PneuMesh: Pneumatic-driven Truss-based Shape Changing System

Jianzhe Gu
jianzheg@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

Yuyu Lin
linyuyu@stanford.edu
Zhejiang University
Hangzhou, China
Stanford University
San Francisco, USA

Qiang Cui
cui-q18@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Xiaoqian Li
xiaoqian_li@mymail.sutd.edu.sg
Zhejiang University
Hangzhou, China

Jiaji Li
lijiaji@zju.edu.cn
Zhejiang University
Hangzhou, China

Lingyun Sun
sunly@zju.edu.cn
International Design Institute,
Zhejiang University
Hangzhou, China

Fangtian Ying
yingft@zju.edu.cn
Zhejiang University
Hangzhou, China

Guanyun Wang
guanyun@zju.edu.cn
Zhejiang University
Hangzhou, China

Lining Yao
liningy@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, USA

ABSTRACT

From transoceanic bridges to large-scale installations, truss structures have been known for their structural stability and shape complexity. In addition to the advantages of static trusses, truss structures have a large degree of freedom to change shape when equipped with rotatable joints and retractable beams. However, it is difficult to design a complex motion and build a control system for large numbers of trusses. In this paper, we present PneuMesh, a novel truss-based shape-changing system that is easy to design and build but still able to achieve a range of tasks. PneuMesh accomplishes this by introducing an air channel connection strategy and reconfigurable constraint design that drastically decreases the number of control units without losing the complexity of shape-changing. We develop a design tool with real-time simulation to assist users in designing the shape and motion of truss-based shape-changing robots and devices. A design session with seven participants demonstrates that PneuMesh empowers users to design and build truss structures with a wide range of shapes and various functional motions.

CCS CONCEPTS

• **Human-centered computing**;

KEYWORDS

Shape-changing Interface, Computational fabrication

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04.
https://doi.org/10.1145/3491102.3502099

ACM Reference Format:
Jianzhe Gu, Yuyu Lin, Qiang Cui, Xiaoqian Li, Jiaji Li, Lingyun Sun, Fangtian Ying, Guanyun Wang, and Lining Yao. 2022. PneuMesh: Pneumatic-driven Truss-based Shape Changing System. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3491102.3502099

1 INTRODUCTION

Among various shape-changing structures such as linear pin arrays [30], inflatable bladders [22, 37], and origami [28], meshes and trusses have been widely adopted for their advantages in modularity, reconfigurability, and high volume-weight ratio. Mesh or truss-based shape-changing structures have been seen as modeling toolkits computationally augmented for both sensing [16] and actuation [24, 33]. Mesh or truss-based structures have been used to construct dynamic [14] and static [36] artifacts and devices through different actuation and morphing techniques as well. Beyond HCI, truss structures are also commonly used in industry and architecture due to their structural stability and modularity [6, 11, 18]. A truss typically consists of multiple triangular units constructed from straight beams whose ends are connected at joints. By replacing passive beams with linear actuators that can change length independently, researchers enable a truss structure to locomote, change shape drastically, or manipulate objects [31]. Truss devices do not have specific morphology or motions: they can be assembled into arbitrary shapes, and each part of their bodies can be an actuator. By changing the body shape with a large number of actuators, truss devices can execute various motions, including linear or volumetric scaling, rotation, twisting, and adapting to different environments.

Despite their versatility and adaptiveness, truss devices suffer from an increasingly scaled complexity of the control system. As each beam is controlled independently, the number of control units (e.g. air tubings, wires, motors) is proportionally scaled with regard

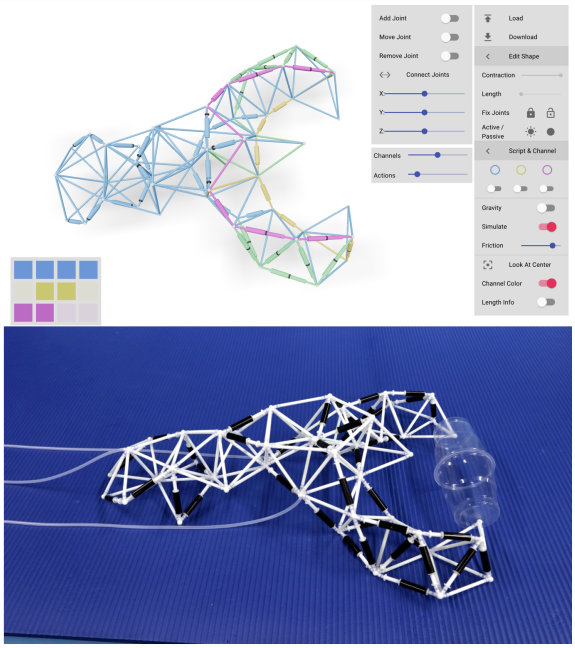

Figure 1: PneuMesh consists of a computation design platform and a set of novel hardware design improvements for Truss-based morphing and locomotion robots.

to the complexity of shape and motion. To reduce the control complexity, researchers use a small number of beams [34], or set most of the beams as passive units [14]. However, these design strategies often compromise the resolution of the shape (i.e., the number of the truss units), or the complexity of the motion. In this paper, we seek an approach that can give us a certain degree of control over every individual beam but still keep a relatively simple physical setup.

We propose PneuMesh, a pneumatic truss-based shape-changing system that can achieve multiple tasks with a small number of control units (Figure 1). PneuMesh is composed of adjustable pneumatic linear actuators (beams) with air channels inside, multi-way joints that direct air to separate channels, and airflow valves controlling each channel. We introduce 1) a partial connection strategy of dividing the entire truss structure into multiple *channel compartments*, with each compartment containing multiple beams that share an airflow valve and connect through algorithmically generated joints, and 2) a replaceable stopper structure that controls the minimum beam length under negative air pressure by restricting contraction. The interconnected beams within one compartment are actuated simultaneously yet have individually reconfigurable contraction ratios. By carefully designing the connection strategy, we allow users to actuate a large number of beams (up to more than 118) with only a few air channel controllers (fewer than or equal to four in this paper), but still create rich motions through the combinations of stopper locations and airflow signals. For example, a six-leg walker can achieve forward motion plus left and right turns with only two air ports and two tubings (Figure 2). Trussformer [14] showed a similar design with 12 individually controllable linear actuator units with correspondingly higher wiring and control complexity, cost, and weight.

As the number of beams and time span increase, the search space of the stopper positions, channel connectivity, and control signals scales up proportionally. Thus, we build an online design tool that allows users to edit and simulate the device motions without try-and-error in the physical world.

Our main contributions are as follows:

- A multifunctional truss-based shape-changing system with small numbers of air channels as the control. The system can act on various motions through adjusting beam contraction ratio, channel configuration, and airflow control signals.
- A computational design tool that assists users to edit, simulate and export the shape-changing device.
- Digital designs and physical prototypes that demonstrate the design potential of the system.

2 RELATED WORK

2.1 Pneumatic Shape-changing Interface

Pneumatic interface design is a fast-growing sphere in HCI. Many previous works [10, 22, 29, 37] have demonstrated shape-changing and deformable interfaces through pneumatically-actuated soft composite materials like rubber, paper and fabrics using actuation and stiffness control techniques. Taking it a step further, pneumatic interfaces have been studied which support sensing [21], haptic feedback [7], tunable stiffness [5, 23] and physical affordances [32]. Other researchers applied feedback mechanisms to improve the health of people through rehabilitation and physical fitness [25].

In terms of the shape design and control strategy, most of the existing works can be divided into two types: one type of structure is fabricated with a specific shape following the targeted function (e.g., a haptic glove that consists of a grid of air bubbles in [22]), thus requiring only a single air channel; the second type consists of

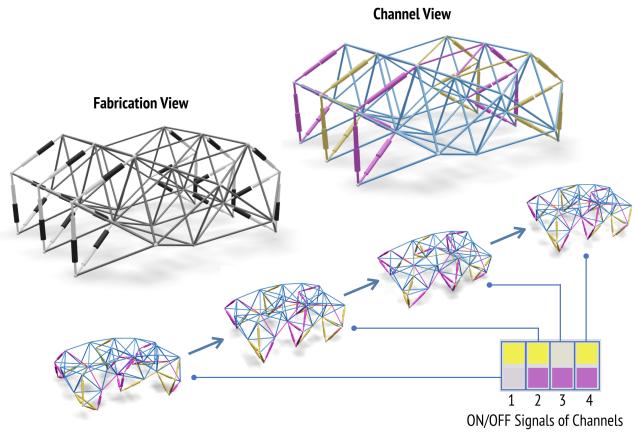

Figure 2: A six-leg walker with only two channels compartments can move ahead effectively, designed in the PneuMesh platform. In contrast, most typical six-legged robots have more control units (12 linear actuators in TrussFormer [14])

multiple modules with each module individually controlled by an air channel [21, 32]. While the modular approach provides flexibility and shape reconfiguration, it often requires many air tubes. To leverage the advantages of the modular design yet tackle the challenge of complexing tubing arrangements, our research explores the pneumatic actuation control on piston actuators to form a truss structure deformable robotic system. The additional design features such as channel compartment and adjustable beam contraction ratio help to minimize the total number of channels needed. For example, we have demonstrated that a truss structure made of 24 active linear actuators only needs two air channels to achieve target locomotion (Figure 2).

2.2 Truss Robots and Morphing Structure Design

Variable geometry trusses (VGT) [27, 35] have been introduced previously to HCI for computational fabrication [15] and large-scale kinetic structure designs [14]. VGT structures change member beams' lengths resulting in changes in geometry, and have been proposed as robotic manipulators [12], locomotion systems [39], and adaptive morphing structures [2, 8, 31]. Some truss robots are developed to be partially modular and reconfigurable as well, such as Morpho and Tetrobots [9, 38]. Traditional VGT robots are limited by the small extension ratio of the linear actuating beams. The untethered isoperimetric robot from Usevitch et. al.[34] is an example of a pneumatic reel actuator that is an extensible design of a self shape deforming robot used for modular locomotive functions based on its high extension ratio. Roller modules were also used to design compressible truss robots [34].

However, while a handful of virtual VGTs have been envisioned with a great geometrical complexity [20], the physical prototypes are often relatively simple and consist of a much smaller number of beams or units. This is due to multiple requirements of the actuators, including a high extension ratio, a decent load carrying capacity, and a lightweight structure. Also, if each actuator has to be controlled individually for each beam, wiring becomes very complex for a VGT structure that consists of tens or hundreds of beam units.

In our work, we try to balance multiple factors including the controllability (i.e., controlled extension ratio of each beam), the shape complexity (i.e., aiming at structures that consist of large numbers of beams), and the lightweight control system with minimized wiring tasks. We do so by introducing the idea of channel compartments and manually reconfigurable contraction ratio of each beam.

2.3 Computational Design of VGT and Other Robots in HCI

In robotics, a lot of computational design efforts for VGT robots focus around control policy optimization [1, 17]. In the context of HCI, the major focus has been around user-centered interactive tools that provide both simulation and fabrication instructions, such as in Trussformer [14]. In addition, HCI researchers have developed robotic systems for other interaction purposes including physical embodiments for mobile communication agents by mimicking animal motions [19], educational and tangible toys[26], and construction toolkits[24].

Our computational workflow is highly inspired by some existing computational tools for designing robotic systems in HCI. For example, the keyframe animation interface is inspired by tools such as Trussformer [14] and the movement pattern sequencer in [19]. Our idea of building up a truss structure with a user-directed forward process is also inspired by Trussformer. However, Trussformer limited the truss units to tetrahedra and octahedra to ensure decent structural stability, whereas we allow users to build up the structure with individual beams to allow larger geometrical flexibility. (Our idea of the channel compartment can also help simplify the control setups.) Additionally, while the majority and default beams in Trussformer are passive, our beams are all active by default, which supporting a larger shape transformation potential. Lastly, while the design goals between Trussformer and our system overlap in targeting truss-based shape-changing structures, Trussformer mostly emphasizes large-sized kinetic structures as its target application context, while we target shape transformation and locomotion behaviors. Our structures are made to be lightweight and small, and we envision some future application examples such as tabletop educational toys that kids can assemble at home, or an animated tabletop agent that can represent digital information (such as a voice agent) and interact with humans.

3 PNEUMESH SYSTEM

We present PneuMesh, a pneumatic truss-based shape-changing system that can achieve multiple tasks with a small number of control units. To change beam lengths on the same channel, we first introduce a passive stopper structure that stops the contraction of the beam at a specific length. The stopper can be manually replaced to adjust the contraction ratio of the beam without any controller modules. Second, to reduce the complexity of the control system of traditional truss devices without losing shape-changing capacity, we introduce the partial connection strategy: each air channel connects a portion of the beams through multi-way joints, where the interconnected beams are actuated simultaneously but independently from beams on other channels. The combination of air channel connection, stopper locations and air flow signals enable PneuMesh to achieve multiple complex motions with a limited number of air channels. Below we will describe the four main design features in more details:

(1) Adjustable Contraction Ratio of Active Beams. Our linear actuators are constructed with a shell, a piston, an adapter, and a stopper structure that can be relocated on the piston (Figure 3a). Each stopper can be manually plugged in to one of three grooves on the 3D printed piston such that the contraction of the piston will be stopped at the stopper location (Figure 3b). In short, when being inflated, every beam actuator will extend to the same full length. When deflated, they will contract by a certain percentage based on the constraint position. Based on our design and measurement, the actuator can be set to one of four different contraction ratios under deflation as shown in Figure 3c. Figure 3d, e show the designed and fabricated single actuator unit, respectively. We demonstrate how

the adjustable contraction ratio of each beam can be leveraged to transform a tetrahedron in Figure 4a, b.

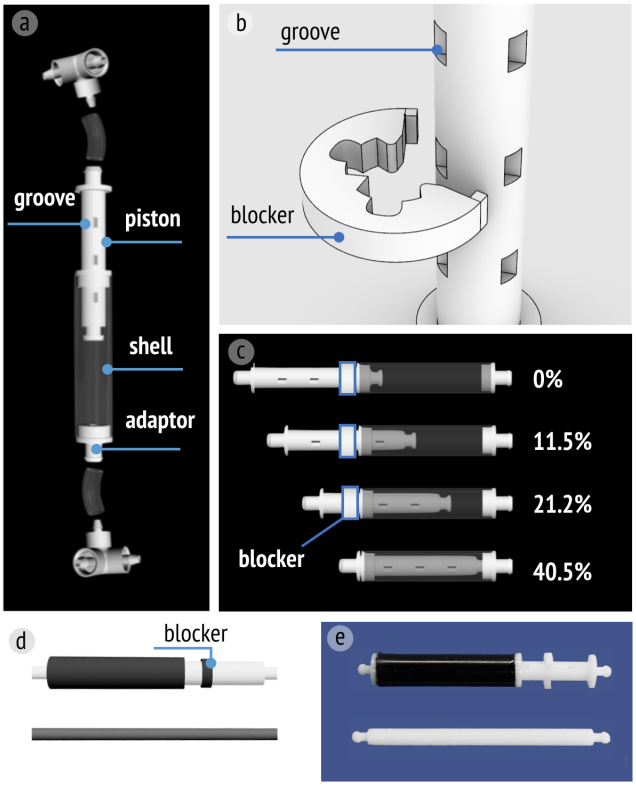

Figure 3: (a) Each pneumatic linear actuator is composed of a shell, a customized piston, and a tubing adaptor. Flexible rubber tubing connects the adapter to the multiway joints. (b) The stopper structure can be placed onto any of the three grooves on the piston to control the contraction ratio in (c). (d) The physically fabricated structure. (e) The computer model of the actuator unit.

(2) Passive Beams. To accommodate complex geometry while simplifying the fabrication tasks, we allow users to convert actuation (active) beams that have been set with full contraction into geometrically equivalent passive beams, Figure 4c.

(3) Reconfigurable Channel Compartment. We design 3D printable multi-way joints each allowing different channels to go through separately.

(4) Control Signals. Since we allow reconfigurable channel compartments, each compartment can be actuated separately. Users can design a set of motions or transformations sequentially by merely changing the control signals which change the state (ON or OFF) of each compartment (Figure 5).

Through 3D printable multi-way joints, our system allows multiple channel compartment design. Users can individually actuate each compartment, or combine multiple compartments at the same time. Users can also manually adjust the contraction of every single beam with the adjustable actuator design. With the combination of

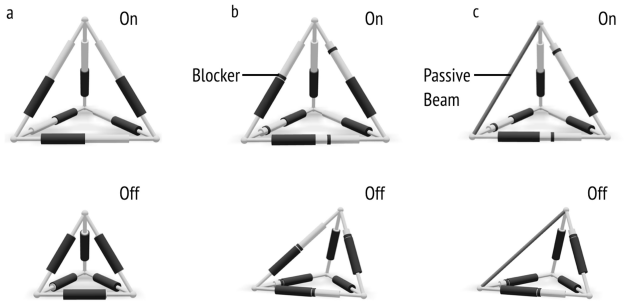

Figure 4: Examples of different design variables enabled by the platform. (a) A tetrahedron with no blockers (top) uniformly contracts after deflation. (b) A tetrahedron with blockers deforms into an irregular shape. (c) An actuator with full contraction is replaced with a passive beam and keeps the same transformation. This is our strategy of keeping the desired transformation behaviors as designed while decreasing the weight of the structure.

the channel connection and stopper replacement, users can explore various geometries and shape-changing behaviors.

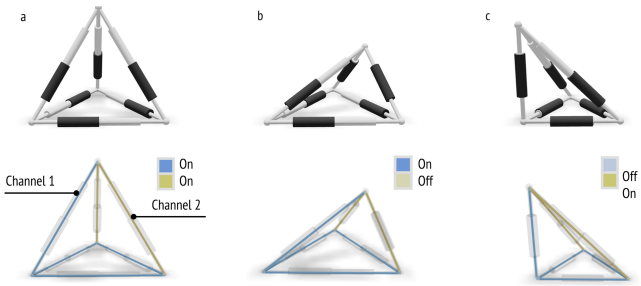

Figure 5: Partial connection strategy. A tetrahedron is composed of beams from two channel compartments. (a) Both channels are actuated. (b) Channel 2 is deflated. (c) Channel 1 is deflated.

4 USER WORKFLOW

To assist novice users to design PneuMesh artifacts, we developed an online design tool that allows editing and simulating the PneuMesh setting, as well as exporting 3D printable joint models and assembly instructions. We will go through the user's workflow in the bellow section.

Step 1: Shape design. Users can build arbitrary truss structures by using five interactive tools iteratively (Figure 6). The shape design starts with a single tetrahedron under the deflated status. Users can 1) click "Add Beam" to add one actuator with a new joint onto an existing joint, 2) drag the new joint to change its location, 3) select multiple joints connect an isolated joint to other joints with new beams (the joint position would automatically adjust such that beams are of the same minimum length, 4) undo/redo edits with "ctrl-z" and remove a joint and all connected beams with

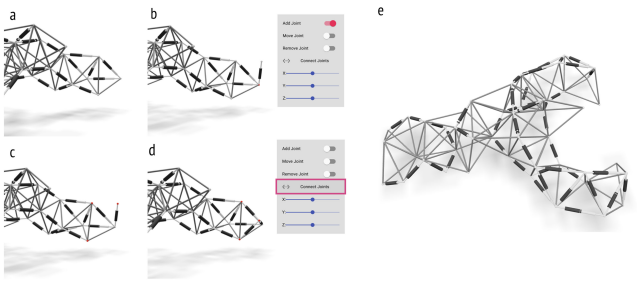

Figure 6: Shape editing function. (a-b) Choose "Add Joint" and click a joint to append a beam. (c-d) Choose multiple joints and click "Connect Joints" to add beams between each pair of selected joints. Iteratively doing (a-d) and eventually finish a shape design, e.g., a lobster (e).

"Remove Joint", and finally 5) fix the positions of joints by selecting joints and clicking "Fix", which is useful when creating stationary applications.

Step 2: Channel compartment configuration. Users can assign each beam with a channel compartment to connect by selecting the beams and corresponding compartments (Figure 7). A channel validation algorithm will reject edits causing discontinuous channels (Figure 7 c). Users can change the air port locations of each channel by selecting a joint and the corresponding air channel of the port.

Step 3: Stopper locating and length setting. Users can change minimum beam lengths as well as the PneuMesh shape at fully contracted status by either setting a beam as passive by clicking "Active/Passive", or changing the contraction ratio of an active beam. For passive beams, users can further change the static length of the beam. For active beams, a stopper will be visualized on the corresponding position (Figure 8).

Step 4: Airflow signal editing. The signal editor is in the left bottom corner of the screen. Users can first change the number of time frames of the signal, with each time frame lasting for 2.5 seconds (500 simulation time steps). Users can then toggle the inflation/deflation status of each channel at every time frame. The PneuMesh will repeat the control signals according to the script (Figure 9).

Step 5: Simulate. If "Simulate" is toggled on, a mass-spring-based numerical simulator will run in the background and animate the PneuMesh structure. The simulator includes the tensile energy of beams, bending energy of joints, the air volume in the channels. The terrain mode includes gravity, ground, and friction. Real-time simulation can be run at any step during the design process, allowing interactive design.

Step 6: Export Fabrication Instructions. The "Export" function uses an air channel generation algorithm to create 3D printable models of multiway joints incorporating our optimized channel geometries (Figure 10). The air channels are routed to prevent them from colliding or intersecting. To enable assembly without it "being a puzzle," a number is shown on each printed joint model as well as in the design tool. Users can assemble the PneuMesh referring to the numbers on the joints and the positions of stoppers.

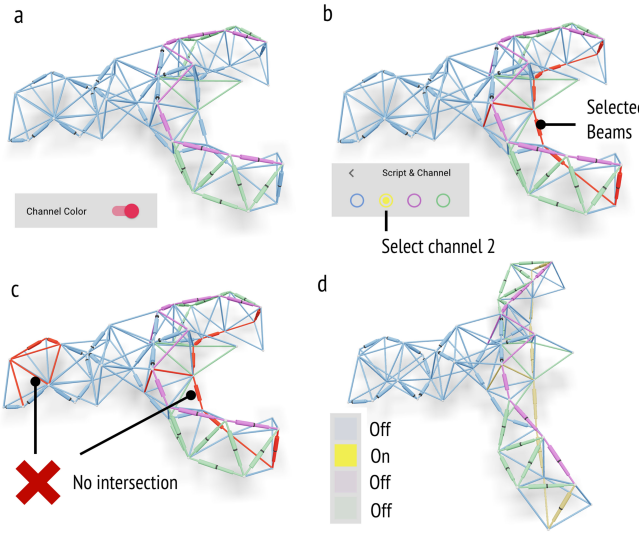

Figure 7: Channel setting and validation. (a) Check "Channel Color" to visualize the channel compartment assignment. Beams of the same color belong to the same compartment and will be actuated by a single air channel. (b) Choose beams and assign them with corresponding channel compartments by clicking the compartment icons. (c) Beams belonging to the same channel compartment must be connected through neighboring joints. (d) Users can then switch on the control signal of each channel channel compartment and simulate the transformation.

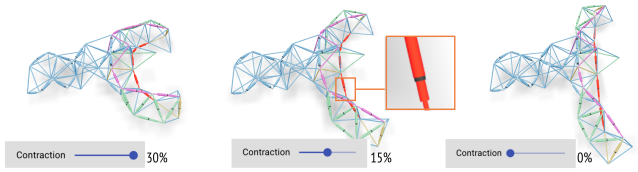

Figure 8: Beam contraction ratio. By selecting actuators and moving the slider of "Contraction", users can change the contraction ratio. The location of the stopper is shown in black.

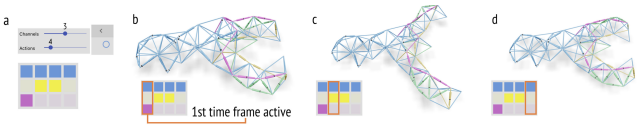

Figure 9: Airflow control signals. Users can edit the control signal panel to control the inflation/deflation status of each compartment at each time frame. The colors of the squares correspond to the color of channel compartments. Each column indicates a time frame.

Step 7: Physical Assembly and Control. Figure 11 show the fabricated and assembled structure of a lobster. By manually switching the blockers of the yellow channel, the lobster can be converted

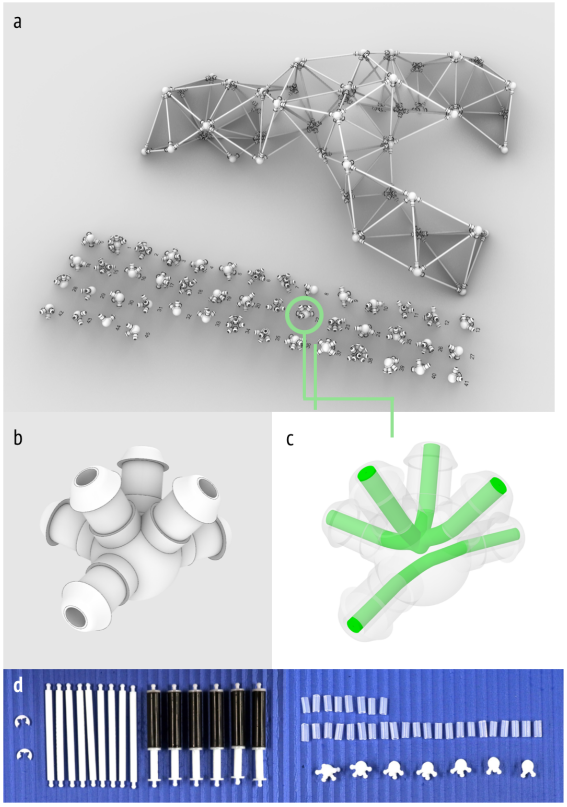

Figure 10: (a-c) Multi-way joints generated for additive manufacturing. (d) Hardware components are fabricated before the final assembly.

between two modes. By receiving the actuation signal for different channels, the lobster can enact 1) grabbing plus slowly moving (Figure 11 a) in the first mode or 2) quickly moving in the other mode (Figure 11 b).

5 DEMONSTRATION OF DESIGN SPACE

Although the basic building block is simply a length-changing beam, when a large number of such beams assemble and move programmatically under a temporal sequence, the design space increases exponentially. As Figure 12 shows, PneuMesh can potentially be used to design structures for locomotion, manipulation, and shape transformation. The shape change behavior can also act as a way to interact with the environment, people, or other objects nearby.

Pill-bug We showed a basic crawling robot designed to move forward with three control channels (Figure 13). The bug uses crawling motion and gait to locomote on the ground.

Turtle The turtle shows the capability of switching locomotion behavior by merely changing the control signals Figure 14. In this case, neither the contraction ratio of the beams nor the initial geometry (physical assembly) has been changed. Figure 14 b also shows additional sensors, such as an IR sensor, that can be attached to the robot to make it interact with the human hand.

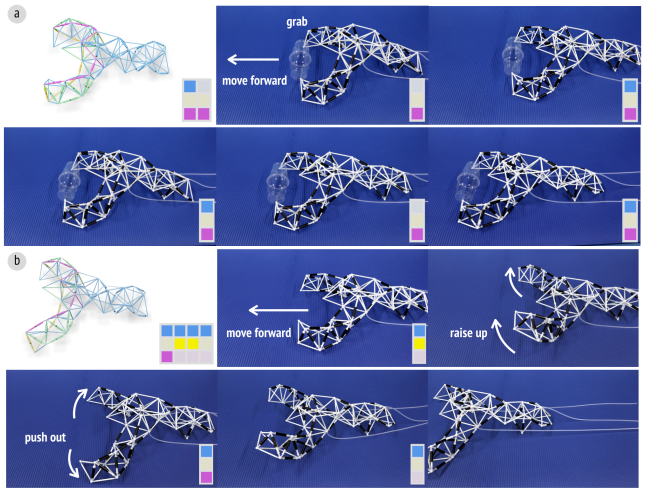

Figure 11: (a) With the top-left configuration, The lobster grabs two cups and slowly moves forward (12 cycles, 47 seconds). (b) The lobster quickly moves forward (5 cycles, 22 seconds).

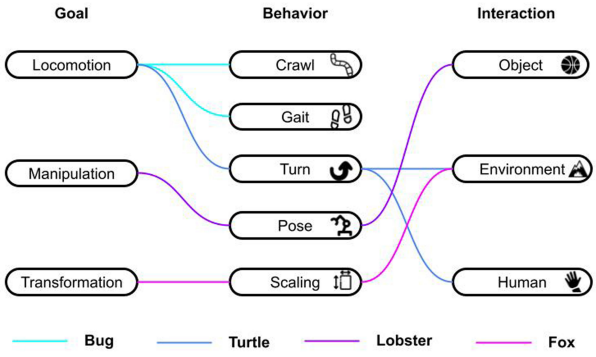

Figure 12: The design space of PneuMesh for different goals, behaviors, and interactions. Lines with the same color indicate the design space of one of the four shapes (bug, turtle, lobster, and fox).

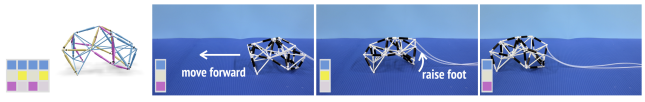

Figure 13: A crawling pillbug robot moving forward.

Fox We implemented the fox to show the capabilities of switching transformation behaviors by changing both the contraction ratio of selective beams and the control signals. While Figure 15 a shows the fox is moving forward, Figure 15 b shows that the fox can transform shape by bending down its head to travel through a constrained space.

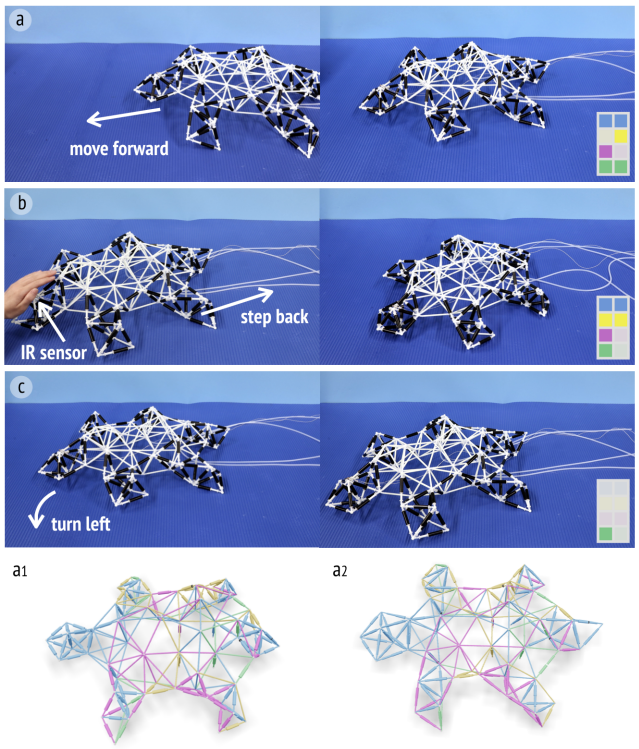

Figure 14: The turtle switches locomotion modes (a - c) by the changes of control signals.

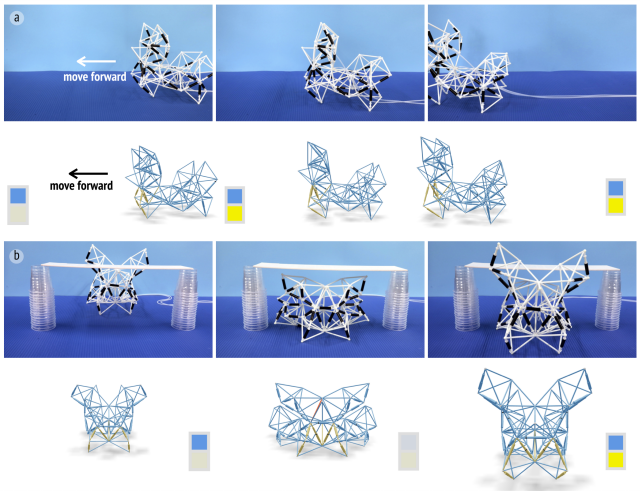

Figure 15: The fox switches geometry and locomotion by changing control signals. (a) Moving ahead. (b) Lowering the head and traveling through a constrained space.

6 QUALITATIVE DESIGN SESSION

To validate the usability of PneuMesh, we conducted a qualitative design session with seven participants. In the design session, users were asked to create shape-changing devices with the PneuMesh design tool and share their feedback to reveal existing problems or show future improvements.

Participants. We recruited seven participants(two males, aged 19-26). Four of them were students majoring in industry designs, one in architecture, one in material science, and one in art. Four participants reported to have basic CAD design experience and three were proficient in CAD modeling and design. During each session, each participant created one to four designs except for P7 who didn't finish the design within the given time.

Process. The design session took 2 hours. Participants first went through a tutorial to understand the basic knowledge of the shape-changing truss and the challenges when building large shapes. They were then asked to build three basic shape-changing structures (twisting column, bending strip, expanding sphere) to get familiar with the tool and its functions. Next, users were guided to build several basic shapes and get familiar with the tool. After that, they were asked to build their own design. Following the design session, participants went through a semi-structured interview. Questions included: "What problems did you find?","Could you create the design you had in mind?", "Without PneuMesh, how would you design the device you want?".

Overall, participants responded positively to PneuMesh's design tool: "I appreciate the tool being intuitive and easy to use, but can still create many complicated designs" (P3). "Pretty interesting project that gives the user easy access to the design of a complex interface." (P6). "I really enjoy the pipeline of interactively editing and seeing the result in the real-time, it made my life easier in designing shape-changing interface". (P1). They found the tool "easy and fun to learn" (P2). They thought "the color-coding of different channels and the correspondence between the channel color and the control script block color gives an intuitive visual cue that helps me to understand the channel connection strategy and improve my design." (P4)

Learning curve. Participants improved greatly during the design session. In under two hours, P1 was able to finish five designs of increasing shape and motion complexity. (Figure 16 P1). Participants started to understand "how to build and allocate actuating beams if I want to build a twisting beam or a bending flower after using the tool for a while" (P2).

Design space. Participants appreciated the large design space of the system and the design tool that lowers the barrier to the PneuMesh system. To build the same design without PneuMesh, P4 would "first use some plasticine and sticks to prototype the shape and then use Arduino with motor and linkage system to implement." But they could imagine that would require "tedious and repetitive design iterations" (P4). P1 mentioned that without the editing tool, they would use a traditional CAD tool to sketch and iterate on the shape, but it would be non-intuitive and error-prone as the structure scales up and the number of units increases.

Feedback and improvements. Participants also pointed out existing problems and gave suggestions for new features in the tool. One main complaint was the lack of common CAD functions, for example, mirroring and trimming (P7, P3, P2). Users mentioned that the process of building a large number of beams was tedious, and suggested a tool that automatically converts mesh into tetrahedrons(P7, P5). They also mentioned that it would be helpful to give

a library of basic shapes to save time, such as twisting units, elongation units, and expansion units. Finally, participants proposed some inverse kinematics algorithms and optimization methods that could automatically generate the channel connection by assigning a target shape or target motion.

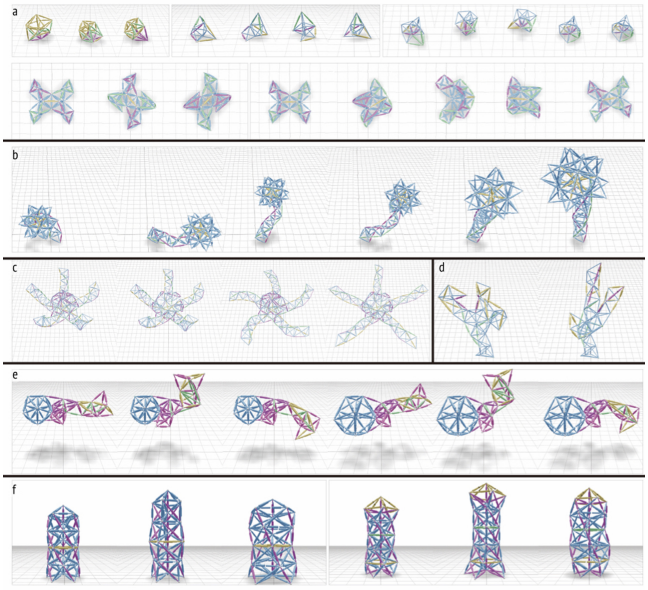

Figure 16: Designs created by six participants with the PneuMesh editing tool. P1 designed three self-rolling polygons and two dancing robots; P2 and P3 designed two flowers; P4 designed a claw; P5 designed a globefish; P6 designed two adjustable pillars.

Participants' designs. Figure 16 shows a series of designs by participants. We observed that users like to build things that are related to their life or work. For example, P6 is an architect who built Figure 16 P6 which represents an interactive column that changes height and diameter and can twist. It turned out that users who want to build a motion prefer changing the channel connection, while users who want to build specific shapes have a higher chance to change the contraction ratio. The reason might be that channel connection change gives a large change in the global shape and motion but might miss the shape detail, while contraction ratio change enables users to better control the shape details. We also noticed that participants like to use the tool to build bio-inspired objects that have organic shapes and irregular motions, such as flowers (Figure 16 P2, P3), globefish (Figure 16 P5) and claws (Figure 16 P4).

7 IMPLEMENTATION

7.1 Characterization

To determine the parameters of the simulation, we measured essential mechanical parameters of the basic units with a constant-pressure pump with positive air pressure at 7.5 ± 0.3 psi and negative pressure at -7.5 ± 0.3 psi.

Theoretically, the actuating force from the air pressure is constantly $f_a = p \cdot s$, where p is the air pressure and s is the inner cross section area of the linear actuator. If the loading force expanding (if the pressure is inner negative) or compressing (if the inner pressure is positive) the actuator is larger or equal to the actuating force, the actuating status of the actuator will be switched. To verify this theory, we first fixed the end of the linear actuators with negative pressure, applied an expanding/tension force with a thrust meter, slowly increasing the force (Figure 17 a). We then fixed the body of the linear actuators with positive pressure, applied a compressing force along the axis, and slowly increased the force (Figure 17 b). The results aligned with our assumption. We did 18 tests on three linear actuators, including nine with positive pressure and nine with negative pressure. The average maximum tension force for negative pressure is 9.2 ± 0.03 N and maximum compression force for positive pressure is 8.3 ± 0.03N.

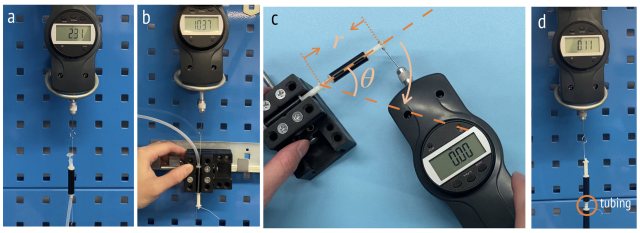

Figure 17: (a) The threshold tension force of the actuator through a tension test, (b) the threshold compressing force of the actuator through a compression test, (c) measurement of the maximum and minimum bending angles, and the bending force to deform the rubber tubing, and (d) measurement of the shear friction between the rubber tubing and joints.

The joint is composed of multi-way joints and flexible rubber tubings. The joints and the tubings are firmly connected by the friction and tension of the tubing. Each joint has a maximum and minimum bending angle and maximum shear friction. We first measure the maximum bending angle of the joints (Figure 17 c). We did 15 tests on five 3-way joints. We first fixed one side of the beam on the table, and slowly changed the angle of the free beam until it broke. We recorded the average maximum bending angle as 91.2 ± 0.2°. We then measure the maximum shear friction of the tube, over which the joint will detach (Figure 17 d). We did nine tests on three tubings and recorded the average maximum shear force is 3.42 ± 0.03 N with a standard deviation of 0.42N. The measured bending force (Figure 17 c) is smaller than 0.1 N at a radius of 51 mm, which is ignorable compared to the shear friction and is therefore left out of the simulation.

7.2 Simulation

We used JavaScript to develop a position-based dynamics model for PneuMesh structures. The PneuMesh structure uses rigid plastic for the beams and joints and flexible rubber tubing to connect the beams and joints. The self-deformation is driven by the pneumatic actuation force along the axial direction of actuators as well as

the bending resistance force from the tubing. However, due to the softness of the tubing and the structural stability of triangles, we consider the self-deformation only governed by the axial actuation force. In addition to the axial actuation force, the simulation also includes the ground contact and friction.

To simplify the model, we consider the beams as rigid bodies and joints as point masses. During simulation, we update the target length of each beam based on the actuation speed and update the joint positions following the length constraint. To achieve real-time performance, we allow deviation from the length constraint and use the beam strain energy as a soft penalty. We compute forces applied to joints by minimizing the beam strain energy and updated velocity and displacement of each joint by forward Euler integration.

Target Length. At every time step, each actuator has a target length l_g based on the volume inside the beam and the blocker position. We define $l_g = k_a l_M + (1 - k_a)(l_M \sigma_M, l_M - d_b)$, where k_a equals to 1 if the channel is under inflation or 0 otherwise, l_M is the maximum length of actuators, and σ_M is the maximum contraction ratio of actuators. d_b is the beam displacement which equals to $t_a \cdot v_b$, where t_a is the time of the current actuation, v_b is the expanding/contraction speed of beams. For passive beams, σ_M and d_b are constantly 0.

Beam Strain Energy. To keep the length of the beams approximating their target lengths, we introduce the strain energy $E_s = k_s(l - l_g)^2$, where k_s is the weight of the energy. In the experiment, we use a rubbery mat as the ground, which gives enough friction and stabilizes the device. Accordingly, we assume that the ground absorbs all impact by setting the z-component of velocity to 0 for the joints moving against the ground. We use Coulomb's model of friction with the static friction factor as 0.72 and the dynamic friction factor as 0.36.

Durability validation. Tubing detachment and the inversion of actuators' actuation are two factors that cause the most instability. The tubing is the weakest part of the PneuMesh structure, and it is fixed to adapter structures by friction and internal tension. The structure may break down due to the detachment of the tubing rather than other parts. Although the two parts of an actuator are considered to be rigid bodies, the movement between the two parts might be inverted due to the load on the beam (e.g. a tall structure gives a huge load on the beams at the bottom). Specifically, an inflating actuator can be compressed and a deflated actuator can be stretched. We only consider the former case because, under large tension, the tubing will be detached before the latter case happens. At the end of each time step, we validate the force applied on the beam and the tube. If they exceed the threshold value we found through characterization, the tool throws a warning. Similarly, the tool evaluates the bending angle of each beam attached to the joint and throws a warning once any angle exceeds the threshold angle.

Directional surface. We found that adding a directional friction surface 3D-printed with PLA, which only provides friction in one defined direction, on the bottom of the mesh makes the locomotion more efficient. To simulate it, for each joint with a directional surface, we calculate a forward direction by optimizing a vector that minimizes the angle difference between the vector and all the neighboring edges compared to the initial setting.

7.3 Joint Generation

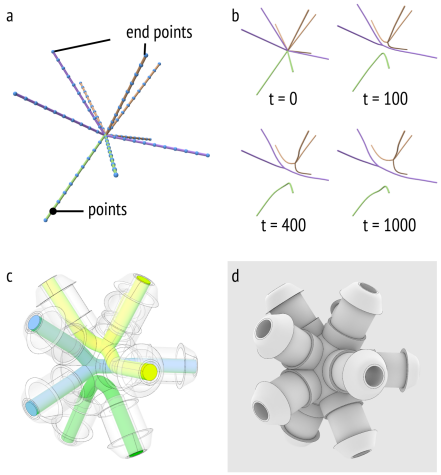

Figure 18: Joint generation. (a) The tunnels are initialized as straight lines sharing nodes in the middle, with endpoints fixed. (b) The optimization pushes channels apart, where t represents time steps. (c,d) Grasshopper generates the 3D model with separated channels for 3D printing.

A joint is composed of ports connecting beams to a joint sphere, and air tunnel networks inside the sphere. The air tunnel network consists of multiple air tunnels intersecting at one point, where each air channel connects one beam on the joint. Each air tunnel network connects all the beams belonging to the same channel. In other words, if beams on the joint belong to the same channel, only one air tunnel network will be generated, where all the air tunnels will simply intersect in the center and remain straight. Otherwise, multiple air tunnel networks will coexist inside the joint sphere and must be separated from each other. There are a few requirements to ensure joint quality. First, there is a minimum distance between tunnels from different networks to ensure isolation. Second, a larger radius of the channel gives rise to a larger airflow. Third, a shorter and straighter tunnel has a higher printing success rate and larger airflow. We first initialize all the air tunnels straight and connect them to the center point of the sphere. We then discretize each air tunnel into N points. Air tunnels in the same network have a shared point in the center of the sphere. We then optimize the nodes' locations by minimizing a weighted sum of tensile energy and repulsive energy $P = \text{argmin}_P \, k_e E_e + k_r E_r$, where P is the set of all point positions (Figure 18 a,b).

Tensile energy. We set tensile energy to every neighboring pair of points in the same network as $E_e = d^2$, where d is the distance between the points and is initialized as 0.05mm. Minimizing this energy gives relatively short and smooth channels, which have a higher fabrication success rate.

Repulsive energy. We set repulsive energy to every pair of points belonging to different networks as $E_r = -1/d, d < d_m, Er = 0, d \geq d_m$, where d_m is a minimum interference distance we set as 0.4mm. This resembles the electrostatic potential energy following Coulomb's law, which pushes networks apart from each other. The

cutoff distance reduces the computational cost considering the $O(N^2)$ complexity of the long-range force energy.

We do not update the position of the points on the end, which will be connected to the channel of beams. We used Python with the NumPy library to implement air tunnel optimization then store the result as a JSON file. We use Rhino 6 with Grasshopper to read the JSON file in real-time and generate the joint models for 3D printing (Figure 18 c,d).

7.4 GUI Implementation Details

We used JavaScript with React-Three-Fiber library (a WebGL-based graphics library) [4] to implement the design tool, simulator, durability validation, and GUI. The whole platform is implemented and tested on a quad-core Macbook Pro. The browser-based design tool has been tested on Google Chrome, Safari 14, and Edge. The code has been released open-sourced (https://github.com/riceroll/PneuMesh) and a demo can be accessed through (https://riceroll.github.io/pneumesh/).

7.5 Fabrication Details

For positive pressure, we used ASLONG AP-370 air compressor with $7.73psi$ maximum pressure, and $0.5 \sim 1.5L/min$; for negative pressure, we used PYP370 vacuum pump rated at $-7.15psi$ and $0.5 \sim 2.5L/min$. We used Arduino UNO and electromagnetic valves to control the pneumatic actuation. We used Z Rapid iSLA600 printer to print the joints, pistons, and adaptors and used silicone tubing with a 50 durometer and 1.5 mm inner diameter. We used polypropylene black pipe with 6 mm inner diameter and used a manual electric grinder machine to cut the pipes.

8 EVALUATION

8.1 Actuation Speed

The speed of actuation is affected by both the structural complexity and power of the pumps we used. In our test, we use the same physical setup detailed in the *Fabrication Details* section. Theoretically, the time from starting actuation to finishing equals $t_a = v/k$, where v is the volume of the channels, including the beams, joints, and connection parts, and k is the flow rate of the pump. In reality, the actuation time varies due to the friction and structure deformation as shape complexity increases. To verify our assumption and quantify the performance, we measured the time of actuation with regard to the number of beams. We used five structures, each with nine active actuators without blockers, as well as a varied number of passive beams. We tested the time of a cycle of full actuation. The time varies from 1.2s to 20.8s. We posit two major reasons for the variation of the actuation speed: 1) Fabrication inconsistency, including printing and assembly qualities. 2) Small distortion in the tubing changes the friction of the linear actuator which requires higher accumulated air pressure to actuate and takes more time.

8.2 Simulation Accuracy

We evaluate the accuracy of our simulation by measuring the difference of corresponding joint positions between experiments and simulations after actuation. For simple shapes such as a single tetrahedron, the experiment result aligns well with our simulation. As the complexity of the shape and time of actuation increases, the discrepancy between the experiment and simulation also increases. For example, for the fox in Figure 15, the position of the left-front node was displaced for (34.6 ± 1.8, -4.5 ± 0.4) mm after 5 cycles of actuation, while in simulation the locomotion is (36.4, 0.3) mm. The difference is 5.3% (compared to the total locomotion distance). To improve our accuracy in the future, in addition to implementing a high-order simulator with smaller time steps, we could add acceleration sensors to measure the location and adjust the simulator on the fly.

8.3 Fabrication Parameters

We reported the fabrication parameters of each application in Table 1. From the result, we can see that we use a very limited number of channels to actuate large numbers of actuators.

Name	T_build	T_design	# beam	# actuator	# joint	# channel	# shape	W
Lobster	2h	1.5h	137	44	46	4	8	179.1g
Turtle	3h	2h	201	107	63	4	6	296.6g
Fox	1.5h	0.5h	126	33	44	2	4	164.5g
Six-feet Robot	/	0.5h	101	24	33	2	2	/
Pill-bug	0.5h	0.5h	57	18	18	3	5	72.6g`

Table 1: The parameters of each application, including (from left) the building time, the time using the design tool, number of beams, number of linear actuators, number of joints, number of channel compartments, number of static shapes, and weights.

9 LIMITATIONS

Speed As reported previously, there is a limit in speed due to the flow rate of the pumps and the maximum air pressure the structure can withstand. Currently, the physical prototypes are aimed at tabletop interfaces, such as construction toolkits for design, education, or physically embodied agents. Higher speeds would require increasing the power of the pump and changing the design of the connecting parts to withstand higher air pressure.

Unit Size and Shape Resolution To increase the number of beam units and shape resolution for more complex geometries without sacrificing actuation speed, each unit must be miniaturized. Currently, the unit size is limited by both the beam and joint designs. Each requires an air channel; this channel cannot be too narrow due to both our 3D printing resolution and the required flow rate of air for a given actuation speed. Different printing methods, or different speed goals, could allow further miniaturization.

Asynchronous Actuation of Different Beam Units We have mentioned in our evaluation section how the speed of actuation could vary given the same beam number and geometrical complexity, and this is largely due to the fabrication quality and varied extent of deformation at joints (due to the different transformation design). Indeed, these two factors also cause some asynchronous actuations of different beam units. For relatively complex structures shown in this paper, some beams will take a longer time to actuate than

others. For now, our simulation does not take this factor into account, and this will sometimes cause discrepancies between the simulation results and the physical performance. We believe both improving hardware design (e.g., removing the soft tubing at each joint) and adapting more sophisticated fluid mechanics simulations to better capture how air flows within our channels will help tackle this challenge.

Load Carrying Capacity This is a lightweight structure and not optimized for loading. The load-carrying capability is largely limited by the soft rubber tube connections we are using. We may improve the joint design with methods introduced in previous truss-based robotics such as the flexible joint design in Trussformer[14], so we can have flexible joints with varying deformation angles yet being rigid and load-bearing. Instead of printing communicating tubes as joints, we 3D print a scaffold for structural stability and connecting flexible tubes for air to pass through respectively. However, a more sophisticated (or articulated) joint design may bring higher requirements for the printing resolution and limit the minimum size of our structure.

Optimization and Inverse Design

Since users might wish to design quite complex truss structures with many adjustable beam units, it could take the users a lot of trial-and-error in a forward design process until they achieve a transformation or locomotion behavior they like. In addition, what the users have designed may not be most optimized in the sense that fewer air channels may be able to reach the same locomotion behavior given more time to try other design options. For example, the lobster design took us about 1.5 hours. For the next step, to better facilitate the design process of complex truss structures, we will implement an inverse design process with tailored optimization in the future. Inverse kinematics [13] or evolutionary robotics design [3] are potential references.

10 CONCLUSION

In this paper, we showed a novel design concept of truss-based shape-changing systems with a small number of control units for complex geometries, morphing and locomotion behaviors. While this paper focuses on the computational design of basic shape and morphing (output), we see a great design potential to add interactivity to augment truss-based shape-changing structures. For example, as we briefly showed in our design example of the turtle, we could develop modular sensors as attachments to the structure and make the robots interactive to human inputs or environmental stimuli. Inspired by previous works that have shown how different morphing behaviors of shape-changing interfaces can convey different emotions, physically embody virtual information, and augment human-information or human-robot interactions, we could envision various desktop interface designs made with PneuMesh platform. We will further explore potential application scenarios and we believe the technique detailed in this paper will enrich the toolboxes for such shape-changing interfaces

REFERENCES

[1] Miguel Abrahantes, Aaron Silver, and Luke Wendt. 2007. Gait design and modeling of a 12-tetrahedron walker robot. In *2007 Thirty-Ninth Southeastern Symposium on System Theory*. IEEE, 21–25.

[2] Steven Curtis, Matthew Brandt, Greg Bowers, Gary Brown, Cynthia Cheung, Caner Cooperider, Mike Desch, Noah Desch, John Dorband, Kyle Gregory, et al. 2007. Tetrahedral robotics for space exploration. In *2007 IEEE Aerospace Conference*. IEEE, 1–9.

[3] Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret, and Agoston E Gusz Eiben. 2015. Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI* 2 (2015), 4.

[4] drcmda. 2021. React-Three-Fiber, github repository. https://github.com/pmndrs/react-three-fiber.

[5] Sean Follmer, Daniel Leithinger, Alex Olwal, Nadia Cheng, and Hiroshi Ishii. 2012. Jamming user interfaces: programmable particle stiffness and sensing for malleable and shape-changing devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 519–528.

[6] Kevin C. Galloway, Rekha Jois, and Mark Yim. 2010. Factory floor: A robotically reconfigurable construction platform. In *2010 IEEE International Conference on Robotics and Automation*. 2467–2472. https://doi.org/10.1109/ROBOT.2010.5509878

[7] Kristian Gohlke and Eva Hornecker. 2018. A stretch-flexible textile multitouch sensor for user input on inflatable membrane structures & non-planar surfaces. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. 191–193.

[8] Sachin Sean Gupta, Dhileep Kumar Jayashankar, Naresh D Sanandiya, Javier G Fernandez, and Kenneth Tracy. 2019. Prototyping of chitosan-based shape-changing structures. (2019).

[9] Gregory J Hamlin and Arthur C Sanderson. 2013. *Tetrobot: A modular approach to reconfigurable parallel robotics*. Vol. 423. Springer Science & Business Media.

[10] Takayuki Hirai, Satoshi Nakamaru, Yoshihiro Kawahara, and Yasuaki Kakehi. 2018. xslate: A stiffness-controlled surface for shape-changing interfaces. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–4.

[11] David Hjelle and Hod Lipson. 2009. A robotically reconfigurable truss. In *2009 ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*. 73–78.

[12] Peter C Hughes, Wayne G Sincarsin, and Kieran A Carroll. 1991. Trussarm—a variable-geometry-truss manipulator. *Journal of Intelligent Material Systems and Structures* 2, 2 (1991), 148–160.

[13] S Jain and SN Kramer. 1990. Forward and inverse kinematic solution of the variable geometry truss robot based on an N-celled tetrahedron-tetrahedron truss. (1990).

[14] Robert Kovacs, Alexandra Ion, Pedro Lopes, Tim Oesterreich, Johannes Filter, Philipp Otto, Tobias Arndt, Nico Ring, Melvin Witte, Anton Synytsia, et al. 2018. TrussFormer: 3D printing large kinetic structures. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 113–125.

[15] Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, et al. 2017. Trussfab: Fabricating sturdy large-scale structures on desktop 3d printers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2606–2616.

[16] Vincent LeClerc, Amanda Parkes, and Hiroshi Ishii. 2007. Senspectra: A computationally augmented physical modeling toolkit for sensing and visualization of structural strain. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 801–804.

[17] Woo Ho Lee and Arthur C Sanderson. 1999. Dynamics and distributed control of Tetrobot modular robots. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, Vol. 4. IEEE, 2704–2710.

[18] 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*. 471–479.

[19] Stefan Marti and Chris Schmandt. 2005. Physical embodiments for mobile communication agents. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. 231–240.

[20] Andrea Mazzone and Andreas Kunz. 2005. Sketching the future of the SmartMesh wide area haptic feedback device by introducing the controlling concept for such a deformable multi-loop mechanism. In *Proceedings of theFirst Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE.

[21] Ryosuke Nakayama, Ryo Suzuki, Satoshi Nakamaru, Ryuma Niiyama, Yoshihiro Kawahara, and Yasuaki Kakehi. 2019. MorphIO: Entirely Soft Sensing and Actuation Modules for Programming Shape Changes through Tangible Interaction. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 975–986.

[22] 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*. 121–132.

[23] Jifei Ou, Lining Yao, Daniel Tauber, Jürgen Steimle, Ryuma Niiyama, and Hiroshi Ishii. 2014. jamSheets: thin interfaces with tunable stiffness enabled by layer jamming. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*. 65–72.

[24] Amanda Parkes and Hiroshi Ishii. 2010. Bosu: a physical programmable design tool for transformability with soft mechanics. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*. 189–198.

[25] Panagiotis Polygerinos, Frederick Sebastian, Qiushi Fu, and Marco Santello. 2019. Soft robotic haptic interface with variable stiffness for rehabilitation of sensorimotor hand function. US Patent App. 16/212,205.

[26] Hayes Solos Raffle, Amanda J Parkes, and Hiroshi Ishii. 2004. Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 647–654.

[27] Marvin D Rhodes and MM Mikulas Jr. 1985. Deployable controllable geometry truss beam. (1985).

[28] Anne Roudaut, Abhijit Karnik, Markus Löchtefeld, and Sriram Subramanian. 2013. Morphees: toward high" shape resolution" in self-actuated flexible mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 593–602.

[29] 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*. 3669–3680.

[30] Alexa F Siu, Son Kim, Joshua A Miele, and Sean Follmer. 2019. shapeCAD: An accessible 3D modelling workflow for the blind and visually-impaired via 2.5 D shape displays. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 342–354.

[31] AYN Sofla, DM Elzey, and HNG Wadley. 2009. Shape morphing hinged truss structures. *Smart Materials and Structures* 18, 6 (2009), 065012.

[32] 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*. 143–151.

[33] Yasaman Tahouni, Isabel PS Qamar, and Stefanie Mueller. 2020. NURBSforms: A Modular Shape-Changing Interface for Prototyping Curved Surfaces. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 403–409.

[34] Nathan S Usevitch, Zachary M Hammond, Mac Schwager, Allison M Okamura, Elliot W Hawkes, and Sean Follmer. 2020. An untethered isoperimetric soft robot. *Science Robotics* 5, 40 (2020).

[35] Ben K Wada, James L Fanson, and Edward F Crawley. 1990. Adaptive structures. *Journal of Intelligent Material Systems and Structures* 1, 2 (1990), 157–174.

[36] Guanyun Wang, Humphrey Yang, Zeyu Yan, Nurcan Gecer Ulu, Ye Tao, Jianzhe Gu, Levent Burak Kara, and Lining Yao. 2018. 4DMesh: 4D printing morphing non-developable mesh surfaces. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 623–635.

[37] 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*. 13–22.

[38] Chih-Han Yu, Kristina Haller, Donald Ingber, and Radhika Nagpal. 2008. Morpho: A self-deformable modular robot inspired by cellular structure. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3571–3578.

[39] Juan Cristobal Zagal, Cristobal Armstrong, and Shuguang Li. 2012. Deformable octahedron burrowing robot. In *Artificial Life Conference Proceedings 12*. MIT Press, 431–438.