistent 90-degree θ was opted to provide flexibility (Fig. 12). This flexibility allows the structure to better conform to the wearer's spine curvature, permit small amounts of movement, and increases contact area for more distributed support.

8 EVALUATION

To assess the performance of the interlocking mechanism during use, we conducted a series of strength tests on our StructCurve design. We report results of physical deformation tests on 3D printed segments, and simulated stress tests on the individual joints using finite element analysis.

Figure 11: Transformable clothing. (Left) Rigid state: Dress elevated by underlying structure to create a floating appearance. Inspired by Loewe's "illusion" dresses [4]. (Right) Flexible state: The skirt hangs naturally when the structure is disassembled. The inset figure shows the disassembled chains.

Figure 12: Personalized posture correction. (Left) The disassembled chain permits flexible movement. (Middle) Assembled structure providing rigid back support. (Right) Side view of the assembled structure highlighting the customization to the curve of the wearer's spine.

8.1 Fabrication Accuracy

To evaluate the accuracy of our 3D printed and assembled structures, we conducted tests with a variety of sample segments and compared them to their original design. We found that fabrication errors vary with the orientation of the teeth, 90° samples exhibiting a larger range of errors in comparison to those 0° samples. This underscores the significance of tooth orientation in minimizing angular errors during fabrication. Note each sample was printed five times to compute error ranges.

Furthermore, we investigated twisted structures, where the tooth orientation changes along the spline. Here, we observed an increase in error as the twist angle approached 90°, which we attribute to greater lateral bending tolerance, particularly pronounced in StructCurves with 90° orientation compared to those at 0°. Refer to Appendix A for experimental results.

8.2 FEM Stress Analysis

we conducted Finite Element Method (FEM) simulation of the interlocking teeth to identify potential weak areas in our revolute joint design. Fig. 13 shows a visualization of the stress distribution and deformation of a 4-block test. We anchored one end of the model and applied a 50N force on the other end (Fig. 13). The results showed that the highest stress and deformation occurred at the joint's base, potentially causing gaps or breakage under high force. These findings align with our experiments and suggest the

need for further reinforcement in these areas to enhance durability and performance.

Simulations were also conducted on sample models with twisting orientations. Five samples were prepared, each featuring a straight central spline with identical initial orientations of 0° at one end. The orientations at the opposite ends varied from 15° to 90°. Despite these variations, all samples shared the same simulation setup: their ends were fixed, and a force was applied at the midpoint in the direction opposite to the face normal. It was observed that the displacement at the midpoint increases with the orientation difference which is demonstrated in Figure 14(b).

8.3 Physical Experiments on Deformation

StructCurves without an embedded mesh show more deformation under external forces but maintain their interlocked connections (see the comparison in Fig. 17). This deformation largely results from tolerances between teeth, which affect the rotational motion and weaken the connection. The embedded mesh helps maintain tooth spacing and shape under applied force.

Our experiments, pictured in Figure 15, explored the influence of tooth orientation and spline curvature on structural strength. With a load cell and caliper, we evaluated eight shapes: four arcs, each with both 90° and 0° tooth orientations, in two variations – one made of a single segment on each side and the other made of two segments on each side with a breakpoint at the arc's midpoint. Results in Figure 16 indicated that 0° StructCurves respond linearly to external forces, while 90° StructCurves show more complex behavior, initially more flexible but eventually stiffening. Two-segment StructCurves, compared to one-segment versions, showed slightly reduced initial rigidity and early joint dislodging.

9 LIMITATIONS & FUTURE WORK

Fabrication Method. There are many interesting ways to approach fabrication of the connections between blocks. Using hinges could

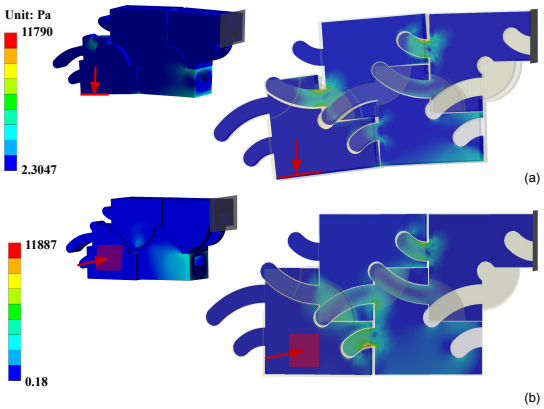


Figure 13: Stress distribution results obtained from FEM analysis with Ansys. The right-most faces on the teeth are fixed (dark gray) and a force (red arrow) is applied on the left-most tooth. In (a), the force is applied downward at the tooth's base; in (b), the force is applied inward. We include a 3D view and a cross-section view to show the joint geometry.

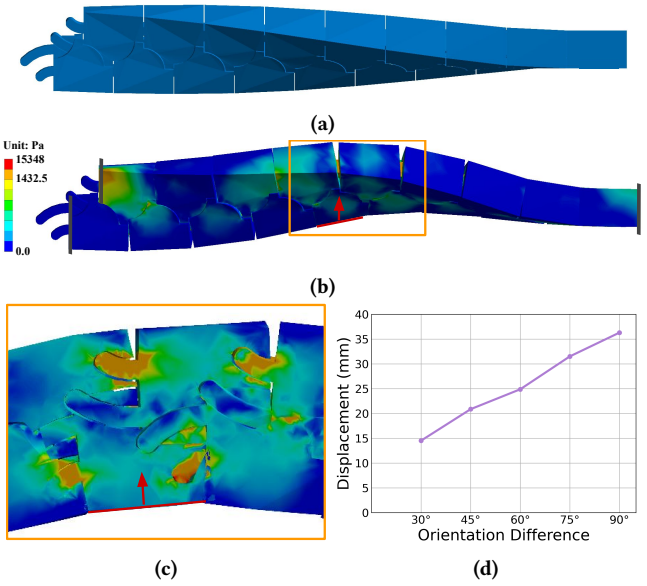


Figure 14: (a) Initial segment with orientation changing from 0° to 90°. (b) Stress distribution from FEM conducted with Ansys. Ends are fixed (dark gray), a 50N force (red) is applied to the midpoint. (c) Cross-sectional view at the midpoint. (d) Plot of midpoint displacement vs. twist angle for 5 different samples (applied force of 50N).

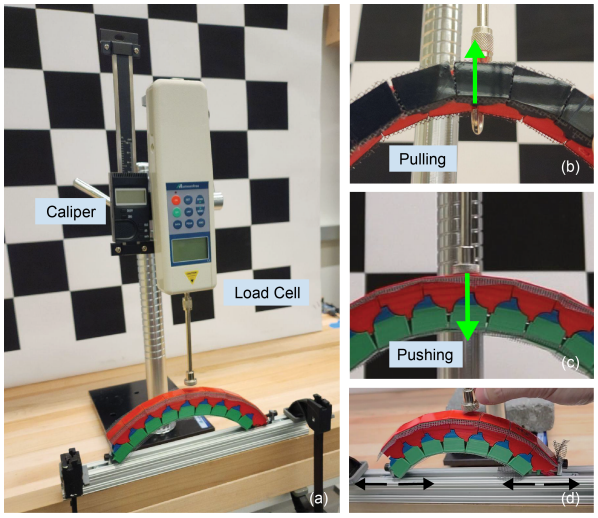


Figure 15: (a) Experimental setup for strength tests. External forces are applied to the structure by pulling (b) or pushing (c) at the middle of the arc. (d) The sample is secured to the frame with two ends attached to slidable brackets, allowing free horizontal movement during deformation.

be another method to connect the teeth, but it would require a careful design to achieve the same range of motion and high tensile strength provided by the embedded mesh. We also considered dual material printing with flexible material as the backbone, but our

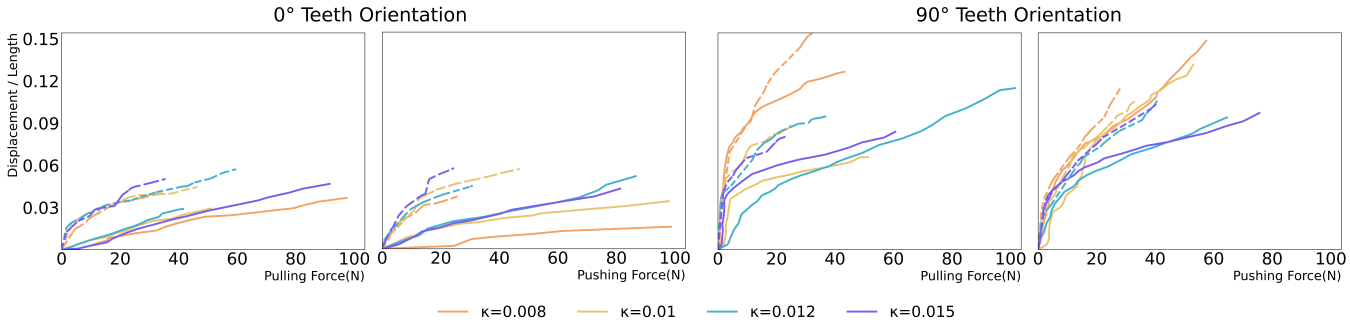
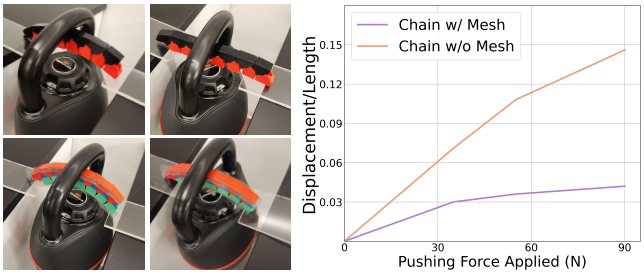


Figure 16: Strength test results for arc-shaped sample structures under applied pulling forces and pushing forces. See setup in Fig. 15. The y-axis is relative displacement (vertical deformation at midpoint divided by total length). Line color indicates different curvatures (see Fig. 18). Results are shown for one-segment samples that have a continuous mesh backbone (solid lines), and samples with two segments on each side that meet at the center (dashed lines). The plots show results for 0° tooth orientation (left) and 90° tooth orientation (right). κ is curvature in unit mm^{-1}.



(a) 3.5 kg **(b)** 9.0 kg **(c)**

Figure 17: Effect of embedded mesh on deformation. (a,b) Bottom: Teeth with embedded mesh. (a,b) Top: Segments without any mesh exhibit higher deformation but remain interlocked. (c) Plot of deformation vs. force with and without the mesh. Samples were tested at 3.5 kg, 5.5 kg, **and** 9.0 kg.

initial experiments found this approach too prone to breakage. We hope to investigate this direction more in future work as it would significantly simplify the fabrication process. While the mesh requires extra work in the printing process, we found it to be effective as a way to maintain flexibility and durability. Another direction we hope to explore is belt 3D printing, which may have the capability of creating the entire zipper chain in a single print job, removing the need for partitioning into segments.

Curvature Limit. Splines that exhibit sharp turns can lead to thin socket walls for joints and interlocking failures at certain curvature thresholds. If the curvature at any point on the spline exceeds this limit, the teeth in the generated StructCurve will penetrate through the neighboring teeth's walls.

Scalability. Scaling up to higher block counts would present new challenges, such as increased flexibility due to tolerances between teeth and the need for extensive post-processing. Addressing these issues requires new fabrication methods and design modifications, like reinforcing joints, to maintain the precision and integrity of larger structures.

Reusability. The tooth geometries in our examples are generally unique, making them non-reusable for other curved structures. Improving reusability is a potential area for future work, for example, using pre-fabricated segments with fixed turning or twisting angles (similar to interchangeable Lego pieces) to approximate a given spline input.

Inverse Design. In our current implementation the θ parameter for block orientation is selected manually by user input. In future work we are interested in developing inverse design methods that optimize orientation based on properties of strength or other objectives.

Other Challenges. Assembly is done by hand which can be time-intensive. In future work we would like to design a slider that would make our structures truly zippable and make assembly more efficient. Additionally, interlocking structures with branching or loop splines are an area needing further investigation for improved connections at open ends and junctions.

10 CONCLUSION

We present StructCurves, a novel modeling and fabrication pipeline for 3D-printed line structures that interlock into curved forms that are load-bearing. The line segments are printed piecewise in a flexible state while maintaining rigidity when assembled. This enables compact storage and transport, fabricating objects larger than the 3D printer volume, and eases assembly with a continuous chain of blocks. The assembly of our structures aims to be straightforward. Unlike complex interlocking designs that demand specific assembly sequences, our design simplifies this process, enabling users to assemble the structure with the help of the mesh guide, eliminating the need for detailed instructions.

The ability to actuate from a flexible to rigid state makes our design applicable to deployable structures, as they may be shipped compactly and assembled on site. We have demonstrated scenarios where the structures can serve as a skeleton, providing the framework for secondary materials such as tensioned fabric. Ultimately, we hope to enable future work on the design of large-scale, durable, deployable structures, that may be quickly assembled by humans or robots.

REFERENCES

[1] Muhammad Abdullah, Romeo Sommerfeld, Laurenz Seidel, Jonas Noack, Ran Zhang, Thijs Roumen, and Patrick Baudisch. 2021. Roadkill: Nesting laser-cut objects for fast assembly. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 972–984.

[2] Muhammad Abdullah, Martin Taraz, Yannis Kommana, Shohei Katakura, Robert Kovacs, Jotaro Shigeyama, Thijs Roumen, and Patrick Baudisch. 2021. FastForce: Real-Time Reinforcement of Laser-Cut Structures. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[3] Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofen, Hsiang-Ting Chen, Stefanie Mueller, and Patrick Baudisch. 2015. Protopiper: Physically Sketching Room-Sized Objects at Actual Scale. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 427–436. https://doi.org/10.1145/2807442.2807505

[4] Jonathan Anderson. 2023. Loewe: Spring Summer 2023 Women's Show. https://www.loewe.com/usa/en/stories-collection/ss23-women-runway.html. Accessed: 2023-09-14.

[5] Chrystiano Araújo, Daniela Cabiddu, Marco Attene, Marco Livesu, Nicholas Vining, and Alla Sheffer. 2019. Surface2Volume: Surface segmentation conforming assemblable volumetric partition. *arXiv preprint arXiv:1904.10213* (2019).

[6] Rulin Chen, Ziqi Wang, Peng Song, and Bernd Bickel. 2022. Computational Design of High-Level Interlocking Puzzles. *ACM Trans. Graph.* 41, 4, Article 150 (jul 2022), 15 pages. https://doi.org/10.1145/3528223.3530071

[7] Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. 2014. Field-aligned mesh joinery. *ACM Transactions on Graphics (TOG)* 33, 1 (2014), 1–12.

[8] Foster Collins and Mark Yim. 2016. Design of a spherical robot arm with the spiral zipper prismatic joint. In *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2137–2143.

[9] Mario Deuss, Daniele Panozzo, Emily Whiting, Yang Liu, Philippe Block, Olga Sorkine-Hornung, and Mark Pauly. 2014. Assembling self-supporting structures. *ACM Trans. Graph.* 33, 6 (2014), 214–1.

[10] Sara Elizabeth Falcone. 2020. *Zipped assembly.* Master's thesis. Massachusetts Institute of Technology.

[11] Chi-Wing Fu, Peng Song, Xiaoqi Yan, Lee Wei Yang, Pradeep Kumar Jayaraman, and Daniel Cohen-Or. 2015. Computational interlocking furniture assembly. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–11.

[12] Somaye Ghandi and Ellips Masehian. 2015. Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design* 67 (2015), 58–86.

[13] Liang He, Huaishu Peng, Michelle Lin, Ravikanth Konjeti, François Guimbretière, and Jon E. Froehlich. 2019. Ondulé: Designing and Controlling 3D Printable Springs. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 739–750. https://doi.org/10.1145/3332165.3347951

[14] Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

[15] Simon Kolmanič and Nikola Guid. 2002. The flattening of arbitrary surfaces by approximation with developable stripes. In *From Geometric Modeling to Shape Modeling: IFIP TC5 WG5. 2 Seventh Workshop on Geometric Modeling: Fundamentals and Applications October 2–4, 2000, Parma, Italy*. Springer, 35–46.

[16] Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. 2017. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2606–2616. https://doi.org/10.1145/3025453.3026016

[17] Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. 2020. Tsugite: Interactive Design and Fabrication of Wood Joints. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 317–327. https://doi.org/10.1145/3379337.3415899

[18] Wallace Lira, Chi-Wing Fu, and Hao Zhang. 2018. Fabricable eulerian wires for 3D shape abstraction. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–13.

[19] Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. 2012. Chopper: Partitioning models into 3D-printable parts. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–9.

[20] Matthew T Mason. 2001. *Mechanics of robotic manipulation.* MIT press.

[21] James McCrae, Nobuyuki Umetani, and Karan Singh. 2014. FlatFitFab: interactive modeling with planar sections. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 13–22.

[22] Eder Miguel, Mathias Lepoutre, and Bernd Bickel. 2016. Computational design of stable planar-rod structures. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

[23] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. 2014. FaBrickation: Fast 3D Printing of Functional Objects by Integrating Construction Kit Building Blocks. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3827–3834. https://doi.org/10.1145/2556288.2557005

[24] Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. 2021. Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.

[25] Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

[26] Keunwoo Park, Conrad Lempert, Muhammad Abdullah, Shohei Katakura, Jotaro Shigeyama, Thijs Roumen, and Patrick Baudisch. 2022. FoolProofJoint: Reducing Assembly Errors of Laser Cut 3D Models by Means of Custom Joint Patterns. In *Proc. 2022 CHI Conference on Human Factors in Computing Systems*. 1–12.

[27] Yingying Ren, Julian Panetta, Tian Chen, Florin Isvoranu, Samuel Poincloux, Christopher Brandt, Alison Martin, and Mark Pauly. 2021. 3D weaving with curved ribbons. *ACM Transactions on Graphics* 40, ARTICLE (2021), 127.

[28] Michael L. Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E. Hudson. 2017. Stretching the Bounds of 3D Printing with Embedded Textiles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 497–508. https://doi.org/10.1145/3025453.3025460

[29] Thijs Roumen, Yannis Kommana, Ingo Apel, Conrad Lempert, Markus Brand, Erik Brendel, Laurenz Seidel, Lukas Rambold, Carl Goedecken, Pascal Crenzin, et al. 2021. Assembler3: 3d reconstruction of laser-cut models. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.

[30] Thijs Roumen, Conrad Lempert, Ingo Apel, Erik Brendel, Markus Brand, Laurenz Seidel, Lukas Rambold, and Patrick Baudisch. 2021. autoAssembler: Automatic Reconstruction of Laser-Cut 3D Models. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 652–662.

[31] Christian Schüller, Roi Poranne, and Olga Sorkine-Hornung. 2018. Shape Representation by Zippables. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)* 37, 4 (2018).

[32] Yuliy Schwartzburg and Mark Pauly. 2013. Fabrication-aware Design with Intersecting Planar Pieces. *Computer Graphics Forum* (2013). https://doi.org/10.1111/cgf.12051

[33] Amy Sniffen, Zezhou Sun, Samuel Lensgraf, Emily Whiting, Alberto Li, and Devin Balkcom. 2021. Falling Into Place: Drop Assembly of Interlocking Puzzles. In *Robotics: Science and Systems*.

[34] Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. 2016. CofiFab: coarse-to-fine fabrication of large 3D objects. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

[35] Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. 2012. Recursive interlocking puzzles. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–10.

[36] Lingyun Sun, Jiaji Li, Yu Chen, Yue Yang, Zhi Yu, Danli Luo, Jianzhe Gu, Lining Yao, Ye Tao, and Guanyun Wang. 2021. FlexTruss: A Computational Threading Method for Multi-material, Multi-form and Multi-use Prototyping. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[37] Ryo Suzuki, Junichi Yamaoka, Daniel Leithinger, Tom Yeh, Mark D. Gross, Yoshihiro Kawahara, and Yasuaki Kakehi. 2018. Dynablock: Dynamic 3D Printing for Instant and Reconstructable Shape Formation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 99–111. https://doi.org/10.1145/3242587.3242659

[38] Keke Tang, Peng Song, Xiaofei Wang, Bailin Deng, Chi-Wing Fu, and Ligang Liu. 2019. Computational design of steady 3D dissection puzzles. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 291–303.

[39] Skylar Tibbits, Marcelo Coelho, and FormLabs inc. 2013. Hyperform - self-assembly lab. https://selfassemblylab.mit.edu/hyperform/

[40] Skylar Tibbits and MIT Self-Assembly Lab. [n. d.]. Folding proteins - self-assembly lab. https://selfassemblylab.mit.edu/proteins

[41] Guanyun Wang, Ye Tao, Ozguc Bertug Capunaman, Humphrey Yang, and Lining Yao. 2019. A-Line: 4D Printing Morphing Linear Composite Structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300656

[42] Ziqi Wang, Peng Song, Florin Isvoranu, and Mark Pauly. 2019. Design and Structural Optimization of Topological Interlocking Assemblies. *ACM Transactions on Graphics (SIGGRAPH Asia 2019)* 38, 6 (2019). Article No. 193.

[43] Ziqi Wang, Peng Song, and Mark Pauly. 2018. DESIA: A General Framework for Designing Interlocking Assemblies. *ACM Trans. Graph.* 37, 6, Article 191 (dec 2018), 14 pages. https://doi.org/10.1145/3272127.3275034

[44] Shiqing Xin, Chi-Fu Lai, Chi-Wing Fu, Tien-Tsin Wong, Ying He, and Daniel Cohen-Or. 2011. Making burr puzzles from 3D models. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–8.

[45] Zhijin Yang, Pengfei Xu, Hongbo Fu, and Hui Huang. 2021. WireRoom: model-guided explorative design of abstract wire art. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.
[46] Jiaxian Yao, Danny M Kaufman, Yotam Gingold, and Maneesh Agrawala. 2017. Interactive design and stability analysis of decorative joinery for furniture. *ACM Transactions on Graphics (TOG)* 36, 2 (2017), 1–16.
[47] Yinan Zhang and Devin Balkcom. 2018. Interlocking block assembly. In *Algorithmic Foundation of Robotics (WAFR)*.
[48] Yunbo Zhang, Wei Gao, Luis Paredes, and Karthik Ramani. 2016. Cardboardizer: Creatively customize, articulate and fold 3d mesh models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 897–907.

A FABRICATION ACCURACY RESULTS

In this appendix section, we provide additional images and data plots of our fabrication accuracy tests as discussed in Sec. 8.1.

Figure 18 shows the arc-shaped samples we used for fabrication accuracy and strength tests with four different curvature values. We created a set of segments with teeth at 90° orientation, where the curve normal vector N_F is perpendicular to the orientation vector N_Z, and another set of segments with 0° orientation, where N_Z is aligned with N_F. Note that each sample was printed and tested five times.

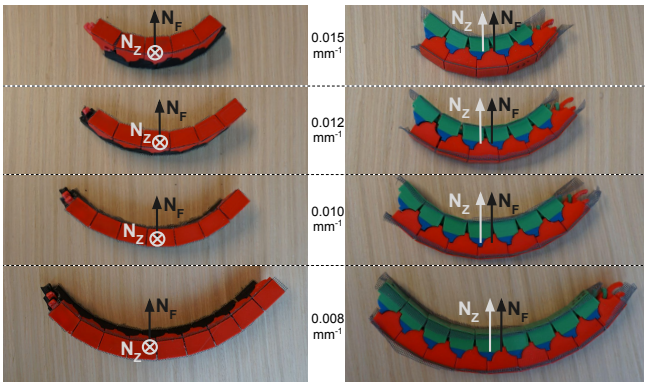


Figure 18: Arc-shaped samples for fabrication accuracy and strength tests. Four different curvatures (values shown in the middle, in units mm^{-1}). (Left) Segments with $90°$ orientation (N_Z points into image plane). (Right) Segments with $0°$ orientation.

Figure 19 shows the results of the fabrication accuracy experiments, along with the measurement setup. We tested four arc shapes at different curvatures, as shown in Fig. 18.

Figure 20 displays the results for accuracy measurements of twisting segments, along with the 3D printed samples. The twist angle refers to the change in θ orientation between the start and end of the segment.

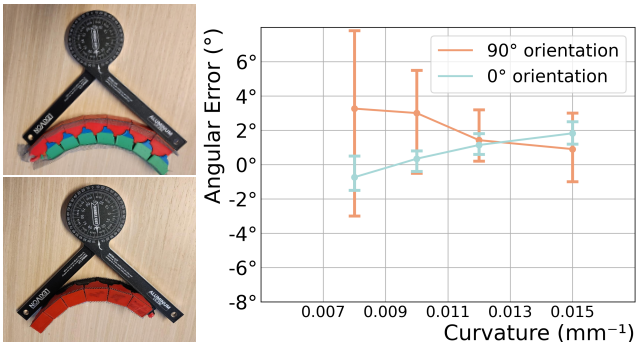


Figure 19: Results of fabrication accuracy tests. Each curvature had both a $90°$ tooth orientation sample and $0°$ tooth orientation sample. (Left) Protractor setup to measure turning angle accuracy. (Right) Plot of angular error vs. curvature.

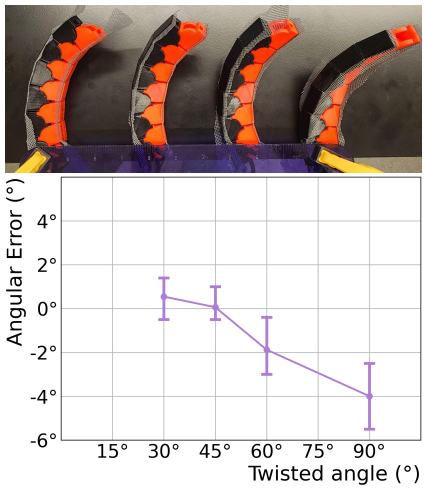


Figure 20: (Top) Four twisted samples with the same central spline shape, each with a curvature of $0.012\,\mathrm{mm}^{-1}$. All samples start with tooth orientation of $0°$. The ending tooth orientations are $30°$, $45°$, $60°$, and $90°$ (left to right). (Bottom) Plot of angular error vs. twist angle.