PopTuber: Discretely Reshaping High Resolution 3D Curves with "Serial Multistable" Tubes

Willa Yunqi Yang
University of Chicago
Chicago, Illinois, USA
yunqiy@uchicago.edu

Yifan Zou
University of Chicago
Chicago, Illinois, USA
anglezouyifan@uchicago.edu

Anup Sathya
University of Chicago
Chicago, Illinois, USA
anups@uchicago.edu

Ken Nakagaki
University of Chicago
Chicago, Illinois, USA
knakagaki@uchicago.edu

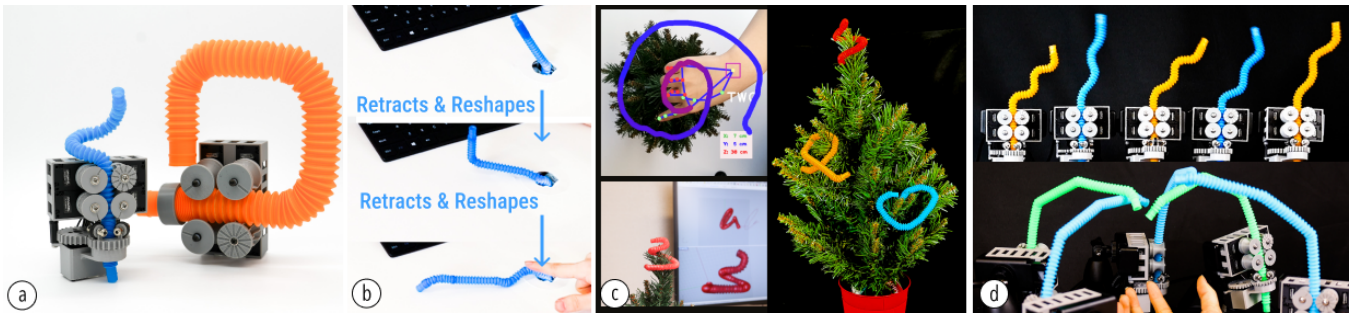


Figure 1: (a) PopTuber's shaping devices, (b) reconfigurable curve-based tangible UIs, (c) physicalizing mid-air drawing, (d) a multi-module shape display.

ABSTRACT

The vision of programmable matter — materials that can change shape or properties in a programmable way — has inspired decades of research across robotics, materials science, and HCI. However, many line-based actuated systems struggle to achieve fine-grained geometries, due to motor size, cost, and control complexity. In this work, we introduce PopTuber, a novel approach to dynamically generate high-resolution 3D curves using passive, low-cost, multistable pop tubes. Instead of chaining actuators, our system leverages the densely arranged bellows with discrete mechanical states, collapsed, expanded, or folded, to represent and construct arbitrary curves. We present the Shaper, a device that uses only five servo motors to reconfigure tubes of any length and curved segments. Alongside the hardware, we provide design guidelines that allow our shaping principle to scale for different tube sizes, a software pipeline that can simulate the tube geometry and execute motor cmmands, and an evaluation of shaping performance and scalability. Our work challenges conventional actuator-intensive and material-oriented approaches in programmable matter, showing how passive, discretely shapable materials combined with minimal actuation can enable versatile, low-cost shape transformations along with a variety of applications.

CCS CONCEPTS

• **Human-centered computing** → **Interaction devices**.

KEYWORDS

Electronics Prototyping Toolkits, Interaction Design

ACM Reference Format:
Willa Yunqi Yang, Yifan Zou, Anup Sathya, and Ken Nakagaki. 2026. PopTuber: Discretely Reshaping High Resolution 3D Curves with "Serial Multistable" Tubes. In *Twentieth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '26), March 08–11, 2026, Chicago, IL, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3731459.3773321

1 INTRODUCTION

In the past thirty years, the vision and concept of Programmable Matter have attracted attention across robotics, materials science, and Human-Computer Interaction (HCI), pursuing "materials that can change their physical properties or shape in a controlled, programmable fashion based on external input" [1]. When Programmable Matter was initially proposed by Toffoli and Margolus in 1991 [51], they envisioned such material to be composed of 'an ensemble of fine-grained computing elements,' which led to projects like claytronics that take modular, reconfigurable robotic approaches

[1] https://en.wikipedia.org/wiki/Programmable_matter

This work is licensed under a Creative Commons Attribution 4.0 International License.
TEI '26, March 08–11, 2026, Chicago, IL, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1868-7/26/03.
https://doi.org/10.1145/3731459.3773321

[17]. A number of HCI researchers were also inspired by such vision to construct and design interactive, shape-changing devices, a.k.a Shape Changing Interfaces (SCIs) [2, 23, 43] to broaden tangible and haptic interaction. While this vision branched into diverse form factors (self-foldable sheets [21], pin-array shape-changing surfaces [15, 29]), *lines* have always been a popular form of interest to create versatile motion or curve geometries [8, 9, 27, 50, 58]. Similarly, HCI researchers have also taken such 1D geometry to create dynamic line-based shape changing interfaces [34–36].

When developing an actuated linear system, one challenge is the number of small actuators required to achieve high-resolution geometries. Take LineFORM [36] (a line-based shape-changing interface in HCI) and Milli-Motein [27] (a miniature chain-type programmable matter system in robotics) as examples; the granularity and the smoothness of the linear geometries they can create are inherently limited by the actuators themselves. The fact that each joint requires an electromechanical actuator significantly limits how small the joint can be, and the costs increase with the line's length and granularity. Can we explore alternative strategies to mitigate the granularity and cost constraints introduced by actuators?

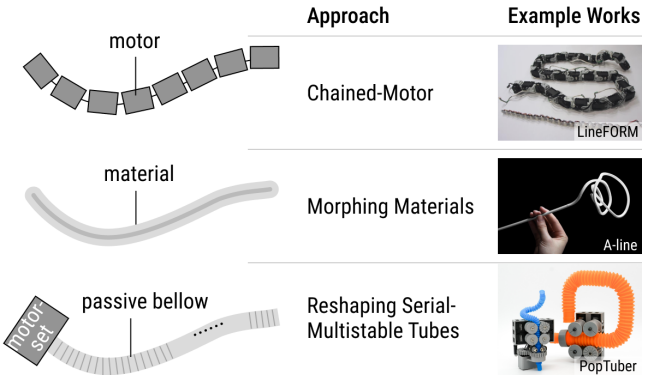


Figure 2: Illustrations of Chained-Motor approach and Morphing Material approach.

To this end, instead of the actuator-intensive approach that requires an actuator for each joint on a line, we explore a novel alternative that uses a shaping device to actively shape passive, connected joints. These passive joints take the form of serially connected multistable structures, which can be found on a widely available linear material, 'pop tubes.' Often sold as fidget toys, pop tubes create a satisfying 'popping' sensation when expanded and collapsed thanks to their multistable buckling behavior. They are mass-produced using corrugated tubing techniques developed since the 1960s,[2] making them widely accessible and inexpensive.

In this paper, we highlight the potential of leveraging pop tubes' unique geometry as a form of programmable matter. Although the pop tubes can be bent and shaped in a free-form way, what's special about pop tubes is that their global geometry is fundamentally determined by the serial arrangement of discrete local states—each bellow being either *collapsed*, *expanded*, or *folded*. These densely connected, 'discretely' defined local bellow states, resembling a bitstream, construct and determine the global geometry. This principle offers a new lens for constructing line-based programmable matter.

With this novel perspective, we have developed a generalizable method to build a 'Shaper' system, a device that dynamically shapes pop tubes based on digital curve data. We show a visual reference in Figure 2, which will be detailed in 2.1. In contrast to the conventional actuator-intensive approaches in programmable matter and SCIs, the Shaper only requires five servo motors to configure tubes of any length. It also offers an alternative to the traditional material-oriented approach, such as 4D printing [50, 55], pneumatic/hydraulic systems [26, 57], etc., which often face difficulties rendering diverse shapes on a material's *continuous* morphing body. Instead, we propose using *discretely* shapable materials with a fixed number of actuators to enable versatile shape transformations.

In what follows, we introduce the overall hardware design and shaping operations. We provide a device design guideline for pop tubes of different sizes, helping researchers, makers, and engineers replicate our approach using any tube available on the market. We also present a software pipeline that translates 3D digital curve data into tube simulations and motor commands. Our evaluation characterizes shaping performance on a small-sized tube (11mm diameter) and demonstrates the scalability of our shaping principle on a larger tube (29mm).

We make the following contributions:

- A concept and approach of employing passive multistable material to be shaped with a separate shaping device (as in Figure 2) as a new strategy of programmable matter.
- A proof-of-concept implementation, including device hardware design and a software pipeline.
- Guidelines and evaluation of shaping accuracy and system scalability for different tube sizes.

2 RELATED WORK

The interdisciplinary field of programmable matter has explored diverse ways to translate digital curve data into dynamic physical lines with controllable shapes. This section reviews approaches from the robotics, materials science, and HCI communities, and situates the proposed approach of PopTuber within this landscape.

2.1 Programmable, Shape-Changing Lines

2.1.1 Actuator-Focused Programmable Lines. Since the vision of Programmable Matter [51], a number of works have investigated approaches to enable lines that can be programmed to form dynamic shapes, including the aforementioned approach of 'chained-motor' [27, 36], chaining electromechanical motors. We show a simplified illustration of this approach in Figure 2. While these line systems can transform into diverse geometries, they are limited in shape *granularity*, one of the key metrics for shape-changing resolution [46], discussed by Roudaut et al. Under that framework, the granularity for a linear system can be calculated as *the number of control points per length*. In the case of chained motors, each motor is counted as a control point, and the size of the motors directly affects the granularity. Some custom miniature motors can be made, such as in [27] where each segment length is only 1 cm. However, due to

[2]Corrugated Tubing – Applications and Benefits https://www.mmnewman.com/blog/corrugated-tubing-applications-and-benefits

the low availability of such small motors, one of the challenges in SCIs summarized by [2], the chained-motor approach often results in low shape resolution geometries.

2.1.2 Material-Oriented Programmable Lines. Instead of embedding mechanically complex, bulky, and expensive motors along the body of the line, engineering passive material's shape-changing behavior has been gaining increasing attention across aforementioned domains [40]—which we refer to as 'morphing material' in this paper. A simplified illustration is shown in Figure 2. Such morphing material systems have employed, for example, thermo-responsive materials (shape-memory alloy [25, 47, 58], or thermo-reactive plastic [50, 55]), humidity/swelling-reactive material (liquid crystal elastomer [45]), magnetically-actuated soft-folding materials [19], and pneumatically/hydraulically actuated materials [8, 12, 26, 53, 54]. In these works, the materials are designed and 'pre-programmed' to respond to external stimuli, allowing dynamic shape-changing capabilities. By offloading the bulky and expensive actuators from the material itself, researchers were able to achieve much smaller [45, 58], smoother (e.g. PneUI [57], OmniFiber [26]), cheaper [9], or easier-to-approach (e.g. MiuraKit [10]) shape-changing lines.

While these prior works inspired our approach to employ passive linear materials shaped by active actuation mechanisms, one major limitation of prior research is that the morphing, shape-changing behavior is generally difficult to control in a precise and discrete manner. Unlike the method of 'chained-motor' methods [27, 36], whose individual discrete segments were able to achieve precise angles to configure the overall geometry, the 'morphing material' approach often relies on the material's continuous morphing behavior, which essentially has infinite degrees of freedom and states. Hence, compared to controlling discrete units with clear mechanical states, it is much more difficult to precisely control the continuous deformation, which limits the dynamic shape-reconfiguration capabilities. As a result, most instances of this 'morphing material' research are only able to transform into limited pre-fixed shapes.

2.1.3 Actuating Shapeable Mechanical Structures. To give morphing materials more diverse shape transformations and discrete controls, researchers have begun to leverage shapeable mechanical structures. Among them, bistable or multistable structures are especially suited for purposes of shape-changing, as they have clear mechanical states to be repeatedly configured. For example, PopPrint developed foldable and collapsible 3D objects [39], and Kirigami-Haptics introduced bistable haptic features encoded in materials [6]. Other research embedded digital logic into bistable materials to enable material-embedded interactivity [22, 24]. The most relevant to ours is ConeAct, a small multistable actuator capable of expanding, collapsing, and folding [30]. The multistability of a material segment significantly simplifies the actuation electronics, allowing each unit to be small. However, as linear materials are composed of large amounts of dense segments and become higher-resolution, embedding actuation hardware into numerous segments becomes increasingly challenging.

Instead of embedding actuation hardware in each material segment, our approach employs passive linear materials, *pop tubes* (which we see as 'serial-multistable tube'), and serially configures it with a separate shaping device. PopTuber offers key advantages: (1) the use of passive, low-cost, mass-manufacturable materials of various sizes; (2) the use of a fixed and minimal number of actuators to shape a material with unlimited length; (3) scalability of our approach to different material sizes with easy device adaptation, thanks to the fix number of actuators.

2.2 Line-based Fabrication

As lines are fundamental geometries for constructing shapes, various rapid fabrication systems have been developed to create complex shapes from linear material. Protopiper [1] extrudes and shapes tape into rods to prototype human-scale objects. AirTied and Reflatable create large inflated truss structures using flat tubes [33, 42].

Wire bending machines are the most similar to our work. While they can bend metal wires into precise shapes, they are often bulky and limited to 2D bending without additional mechanisms [18, 28]. Reversing bent metal wires is also challenging and requires complex straightening mechanisms [3][4].

Similar to most multistable structures, our tube material is easily hand-bendable. To better communicate the contribution, we contrast PopTuber with manual shaping approaches. Manual shaping relies on global manipulation rather than discrete, bellow-by-bellow control. It is often more suitable for short, simple, or less curved shapes, such as squares or triangles, and can be advantageous for rapid ideation or direct shape edits. Manual bending requires no machine setup and leverages the user's physical skill and spatial intuition. In contrast, our automated shaping system enables discrete control over each bellow, making it well-suited for shaping long, complex, or continuously curved forms such as arcs and spirals. It also supports digital-to-physical integration, allowing CAD-based design, reproducibility, and shape editability. The automated system ensures consistency regardless of user skill and reduces manual labor when shaping multiple tubes, an otherwise time-consuming process. Furthermore, it enables dynamic, real-time reshaping without requiring user intervention, expanding the range of potential applications.

Although pop tubes themselves are hand-bendable, our work does not see them as a fabrication material alone. Instead, we propose a novel programmable matter system that actively configures these serially multistable structures using an external shaping device. This approach combines the high-resolution shape outputs from densely packed mechanical states with the dynamic reconfigurability enabled by a small set of actuators.

2.3 Hybrid "Machine and Material" Systems

Lastly, HCI researchers have developed interactive reconfigurable systems that combine active machines and passive materials or mechanisms. For instance, HERMITS used tabletop robots to reconfigurably actuate passive mechanisms [37]. Dynablock introduced a dynamic 3D printing method that assembles reconstructable elements to form objects that can be disassembled and reused [48]. These hybrid reconfigurable systems allow users to separate passive modules from active ones, enabling them to pick up reconfigured

[3] https://www.wire-straightener.com/wire-straighteners/wire-straightening-machine.html
[4] https://www.durantco.com/standard-wire-straighteners.php

objects and return them to the machine for reshaping or reactuation. PopTuber employs a similar hybrid approach, but to create a line-based actuation and interaction device.

3 POPTUBER: HARDWARE DESIGN AND SHAPING OPERATIONS

The basic architecture of PopTuber requires what we call a *passive serial multistable material* and a shaping device, *Shaper* (Figure 1a). The core idea is that the material itself is reshapable due to its multistability, and Shaper automates the shaping and reshaping processes. In this section, we describe the serial multistable tubes and their core mechanism, followed by the Shaper hardware design and shaping operations. We then provide a parametric design guideline that allows the shaping mechanism to adapt to different tube sizes for reproducibility and scalability. Lastly, we clarify the types of linear geometries this approach can and cannot create.

3.1 Serial Multistable Tubes

Figure 3 shows the three local bellow states of the multistable tube that constitute the overall shape of the tube: (1) **Expanded**, (2) **Collapsed**, and (3) **Folded**. *Expand* and *Collapse* affect the length of the tube, while *Fold* creates and accumulates local folding angles. This local folding angle is fixed to each tube. The primary tube dimensions and folding angle we work with are shown in Figure 3.

Additionally, while each local bellow can be discretely stabilized into the above three states, the global folding direction can be granularly defined for each fold (-180° ≤ α ≤ 180°), which allows the bellow to be bent to different directions granularly.

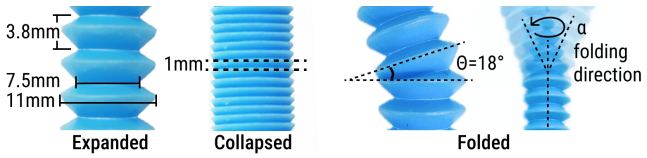


Figure 3: Three stable states of the multistable bellow structure: (1) Expanded, (2) Collapsed, and (3) Folded, with local folding angle θ and folding direction α.

3.2 Shaper Hardware Design

Shaper (Figure 4a) is designed to control the local bellow structure individually in series to shape the overall curve geometry. It requires only five motors (continuous rotation servo motors), featuring rather simple and compact hardware, especially contrasting with systems that require actuators along the shape-changing body [27, 36]. Shaper consists of **a quadra-gear extruder** and **a rotator** to shape multistable bellows on the tube sequentially, which are detailed below.

Quadra-Gear Extruder: The quadra-gear extruder is the central actuation component of our system, shown in Figure 4a. We highlight the custom-designed gears mounted on servo motors. The teeth of all four gears are designed around the tube's inner and outer diameters (ID, OD) (Figure 4b) of the expanded bellows. The top right multi-tooth gear is the key for controlling and shaping the local below states. It does so by allowing an expanded bellow to

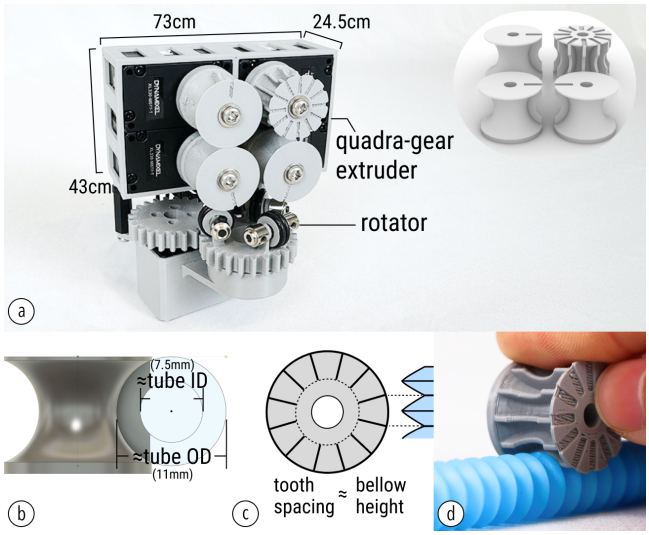


Figure 4: (a) Shaper consists of a quadra-gear extruder and a rotator. (b) The tooth is designed around the tube ID and OD. (c) The multi-tooth gear meshes with the expanded bellows.

fit into each gap on the gear (Figure 4c). The other three gears are single-tooth to flexibly rotate and manipulate the bellows. Further details of the gear design are elaborated in Section 3.4.

Rotator: While the extruder can fold and form different angles unidirectionally, the rotator is designed to rotate the tube in place, allowing the tube to be folded to any direction, which is essential for creating 3D curves.

The in-place rotation of the tube is done by employing a gripping mechanism with three small passive wheels surrounded by an O-ring. The tube's feeding and retracting linear movement aligns with the small wheels' rotation, allowing the tube to feed and retract easily. Conversely, for in-place tube rotation, which occurs perpendicular to the wheels' rotation, the wheels function as a gripping device instead.

All the gears (printed in PLA using FDM 3D printers) were mounted on Dynamixel XL330-M077-T motors, running at 6V for our default 11mm small tube, controlled by an Arduino MKR WiFi 1010 that receives commands wirelessly from a computer. Our system can also control multiple Shaper modules simultaneously, which will be demonstrated in later sections.

3.3 Shaping Operations and Motor Commands

This section outlines how shaping operations are executed using our Shaper hardware by controlling five motors (Quadtra-Gear Extruder and Rotator) in combination, visualized in Figure 5, and detailed below.

3.3.1 Expand and Collapse. The *Expand* operation allows the collapsed tubes to be expanded. This can be used to extend a compactly stored tube, and the tube length can be variable, by controlling how many bellows to be expanded along the tube. To expand an initially collapsed tube, the top two gears first jam the tube, then the bottom gears rotate to pull the tube downward, expanding the bellow layers

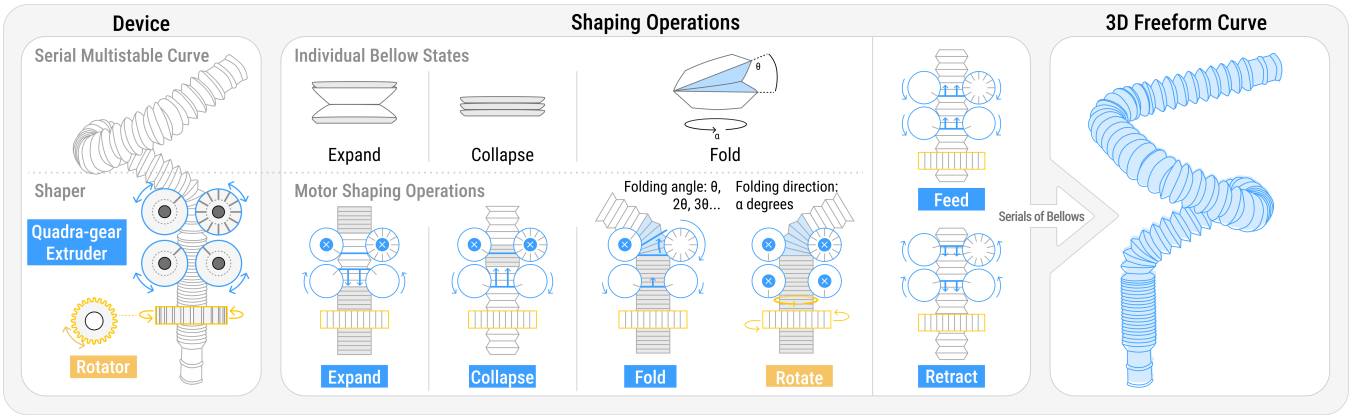


Figure 5: Shaping Architecture and Mechanism of PopTuber

in between. Similarly, to collapse expanded bellows, the bottom gears rotate to push the bellows upward, opposite to expanding.

3.3.2 Rotate and Fold. The *Rotate* operation rotates from -180° to 180° and allows different folding directions. *Fold* can be done by moving the tube upward while blocking the left side of the bellows. With one side blocked, moving N bellows up creates N folds. The current maximum number of continuous folds Shaper can make is four as the gear can no longer block the upward motion when the total fold angle is smaller than 90°. However, sharper angles can be achieved by combining continuous folds, such as six folds being deconstructed into three folds, one straight bellow, and three folds, which we will elaborate in Sec. 3.5.

3.3.3 Feed and Retract. Feed and *Retract* move the tube up and down, where opposite gears rotate at the same but opposite speed, similar to how a 3D printer extruder feeds and retracts filament.

3.3.4 Reset (Retract and Fully Collapse). Lastly, we have developed a special global shaping operation. Reset allows a shaped tube to fully collapse to its most compact state, automating the reshaping process. This is done by resetting all motors to spin back to their starting position and can usually be completed within three seconds for an approximately 27cm tube.

3.3.5 Shaping Conditions and Procedures. We developed two shaping conditions: with the tube (1) initially expanded or (2) initially collapsed, which we visualize in Figure 6. We describe each condition and discuss its pros and cons.

Condition 1 starts with an initially expanded tube. The expanded bellows can be fed to become a straight segment, and folded to become a curved segment. Process [A] and [B] (as labeled in Figure 6 top) will loop until the whole shaping process finishes. As we will show in Evaluation, this condition yields faster and more accurate results.

Condition 2 starts with an initially collapsed tube. Collapsed bellows need to be locally expanded first [C] then [A] and [B] (Figure 6 bottom), and the loop C-A-B-C-... continues until the shaping process finishes. In the end, the process [D] will reset the shape by collapsing all the bellows and retract to the initial position, ready for rendering a new shape directly. This condition allows active reshaping, making it suitable for rendering multiple shapes one after another. However, this shaping condition requires Expand to intertwine with other operations throughout the loop to always make sure the next few bellows are expanded and ready for being folded. Therefore, this condition takes longer than Condition 1 and yields less accurate results because of intertwined Expand, which is detailed in Evaluation.

3.4 Gear Design Guideline

The above device design and shaping principles can be applied and generalized to different tube sizes. For scalability, we introduce a gear design guideline in this section based on a few key parameters extracted from the tubes, as shown in Figure 7. We intend this guideline for fellow researchers, engineers, and makers to replicate our hardware design, adapting to pop tubes of different scales and properties.

The gears are the most crucial components on the Shaper, and they need a number of dimensions to define their geometries, seen earlier in Figure 4. The gear design is informed by key parameters of the tube (OD: tube outer diameter, ID: inner diameter, and bellow height) as well as two user-defined parameters (gear radius r_{gear}, tooth thickness t_{tooth}).

Firstly, OD and ID are used to define the geometry of each tooth (Figure 4b), ensuring each tooth to engage with the bellow.

Next, out of the four gears, one of them is the multi-tooth gear, and the tooth number of this gear needs to be carefully decided so that the gear tooth spacing matches the bellow height to fittingly move the tube bellow by bellow (Figure 4c). To determine the number of teeth, we developed a formula that takes bellow height h_{bellow}, gear radius r_{gear}, and tooth thickness t_{tooth} as inputs, and outputs the number of teeth, as follows:

$$\#\text{teeth} = \text{round}\left(\frac{2\pi}{2\arcsin\left(\frac{t_{\text{thick}}}{2r_{\text{gear}}}\right) + 2\arcsin\left(\frac{h_{\text{bellow}}}{2r_{\text{gear}}}\right)}\right)$$

Here, while gear radius and tooth thickness can be arbitrarily defined by the designer, we recommend setting the gear radius to be about half of tube OD, and the tooth thickness to be thin and

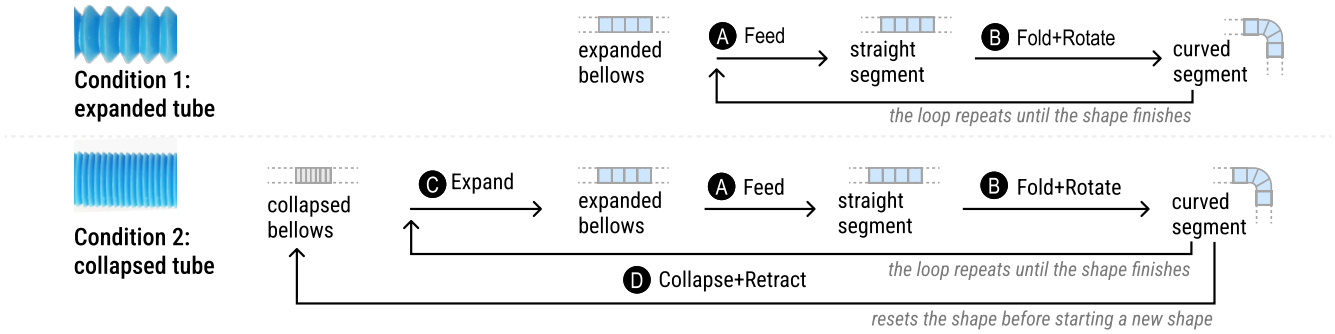


Figure 6: Two shaping conditions. Top: from an expanded tube. Bottom: from a collapsed tube.

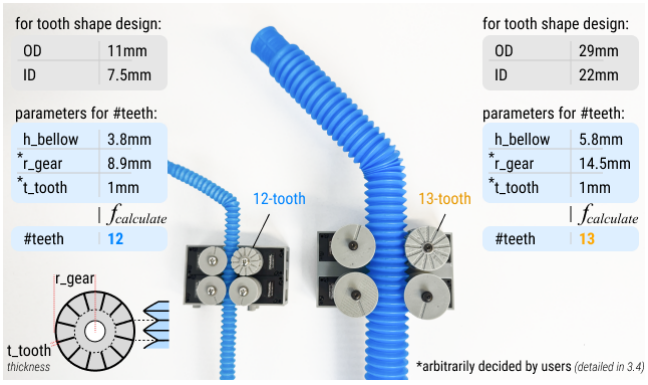


Figure 7: Parametric design tables that inform the gear design for different tube sizes. Left: the tube and device size we primarily focus on. Right: a larger tube and shaping mechanism

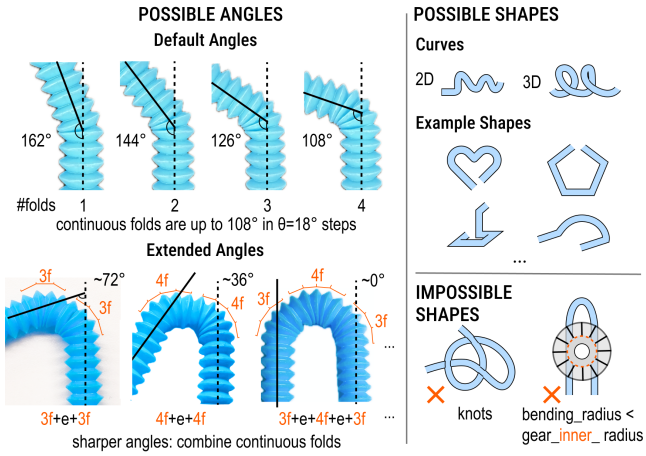


Figure 8: Possible and Impossible Shapes

strong (we use 1mm). In general, the larger the gear radius the user sets, the more teeth the gear will have, and vice versa.

All the gear designs are parametrically designed with Fusion360 and will be open-sourced[5].

Based on this design guide, we designed two device variations for two different tube sizes (OD = 11mm and 29mm, respectively) as shown in Figure 7. The shaping performance of these two designs is evaluated in later sections.

In addition to the gears, the motors need to provide enough torque for bellow expansion. For the larger 29mm tube, We used Dynamixel XC330-M288-T (stall torque is 0.52Nm), which is a higher-torque variant of the motors for the small tube's Shaper (XC330-M077-T, 0.228Nm). This can serve as a reference for component selection.

3.5 Possible and Impossible Shapes

We summarize Figure 8 to illustrate possible and impossible shapes for our system. It is not meant to be a comprehensive list but to give fellow researchers an idea of what PopTuber is capable of. We will use the default small tubes to illustrate the angles and shapes, but the idea can extend to other tube sizes.

[5]https://github.com/AxLab-UofC/PopTuber

The default angles PopTuber can create are increments of θ, where θ depends on the tube design (Figure 3c). For the small tube where $\theta = 18°$, the maximum continuous folds the Shaper can make is 4 (making 108°) due to our current gear design.

For angles that require more than 4 folds, we can combine folds to accumulate the angles. For example, 6 folds can be constructed from 3 + 3 folds, similarly 8 = 4 + 4, and a U-shape that needs 10 folds can be made from 3 + 4 + 3 folds. For angles that are not exact multiples of θ, our software will approximate to the closest fitting.

There is a great variety of shapes PopTuber can generate, including but not limited to regular shapes like polygons, rectangles, a heart, a 3d arc, a stand, etc. There are also shapes that are impossible for PopTuber. It is not possible to generate a curve with a bending radius smaller than the inner radius of the gear. It is also impossible to generate knots, as they require controlled movement of the tube ends while the Shaper loses control of one end once it extrudes out of the device.

4 SOFTWARE PIPELINE

We developed a software pipeline (Figure 9) that consists of three steps: (1) convert a digital 3D curve data into a tube simulation, (2)

generate motor commands, and (3) fabricate the shape. All our code is open-sourced.[6]

Our software pipeline facilitates user interaction in Rhinoceros, as the user input is simply a curve. Tube parameter can also be easily adjusted in Grasshopper, as we show in Figure 9.

4.0.1 Step 1. Convert: A user starts the pipeline by preparing a *freeform 3D curve data* (Figure 9), which can be drawn in Rhinoceros or imported from an external application (e.g., mid-air drawing data based on hand-tracking demonstrated in Section 5.2). The curve is imported to our custom **Curve Interpreter**, which converts the shape of the curve to *high-level shaping guideline data* through segmentation and visualizes the tube (Figure 9).

Figure 10 illustrates how the **Curve Interpreter** works to process freeform 3D curve data to execute a sequence of shaping guidelines along the curve segments. This script serves as a key module for transferring digital geometry to executable motor commands. Generally, similar processes and models have been developed in the past, especially for wire bending CNC machines [3, 20]. We adapted these general methods for our conversion algorithm, which requires additional approximation process of the folded bellows to achieve target bending angles.

Overall, based on the shaping principle of PopTuber, we abstract the tube as a polyline with consecutive straight segments and 3D angles between them. Correspondingly, the input 3D curve is first converted to a fitting polyline. Then, the Curve Interpreter 'scans' through the segment sequentially to determine key parameters for each segment to define the shaping guidelines (as illustrated in Figure 10a). For each straight segment i, we identify the key shaping parameters: the length of the segment, a.k.a. the number of tube layers to *Feed* E_i, the number of *Folds* f_i, and the angle of *Rotation* α_i, specifically from the previous folding direction to the current folding direction.

Curve Interpreter scans through all segments, obtains the key parameters E_i, f_i, α_i for each segment i and converts them to high-level shaping guidelines *Rotate* α_i, *Fold* f_i, and *Feed* E_i, and aggregates the high-level shaping guidelines sequentially in a JSON file. These guidelines are ready to be converted to PopTuber shaping operations and motor commands.

Figure 10 b and c illustrate how the folding number f_i and folding direction α_i are determined based on the geometric relationship between segments to determine each key parameter. The output of the Interpreter is a JSON file that stores key-value pairs with three types of keys "e", "f", and "r" that store corresponding number of expanded straight layers, folded layers, and rotation angles. A simplified example is shown in c2.

In addition, our tube simulation supports parameter variation, which can convert a curve into tubes with different bellow OD, ID, and height, as shown in Figure 11.

4.0.2 Step 2. Generate: Then, our **Shaping Operation Generator** converts the high-level shaping guideline to specific shape operations and motor commands, accounting for the initial condition of the tube (expanded or collapsed) or the maximum continuous folds the Shaper can possibly make. Our software also allows tuning parameters for specific operations, such as breaking down a large Fold or Feed operation into smaller steps to improve performance. If the initial tube condition is collapsed, the Generator also schedules Expand commands to occur before a set number of Feed or Fold. Then it stores the shaping operations and motor commands in a JSON file, which will be executed by the Shaper.

4.0.3 Step 3. Fabricate: Finally, the Shaper will execute motor commands and output the final tube, covered in Section 3.3.

[6] https://github.com/AxLab-UofC/PopTuber

5 APPLICATION EXAMPLES

5.1 Interaction and Display

5.1.1 Reconfigurable Tangible UIs. PopTuber's dynamic reshaping capability presents potential for dynamic physical affordances [15], outputting highly adaptable physical UIs from a single tube, with user input tracked by existing sensing technology such as computer vision. These different physical UIs offer a tactile sense of UI that gestures with a vision system alone cannot serve [13]. Illustrated in Figure 12, by collapsing and then expanding and reshaping the tube, our system could be integrated into everyday desktop environments, offering the possibility of dynamically emergable [44]or (dis)appearable [38] physical interfaces.

5.1.2 On-Body. The mobile-ready design of PopTuber opens up intriguing possibilities for reconfigurable body augmentation. For instance, as explored by researchers in the cosmetic [11] and the beauty technology space [52] users could dynamically augment their headwear with PopTuber devices as shown in Figure 13, allowing for personalized displays of individuality (c) or the real-time expression of emotional states (a,b).

5.1.3 Multi-Unit / Shape Display. The collective operation of multiple PopTuber devices presents a novel approach to shape displays with 'curved pins', thanks to the low-cost components and materials. Illustrated in Figure 14a, when configured in a linear array, the modules can construct organic shapes that defy the limitations of traditional shape displays where fabricating computational curves is essential in visualizing organic structures such as facades [16]. When arranged in a free form (e.g., in a circular pattern as shown in Figure 14b), the tubes can collaboratively construct a dome, creating a volumetric shape out of five curves. Further research is warranted to refine the process of deconstructing 3D models into discrete curve forms, potentially building on programmable material simulations research[7].

5.2 Rapid Fabrication

One of the benefits of PopTuber is that the passive material can be decoupled from the Shaper and becomes a standalone passive prop, which allows a series of rapid fabrication applications with granular shapes.

5.2.1 Data Physicalization. We show PopTuber's potential in data physicalization by producing tangible line charts, as illustrated in Figure 15, which can be read via touch. PopTuber's capacity to render 3D freeform curves also enables the creation of complex line-based 3D shapes, such as representations of DNA or protein folding, further expanding its applicability in data physicalization.

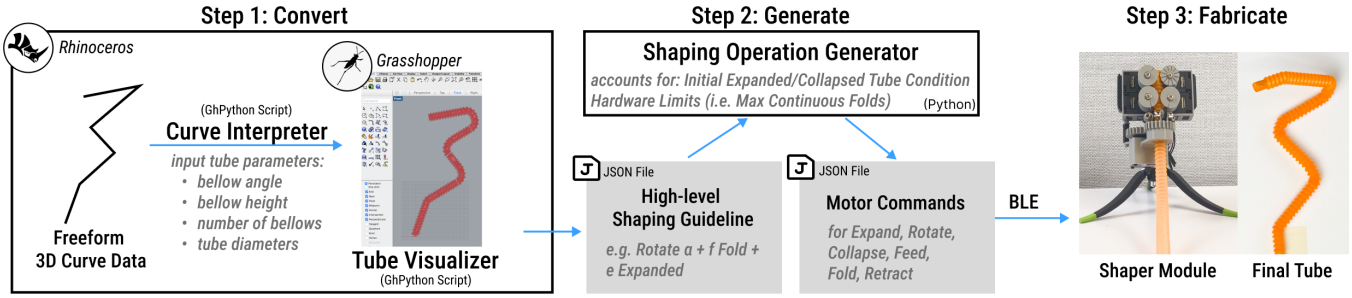


Figure 9: Our software pipeline from free-form curve data on the left to producing physical curves on the right.

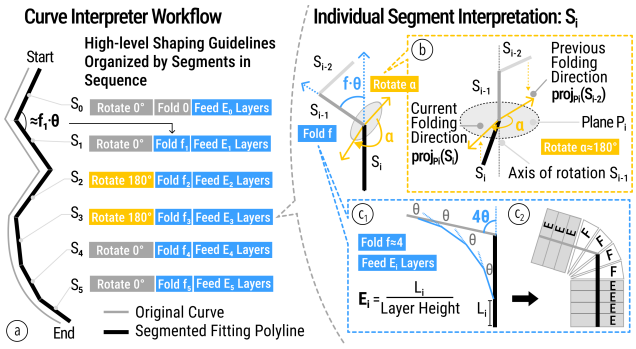


Figure 10: Curve Interpreter Workflow: *left:* (a) Example scanning and ouptuts for a 2D curve, reading the converted polyline segment-by-segment to define shaping guidelines, *right:* example close-up view of processing individual segment i to identify rotation angle α_i, fold number f_i, and length in layers E_i. f_i is the rounded quotient of the vector angle between S_i,S_{i-1} divided by θ. (b) shows that α_i is the vector angle from previous to current folding direction. Current folding direction is the projection of S_i onto P_i . Previous folding direction is effectively the projection of S_{i-2} onto P_i because previous folding happens on the plane shared by S_{i-1},S_{i-2}. (c1) illustrates that replacing a sharp angle with θ-folds shortens E_i, which CurveInterpreter accounts for through trigonometric calculations. (c2) is the tube representation.

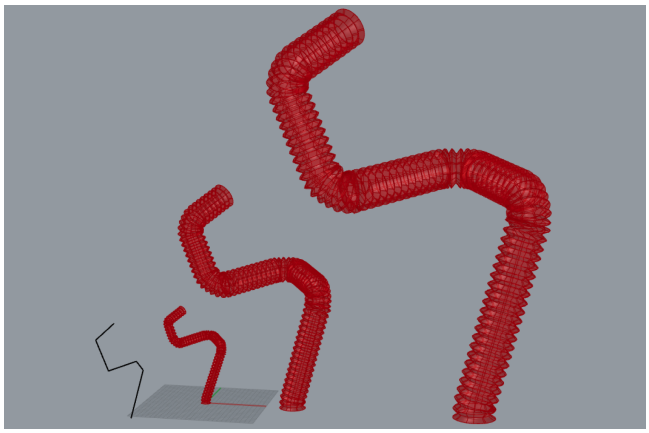


Figure 11: Simulations for different tube sizes. From left to right: the drawn curve, tube OD=11mm, 29mm, 46mm

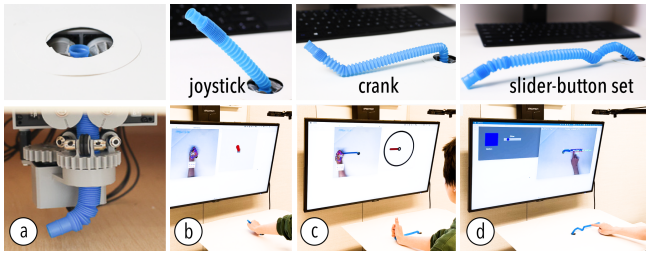


Figure 12: PopTuber can be programmed to shape and reshape into different shapes to allow reconfigurable tangible UIs, with the user input tracked by computer vision technology.

5.2.2 Prototyping. In Figure 16, we explore another scenario where a user is trying out different shapes and sizes to design an ergonomic mouse. From the CAD model, we extract outline curves from the model to create a lo-fi mockup using PopTuber [32], using two tubes to depict the height and width of the mouse. When the 3D model is adjusted, the same two tubes can be quickly reconfigured to test the new shape iteratively. This could potentially allow designers to iterate through multiple variations of organic forms that they want to design in the early stages of prototyping in a rapid and sustainable fashion.

5.2.3 Physicalizing Mid-Air Sketch. [31]

PopTuber can quickly transform digital curves from a computer vision system or CAD software into tangible tubes, which allows researchers and designers to quickly create, reproduce, and iterate their designs through free-form tangible 3D curves. For example, As depicted in Figure 17, users can draw arbitrary 3D free-form lines in the air, which are then tracked by a computer vision system and translated into physical 3D curves. These line-based objects can be potentially drawn directly at the place that needs them.

6 TECHNICAL EVALUATION

This section characterizes the shaping accuracy of basic local operations and global shape outputs using our default tube (OD=11mm) and Shaper. We also evaluate the scalability of our shaping principle

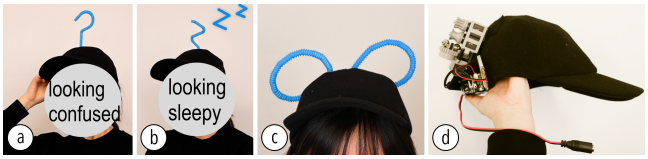


Figure 13: PopTuber mounted on a hat (d) to show different moods (a-b) and as a mode of self-expression through ears (c)

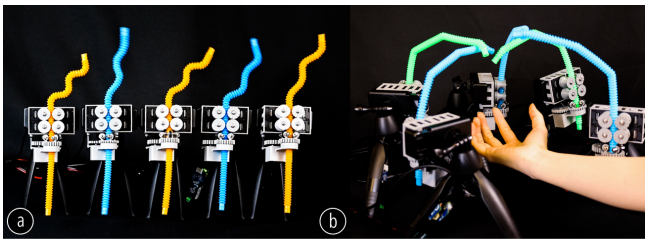


Figure 14: Five shape modules arranged in (a) an array and (b) a circle, creating a dome.

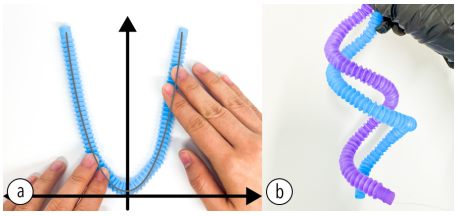


Figure 15: Physicalizing data using PopTuber. a) A parabola and b) a 3D double-helix curve

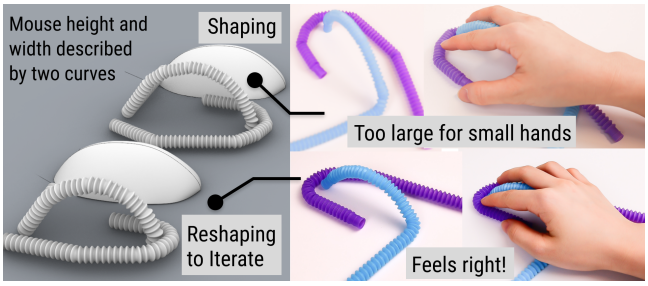


Figure 16: While iterating on the design of a mouse, a user converts the 3D model into a pair of physical curves to try the ergonomics.

with a larger tube (OD=29mm) and device. Finally, we share our preliminary exploration of various tube sizes and stiffness levels using readily available off-the-shelf options.

6.1 Shaping Performance

6.1.1 Basic Operations. With the default tube, we first evaluate basic shaping operations: *Feed*, *Fold*, *Expand*, and *Retract and Collapse*.



Figure 17: A user workflow for designing Christmas ornaments using Poptuber integrated with a CV system

Feed: The Feed operation is the most straightforward operation in our system. It linearly moves a tube bellow by bellow, producing straight segments (process [A] of the shaping cycle in Figure 6).

Each Feed may move multiple bellows. We tested Feed operations for feeding the tube 1 to 6 bellows where the top gears rotate 30 ° (1 step) to 180 ° (6 steps) at a time (n=10). The step angle 30° is determined by the 12-tooth gear. Success is marked by feeding the target number of bellows while keeping them expanded and straight.

Results are reported in Figure 18a. Feeding is rather a simple operation, but when the step is large, such as 6 at a time, it occasionally creates unwanted folds in between. This is likely because the motor tries to complete the large step too quickly, causing the tube to move too fast. The rapid acceleration can cause some of the bellows to collapse under the tube's own weight. Based on this, our software sets a constraint on the largest steps at a time and breaks down larger steps into ones smaller than 6, essentially using smaller steps to slow down the tube motion.

Fold: The number of folds determines the curvature angle, and Fold corresponds to process [B] in Figure 6. We evaluate the continuous folds from 1 to 4 for n=30 times.

Figure 18b shows that our Shaper performs better on smaller folds. For larger folds like three and four, the most common failure is that some bellows are collapsed during the folding process. For example, when aiming to make four folds, one failed result can be two folds and two collapses. Another failure is under-folding, yielding fewer folds than wanted.

These common failures on larger folds may be attributed to two factors: (1) when the angle is approaching 90°, the top left gear becomes less effective in blocking the upward motion of the bellows, and (2) the multi-tooth gear cannot mesh with that many bellows at the same time to keep them expanded while folding, creating collapses.

The result indicates that one strategy to improve shaping accuracy is breaking larger folds into smaller ones. For example, breaking a continuous 4-fold down to a combination of smaller folds, such as 2+straight+2. Customization like this has been incorporated into our software pipeline to optimize global shape generation.

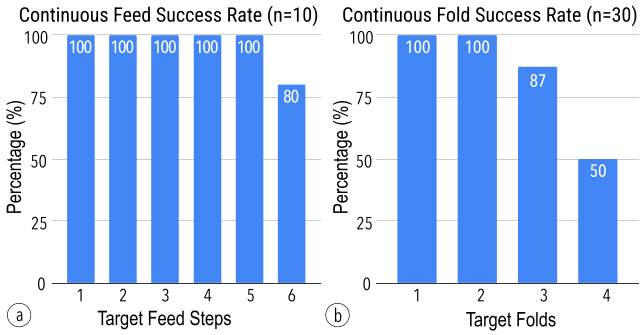


Figure 18: (a) Success Rates for Feed (n=10) and (b) Fold (n=30) on the default small tube.

Expand: The *Expand* operation starts with a collapsed tube, allowing the tube to be stored compact before being expanded. This operation corresponds to process [C] in Figure 6.

We conducted the *Expand* operation 20 times with the test tube in its collapsed state, targeting a consistent expansion of 5 bellows, which is a larger number than the continuous 4-fold. The mean expansion achieved was 5.05 bellows, with STD=0.76.

Retract and Collapse: This corresponds to process [D] in Figure 6. We performed n=20 trials of shape retractions from manually configured random shapes and all tubes trials were able to fully collapse.

6.1.2 Global Shape Generation. In addition to local shape operations, we evaluate the capability of PopTuber to shape tubes to fit target shapes (digital curve data). Figure 19 reports the target shapes (test curves), simulation results, and tube outputs.

We chose to evaluate the following three shapes: (1) a zigzag with various bending angles, involving both different folding angles and repeated rotation angles of 180°, (2) a question mark shape with one part being a circular shape while involving rotation angles of 180°, (3) a 3D arc that involves a circular curve, large continuous folds, and 90-degree rotation angles. These curves were manually created in Rhinoceros.

We tested two conditions, which correspond to the two shaping approaches illustrated in Figure 6 determined by the starting condition of the tube: (1) From Expanded and (2) From Collapsed where (1) evaluates the A-B loop of Condition 1 in Figure 6 and (2) evaluates the C-A-B loop of Condition 2 in Figure 6. We show the setup and result in Figure 19.

Based on two initial conditions of the tube, our software outputs a simulation for each and the corresponding motor commands. We repeated each shape five times using the same tube and device. All shape outputs are reported in Figure 19 for visual comparison.

In Figure 19, each trial has been labeled a percentage value that represents *accuracy*. The accuracy was computed using "edit distance"— since the multistable tubes have clear bellow states, they can be viewed as strings of the letters e, c, and f (expanded, collapsed, and folded). As a result, the input to the Shaper from our software pipeline is essentially a "shape string" as we showed in c2 of Figure 10. Therefore, we adopted "edit distance" to compare the similarity between the actual and target shapes by comparing their "shape strings." Specifically, edit distance computes the minimum edits needed to convert one string to another, commonly used in Natural Language Processing. For example, the edit distance between "eeeff" and "eefff" is 1 because changing 1 character at the third place can make the two strings the same.

The overall average accuracy for Condition 1 is about 94%, and 67% for Condition 2, and we report more detailed observation and analysis of these results below.

Condition 1: Although Condition 1's expanded tube occupies more space, it delivers much more accurate and reproducible shapes compared to the collapsed condition. Additionally, shaping in Condition 1 took significantly less time (2 to 4.3 times faster) as it eliminates the need for Expand operations. Conditon 1, therefore, is more suited for applications that focus on producing shapes fast and accurately, while requiring users to manually expand the tube before each use.

Condition 2: Condition 2 allows active reshaping of the tube while the accuracy is significantly lower in our implementation. The biggest factor contributing to the reduced accuracy in shape rendering in Condition 2 is the Expand operation. The Expand operation introduces complexity, as any inaccurate expansion will affect subsequent operations. Underexpansion, in particular, when happens, fails to expand bellows before the subsequent Fold operations, and can then lead to failures in Fold.

One possible improvement is to fully expand the entire tube all at once, rather than expanding it intermittently throughout the shaping process. This *Auto Pre-Expansion* would reduce both time and the need for precise expansion control. To validate this alternative path, we performed a validation test of this procedure on the small tube: pre-expand the bellows in bulk, followed by 30 Retract. Across 5 trials, the expanded bellow counts were: 29, 30, 28, 29, 29; overall 97% accuracy. The five failures out of 150 were one collapse during Expand and four collapses during Retract. This accuracy shows that *Auto Pre-Expansion* is a viable workaround to improve accuracy with no hardware change. The tradeoff to Condition 2 is a longer shaping process (additional 4.6s to retract 30 bellows).

6.2 Shaping Scalability

To verify the scalability of our shaping principle, we conducted an evaluation for all the basic operations on a larger tube, using a larger gear set as shown in Figure 7 in Section 3.4. The basic operations include Expand, Collapse, Feed, and Fold, following the same procedure as in Section 6.1.1.

For Expand, we set the goal to expand 7 bellows (The space between the top and bottom gears allows 7 expanded bellows). We repeated the operation 20 times; the average expansion is 7.45 bellows, with an STD = 0.89. For Collapse, similar to small tubes, all 20 trials were able to fully collapse an expanded tube.

We would like to highlight the results in Figure 20, which reports the accuracy of Feed and Fold. Both perform better than on the

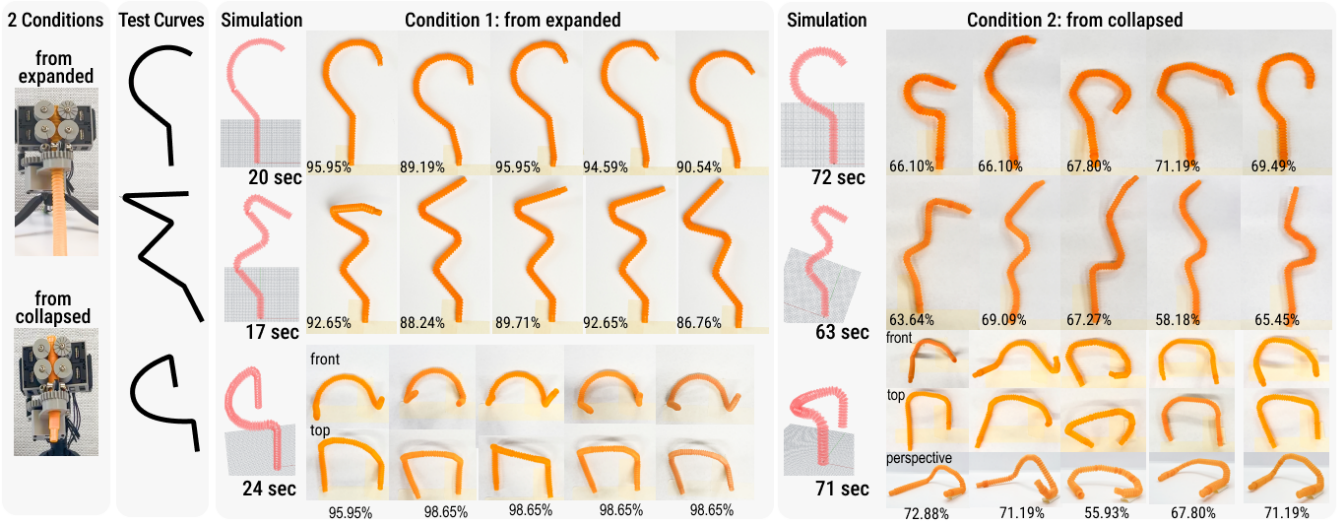


Figure 19: Global Shape Testing Results for From Expanded (left) and From Collapsed (right) conditions. Seconds represent shape completion time. Percentage values represent the accuracy for each trial, calculated based on the edit distance.

smaller tubes. Notably, Fold—the core operation delivering shape changes—achieves near-perfect accuracy, achieving and approaching 100% after 30 trials for all numbers of folds tested.

One possible explanation for the better accuracy on the larger tube than the smaller tube is that smaller tubes require greater precision in gear fabrication. In our design, the gear tooth thickness is 1 mm for both small and large gears. With sufficient mechanical strength, thinner or sharper gear teeth reduce the likelihood of slippage when meshing with the bellows. In our case, a 1 mm tooth is relatively sharper for the larger bellows than for the smaller ones, which may have contributed to more effective engagement with the bellows.

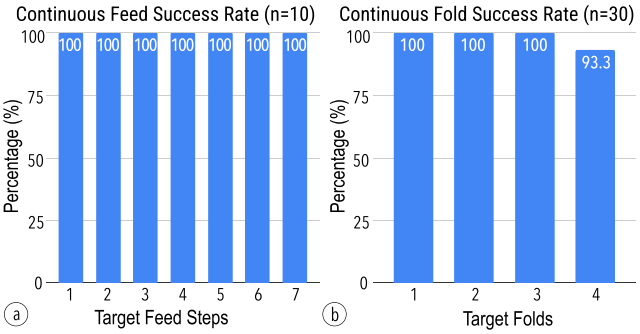


Figure 20: Success rates of Feed and Fold commands for a larger tube, shaped with larger gears.

6.3 Tube Characterization

A range of off-the-shelf tube sizes is available, with outer diameters ranging from 1.1 cm (the smallest we could find and the one we primarily use) to 4.6 cm. While prior work has conducted in-depth analysis and mathematical modeling of the multistability behavior of similar bellow structures [49, 56], here we present a preliminary characterization of several off-the-shelf options, as shown in Figure 21. We measured each tube's outer diameter, inner diameter, and material thickness, and report its vertical deflection under self-weight in Figure 21. Although none of the bellows folded due to the tube's self-weight, each expanded bellow exhibits slight flexibility. This small deformation, when accumulated across the entire tube, can lead to noticeable sagging and deviations from the intended geometry. While our current simulation tool does not account for gravitational effects, future tools could incorporate such factors to better predict and compensate for these deviations.

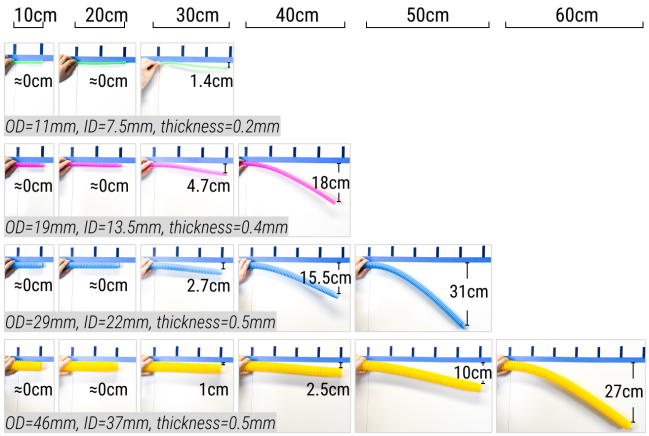


Figure 21: Vertical deflection of four different tubes due to self-weight. Each row is the same tube with OD, ID, and material thickness labeled. Each column is the length of the expanded tube, from 10 cm to 60 cm.

7 DISCUSSION, LIMITATIONS, AND FUTURE WORK

Our work is situated in the field of programmable matter, specifically focusing on the line form factor to create arbitrary 3D curve shapes. By employing widely accessible off-the-shelf linear material, pop tubes, we can create highly granular shapes without relying on chaining many small actuators.

As we mentioned in 2.1.1, the theoretical foundation of our concept is *granularity*, building on the framework defined by [46], which measures "the density of physical actuation points", or control points per centimeter (cp/cm). We share our calculations of granularities of prior SCIs as a reference: ChainFORM [34] (1cp / motor size): 0.41; ConeAct [30] (1cp / cell size): 0.4-0.67 cp/cm; PneUI [57] (1cp / airbag size): typically < 1 cp/cm; OmniFiber [26] (1cp / 1 OmniFiber size): typically < 1 cp/cm. PopTuber (1cp / bellow size): 2.86 cp/cm. This means our approach provides a viable path that allows control points to go from centimeter-scale towards millimeter-scale.

While our work achieves high granularity, emphasizing granularity alone as the sole metric is limited. It is important to note that **the angle of each bellow is limited to** θ, where $\theta = 18^{\circ}$ for the small tube (as elaborated in Section 3.5), in contrast to the broader range of π by the motors in LineFORM [36] for instance. This highlights a tradeoff: although our system offers densely distributed bellow units, the angle of each unit is limited. To improve the overall granularity of the PopTuber system, there need to be tubes with smaller θ supported by more refined gears. Notably, recent research has made exciting progress in fine-tuning the tube structures and fabricating such bellows [4, 41]. This shows promising potential for researchers to customize them for desired applications, scaling to different sizes, granularities, and structural properties (Figure 22b).

In addition, our system **requires more time to shape the curve** in contrast to the 'chained-motor' approach; the more joints or curved segments the target shape has, the longer it takes to shape the tube. For example, a spiral takes more time to shape than a rectangle. Future work could explore software strategies that prioritize shaping speed by allowing approximate shape fitting—for example, by combining small folds into coarser segments when possible.

Our system is yet to incorporate **closed-loop sensing**, which is another exciting opportunity for future work. For example, the motors on the Shaper can potentially verify the shaping process, by reading the force feedback on the motor to catch any shaping errors to self-correct. Alternatively, incorporating computer vision systems to classify and verify the Expand and Fold operations can also be a way of verifying if the bellows are correctly shaped, signaling to the Shaper to redo the operations if needed.

Enhancing **load-bearing capacity** is another promising avenue for future research. The current tube stiffness we use is not suitable for applications requiring structural rigidity. However, the hollow nature of the tube allows possibilities for internal injection to reinforce the rigidity. As Figure 22a shows, we injected polyurethane foam (liquid foam that expands and solidifies quickly) into the tube, which made the tube strong enough to hold a 1kg load. While this injection is non-reversible, it demonstrates the potential of integrating with other material systems that allow controllable stiffness [5, 14]. Future work of our software simulation could incorporate the material's self-weight and the resulting flexion to achieve more accurate results.

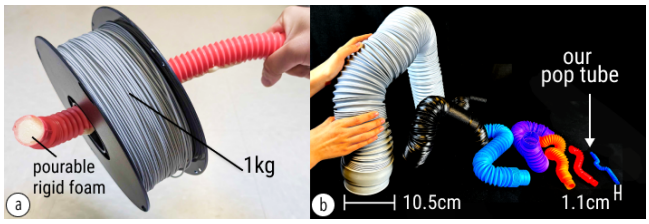


Figure 22: (a) Tube injected with pourable foam liquid. (b) Tubes of varying sizes.

Lastly, our system decouples the line length from actuator count. However, with the off-the-shelf tubes, a lot of the shapes generated are limited by the tube length. Therefore, another future work is to have a set of **tube connectors** that support extending the tube length or connecting tubes at different angles like T-shaped or crosshair-shaped connectors for more complex structures.

8 CONCLUSION

In this paper, we introduced PopTuber, a novel programmable matter system for discretely shaping and reshaping dense, multistable bellows to create high-granularity 3D curves. By creating a Shaper device that edits the discrete material units with clear mechanical states, we increase the shape variety compared to the traditional morphing material approach. We believe that PopTuber contributes a novel alternative to the existing repertoire of programmable matter. Despite current limitations, we believe PopTuber offers a meaningful addition to the landscape of programmable matter. By discretely shaping and reshaping passive multistable materials with a fixed number of actuators, we broaden how linear programmable matter systems can be imagined and implemented. We hope this work encourages further exploration of such discrete, reconfigurable mechanisms and opens up new possibilities for programming physical matter across domains.

ACKNOWLEDGMENTS

We are grateful to Ran Zhou for her exceptional system-architecture illustration and valuable feedback on the paper. We also thank the AxLab members for their assistance with proofreading.

REFERENCES

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

[2] Jason Alexander, Anne Roudaut, Jürgen Steimle, Kasper Hornbæk, Miguel Bruns Alonso, Sean Follmer, and Timothy Merritt. 2018. Grand challenges in shape-changing interface research. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–14.

[3] Andrea Baraldo, Luca Bascetta, Fabrizio Caprotti, Sumit Chourasiya, Gianni Ferretti, Angelo Ponti, and Basak Sakcak. 2022. Automatic computation of bending sequences for wire bending machines. *International Journal of Computer Integrated Manufacturing* 35, 12 (2022), 1335–1351.

[4] Ezra Ben-Abu, Anna Zigelman, Yaron Veksler, Sefi Givli, Evgueni Filipov, Hod Lipson, and Amir D Gat. 2025. Reprogrammable 3D Shapes from 1D Metamaterial. *Advanced Materials Technologies* 10, 4 (2025), 2401113.

[5] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R Zakin, Hod Lipson, and Heinrich M Jaeger. 2010. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences* 107, 44 (2010), 18809–18814.

[6] Zekun Chang, Tung D Ta, Koya Narumi, Heeju Kim, Fuminori Okuya, Dongchi Li, Kunihiro Kato, Jie Qi, Yoshinobu Miyamoto, Kazuya Saito, et al. 2020. Kirigami haptic swatches: Design methods for cut-and-fold haptic feedback mechanisms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[7] Kenneth C Cheung, Erik D Demaine, Jonathan R Bachrach, and Saul Griffith. 2011. Programmable assembly with universally foldable strings (moteins). *IEEE Transactions on Robotics* 27, 4 (2011), 718–729.

[8] Margaret M Coad, Laura H Blumenschein, Sadie Cutler, Javier A Reyna Zepeda, Nicholas D Naclerio, Haitham El-Hussieny, Usman Mehmood, Jee-Hwan Ryu, Elliot W Hawkes, and Allison M Okamura. 2019. Vine Robots. *IEEE Robotics & Automation Magazine* 27, 3 (2019), 120–132.

[9] William Coral, Claudio Rossi, Julian Colorado, Daniel Lemus, and Antonio Barrientos. 2012. SMA-based muscle-like actuation in biologically inspired robots: a state of the art review. *Smart Actuation and Sensing Systems–Recent Advances and Future Challenges* (2012), 53–82.

[10] Zhitong Cui, Shuhong Wang, Junxian Li, Shijian Luo, and Alexandra Ion. 2023. MiuraKit: A Modular Hands-On Construction Kit For Pneumatic Shape-Changing And Robotic Interfaces. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. 2066–2078.

[11] Christine Dierk, Tomás Vega Gálvez, and Eric Paulos. 2017. AlterNail: Ambient, batteryless, stateful, dynamic displays at your fingertips. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 6754–6759.

[12] Nicholas Farrow and Nikolaus Correll. 2015. A soft pneumatic actuator that can sense grasp and touch. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2317–2323.

[13] Stephanie Foehrenbach, Werner A. König, Jens Gerken, and Harald Reiterer. 2009. Tactile feedback enhanced hand gesture interaction at large, high-resolution displays. *J. Vis. Lang. Comput.* 20, 5 (Oct. 2009), 341–351. https://doi.org/10.1016/j.jvlc.2009.07.005

[14] 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.

[15] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation.. In *Uist*, Vol. 13. Citeseer, 2501–988.

[16] Sergio Araya Goldberg. 2006. Computational Design of Parametric Scripts for Digital Fabrication of Curved Structures. *International Journal of Architectural Computing* 4, 3 (2006), 99–117. https://doi.org/10.1260/147807706778658801 arXiv:https://doi.org/10.1260/147807706778658801

[17] Seth Copen Goldstein and Todd Mowry. [n. d.]. Claytronics: A Scalable Basis For Future Robots. ([n. d.]), 6.

[18] BLM Group. 2024. Wire Bending Machines. https://www.blmgroup.com/wire-bending-machines

[19] Hongri Gu, Marino Möckli, Claas Ehmke, Minsoo Kim, Matthias Wieland, Simon Moser, Clemens Bechinger, Quentin Boehler, and Bradley J Nelson. 2023. Self-folding soft-robotic chains with reconfigurable shapes and functionalities. *Nature Communications* 14, 1 (2023), 1263.

[20] Rahimah Abdul Hamid and Teruaki Ito. 2018. Integration of CAD/CAM in developing the CNC dental wire bending machine. *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 12, 3 (2018), JAMDSM0079–JAMDSM0079.

[21] Elliot Hawkes, B An, Nadia M Benbernou, H Tanaka, Sangbae Kim, Erik D Demaine, D Rus, and Robert J Wood. 2010. Programmable matter by folding. *Proceedings of the National Academy of Sciences* 107, 28 (2010), 12441–12445.

[22] Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. 2017. Digital mechanical metamaterials. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 977–988.

[23] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical atoms: beyond tangible bits, toward transformable materials. *interactions* 19, 1 (2012), 38–51.

[24] Yu Jiang, Shobhit Aggarwal, Zhipeng Li, Yuanchun Shi, and Alexandra Ion. 2023. Reprogrammable Digital Metamaterials for Interactive Devices. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–15.

[25] Hu Jin, Erbao Dong, Gursel Alici, Shixin Mao, Xu Min, Chunshan Liu, KH Low, and Jie Yang. 2016. A starfish robot based on soft and smart modular structure (SMS) actuated by SMA wires. *Bioinspiration & biomimetics* 11, 5 (2016), 056012.

[26] Ozgun Kilic Afsar, Ali Shtarbanov, Hila Mor, Ken Nakagaki, Jack Forman, Karen Modrei, Seung Hee Jeong, Klas Hjort, Kristina Höök, and Hiroshi Ishii. 2021. OmniFiber: Integrated fluidic fiber actuators for weaving movement based interactions into the 'fabric of everyday life'. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 1010–1026.

[27] Ara N Knaian, Kenneth C Cheung, Maxim B Lobovsky, Asa J Oines, Peter Schmidt-Neilsen, and Neil A Gershenfeld. 2012. The Milli-Motein: A self-folding chain of programmable matter with a one centimeter module pitch. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1447–1453.

[28] PENSA LABS. 2024. The D.I.Wire Pro. https://www.pensalabs.com/diwire-pro

[29] Daniel Leithinger, Sean Follmer, Alex Olwal, and Hiroshi Ishii. 2015. Shape displays: Spatial interaction with dynamic physical form. *IEEE computer graphics and applications* 35, 5 (2015), 5–11.

[30] Yuyu Lin, Jesse T Gonzalez, Zhitong Cui, Yash Rajeev Banka, and Alexandra Ion. 2024. ConeAct: A Multistable Actuator for Dynamic Materials. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–16.

[31] Zhihao Liu, Fanxing Zhang, and Zhanglin Cheng. 2021. BuildingSketch: Freehand Mid-Air Sketching for Building Modeling. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 329–338. https://doi.org/10.1109/ISMAR52148.2021.00049

[32] Stefanie Mueller, Dustin Beyer, Tobias Mohr, Serafima Gurevich, Alexander Teibrich, Lisa Pfistere, Kerstin Guenther, Johannes Frohnhofen, Hsiang-Ting Chen, Patrick Baudisch, et al. 2015. Low-fidelity fabrication: Speeding up design iteration of 3D objects. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. 327–330.

[33] 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*. 1–8.

[34] Ken Nakagaki, Artem Dementyev, Sean Follmer, Joseph A Paradiso, and Hiroshi Ishii. 2016. ChainFORM: a linear integrated modular hardware system for shape changing interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 87–96.

[35] Ken Nakagaki, Sean Follmer, Artem Dementyev, Joseph A. Paradiso, and Hiroshi Ishii. 2017. Designing Line-Based Shape-Changing Interfaces. *IEEE Pervasive Computing* 16, 4 (Oct. 2017), 36–46. https://doi.org/10.1109/MPRV.2017.3971127 Conference Name: IEEE Pervasive Computing.

[36] Ken Nakagaki, Sean Follmer, and Hiroshi Ishii. 2015. Lineform: Actuated curve interfaces for display, interaction, and constraint. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 333–339.

[37] Ken Nakagaki, Joanne Leong, Jordan L Tappa, João Wilbert, and Hiroshi Ishii. 2020. Hermits: Dynamically reconfiguring the interactivity of self-propelled tuis with mechanical shell add-ons. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 882–896.

[38] Ken Nakagaki, Jordan L Tappa, Yi Zheng, Jack Forman, Joanne Leong, Sven Koenig, and Hiroshi Ishii. 2022. (Dis) Appearables: A Concept and Method for Actuated Tangible UIs to Appear and Disappear based on Stages. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–13.

[39] Yuta Noma, Koya Narumi, Fuminori Okuya, and Yoshihiro Kawahara. 2020. Pop-up print: Rapidly 3D printing mechanically reversible objects in the folded state. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 58–70.

[40] Isabel PS Qamar, Rainer Groh, David Holman, and Anne Roudaut. 2018. HCI meets material science: A literature review of morphing materials for the design of shape-changing interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–23.

[41] Zhe Qiu, Shengyang Zhang, Yitong Xue, Yang Zhang, Yoshiki Mori, Shinichi Hirai, Sadao Kawamura, and Zhongkui Wang. 2024. An empirical model of soft bellows actuator. *Scientific Reports* 14, 1 (2024), 28681.

[42] Lukas Rambold, Robert Kovacs, Conrad Lempert, Muhammad Abdullah, Helena Lendowski, Lukas Fritzsche, Martin Taraz, and Patrick Baudisch. 2023. AirTied: Automatic Personal Fabrication of Truss Structures. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–10.

[43] Majken K Rasmussen, Esben W Pedersen, Marianne G Petersen, and Kasper Hornbæk. 2012. Shape-changing interfaces: a review of the design space and open research questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 735–744.

[44] Simon Robinson, Céline Coutrix, Jennifer Pearson, Juan Rosso, Matheus Fernandes Torquato, Laurence Nigay, and Matt Jones. 2016. Emergeables: Deformable displays for continuous eyes-free mobile interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3793–3805.

[45] Mikołaj Rogóż, Klaudia Dradrach, Chen Xuan, and Piotr Wasylczyk. 2019. A millimeter-scale snail robot based on A light-powered liquid crystal elastomer continuous actuator. *Macromolecular rapid communications* 40, 16 (2019), 1900279.

[46] 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* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 593–602. https://doi.org/10.1145/2470654.2470738

[47] Tomoyuki Saito, Tadao Kagiwada, and Hiroyuki Harada. 2009. Development of an earthworm robot with a shape memory alloy and braided tube. *Advanced Robotics* 23, 12-13 (2009), 1743–1760.

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

[49] Weijia Tao, Zhi Qiao, and Wenlong Zhang. 2022. Design, Characterization, and Dynamic Modeling of BEAST: a Bistable Elastomeric Actuator for Swift Tasks. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*. IEEE, 390–395.

[50] Skylar Tibbits. 2014. 4D printing: multi-material shape change. *Architectural Design* 84, 1 (2014), 116–121.

[51] Tommaso Toffoli and Norman Margolus. 1991. Programmable matter: concepts and realization. *Physica. D, Nonlinear phenomena* 47, 1-2 (1991), 263–272.

[52] Katia Vega and Hugo Fuks. 2014. Beauty Technology: Body Surface Computing. *Computer* 47, 4 (2014), 71–75. https://doi.org/10.1109/MC.2014.81

[53] Shuichi Wakimoto, Koichi Suzumori, and Jungo Takeda. 2011. Flexible artificial muscle by bundle of McKibben fiber actuators. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 457–462.

[54] James Walker, Thomas Zidek, Cory Harbel, Sanghyun Yoon, F Sterling Strickland, Srinivas Kumar, and Minchul Shin. 2020. Soft robotics: A review of recent developments of pneumatic soft actuators. In *Actuators*, Vol. 9. MDPI, 3.

[55] Guanyun Wang, Ye Tao, Ozguc Bertug Capunaman, Humphrey Yang, and Lining Yao. 2019. A-line: 4D printing morphing linear composite structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[56] Zhongyuan Wo and Evgueni T Filipov. 2022. Bending stability of corrugated tubes with anisotropic frustum shells. *Journal of Applied Mechanics* 89, 4 (2022), 041005.

[57] 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.

[58] Ketao Zhang, Chen Qiu, and Jian S Dai. 2015. Helical kirigami-enabled centimeter-scale worm robot with shape-memory-alloy linear actuators. *Journal of Mechanisms and Robotics* 7, 2 (2015), 021014.