AirTied: Automatic Personal Fabrication of Truss Structures

Lukas Rambold, Robert Kovacs, Conrad Lempert, Muhammad Abdullah, Helena Lendowski, Lukas Fritzsche, Martin Taraz, Patrick Baudisch
Hasso-Plattner-Institute, Potsdam, Germany
firstname.lastname@hpi.de

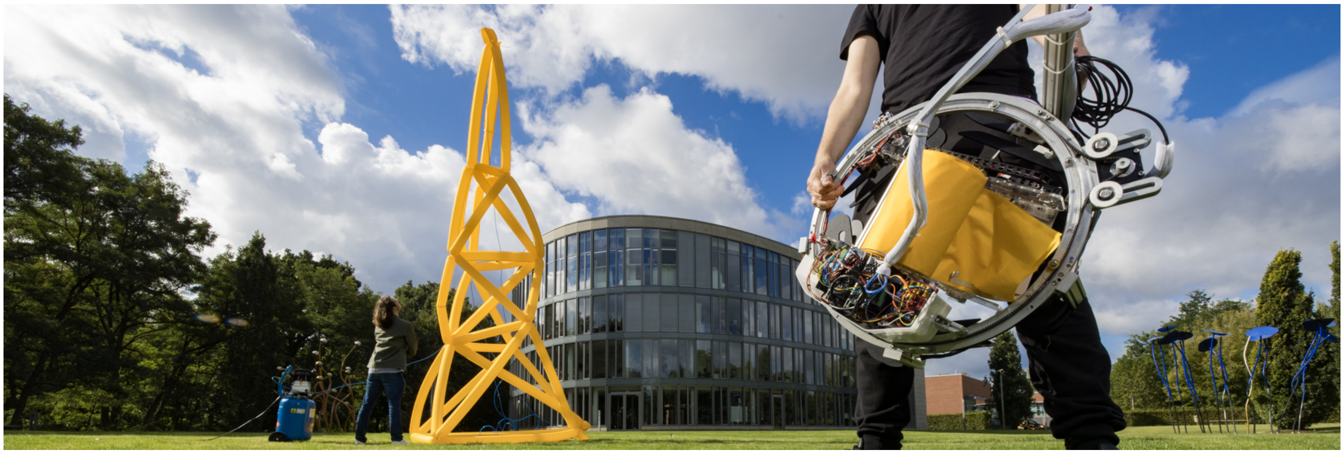

Figure 1: The AirTied device (right) fabricates large truss structures in fully automated fashion, such as this 6 m tower designed to hold a WIFI beacon.

ABSTRACT

We present *AirTied*, a device that fabricates truss structures in a fully automatic fashion. AirTied achieves this by unrolling a 20cm-wide inflatable plastic tube and tying nodes into it. AirTied creates nodes by holding onto a segment of tube, stacking additional tube segments on top of it, tying them up, and releasing the result. The resulting structures are material-efficient and light as well as sturdy, as we demonstrate by creating a 6m-tower. Unlike the prior art, AirTied requires no scaffolding and no building blocks, bringing automated truss construction into the reach of personal fabrication.

CCS CONCEPTS

• **Human-centered computing** → Human computer interaction (HCI); Interactive systems and tools..

KEYWORDS

inflatables, personal fabrication, truss, human-scale

ACM Reference Format:
Lukas Rambold, Robert Kovacs, Conrad Lempert, Muhammad Abdullah, Helena Lendowski, Lukas Fritzsche, Martin Taraz, Patrick Baudisch. 2023. AirTied: Automatic Personal Fabrication of Truss Structures. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), October 29–November 01, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3586183.3606820

UIST '23, October 29–November 01, 2023, San Francisco, CA, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0132-0/23/10.
https://doi.org/10.1145/3586183.3606820

1 INTRODUCTION

For several decades, researchers in architecture have been exploring techniques for creating truss structures [7]. The interest in trusses stems from the fact that they allow producing large and sturdy, yet material-efficient structures. Trusses achieve this by breaking large volumes down into triangular sub-structures made from edges and nodes, and thus, are used for bridges, concert halls or space exploration, among many other examples [10].

More recent research explores how to create truss structures automatically, either by automatically stacking pre-made building blocks (Doggett et al. [9], *BILL-E* [15], *Material-Robot-System* [16]) or by setting up large scaffolded volumes and then spanning and intertwining edges from top to bottom (e.g., *Aerial Construction* [2], *Spatial Winding* [11]).

Today, with researchers in HCI exploring truss construction from a *personal* fabrication perspective (e.g., *TrussFab* [23]), the question arises whether *automatic* truss construction might be possible without the comparably complex resources commonly used in architecture, i.e., without pre-made building blocks and without scaffolded spaces.

In this paper, we thus explore how to automatically fabricate truss structures "from scratch". We are building on research in inflatables [40], in particular, inflatable structures based on creasing and folding long tubes, such as *Aeromorph* [32] and *Printflatables* [37].

We present *AirTied*, a device that fabricates truss structures in a fully automated fashion. AirTied achieves this by unrolling a 20cm-wide inflatable plastic tube and tying nodes into it. AirTied creates nodes by holding onto a segment of tube, stacking additional tube segments on top of it, tying them up, and releasing the result. This allows automated truss construction "from scratch", i.e., without pre-made truss building blocks and without a scaffolded volume, affording a device smaller than its artifacts. At the same time, the resulting truss structures allow creating *much* larger structures than the prior art in inflatables, as we demonstrate by creating a 6m tower (Figure 1).

2 CONTRIBUTION, BENEFITS, AND LIMITATIONS

Our main contribution is a mechanism that creates nodes by holding onto a tube segment, stacking additional tube segments on top of it, tying them up, and releasing the result. We have implemented this design in the form of the AirTied device shown in Figure 1. It produces fully assembled large-scale truss structures

We furthermore present an algorithm that accepts 3D truss designs as input and generates the series of instructions required for AirTied (or a human) to fabricate it. We also present an editor (based on *TrussFab* [23]) that allows users to generate AirTied structures interactively. Finally, we validate the concept and the device by producing several inflatable truss structures.

AirTied structures are *very* material efficient, as (1) truss structures are inherently hollow and (2) each inflated member itself is also hollow. Furthermore, AirTied creates trusses in a non-destructive way, which allows users to re-use material by removing ties, rolling up the tube, and re-inserting it into the AirTied device.

Limitations*:* The current prototype (1) still requires users to attach air inlets manually, (2) cannot fabricate edges shorter than ~50cm, as they need to be long enough to reach across the device, (3) cannot hold more than five nodes at a time, while models with large cross sections can require more, (4) offers a 97 % per-operation success rate (edges require one operation, nodes two; see Technical Evaluation) and (5) cannot reliably fabricate nodes connecting more than eight edges.

3 RELATED WORK

This work builds on research on truss structures, inflatables, and construction using textile processes.

3.1 Trusses

Truss structures form rigid frameworks from an assembly of beams. They are ubiquitous in architecture and construction and used for e.g., concert halls, airport roof constructions, bridges, pylons, and towers. While, traditionally, these structures have been constructed manually by artisans, systems of pre-made components have commoditized construction (e.g., *Mero* [27]).

To access the full design space, architects and researchers turn to 3D printing for bespoke connection nodes that are not subject to the geometric restrictions faced by previous systems (*Digital Bamboo* [22], *Redefining Polyhedral Space* [21]).

More recent systems empower novice users and allow them to model and fabricate truss structures, albeit still requiring manual assembly, e.g., *TrussFab* [23], *Strawctures* [53], *StrutModeling* [24], *computational truss generator* [1].

We build on these design tools and borrow the idea from *FlexTruss* [42] to fabricate trusses from continuous 1D filament to enable *automatic* fabrication of trusses.

3.2 Inflatables

Since the 1960s, architects have used modern plastics to create light and material-efficient structures [12].

The advent of digital fabrication machines and advanced design software has given designers additional control over the shape of inflatables. Systems that demonstrate this are, e.g., *Plushie* [28], *Designing Inflatables* [32], *Poimo* [31], *BalloonFab* [51], *Single-Stroke-Structures* [19], *Knitflatables* [3], and *KnitCandela* [34]. The latter builds permanent concrete structures using digitally fabricated inflatables as formwork.

Recently, researchers have started to explore inflatables for soft robots (e.g., *vine robots* [36], *octahedron robots* [48]) and shape-changing interfaces (e.g., *PneUI* [52], *TilePop* [45], *LiftTile* [43], *PneuSeries* [5], *Roomscale Pneumatic Structures* [44]). More specifically, inflatable trusses have been used for commercial tents [14], lightweight pavilions [6], deployable satellites [17], and artistic installations [26].

3.3 Automatic Fabrication of Inflatables

To automate the fabrication pipeline for inflatables, *Aeromorph* [32] uses a CNC-controlled welding device to create software-defined air chambers between the two sheets. Inflation causes the material sheet to bend selectively.

Printflatables [37] and *Reflatables* [30] add a continuous material feed and the capability to form "mountain folds" for articulating bends.

AirTied builds on these projects and adds the ability to produce trusses, which it achieves by holding onto material segments and tying them together, thereby forming nodes.

3.4 Automatic Fabrication using Pre-Made Scaffolding

A number of research projects tackle automatic fabrication in three steps: (1) set up scaffolding, (2) Span edges between parts of the scaffolding and connect them with each other, and (3) remove scaffolding.

The project *Aerial Construction* [2], e.g., demonstrated how to construct a suspension bridge using drones that carry and unroll yarn robotically.

Reichert et al. [35] pioneered the idea of robotically producing architectural structures from a single thread of yarn. A robotic arm spins an epoxy-soaked carbon fiber yarn around a steel frame. An operator removes the steel frame and lets the epoxy cure, resulting in a rigid structure.

Kayser et al. [20] and Beyer [4] fabricated columns without a fixed steel frame: Their robot designs double as a temporary frame for section under fabrication at the moment.

The *Spatial Winding System* [11] and *Spatial Lacing Robot Concept* [47] are robotic systems that use a steel frame *encompassing* the

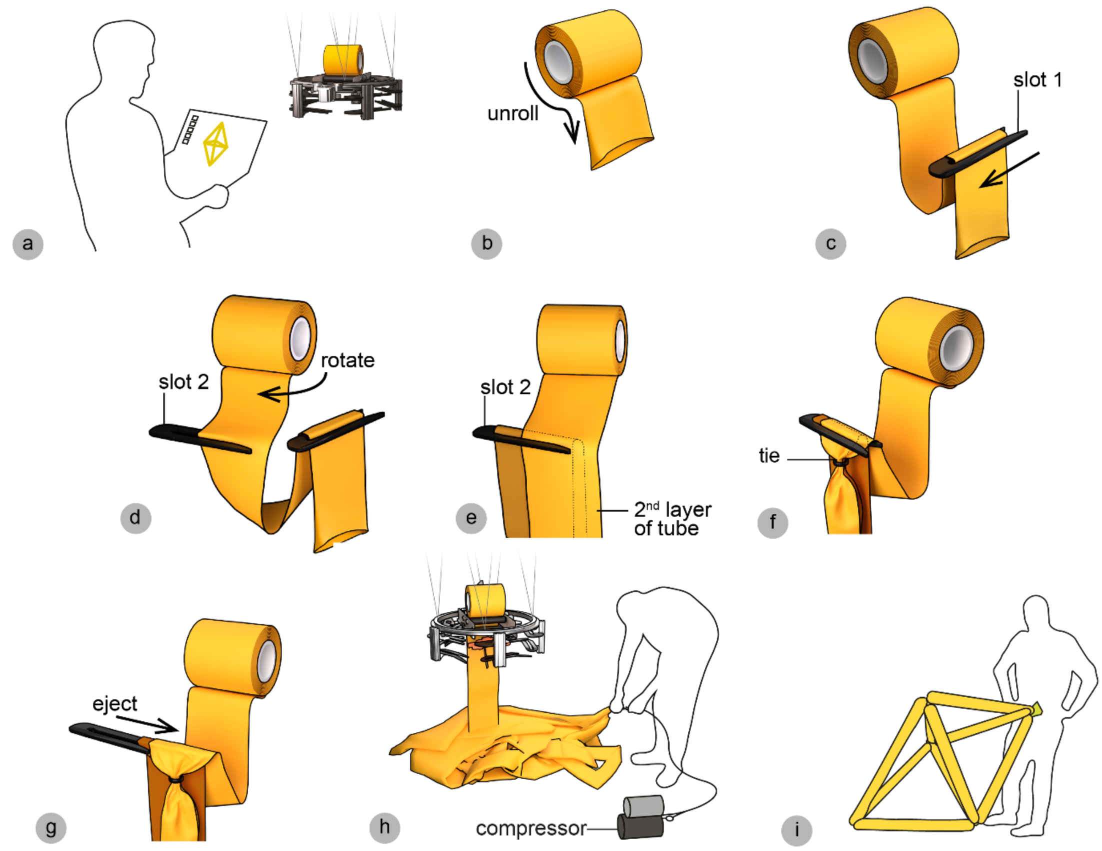

Figure 2: (a) Users set off the fabrication process with their design: The AirTied device starts by (b) unrolling and (c) bookmarking the first segment of tube using a *bookmarking slot.* (d) It repeats this for other segments on different bookmarking slots. (e) To form a connection, the AirTied device inserts a second segment on top of the existing one, (f) ties both together, (g) and ejects the completed node. (h) Once AirTied completes the last node, (i) users inflate the structure.

artifact. They use mobile robots, or an industrial robot arm / gantry combination to route an epoxy-soaked thread in 3D space.

One alternative to scaffolding is to produce truss members using very large robot arms that essentially act as very large 3D printer (*metal 3D printing trusses* [18]).

While the scaffold-based approach is very common in architecture, one would arguably want to see the scaffolding (or robot arm) to be set up (and removed) automatically as well. AirTied overcomes this by creating truss structures without scaffolding.

3.5 Automatic Assembly from Pre-Made Truss Components

Doggett et al. [9] assemble pre-made components using a stationary industrial robot arm. In this work, the robot arm picks up node and edge components from a magazine and clips them together on a turntable platform. Leung et al. [25] applied robotic assembly to pre-made wooden beams with lap joints.

BILL-E [15] and the *Material-Robot-System* [16] use two-legged mobile robots that can pick up a pre-made truss voxel—8 x 8 x 8 cm—from a magazine, "climb" the structure with it and place it in its designated position. This affords structures larger than the size of the original robot.

In contrast, AirTied constructs trusses from a generic material, i.e., a tube, thereby constructing its trusses end-to-end. This also allows AirTied to produce trusses with a variety of different edge lengths, graph structures, and angles.

4 THE AIRTIED SYSTEM

In the following, we give an overview of how AirTied fabricates truss structures from a single tube.

4.1 Walkthrough

As illustrated by Figure 2a, users start by modelling a truss structure using our custom editor. Its *exporter* converts the model into executable instructions and sends them to the AirTied device.

(b) In order to fabricate the structure, AirTied device starts by unrolling some tube. (c) AirTied pushes the first tube segment into a physical bracket, which we will refer to as a *bookmarking slot* since it marks the start of a segment. (d) AirTied now unrolls more tube and holds on to it using a second bookmarking slot. AirTied continues this process and fills another two bookmarking slots (the double tetrahedron requires a minimum of three bookmarking slots). (e) When AirTied re-encounters the second node bookmarked earlier, AirTied inserts the tube again into the second bookmarking slot, forming a second layer, (f) wraps a wire-tie around both tube layers, and (g) ejects the completed node from the bookmarking slot. After similarly constructing the remaining five nodes, the AirTied system ejects the last node, completing the model. (h) The user now cuts off the tube, attaches a compressor using a hose clamp, sealing off the tube (as shown in Figure 8), and (i) inflates the structure.

The supplemental video, available on the DOI-URL of this paper, shows the process on the AirTied device.

4.2 The Bookmarking, Tying, and Ejecting Mechanism

Figure 3 shows the AirTied device. Its central element is a mechanism that allows it to "bookmark" and tie tube segments.

AirTied achieves this with the help of two major functional groups. We reveal them by removing the *tube unroller*, i.e., two actuated rubber rollers that the spool rests on. (b) **Group 1:** A set of (here, five) *bookmarking slots* are organized in a rotationally symmetric arrangement around the perimeter of the device. Each slot features a *lock*, i.e., a small servo-actuated lever that allows fixating the tube in the slot. (c) **Group 2:** The *rotary mechanism* rotates the components located at the center of the device, i.e., the *tube unroller*, the *tube pusher*, and the *wire tie gun*.

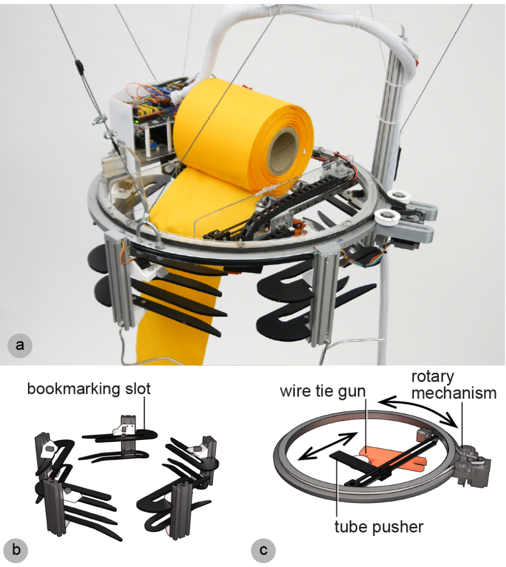

Figure 3: (a) The AirTied device with (b) multiple (here five) bookmarking slots for holding and tying material and (c) 2DOF rotary gantry to position the *tube pusher* and *wire tie gun*.

Figure 4 shows the bookmarking mechanism in use: (a) The *tube pusher* slides outwards to align the tube with the orbit of the bookmarking slot. (b) The device then rotates the tube into the bookmarking slot, retracts the pusher, and (c) the slot locks the tube in place by actuating its *lock*. Finally, it retracts the pusher to (d) unrolls the next tube segment. Since the lock holds the tube in place as it unrolls, the tube folds over the slot.

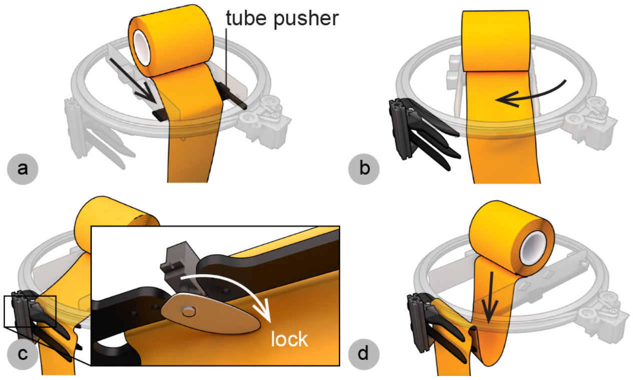

Figure 4: The device bookmarks a tube segment by (a-b) inserting it into a bookmarking slot, (c) locking, (d) and unrolling.

Figure 5 shows how AirTied brings together a second piece of tube to form a node. It does so by (a) inserting the second layer of tube into a bookmarking slot: (b) Since the tube pusher is mounted slightly above the slot, it can push the tube over the existing tube material on the slot. After (c) actuating the lock and unrolling the next piece of tube, two layers of tube are now on the same bookmarking slot on top of each other, prepared for tying. The device can add more layers for larger nodes in the same fashion.

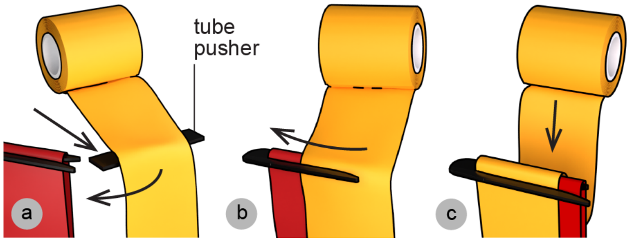

Figure 5: The device (a-b) inserts new material on top of the existing material and (c) unrolls to prepare tying multiple segments together.

Figure 6, finally, illustrates how AirTied completes a node: (a) When all intended layers of tube have been collected in a bookmarking slot, the device places the wire tie gun next to the slot. (b) It rotates the wire tie gun clockwise to funnel all layers in the bookmarking slot into the wire tie gun's "mouth". Finally, it (c) triggers the wire tie gun, which wraps a metal wire tie around all material contained in its mouth. (d) The device now rotates counterclockwise to eject the completed node.

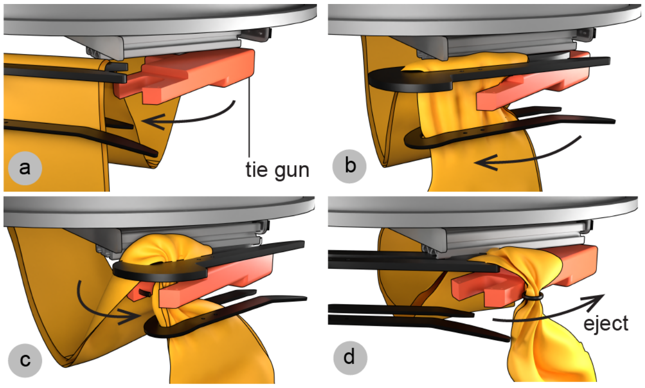

Figure 6: The AirTied device ties and ejects a node by (a) positioning the tying gun, (b) then squishing the material into the end effector to (c) wrap the wire around it. (d) Finally, the device ejects material from the slot using the back of the wire tying gun.

During usage, users suspend AirTied. As illustrated by Figure 7, this allows the tube material to drop and accumulate under the device. While not in use, AirTied can be carried, as shown in Figure 1 (60 x 60 x 50 cm, 14.0 kg).

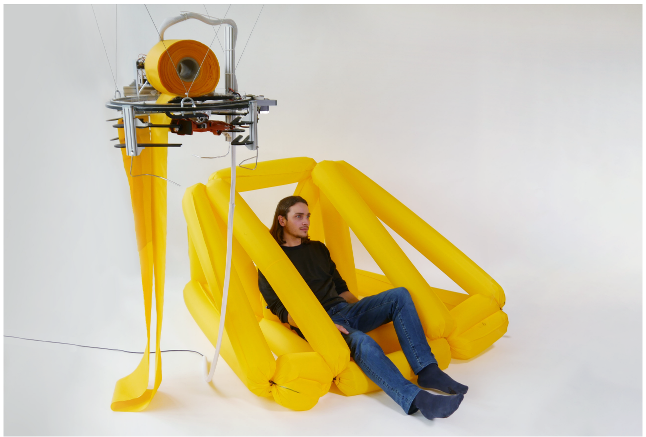

Figure 7: During use, we suspend the device to provide space for the dropping tube.

4.3 Inflating

Once fabricated, users inflate the truss using a compressor. To enable this, users insert an air inlet to either end of the tube, as shown in Figure 8. The inlet connects on one side to the fabric tube and on the other side to standard 6mm pneumatic tubes. Adding a check valve on the pneumatic tube leading to the compressor allows users to disconnect it again without releasing the air. Inflating very large structures, such as the tower from Figure 1, requires an additional inlet in the middle of the tube.

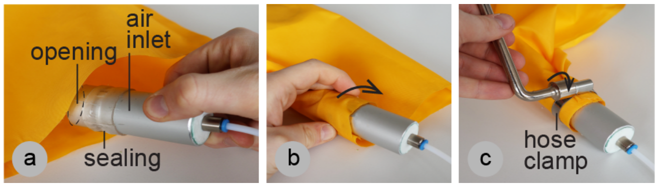

Figure 8: To install the air inlet, the user (a) inserts it into the tube, (b) wraps the tube around, (c) secures it with a hose clamp.

4.4 System Components

To help readers reproduce the device, Figure 9 summarizes AirTied's hardware design. The device uses conventional stepper motors for the rotary (*Stepperonline 23HS22-1504S*) and linear motion of the platform and for unrolling the tube (*Stepperonline 17HS15-1504S1*). The tying gun *F-Binder 350* from *Hana Corporation* [13] is used to form nodes. Standard RC servos are used to trigger the tying gun (*BMS-620MG*) as well as to hold the tube in the bookmarking slots (*MG90S*). The following components, designed to aid with material control, are made from bent aluminum: The *retainer* ensures that material unrolls downwards, the *clamping springs* fixate the material on the tube pusher for easing bookmarking, and the *kicker* guides around previous layers on the bookmarking slot using an actuated rod.

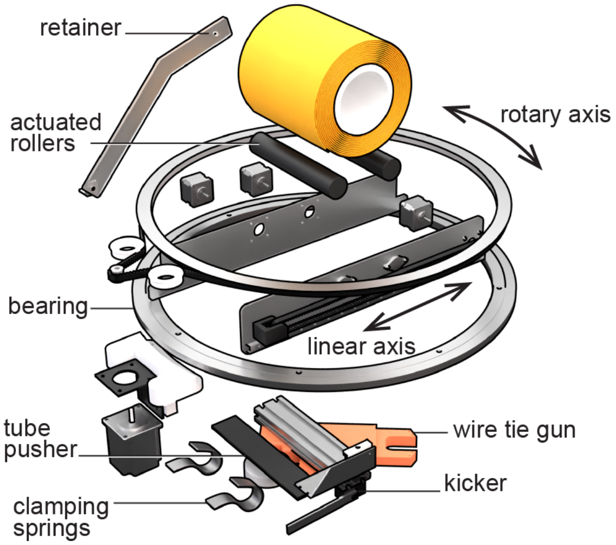

Figure 9: Exploded view drawing of the mechanical design, bookmarking slots removed for visual clarity.

AirTied is driven using an *Arduino Uno* with a custom version of the *GRBL* CNC control firmware [39]. GRBL is responsible for interpreting G-Code commands from a controlling PC over a USB serial connection, planning the motion paths, and forwarding appropriate control signals to the stepper drivers. We extend GRBL to enable servo motor support over a dedicated driver board and expose homing functionality as a G-Code command. The latter enables homing of the two axes during operation using limit switches, as step loss can occur while tying. For streaming G-Code to the device, we use the *Universal G-Code Sender*-package [50].

The yellow tube material is a woven nylon fabric with a thermoplastic polyurethane (TPU) coating. The coating serves both as the sealant layer for the woven fabric, as well as the bonding layer for welding. (210 den, TPU-coated one side, 275g/sqm). The 20 cm wide tube weighs 110g per meter. Other materials with similar friction and stiffness properties, such as polyethylene (PE), are also suitable.

While many tube materials are ready for use, this specific one comes in the form of 150cm-wide sheets. To turn it into a tube, we score the fabric in 20cm intervals at the beginning of the roll and rip it into seven strips. As we are working with a woven fabric, ripping produces perfectly straight edges along the full length of the strips. We then layer strips pairwise on top of each other—coated side on coated side—and weld them along both edges with a heat sealer (*airCano device* [38]).

For the valves, we use standard 6mm pneumatic tubing and *IQSDRV60* throttle check valves. We use an *Airpress HL 360 50 compact* compressor.

5 INTERACTIVE EDITING FOR AIRTIED

Our interactive editor, shown in Figure 10, allows novice users to design structures for AirTied. We build the editor based on the *TrussFab* editor [23], which in turn builds on the general-purpose 3D authoring tool SketchUp [46].

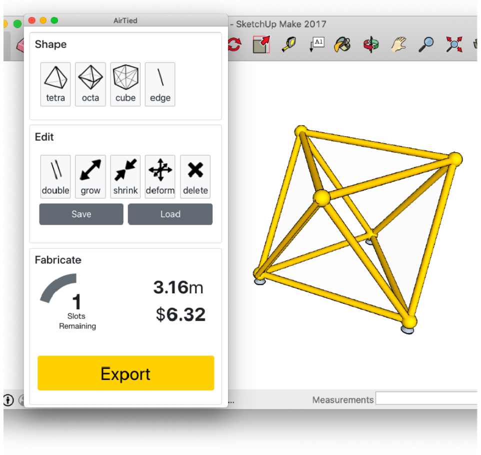

Figure 10: AirTied editor application to design structures and real-time preview fabrication parameters such as tube length, cost, and remaining bookmarking slots.

Figure 11 shows a user modeling a 5m "Loch Ness" monster for an outdoor waterpark. (a) The user selects the *octa tool* and fleshes out the rough shape of the monster by attaching octahedron onto octahedron. (b) Next, the user picks the *tetrahedron tool*, but when the user (c) tries to use it to attach wings to the monster, the system responds with an error (by turning the clicked triangle surface red), indicating that this addition would requires more bookmarking slot than currently configured (see Section "The Algorithm"). The user could upgrade to a device with more bookmarking slots and update the editor's settings accordingly. Here, however, the user decides to instead focus on the monster's mouth by adding two tetrahedra here and (e) adjusts the head's pose using the *shrink edge* tool.

(f) Finally, the user clicks *export*. Our system responds by displaying estimates for time, length of required material, and cost. It then automatically converts the model into instructions for an AirTied device, as we describe in the following.

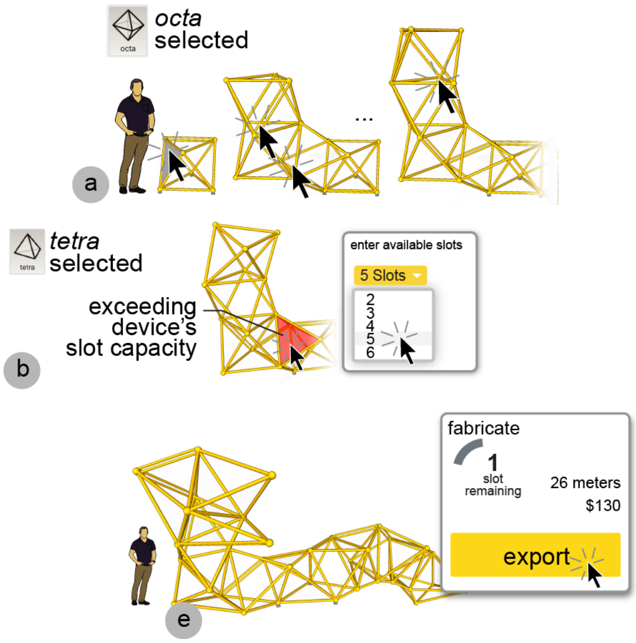

Figure 11: Constructing a water park monster for AirTied

6 THE ALGORITHM

The *AirTied exporter* converts AirTied models into instructions for an AirTied device.

The AirTied exporter accepts as input Wavefront OBJ from a standard 3D modelling tool and JSON models from the AirTied editor. Both formats encode models as a list of edges corresponding to adjacent nodes and a list of nodes described by coordinates. Node coordinates implicitly contain topology information, i.e., how long every edge is and how edges are oriented with respect to each other in space. The AirTied exporter uses this information to verify that the generated fabrication instructions will produce the correct topology.

Based on this input, computation takes place in three main steps: (1) The AirTied exporter generates a continuous *path* for the tube that visits every edge of the graph at least once, while minimizing the number of bookmarking slots, (2) the AirTied exporter *maps* every node in the graph to a bookmarking slot, and finally, (3) the exporter generates the machine instructions (G-Code) to drive the AirTied device.

6.1 Step 1: Generating the Path while Minimizing the Number of Required Bookmarking Slots

In this first step, the AirTied exporter generates a continuous path through the graph that traverses each edge. This represents the order in which the device unrolls and ties the tubes. A continuous path ensures that the model will be created in a single go.

Our algorithm builds on *Fleury's algorithm*, which is a method that finds a continuous path that traverses each edge of a graph *exactly once* (aka. the *Eulerian* path) [49]. Fleury's algorithm works by traversing the graph along edges not yet visited that do not disconnect the graph.

However, the majority of AirTied models do not have an Eulerian path. As illustrated by Figure 12, our algorithm resolves this by introducing double edges for graphs that do not have an Eulerian path.

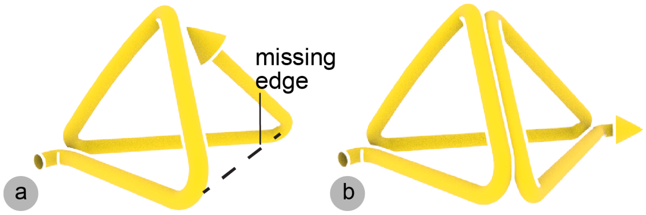

Figure 12: (a) While the unmodified Fleury's algorithm would run into a dead-end on graphs without an Eulerian path, (b) AirTied exporter greedily doubles edges in these situations.

To enable complex models on AirTied devices with small numbers of bookmarking slots, the AirTied exporter now optimizes its path so to minimize the number of required bookmarking slots. Our algorithm, for example, allows fabricating the 13-node tower model from Figure 1 on AirTied devices with 4 or more bookmarking slots.

Our algorithm minimizes the number of bookmarking slots in concurrent use by completing and ejecting nodes whenever possible to allow reuse of the bookmarking slot, as illustrated by Figure 13. Our algorithm accomplishes this by focusing its traversal of the graph on nodes that are already "in progress". It does so by running a breadth-first-search at every node during the path generation, thereby prioritizing edges that lead to the closest "in progress" node.

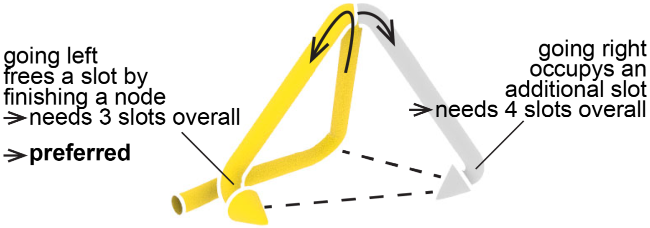

Figure 13: Finishing the node on the left before starting the node on the right saves one slot.

The algorithm employs a second heuristic to minimize slot usage by deliberately picking a starting point for the path: Models with a high width-to-height ratio, such as the tower from Figure 1, benefit from picking a starting point at the top or the bottom of the model. This avoids paths that revisit sections of the model that are only partially fabricated. The algorithm accomplishes this by computing all distances between nodes using the *Floyd-Warshall Algorithm* [8] and picking one of the nodes furthest apart as a starting node.

6.2 Step 2: Mapping Nodes to Bookmarking Slots while Avoiding Tangling

The AirTied exporter now maps nodes to bookmarking slots on the machine. Figure 14 shows why proper node to slot mapping is essential, as the same path can either be mapped to (a) a valid topology or (b) a topology subject to tangling.

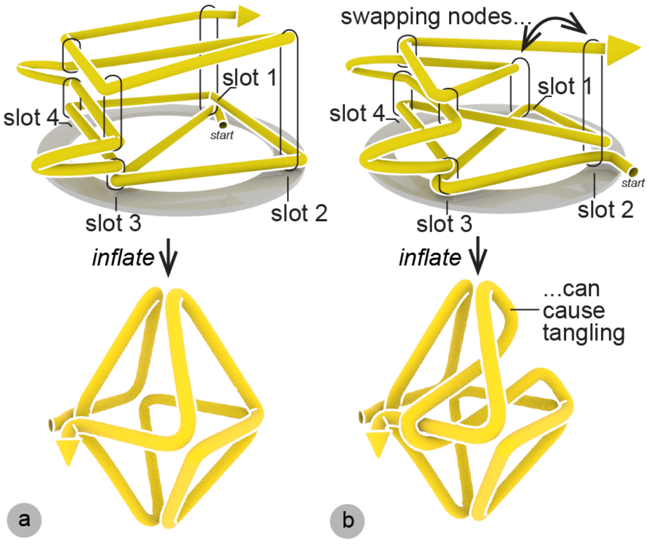

Figure 14: (a) Our algorithm maps nodes to bookmarking slots to avoid (b) tangling.

Our algorithm starts by creating a naïve node-to-slot-mapping. Starting with a default state in which nodes have not yet been mapped to bookmarking slots, the algorithm traverses the path and maps new nodes to free slots as it encounters them (e.g., node A→ slot 1, node B → slot 2, etc.). When it completes a node, i.e., encounters a node for the last time, it reuses the slot for new upcoming nodes. Once it has traversed the entire path, every node has a corresponding slot, and the mapping is complete.

Next, our algorithm exporter then detects tangling. It does so by anticipating the topology that the current mapping will produce during fabrication: as illustrated by the upper row of Figure 14, the algorithm "simulates" fabrication by constructing a line along the generated path, passing through the locations of the respective bookmarking slots. The algorithm now compares whether the users' design and the simulated result have the same *Yamada Polynomial* [29] (Python implementation [33]). The Yamada Polynomial is the same if both are topologically equivalent, i.e., the fabrication result can be transformed into the target model without cuts.

If Yamada Polynomials differ our algorithm has detected tangling. It resolves it by generating a new mapping that maps nodes to other slots and rechecks for tangling until it finds a valid mapping.

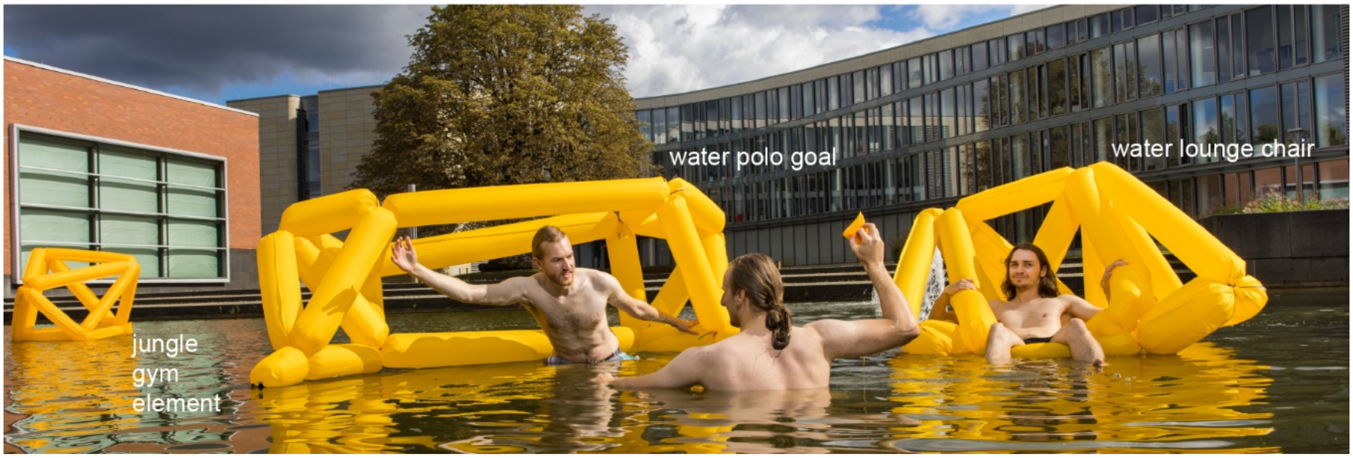

Figure 15: Water polo scene fabricated using AirTied

6.3 Step 3: Generating G-Code

Finally, our algorithm converts the output of the previous step into the G-Code to drive the AirTied device. It accomplishes this by traversing the path and creating *bookmark* and *unroll* statement for every node along the path, followed by a *tie* and *eject* statement, in case the respective node occurs for the last time in the sequence. It maps these instructions to G-Code using parametrizable string templates for each operation and streams the resulting G-Code file to the AirTied device.

6.4 Computation Complexity

The computational complexity of the path generation step of our algorithm is O(n^2), where n is the number of edges in the model. This complexity can be broken down as follows: Floyd-Warshall runs in O(n^2). The algorithm then traverses the path in O(n) and runs in every traversal step a graph disconnection check (O(n)) and a breadth-first search (O(n)). This generally allows us to run our algorithm in real-time and thus interact with users at interactive rates, e.g., to provide warnings during editing (Figure 11c).

For detecting tangling there is, however, no known polynomial runtime algorithm to validate the topology of a generated truss structure against the users' model. While computing the Yamada Polynomial has exponential runtime complexity, our demo models containing only 6-13 nodes can be checked in less than 2 minutes on a 2018 ThinkPad X1 laptop.

For larger models, the linking invariant approach described by Spinos et al. [41] can be used as an alternative to the Yamada Polynomial for comparing spatial graph topologies. While it does not provide the same knot-theoretic guarantees, it works well in practice and is significantly faster.

7 TECHNICAL EVALUATION

To evaluate the functionality of the system, we fabricated five models of varying complexity, tube length, scale, and bookmarking slot requirements, i.e., the three models from the water polo scene shown in Figure 15 and the tower shown in Figure 1. AirTied fabricated all models in 15-33min with an additional 1-7min for inflating the model, as shown in Table 1.

Table 1: Overview of fabrication results

	slots	fabricate	inflate
double tetrahedron 2.0 x 3.0 x 2.0 m, 5 nodes, 9 edges, 14m tube (1.5kg)	3	15min	40s
octahedron 1.7 x 1.4 x 1.3 m, 6 nodes, 12 edges, 12m tube (1.3kg)	5	18min	50s
water polo goal 1.0 x 2.4 x 0.7m, 8 nodes, 16 edges, 10m tube (1.1kg)	4	23min	2min
water lounge chair 1.2 x 1.5 x 0.9m, 12 nodes, 30 edges, 25m tube (2.8kg)	5	32min	3min
tower 2.7 x 2.4 x 6m, 13 nodes, 33 edges, 60m tube (6.6kg)	4	33min	7min

To test reliability, we furthermore produced 10 double tetrahedrons (19 operations each), resulting in 97% reliability per operation (i.e., bookmarking, tying, or ejecting).

Truss members of 1m length and 0.25bar of pressure exhibit a critical compression load of 0.34kN before buckling (as tested in a load test), and a tension load of 7.20kN (according to the material spec) before failing.

8 CONCLUSION

In this paper, we presented AirTied, a personal fabrication device that produces truss structures in fully automatic fashion. The main engineering contribution is our mechanism that creates nodes by holding onto a segment of tube, stacking additional tube segments on top of it, tying them up, and releasing the result. AirTied achieves this without scaffolding and without pre-made building blocks, thereby bringing truss fabrication into the reach of *personal* fabrication.

As future work, we are planning to explore ways of increasing the load-bearing capabilities of structures generated using AirTied.

ACKNOWLEDGMENTS

We thank Daniela Vogel for the renderings, Margarete Wohlleber and Jingchen Han for their expertise on tube production, Robert Kosmehl for helping on hardware design, Tarik Alnawa and Joana Bergsiek for working on simulations, Shohei Katakura for helping with the demo models, and Antonius Naumann and Julian Arnold for making a cold day in fall look like a hot day in summer while playing water polo. Furthermore, we would like to thank Mike Sinclair invaluable input during conception of the project.

REFERENCES

[1] Rahul Arora, Alec Jacobson, Timothy R. Langlois, Yijiang Huang, Caitlin Mueller, Wojciech Matusik, Ariel Shamir, Karan Singh, and David I. W. Levin. 2019. Volumetric Michell trusses for parametric design & fabrication. In *Proceedings of the ACM Symposium on Computational Fabrication* (SCF '19), Association for Computing Machinery, New York, NY, USA, 1–13. DOI:https://doi.org/10.1145/3328939.3328999

[2] Frederico Augugliaro, Sergei Lupashin, Michael Hamer, Cason Male, Markus Hehn, Mark W. Mueller, Jan Sebastian Willmann, Fabio Gramazio, Matthias Kohler, and Raffaello D'Andrea. 2014. The Flight Assembled Architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine* 34, 4 (August 2014), 46–64. DOI:https://doi.org/10.1109/MCS.2014.2320359

[3] Yuliya Baranovskaya, Marshall Prado, Moritz Dörstelmann, and Achim Menges. 2016. Knitflatable Architecture - Pneumatically Activated Preprogrammed Knitted Textiles. Oulu, Finland, 571–580. DOI:https://doi.org/10.52842/conf.ecaade.2016.1.571

[4] Bastian Beyer. 2018. project turm2. Retrieved from https://www.bastianbeyer.com/project-udk

[5] Yu-Wen Chen, Wei-Ju Lin, Yi Chen, and Lung-Pan Cheng. 2021. PneuSeries: 3D Shape Forming with Modularized Serial-Connected Inflatables. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (UIST '21), Association for Computing Machinery, New York, NY, USA, 431–440. DOI:https://doi.org/10.1145/3472749.3474760

[6] Hsu Yi Chia and Hsu Pei Hsien. 2018. The fabrication and application of parametric inflatable structure. *Blucher Design Proceedings* (November 2018), 684–689. DOI:https://doi.org/10.5151/sigradi2018-1609

[7] John Chilton. 1999. *Space Grid Structures.* Taylor & Francis Ltd.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. The Floyd-Warshall Algorithm. In *Introduction to Algorithms, 3rd Edition* (3rd Edition). The MIT Press, Cambridge, MA, USA, 558–565, 570–576.

[9] W. Doggett. 2002. Robotic assembly of truss structures for space systems and future research plans. In *Proceedings, IEEE Aerospace Conference*, 7–7. DOI:https://doi.org/10.1109/AERO.2002.1035335

[10] James F. Doyle. 1991. Truss and Frame Analysis. In *Static and Dynamic Analysis of Structures: with An Emphasis on Mechanics and Computer Matrix Methods*, James F. Doyle (ed.). Springer, Dordrecht, Netherlands, 66–96. DOI:https://doi.org/10.1007/978-94-011-3420-0_4

[11] Rebeca Duque Estrada, Fabian Kannenberg, Hans Jakob Wagner, Maria Yablonina, and Achim Menges. 2020. Spatial winding: cooperative heterogeneous multi-robot system for fibrous structures. *Construction Robotics* 4, 3 (December 2020), 205–215. DOI:https://doi.org/10.1007/s41693-020-00036-7

[12] Sharon Francis. 2019. *Bubbletecture: Inflatable Architecture and Design* (Illustrated Edition). Phaidon Press, London.

[13] Hana corp. Hana FB-350. Retrieved from http://www.r-binder.co.kr/

[14] Heimplanet Development GmbH. Mavericks. Retrieved from https://en.heimplanet.com/products/shelters-zelte-mavericks

[15] Benjamin Jenett. 2017. BILL-E: Robotic Platform for Locomotion and Manipulation of Lightweight Space Structures. In *25th AIAA/AHS Adaptive Structures Conference*, American Institute of Aeronautics and Astronautics, Grapevine, TX, USA. DOI:https://doi.org/10.2514/6.2017-1876

[16] Benjamin Jenett, Amira Abdel-Rahman, Kenneth Cheung, and Neil Gershenfeld. 2019. Material–Robot System for Assembly of Discrete Cellular Structures. *IEEE Robotics and Automation Letters* 4, 4 (October 2019), 4019–4026. DOI:https://doi.org/10.1109/LRA.2019.2930486

[17] Christopher Jenkins. 2001. *Gossamer Spacecraft: Membrane And Inflatable Structures Technology For Space Applications.* American Institute of Aeronautics and Astronautics, Inc. Retrieved from https://doi.org/10.2514/4.866616

[18] Alexander Hillary. Metal Printing into the Great Wide Open. Retrieved from https://alexanderhillary.com/metal-printing-into-the-great-wide-open

[19] Yasuaki Kakehi and Takahiro Hasegawa. 2016. *Single Stroke Structures.*

[20] Markus Kayser, Levi Cai, Sara Falcone, Christoph Bader, Nassia Inglessis, Barrak Darweesh, and Neri Oxman. 2018. FIBERBOTS: an autonomous swarm-based robotic system for digital fabrication of fiber-based composites. *Construction Robotics* 2, 1 (December 2018), 67–79. DOI:https://doi.org/10.1007/s41693-018-0013-y

[21] Marirena Kladeftira, Matthias Leschok, Eleni Skevaki, and Benjamin Dillenburger. 2021. Redefining Polyhedral Space Through 3D Printing. In *Advances in Architectural Geometry 2020*, Presses des Ponts, Online, 306–329. Retrieved from https://www.research-collection.ethz.ch/handle/20.500.11850/452213

[22] Marirena Kladeftira, Matthias Leschok, Eleni Skevaki, and Yael Ifrah. 2021. Digital Bamboo. *Biennale Architettura 2021,* Venice, Italy. Retrieved from https://dbt.arch.ethz.ch/project/digital-bamboo/

[23] 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* (CHI '17), Association for Computing Machinery, New York, NY, USA, 2606–2616. DOI:https://doi.org/10.1145/3025453.3026016

[24] Danny Leen, Raf Ramakers, and Kris Luyten. 2017. StrutModeling: A Low-Fidelity Construction Kit to Iteratively Model, Test, and Adapt 3D Objects. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (UIST '17), Association for Computing Machinery, New York, NY, USA, 471–479. DOI:https://doi.org/10.1145/3126594.3126643

[25] Pok Yin Leung, Aleksandra Anna Apolinarska, Davide Tanadini, Fabio Gramazio, and Matthias Kohler. 2021. Automatic Assembly of Jointed Timber Structure using Distributed Robotic Clamps. In *PROJECTIONS – Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design*, Association for Computer Aided Architectural Design Research in Asia, 583–592. DOI:https://doi.org/10.3929/ethz-b-000481928

[26] Chico MacMurtrie. 2008. *Inflatable Architectural Body.* Retrieved from http://amorphicrobotworks.org/inflatable-architectural-body

[27] MERO-TSK International GmbH & Co. KG. *MERO.* Retrieved from https://www.mero.de/en/

[28] Yuki Mori and Takeo Igarashi. 2007. Plushie: an interactive design system for plush toys. *ACM Trans. Graph.* 26, 3 (July 2007), 45-es. DOI:https://doi.org/10.1145/1276377.1276433

[29] Jun Murakami. 1993. The Yamada polynomial of spacial graphs and knit algebras. *Commun.Math. Phys.* 155, 3 (August 1993), 511–522. DOI:https://doi.org/10.1007/BF02096726

[30] Takumi Murayama, Junichi Yamaoka, and Yasuaki Kakehi. 2020. Reflatables: A Tube-based Reconfigurable Fabrication of Inflatable 3D Objects. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI EA '20), Association for Computing Machinery, New York, NY, USA, 1–8. DOI:https://doi.org/10.1145/3334480.3382904

[31] Ryuma Niiyama, Hiroki Sato, Kazzmasa Tsujimura, Koya Narumi, Young ah Seong, Ryosuke Yamamura, Yasuaki Kakehi, and Yoshihiro Kawahara. 2020. poimo: Portable and Inflatable Mobility Devices Customizable for Personal Physical Characteristics. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* Association for Computing Machinery, New York, NY, USA, 912–923. Retrieved from https://doi.org/10.1145/3379337.3415894

[32] Jifei Ou, Mélina Skouras, Nikolaos Vlavianos, Felix Heibeck, Chin-Yi Cheng, Jannik Peters, and Hiroshi Ishii. 2016. aeroMorph - Heat-sealing Inflatable Shape-change Materials for Interaction Design. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (UIST '16), Association for Computing Machinery, New York, NY, USA, 121–132. DOI:https://doi.org/10.1145/2984511.2984520

[33] Chad Peterson. 2023. Yamada: The Python Library for Calculating the Yamada Polynomial of Spatial Graphs. Retrieved from https://github.com/Chad-Peterson/Yamada

[34] Mariana Popescu, Matthias Rippmann, Andrew Liew, Lex Reiter, Robert J. Flatt, Tom Van Mele, and Philippe Block. 2021. Structural design, digital fabrication and construction of the cable-net and knitted formwork of the KnitCandela concrete shell. *Structures* 31, (June 2021), 1287–1299. DOI:https://doi.org/10.1016/j.istruc.2020.02.013

[35] Steffen Reichert, Tobias Schwinn, Riccardo La Magna, Frédéric Waimer, Jan Knippers, and Achim Menges. 2014. Fibrous structures: An integrative approach to design computation, simulation and fabrication for lightweight, glass and carbon fibre composite structures in architecture based on biomimetic design principles. *Computer-Aided Design* 52, (July 2014), 27–39. DOI:https://doi.org/10.1016/j.cad.2014.02.005

[36] Ali Sadeghi, Alessio Mondini, and Barbara Mazzolai. 2017. Toward Self-Growing Soft Robots Inspired by Plant Roots and Based on Additive Manufacturing Technologies. *Soft Robotics* 4, 3 (September 2017), 211–223. DOI:https://doi.org/10.1089/soro.2016.0080

[37] Harpreet Sareen, Udayan Umapathi, Patrick Shin, Yasuaki Kakehi, Jifei Ou, Hiroshi Ishii, and Pattie Maes. 2017. Printflatables: Printing Human-Scale, Functional and Dynamic Inflatable Objects. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), Association for Computing Machinery, New York, NY, USA, 3669–3680. DOI:https://doi.org/10.1145/3025453.3025898

[38] SebaPack GmbH. airCano Device. Retrieved from https://www.sebapack.de/en/inflatable-packaging-systems/air-cushion-machines/260/aircano-air-pillow-

machine

[39] Simen Svale Skogsrud and Sungeun Jeon. 2019. GRBL. Retrieved from https://github.com/gnea/grbl

[40] Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing inflatable structures. *ACM Transactions on Graphics* 33, 4 (July 2014), 63:1-63:10. DOI:https://doi.org/10.1145/2601097.2601166

[41] Alexander Spinos and Mark Yim. 2022. A Linking Invariant for Truss Robot Motion Planning. *IEEE Robotics and Automation Letters* 7, 2 (April 2022), 1424–1430. DOI:https://doi.org/10.1109/LRA.2021.3139941

[42] 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* (CHI '21), Association for Computing Machinery, New York, NY, USA, 1–12. DOI:https://doi.org/10.1145/3411764.3445311

[43] Ryo Suzuki, Ryosuke Nakayama, Dan Liu, Yasuaki Kakehi, Mark D. Gross, and Daniel Leithinger. 2020. LiftTiles: Constructive Building Blocks for Prototyping Room-scale Shape-changing Interfaces. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '20), Association for Computing Machinery, New York, NY, USA, 143–151. DOI:https://doi.org/10.1145/3374920.3374941

[44] Saiganesh Swaminathan, Michael Rivera, Runchang Kang, Zheng Luo, Kadri Bugra Ozutemiz, and Scott E. Hudson. 2019. Input, Output and Construction Methods for Custom Fabrication of Room-Scale Deployable Pneumatic Structures. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 62:1-62:17. DOI:https://doi.org/10.1145/3328933

[45] Shan-Yuan Teng, Cheng-Lung Lin, Chi-huan Chiang, Tzu-Sheng Kuo, Liwei Chan, Da-Yuan Huang, and Bing-Yu Chen. 2019. TilePoP: Tile-type Pop-up Prop for Virtual Reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (UIST '19), Association for Computing Machinery, New York, NY, USA, 639–649. DOI:https://doi.org/10.1145/3332165.3347958

[46] Trimble, Inc. SketchUp. Retrieved from https://www.sketchup.com/

[47] Cody Tucker, Xiliu Yang, August Lehrecke, Mathias Maierhofer, Rebeca Duque Estrada, and Achim Menges. 2022. A Collaborative Multi-robot Platform for the Distributed Fabrication of Three-dimensional Fibrous Networks (Spatial Lacing). In *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication* (SCF '22), Association for Computing Machinery, New York, NY, USA, 1–18. DOI:https://doi.org/10.1145/3559400.3561995

[48] Nathan S. Usevitch, Zachary M. Hammond, Mac Schwager, Allison M. Okamura, Elliot W. Hawkes, and Sean Follmer. 2020. An untethered isoperimetric soft robot. *Sci Robot* 5, 40 (March 2020), eaaz0492. DOI:https://doi.org/10.1126/scirobotics.aaz0492

[49] Robin J. Wilson. 2010. *Introduction to Graph Theory* (5th ed.). Prentice Hall, Harlow, United Kingdom.

[50] Will Winder. 2022. Universal-G-Code-Sender. Retrieved from https://github.com/winder/Universal-G-Code-Sender

[51] Haoran Xie, Yichen Peng, Naiyun Chen, Dazhao Xie, Chia-Ming Chang, and Kazunori Miyata. 2019. BalloonFAB: Digital Fabrication of Large-Scale Balloon Art. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (CHI EA '19), Association for Computing Machinery, New York, NY, USA, 1–6. DOI:https://doi.org/10.1145/3290607.3312947

[52] Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneUI: pneumatically actuated soft composite materials for shape changing interfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST '13), Association for Computing Machinery, New York, NY, USA, 13–22. DOI:https://doi.org/10.1145/2501988.2502037

[53] Jin Yu, Prabodh Sakhardande, Ruchita Parmar, and HyunJoo Oh. 2022. Strawctures: A Modular Electronic Construction Kit for Human-Scale Interactive Structures. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '22), Association for Computing Machinery, New York, NY, USA, 1–10. DOI:https://doi.org/10.1145/3490149.3501322