# MorphIO: Entirely Soft Sensing and Actuation Modules for Programming Shape Changes through Tangible Interaction


**Ryosuke Nakayama**[*1], **Ryo Suzuki**[*2], **Satoshi Nakamaru**[1],
**Ryuma Niiyama**[3], **Yoshihiro Kawahara**[3], **Yasuaki Kakehi**[3]
[1]Keio University, [2]University of Colorado Boulder, [3]The University of Tokyo
ryosk611@sfc.keio.ac.jp, ryo.suzuki@colorado.edu, s.nakamaru@keio.jp,
{niiyama@isi.imi.i, kawahara@akg.t, kakehi@iii}.u-tokyo.ac.jp



**Figure 1. MorphIO is a modular and entirely soft sensing actuation unit for programming by demonstration of soft robots. (A) Each MorphIO unit consists of a pneumatic actuator containing a conductive elastic sponge sensor, which both senses and actuates three-dimensional deformation. The modular design of MorphIO's unit allows the user to construct various shapes and topologies through magnetic connection. (B) The user can demonstrate and record motion through tangible interaction. (C) The recorded motion can later be played back through pneumatic actuation.**


## ABSTRACT

We introduce MorphIO, *entirely soft sensing and actuation modules* for programming by demonstration of soft robots and shape-changing interfaces. MorphIO's hardware consists of a soft pneumatic actuator containing a conductive sponge sensor. This allows both input and output of three-dimensional deformation of a soft material. Leveraging this capability, MorphIO enables a user to record and later playback physical motion of programmable shape-changing materials. In addition, the modular design of MorphIO's unit allows the user to construct various shapes and topologies through magnetic connection. We demonstrate several application scenarios, including tangible character animation, locomotion experiment of a soft robot, and prototyping tools for animated soft objects. Our user study with six participants confirms the benefits of MorphIO, as compared to the existing programming paradigm.



### CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI);**

### Author Keywords

shape-changing interfaces; programming by demonstration; soft robots; pneumatic actuation; tangible interactions


*** The first two authors contributed equally to this work.**




*DIS '19,* June 23–28, 2019, San Diego, CA, USA





ISBN 978-1-4503-5850-7/19/06...$15.00

DOI: https://doi.org/10.1145/3322276.3322337


## INTRODUCTION

Pneumatically actuated soft materials have attracted much attention in the HCI and robotics communities. In contrast to traditional rigid materials, soft materials have a unique capability of deformation, movement, and flexibility, which enables the rich expression and function of shape-changing physical objects. While the idea of pneumatic actuation was originally explored in soft robotic literature, recent research in HCI has demonstrated the great potential of soft actuation methods for user interaction. For example, pneumatic shape-changing materials are a promising approach to adaptive affordances and functionalities [40, 25, 32], shape displays [10], accessibility [2], and haptic interfaces [5, 37].

When we interact with these soft material interfaces, they allow rich embodiment and intuitive manipulation through tangible interactions. However, when we program them, such intuitive operations disappear; the dominant programming paradigm of soft robots and material interfaces is largely confined within a digital screen, leaving little room for users to interactively explore physical motion through tangible interaction. In such a workflow—compiling code on a digital screen then transferring it into the physical object—users need to repeatedly switch between the digital and physical worlds. This leaves a large gulf of execution in their programming experiences [22]. Thus, the traditional programming paradigm significantly limits the user's ability to experiment with the design of expressive motion. Moreover, due to this barrier, such an opportunity is largely limited to highly skilled programmers and researchers who are proficient in hardware programming.

To lower this barrier, recent works have explored the *programming by demonstration* approach to leverage tangible

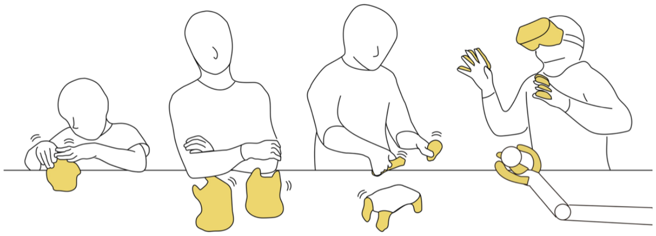

**Figure 2. A future vision of MorphIO, where users can explore the various behaviors and interactions of soft objects through tangible interaction, just like sculpting a shape with clay.**

interactions for programming soft interfaces [28, 34]. However, most of the existing tools utilize rigid components (e.g. accelerometers or strain sensors) or external tracking systems (e.g., cameras and markers) to sense deformation. These rigid materials impair the benefits of soft tactile sensation, and external sensors require complex setup and introduce other problems such as occlusion or immobility.

This paper introduces MorphIO, entirely soft sensing and actuation modules for programming by demonstration of soft robots and shape-changing interfaces (Figure 1). MorphIO's hardware consists of a soft pneumatic actuator containing a conductive sponge sensor. This allows for integrated and entirely soft shape-changing modules that can both sense and actuate a variety of three-dimensional deformations. Leveraging this capability, MorphIO enables the user to program behaviors by recording and later playing back physical motions through tangible interaction. In addition, the modular design of MorphIO's unit allows the user to construct various shapes and topologies through magnetic connection, then synthesize multiple recorded motions to achieve more complex behaviors, such as bending, gripping, and walking.

In our hardware contribution, we describe a design and fabrication method for a conductive sponge sensor that can be embedded into an air chamber in the pneumatic actuator. The conductive sponge sensor leverages the porous structure to sense the three-dimensional deformation by measuring the internal resistance value; when contracted, the resistance value between the top and bottom surfaces drops, and when extended, it increases. In contrast to existing sensing techniques [20], an elastic sponge allows for a higher degree of freedom in sensing capability (e.g., stretching, bending, and compression) without sacrificing the softness of the interface. Moreover, our modular design and graphical interface allows for easy experiments involving multiple units. For example, the system can visualize multiple recorded sensor values, so that the user can see, customize, and synthesize recorded motion to construct more complex behaviors. These hardware and software designs were informed by our formative study, wherein we interviewed five experienced researchers from the robotics and HCI communities.

To evaluate the effectiveness of MorphIO, we conducted a control experiment where we compared MorphIO's workflow with traditional programming paradigms using the Arduino IDE. We recruited six participants and asked them to program an abstract emotion of an animated character, such as happiness, anger, and sadness. The study results validate MorphIO's programming experience as being significantly efficient in terms of task completion time (MorphIO: 2 m 19 s Arduino: 5 m 21 s) and the number of trials and errors (MorphIO: 4.4 times, Arduino: 6.4 times). The qualitative feedback validates the benefits of tangible interaction, while also revealing the limitations of the MorphIO's current implementation, such as hardware capability and agility.

Finally, we demonstrate several possible application scenarios leveraging MorphIO's hardware and software, including a tangible character animation, the locomotion experiment of soft robots, the remote manipulation of soft robotic grippers, and prototyping tools for animated stuffed animals. We envision this approach's potential for lowering the barrier and opening new opportunities for a larger community to begin designing, prototyping, and exploring soft material motion—not by coding on a screen, but by sculpting behaviors in the physical world (Figure 2).

In summary, this paper contributes:

1. MorphIO, a modular and integrated design of an entirely soft sensor and actuator unit for programming by demonstration of soft robots and shape-changing interfaces
2. A novel design and fabrication of elastic conductive sponge sensors and its technical evaluation
3. A graphical user interface to record, playback, customize, and synthesize the physical motion of multiple MorphIO units
4. Application scenarios that demonstrate how MorphIO can allow the HCI and robotics communities to prototype movements
5. A user evaluation study with six participants that shows the benefits and limitations of MorphIO, as compared to the existing programming approach.

## RELATED WORK

### Shape-changing Interfaces using Soft Materials

Pneumatically-actuated soft materials have demonstrated great potential for shape-changing user interfaces. Unlike traditional rigid materials, soft materials have a unique capability of deformation, movement, and flexibility, which contributes to the ergonomics, functionalities, and aesthetics of such an interface. For example, PneUI [40] is one of the earliest explorations of this class of interfaces. It explores several shape-changing primitives of soft actuation using pneumatic inflation. Sticky Actuator [21] demonstrates a prototyping tool to animate static objects with pneumatic sheet actuators. More recently, aeroMorph [24] explores the programmable design of shape-changing inflatables with computerized numerical control (CNC) machines. It demonstrates how different heat sealing patterns create a unique bending motion of inflatable objects. Printflatables [32] further applies this approach to large-scale inflatable objects and actuation. While these soft materials can be easily deformed with external force, JamminGUI [8], JamSheets [25], and xSlate [13] explore particle or layered jamming in order to change the stiffness of soft materials for different utilities.These shape and stiffness changing capabilities can be applied to many different applications, such as shape displays (e.g., Colorise [10]), interactive toys (e.g., FoamSense [20], SqueezePulse [12]), haptic interfaces for VR (e.g., ForceJacket [5] and PuPoP [37]), and accessibility (e.g., Soft Exosuit [1]).

However, among these interactive interfaces, most systems have either input (e.g., JammingUI [8], FoamSense [20], SqueezePulse [12]) or output (e.g., aeroMorph [24], Colorise [10], ForceJacket [5], PuPoP [37]) capability, while only a few systems have integrated approaches. Yet even with integrated input and output methods (e.g., capacitive sensing in PneUI [40]), the lack of appropriate sensing methods and integrated design makes it difficult for the system to sense the variety of deformation, which is crucial for recording and playing back motion for programming by demonstration.

### Integrated Input and Output in Soft Robotics

Reviewing the soft robotics literature, however, reveals some interesting approaches to integrating sensing and actuation [31]. For example, Park et al. [27] have proposed stretchable sensors using liquid metal. This liquid metal can be embedded into an elastomer, so that the system can measure electric signals to estimate primitive deformation modes such as stretch [19], bend [15], and shear [39]. However, this approach introduces a safety problem; if the surface of the soft robot breaks, the harmful liquid metal can leak. In contrast, Giffney et al. [11] propose a method to embed a conductive polyacrylamide layer inside an air chamber, which allows both sensing and actuation of a pneumatic actuator. Alternatively, recent work has explored various sensing techniques that can be embedded or integrated with the pneumatic actuator. For example, by printing a stretch sensor with carbon nanotubes on the surface of the soft actuator, Robinson et al. [30] demonstrate a method to estimate the degree of bending. Similarly, other works have demonstrated an integrated sensing method using a capacitive sensing [16], an electromagnetic sensing [26], and an optical sensing [41] to sense the deformation. Inspired by these works, we developed a novel sensing technique that leverages an elastic conductive sponge which allows safe, entirely soft, and large deformation capability. By leveraging these characteristics, we explore how this novel sensing method can enhance the programming by demonstration for soft shape-changing materials.

### Programming by Demonstration

Programming by demonstration [4, 17] is a way of programming behaviors through physical demonstration. Although programming by demonstration was first explored in the manipulation of robots [3], researchers in HCI began investigating this approach as a design tool for tangible and shape-changing interfaces. For example, Curlybot [9] is a tangible toy to record simple locomotion. Topobo [29] allows users to construct and demonstrate movement in robotic toys, which can later be played back to animate physical motion. More recently, Reactile [35] further adapts this approach in order to construct collective behaviors of swarm user interfaces.

Several prior works have also investigated the programming by demonstration for soft material interfaces. For example, Pinoky [34] uses photo-reflective sensors and servo motors to record and animate a stuffed animal, and Mirror Puppeteering [33] uses a camera to track the behavior. However, these approaches often either require rigid materials or external tracking systems, which introduces issues such as sacrificing softness, problems with occlusion, and requiring complex and immobile setup.Most similar to our work, Bosu [28] investigated the use of shape-memory alloys and internal bending sensors to enable programming by demonstration, but the lack of modularity limits the flexible customization for different shapes and topologies. Inspired by these works, this work contributes to a *modular* and *entirely soft* sensing and actuation units for programming by demonstration of soft material interfaces.

## FORMATIVE EXPERT INTERVIEW

To better understand the needs and opportunities for support in programming soft robots and interfaces, we first interviewed five researchers from the HCI and soft robotics communities who are proficient in the programming of pneumatic actuation. The purpose of our preliminary interview is to extract current practices in order to draw insights and high-level design goals.

We first briefly ask their past experiences, and then ask open-ended questions about difficulties and challenges when designing and programming the behaviors of soft actuation. Most are using existing programming environments (Arduino: 5, Processing, 2, C/C++: 1, multiple choices are allowed), and average 2.7 years experiences (max: 10 years, min: 0.5 years).

### Insights and Design Goals

#### *Trials over simulation*

In traditional rigid robotics, there are many simulation tools that help users explore different motions. However, in soft robotics, we found that they do not use simulators. *P1. "I don't use a simulator because it's very primitive and notoriously complex."* Rather, they often work with actual robots and iterate trials-and-errors in the physical world. One reason is that soft robots are low-cost, durable, and adaptable for different situations. This leads us to the design decision of leveraging tangible interaction to better support their trials.

#### *Lowering barriers of design iterations*

The researchers often repeat a trial-and-error many times: *P2. "I usually need to iterate trials-and-errors many times, say 100 times, to achieve the desired behavior."* Thus, it is important to lower this barrier by allowing for fast and easy experimentation. When asked about the iteration process, a common workflow is as follows: 1) Set a parameter such as a delay time, 2) compile the code and run, 3) check if the behavior is correct, and 4) adjust the parameter. However, this workflow is very tedious, unintuitive, and uni-directional; there is no way to mentally map the current code to the resulting physical motion. Thus, this leads us the design decision to allow for intuitive mapping between the desired motion and the actual motion.

#### *Supporting focus on higher-level motion design*

Another problem in the current practices is that the users need to focus on tuning lower-level parameters (e.g., changing from delay(1000) to delay(1500) in Arduino) rather than designing higher-level behaviors (e.g., designing happy or angry motion). Particularly, when the number of channels increases, it becomes more complex and difficult to appropriately control the multiple parameters simultaneously. Thus, we decide to support users in focusing on higher-level design, instead of on tuning low-level parameters.

#### *Not sacrificing softness*
Most experts reported that they are not using an existing sensor to track behavior. One important reason was that available electric sensors sacrifice the softness of the material, which is crucially important for the capability, materiality, and aesthetic of soft robots and material interfaces. If required, they may prefer external tracking instead of internal sensors, despite the limited capability. Thus, this leads us to explore a novel sensing technique that does not sacrifice softness, while maintaining the deformation capability.

#### *Applicable for many different applications*
When asked about their needs, experts reported that fabricating pneumatic actuators tends to be a tedious and time-consuming process. Furthermore, application domains can vary depending on the user, including locomotive robots, grippers, shape displays, and haptic interfaces. Thus, this leads us to a modular and reconfigurable design. This allows the user to easily customize and construct actuators for differing applications, which reduces the total cost of actuator design and fabrication.

## MORPHIO: SYSTEM OVERVIEW
Based on these expert insights, we developed MorphIO, an integrated input and output unit for direct and intuitive programming through tangible interactions. In this section, we provide an overview of the system and design, and illustrate how users can program behaviors with MorphIO.

### System Architecture
The MorphIO system consists of the following components: A sensor and actuation unit, a sensing and actuation control unit, a microcontroller, software to control these units, and a visual interface for users to control behaviors. Figure 3 illustrates the overview architecture of MorphIO.

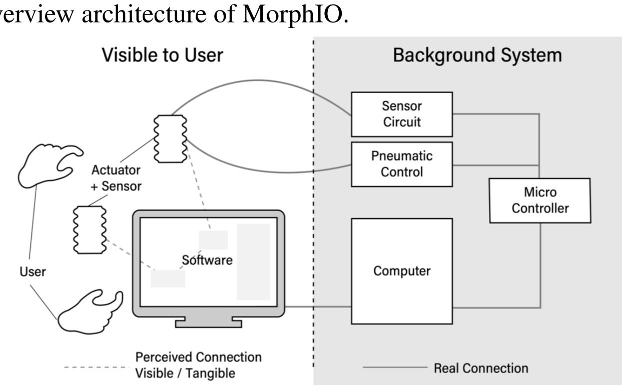


**Figure 3. A system overview of MorphIO. It consists of a unit, a control circuit, a microcontroller, and software.**

The programming workflow with MorphIO is the following: First, a user starts manipulating the MorphIO unit. The demonstrated motion is detected and recorded through internal sensors, and the recorded sensor values are stored in the software. Once the user clicks play in the graphical user interface, the pneumatic pump starts supplying air. By controlling the air flow through switching on and off the solenoid valves, the system can control the behavior of the pneumatic actuator as it plays back the recorded motion.

To enable this workflow, MorphIO has three key components:

1. **Entirely Soft Sensing and Actuation Units**
2. **Modular and Reconfigurable Design**
3. **Graphical User Interface to Record and Playback**

In the following, we describe the details of each component.

### Entirely Soft Sensing and Actuation Units
Each MorphIO unit consists of a pneumatic actuator containing a deformation sensor. The internal sensor is made of a conductive elastic sponge which can detect contraction and expansion. The amount of deformation can change the internal resistance value of the conductive sponge. Thus, by tracking the resistance value between the top and the bottom surfaces, the system can estimate the current degree of deformation.

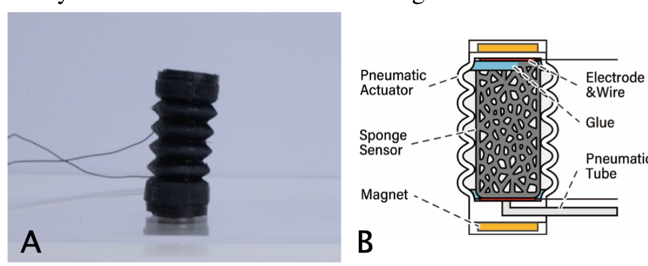


**Figure 4. The MorphIO unit (A). Each unit consists of a bellows pneumatic actuator containing a conductive elastic sponge sensor (B).**

The conductive sponge sensor is enclosed in an air chamber of the pneumatic actuator. The sponge's elasticity and porous structure prevent it from interfering with the capability of the pneumatic actuator, as the pumped air can flow into the porous cells. The sensor is attached to the top and to the bottom of the pneumatic actuator with glue. Figure 4 illustrates the unit and its internal structure.

In our prototype, each MorphIO unit has a length of 6cm and a diameter of 1.6cm, but different sizes can easily be designed and fabricated. In our experiment, each module can contract up to 4.7cm (78%) and expand up to 8.2cm (138%). Figure 5 shows the contacted, neutral, and expanded states.

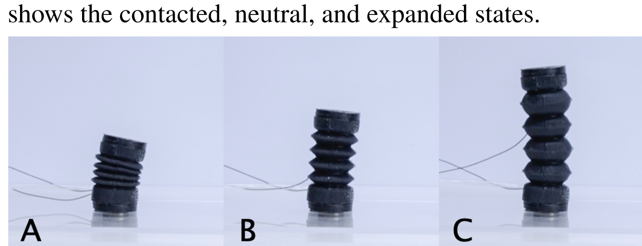


**Figure 5. The contraction, neutral, and expanded states of the MorphIO unit. It can contract and expand up to 78% and 138% respectively.**

### Modular and Reconfigurable Design
While a single module can only have one degree of freedom (i.e., contraction/expansion) in its deformation capability, we designed the MorphIO unit to be modular and reconfigurable, so that the user can connect multiple units to achieve higher

degrees of freedom. For example, by combining two modules in parallel, the MorphIO unit can bend left and right in addition to contracting and expanding (Figure 6 B). Moreover, by combining three modules, the MorphIO unit can now have three-dimensional capability (Figure 6 C).


**Figure 6. By combining multiple MorphIO modules, we can achieve higher degrees of freedom for more complex deformation. Deformation using a single unit (A), two units connected in parallel (B), and three units connected in circle (C).**

For each module, two neodymium magnets (15mm diameter, 3mm thickness) are attached to the top and the bottom surfaces, so that each module can be easily connected and disconnected from the magnetic connection. Thus, the user can construct different topologies with multiple units.

In order to combine multiple MorphIO modules, we created six different types of connectors (Figure 7): 1) A parallel connector of two modules, 2) a parallel connector of three modules, 3) a circular connector with three modules, 4) a bipod connector with a 90-degree angle, 5) a bipod connector with a 120-degree angle, and 6) a tripod connector with a 90-degree angle.

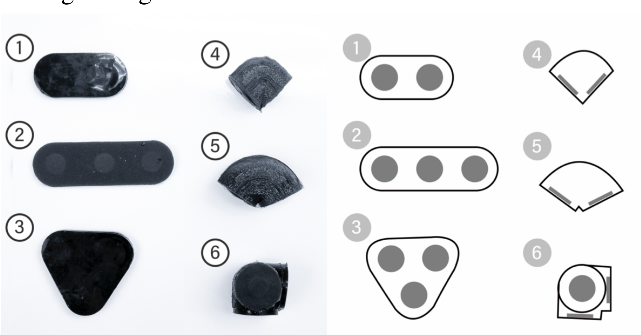


**Figure 7. We designed and prototyped six different types connectors to allow for the easy construction of different toplogies.**

By using these connectors, users can achieve various types of motion for different applications. For example, by using a parallel connector, the user can make the expressive motions of animal characters, and by combining three modules with the tripod connector, users can make a walking robot or a gripper.

### Graphical User Interface to Record and Playback

In our programming workflow, there are basically two ways to record deformation: 1) record and playback, and 2) motion transfer. For example, Figure 8 shows a user recording a single unit, then playing back the same motion. Furthermore, Figure 9 shows a user demonstrating motion which can be simultaneously transferred to another module in real-time.

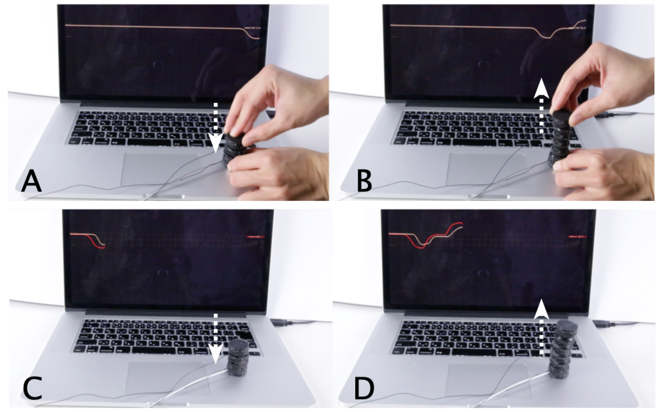


**Figure 8. The user can record motion through tangible interaction (A-B). When the user executes the playback command, the same unit can actuate based on the recorded motion (C-D).**

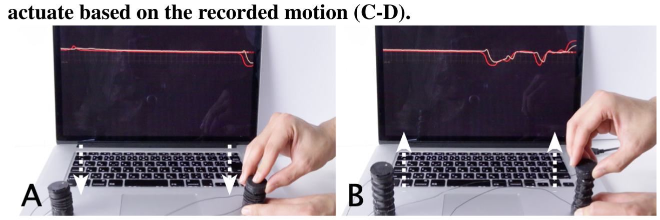


**Figure 9. MorphIO also allows synchronous motion, wherein the deformation of one module can be transferred to another module simultaneously.**

To better support programming experiences, we also developed a graphical user interface that allows the user to view recorded sensor values and change the parameters of playback motion.

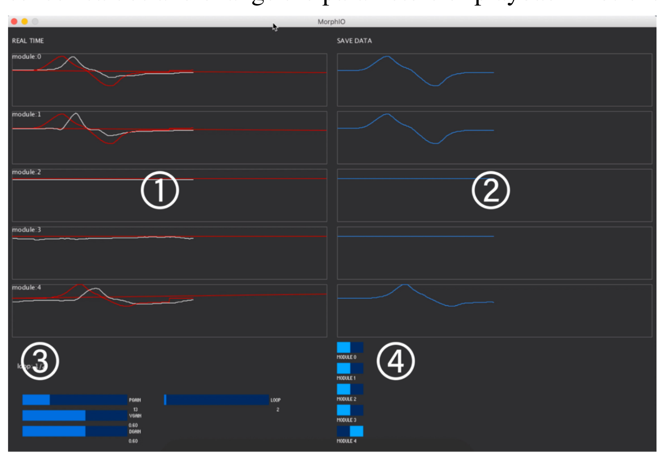


**Figure 10. The graphical user interface to visualize the recorded sensor values in a timeline (1-2), control the parameters (3), and execute the command (4).**

Figure 10 shows the interface to visualize the recorded values. The interface consists of 1) real-time sensor values (white line) and recorded sensor values (red line), 2) a plot of the recorded values (blue line), 3) sliders to change parameters (e.g., loop count, delay time, strength), and 4) a toggle switch to play, pause, and record. In sum, the user can track, customize, and synthesize recorded motion by changing values, the number of loops, and timing.

## IMPLEMENTATION

In this section, we describe the implementation of MorphIO's hardware and software, including 1) the conductive elastic sponge sensor, 2) the integrated sensor and an actuator unit, 3) the pneumatic control circuit, and 4) software for feedback control and visual interface.

### Conductive Elastic Sponge Sensor

#### *Mechanism*

Our novel soft sensor can measure the various deformation states of soft objects. The basic sensing mechanism is similar to a resistance sensor; by measuring the resistance value, we can estimate the current deformation state. A change in the resistance value is caused by a change in the internal electrical property; the external force is applied to the sensor, then the internal porous structure will shrink or expand, which affects the behavior of an electron within the structure.

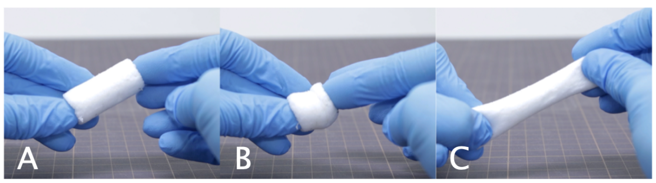


**Figure 11. Due to its elasticity, our proposed sponge sensor can achieve both contraction and expansion, in contrast to prior methods.**

The sensing principle is similar to [20], but our sensor uses an elastic sponge which allows for stretching in addition to compression, which cannot be achieved with the prior work's method (Figure 11). These two types of deformation allow a higher degree of freedom when combining the multiple units, which is a crucial factor for designing more expressive movements.

#### *Fabrication Process*

Here, we describe the fabrication process of a conductive elastic sponge sensor. The fabrication process follows three steps: 1) Fabricate an elastic sponge, 2) impregnate into conductive ink, and 3) attach electrodes and wires.

To fabricate an elastic sponge, we follow the basic procedure proposed by [18]. We first prepare 6.0 g of elastomer prepolymer solution (Smootn-on Inc. Ecoflex 00-50) and 29.1 g of sodium-chloride (Naruto Salt Mfg Co. Ltd), then mix them together by using a planetary centrifugal mixer (Thinky Inc. AR-100). The mixed solution is injected into a 3D printed cylindrical mold (16mm diameter, 40mm height). Then we dry the material with an oven (Yamato Science Co. Ltd, DVS 402) at 100 C degrees for one hour. (While it can be dried in room temperature, we used an oven to shorten the fabrication time.) Once dried, we immerse the sponge in water, so that the sodium chloride can melt, leaving a porous structure within the elastomer sponge (Figure 12).

The next step is to make the elastomer sponge conductive. First, we prepare a conductive ink by mixing 0.8 g of carbon particles (Cabot Inc., Black Pearl 2000) into 16 g of a 2-propanol solution (Kanto Kagaku Co. Ltd.). Then, we pour

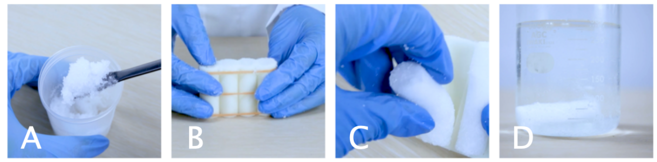


**Figure 12. The fabrication of an elastic sponge.**

the mixed solvent into a 50 ml case with 6 zirconia beads (10 mm diameter) and vibrate for 15 minutes. Next, we dilute 4.0 g of prepolymer silicone solution (Shin-Etsu Chemical Co. Ltd., KE 1316) with 16 g of toluene (Wako Pure Chemical Industries Ltd.). Finally, the carbon solution is mixed with the silicone solution and stirred for 15 minutes.

Once the preparation of the conductive ink in finished, we impregnate the elastomer sponge into the conductive ink (Figure 13 A) Infiltration and extraction need to be repeated several times, so that the conductive ink can completely penetrate into the sponge. We then dry the conductive sponge with a dry oven at 130 C degrees for one hour to affix the conductive ink onto the elastomer sponge through the polymerization of the prepolymer silicone and the evaporation of the solvent.

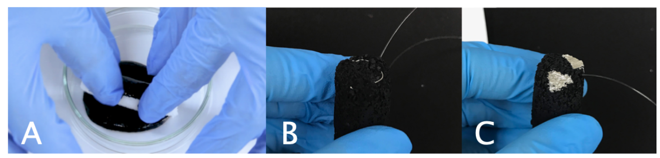


**Figure 13. Impregnation of elastic sponge into conductive ink (A). Bonding the lead wire with the conductive sponge to make electrodes (B-C).**

The next step is to attach electrodes to the conductive sponge. We first prepare the lead wire (Sunhayato Corp, AWG 30) and peel off 5 cm, then weave it into the sponge (Figure 13 B). To attach the electrodes, we use a flexible conductive adhesive (Cemedine Co. Ltd., XX-ECA46L) for bonding the elastomer sponge (Figure 13 C). Finally, we dry the conductive sponge with the dry oven at 100 C degree for 30 minutes.

#### *Technical Evaluation*

We measured the relationship between the deformation ratio and the resistance value of the conductive sponge sensor. In the experiment, the sensor was fixed to the Z-axis stage (Chuo precision industrial Co. Ltd, LV-912) with a length of 40 mm as a neutral. We changed the length of the sponge by 1 mm, then measured the current resistance value. Figure 14 shows the result of the experiment. We observed the change of 5 to 40 kOhm in the resistance value, which is corresponding to the 58% to 142% deformation ratio.

### Integrated Sensor and Actuator Unit

#### *Design*

Our integrated sensor and actuator is a bellows structure which encloses the conductive sponge sensor we fabricated through the above process. The bellows structure is one of the most common actuation designs in soft robotics. By pumping air, it can inflate to extend the actuator; while vacuuming air, it can contract. Due to the porous structure of the conductive sponge, the pumped air can flow into the of the porous cells without interference, thus it can both sense and actuate with a single unit simultaneously.

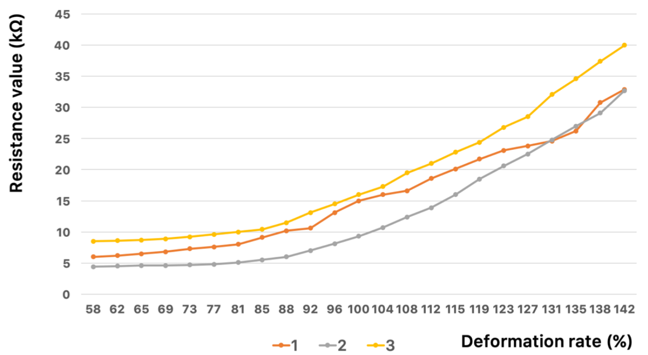


**Figure 14. The technical evaluation of the conductive sponge sensor. The horizontal axis represents the deformation rate (%) and the vertical axis represents the internal resistance value (kΩ) with one (red), two (gray), and three (orange) sponges respectively.**

Our bellows actuator is similar to the design proposed by [6]. However, when the internal air pressure exceeds at a certain point, we observe that the actuator will inflate not only in the vertical direction, but also in the horizontal direction (Figure 15 A).

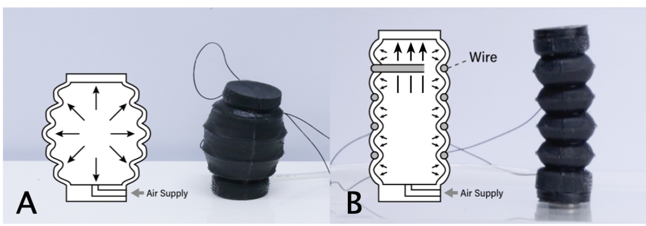


**Figure 15. We wrap the bellows with a wire to ensure that the air pressure can only inflate in the vertical direction. Before using the wire (A). After using the wire (B).**

To prevent the expansion in the horizontal direction, we use a wire to wrap the bellows actuator. Figure 15 shows the inflation state before (A) and after (B) rolling the gut wire. This allows for the appropriate deformation of the inflatable actuator.

### *Fabrication Process*

To create the bellow pneumatic actuator, we first 3D print an outer molding. The 3D printed mold consists of eight parts for two separate sides of the bellows (Figure 16 A). To create the mold, we used an FDM 3D printer with a heat resistant filament; this is in contrast to an SLA or an inkjet 3D printer wherein we observed that the light curable resin was melt when drying with the oven.

For the material, we used the same elastomer (Smooth-on Inc., Ecoflex 00-50) for the inflatable actuator and mixed a black pigment (Smooth-on Inc., Silc Pig) to change the color. We note that the same elastomer can be used as the adhesive for bonding the elastomer parts to each other. We cast the elastomer into the mold, then we dry the elastomer using the oven and a heat gun (Figure 16 D) To bond the sensor and the actuator, we used a commercially available urethane resin adhesive (Konishi, Ultra multi-use SU) instead of using the elastomer.

### *Technical Evaluation*

We measured the deformation ratio in the vertical direction using image processing software and image analysis (Image J).

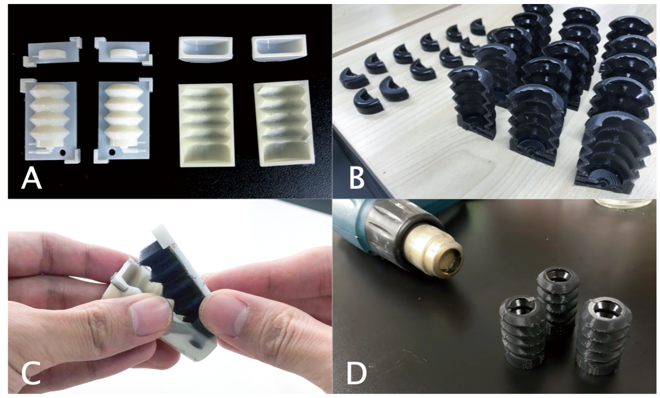


**Figure 16. The fabrication process of the bellows pneumatic actuator. We embed the conductive sponge sensor into the internal air chamber of the bellows.**

The proposed sensor and actuator unit can deform from 78% to 138% (from 46% to 164% when ignoring the fixed length of the top and bottom). We also measured the bending capability by fixing the length of one side of the actuator. The maximum curvature radius of bending was 5.5 cm when supplying 30 cc (Figure 17).

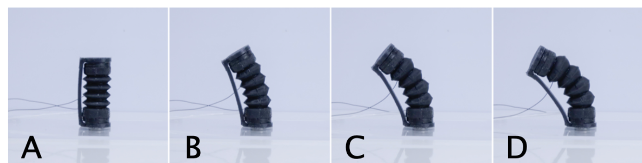


**Figure 17. The technical evaluation of the bending capability.**

### Pneumatic Control Circuit

To control the pneumatic actuator of each module, we use two air pumps (Oken Seiko Co. Ltd, P54A02R, RFP32B03R) and two solenoid valves (SMC, S070C-VDC-32) to supply and vacuum the air pressure. We switch each pump and solenoid valve to target the following four modes: 1) Extension: solenoid valve A ON, positive pressure pump ON, 2) Contraction: solenoid valve B ON, negative pressure pump ON, 3) Fixed: solenoid valve A ON, and 4) Open: All OFF. Figure 18 shows the actual air flow control circuit and its schematic.

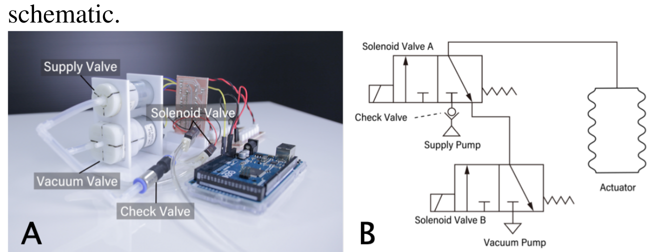


**Figure 18. The pneumatic control circuit, consisting of two air pumps and two solenoid valves for supplying and vacuuming air (A). A schematic of the pneumatic control circuit (B).**

To drive the air pumps and solenoid valves, we use motor drivers (Pololu, DRV8833) through PWM output. To measure the resistance value of the conductive sponge sensor, we use a voltage divider circuit and measure the difference between a fixed resistance value and the current resistance value of the sponge sensor through the analog input. Figure 19 shows the driving circuit using the Arduino Mega microcontroller.

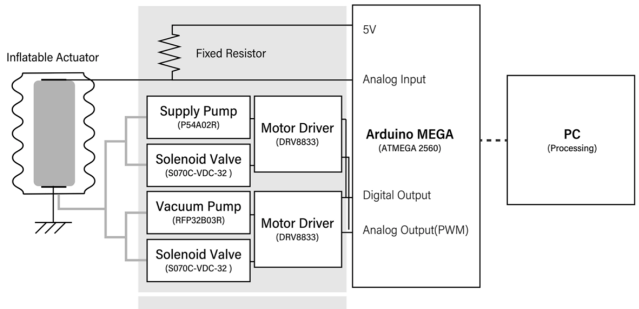


**Figure 19. MorphIO's system design.**

### Feedback Control Software

To enable the playback of the recorded deformation, we need feedback control for each actuation module. To do so, we use the following feedback control algorithm: 1) Compare the current and target deformation value, 2) if there is a positive/negative difference, we set the expansion/contraction mode, 3) if there is no or less difference than the threshold value, we set the closed mode. The system runs this procedure per every one frame (60 fps). Our feedback control software and the graphical user interface were written with C++ and the OpenFrameworks library and communicate with Arduino using the Firmata library.

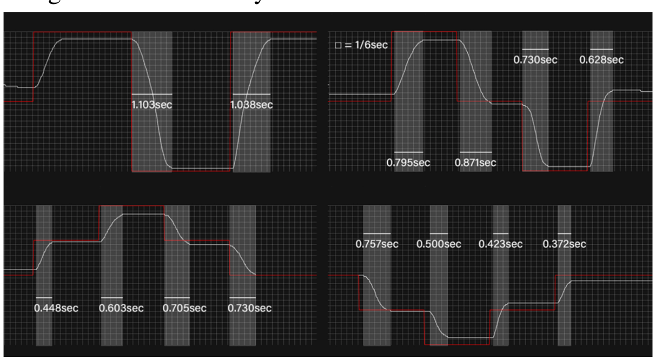


**Figure 20. The evaluation of lag time between differing inflation states.**

We evaluated our feedback control algorithm in terms of the lag time between mode change and the actual deformation. Figure 20 shows a screenshot of the measured lag time with multiple deformations. It took 1.1 seconds from the minimum to the maximum expansion state, and it took 0.6-0.8 seconds from the norm to the minimum/maximum expansion state.

## APPLICATION SCENARIOS

Based on the proposed hardware and software, we demonstrate several possible applications scenarios with MorphIO.

### Tangible Character Animation

With MorphIO, the user can easily create an animation of a tangible character. Taking inspiration from Disney's animation principles [38], such an animated character can react and express emotion to the user through motion.

For example, Figure 21 shows that the user designs the behavior of the communication robot by using three MorphIO modules and connectors. The end user can thus explore expressive motion in the physical world. Similar to [14] and [36], these robots can serve as a communication robot to interact with children.

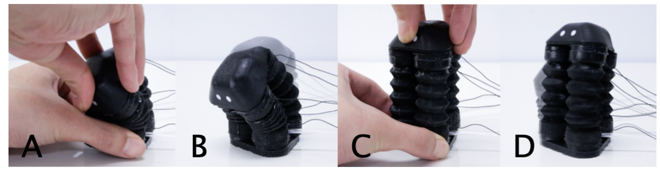


**Figure 21. A user is designing the motion of a tangible animation character. It can be used as a communication robot.**

### Animating Existing Soft Objects

The MorphIO module can be also attached to existing objects to actuate them. For example, Figure 22 shows the user animating a stuffed animal with a MorphIO module.

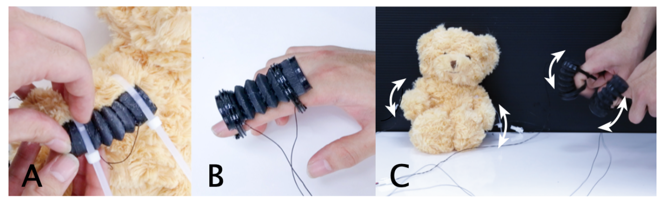


**Figure 22. The MorphIO unit can be attached to the stuffed animal, so that the user can animate and remotely control the existing object.**

The attached module can be synchronized with the other module in real time, so that the user can remotely animate the stuffed animal by binding the control module to her fingers. These animated stuffed animals can be used in communication or storytelling with children.

### Remote Manipulation of Soft Grippers

By leveraging real-time synchronization, the user can also remotely manipulate the soft grippers. By using the tripod connector, three MorphIO modules can be connected, and these modules can grasp a ball (Figure 23).

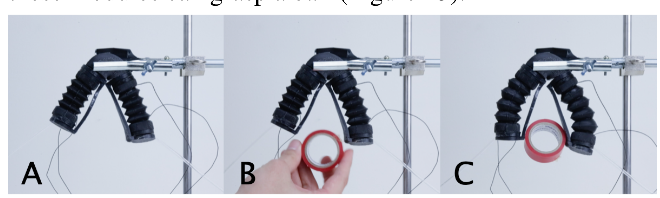


**Figure 23. By using the tripod connector, three MorphIO modules can be used as a soft gripper.**

Another example includes a real-time remote assistant for rehabilitation or training, as accessibility and rehabilitation assistants are a promising application for soft robotics. Thus, by attaching the MorphIO module to the body, a doctor can remotely assist the patient, or a trainer can provide synchronized guidance for movement, dancing, and sports.

### Locomotion Experimentation with Soft Robots

Finally, we demonstrate that MorphIO modules can be applied for the locomotion of soft robots. Usually, it is difficult for robotic researchers to explore diffing locomotion of soft robots. Because of its modularity, MorphIO allows for quick experimentation with different topology and motion designs.

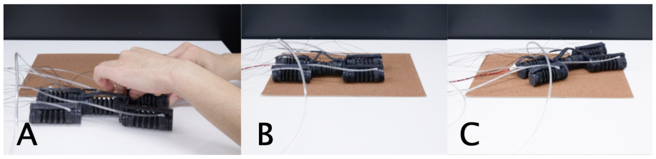


**Figure 24. By combining five MorphIO modules, a user can design and experiment with the locomotion of a soft robot.**

For example, Figure 24 shows a walking robot by using five modules which combine the air chamber at the center. By demonstrating tangible interactions, one can explore differing locomotion, such as speed, orientation, and mechanisms. Since it is usually difficult to effectively control five modules at the same time by traditional programming, tangible demonstration can significantly lower the barrier of iteration and exploration of soft robotic locomotion.

## USER EVALUATION

We conducted a user evaluation study to understand the benefits and limitations of MorphIO. In this study, we focused on answering the following research questions:

- **RQ1**: Does MorphIO save time and reduce the number of iterations to program the target behavior, compared to the existing approach?
- **RQ2**: Does MorphIO increase the expressiveness of the physical motion?

To answer these questions, we conducted a controlled experiment where we compared MorphIO with the current programming approach. We chose Arduino IDE as a base condition for the comparison, as this is the most common programming approach identified through our formative study.

### Participants

For the study, we recruited six participants (5 male, 1 female), ages 22-25 years old (average: 22.8, standard deviation: 1.2) from our local community. All participants were from engineering majors (computer science), having at least 2.8 years prior experience in hardware programming (e.g., Arduino). Two participants had experience in soft robots or pneumatic actuation, while the other participants had limited experience in programming pneumatic actuators. Each session took approximately 60 minutes.

### Method

The goal of this study is to understand the differences between our approach and the existing programming workflow. To do so, we provide three basic tasks to construct a program. For each task, the participants were asked to program three different emotions—happiness, anger, and sadness—of an animated character. We chose these emotions based on Ekman's basic emotions for communication [7]. We instructed the participants to perform these three tasks in two conditions: MorphIO and Arduino IDE. To minimize the effect of knowledge transfer, we randomized the order of the task and condition for each participant.

Since the participant may not be familiar with controlling pneumatic actuators with Arduino, we prepared the basic Arduino code to control the pneumatic actuator with four functions

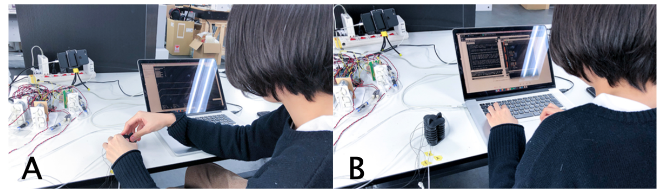


**Figure 25. The controlled user evaluation study. A participant uses MorphIO to program a behavior of an animal character (A). The same participant uses Arduino IDE to program the behavior (B).**

(e.g., expansion, contraction, fixed, and open) so that the user can only control parameters such as delay time or loop condition. For MorphIO, the basic programming workflow is 1) press the record command to switch to the recording mode for 8 seconds, 2) the user tangibly programs the behavior, 3) play the recorded behavior for 8 seconds by pressing the play command.

During the task, we measure the 1) time to complete the task, and 2) the number of iterations. For the task completion time, we let the participants self report the completion; we measured the time until each participant determines that the behavior is close enough to the target motion. For the number of iterations, we measured the number of times the participant pressed the execute button for both conditions. After the task, we conducted a brief interview where we instructed the participants to answer questions on a 9-point Likert scale. We also asked several open-ended questions about their programming experiences.

### Results

Figure 26 shows a summary of the results. The average time of task completion time of MorphIO was 2m 19s, compared to 5m 21s with the control condition. The average number of iterations of MorphIO was 4.4 times, compared to 6.4 with the control condition. The study results validate that MorphIO is significantly efficient in terms of task completion time ($p < 0.05$, $Z = -2.98$) and the number of iterations ($p < 0.10$, $Z = -1.75$) with an Wilcoxon signed-rank test. When asked about

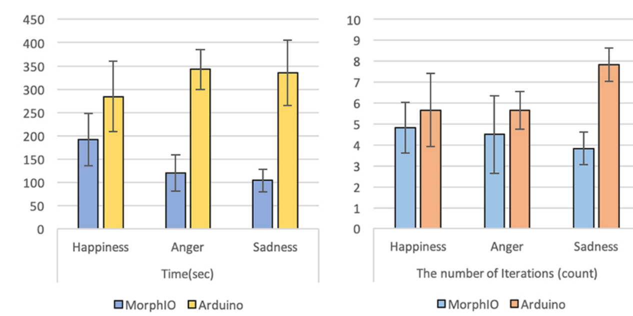


**Figure 26. The user study results.**

the achievement of the expressions using a 9-point Likert scale, the average score with MorphIO was 6.5 (happiness: 6.2, anger: 6.8, sadness: 6.2), compared to 6.3 with the control condition (happiness: 6.2, anger: 6.3, sadness: 6.5). We did not find differences between the two conditions ($p > 0.5$, $Z = -0.69$) using a Wilcoxon signed-rank test. Thus, we conclude the result of our study as follows: **RQ1: Yes, RQ2: No**.

Overall, the participants generally had a positive feeling toward their experience with MorphIO. P1: *"It was a very interesting experience that the tangible motion was recorded and played back."* P2: *"The robot that I made looks like an actual living animal."* P4: *"The experience was not programming, rather more like interactive communication."* On the other hand, we also found some limitations. For example, one participant commented that it was difficult to update the behavior based on the previous attempt, as the demonstrated motion will be overwritten when recording again. Also, two participants mentioned some hardware limitations such as power and agility, which limits the expressiveness of the behavior.

## DISCUSSION

From our post interviews, we realized the advantage of using traditional programming over tangible interactions, which affects the result of RQ2. P3: *"Programming approach allows more precise and slight adjustment."* We found that 1) tangible interactions are suitable for sculpting *rough* motion, 2) programming allows for fine-tuning more precise adjustments. Thus, for future research, systems might allow users to quickly make a rough motion, which can automatically be converted into digital parameters so that the user can also precisely control and adjust the motion. The same *human-computer cooperation* approach can be applied to other design domains: For example, when designing an object, the user can quickly make rough shapes with clay, while letting a machine finish the details. We believe this insight can lead the HCI community to further explore design approaches wherein users and machines cooperate for enhanced interaction design.

## LIMITATIONS AND FUTURE WORK

### Different Tasks and Designs for User Study

There are a number of limitations which need to be addressed in future work. First, in our user study, we focus on one specific task (e.g., expressing emotion) with one specific design (e.g., an animated character). We are interested in future work that explores differing tasks and designs to realize their effect on results (e.g., designing the locomotion for a walking robot).

### Interaction Design for Manipulating Many Modules

We are particularly interested in how the higher degree of freedom will affect results, as there exists a complex interaction design challenge when the number of modules increases. For example, when designing the interaction of a walking robot with five modules or applications like shape displays [10], the user cannot deform more than two separate modules simultaneously. One possible solution is to record a single module one by one, then later synthesize the motion, but there may be better interaction designs to resolve this challenge (e.g., copy-and-paste the recorded motion by touching them).

### Programming Environment for Interactive Applications

We are planning to improve our visual programming environment to make interactive applications. For example, we are interested in allowing MorphIO to connect with IoT devices (e.g., smart speakers) or web applications (e.g., IFTTT). By leveraging the capability of both input and output, the user can easily prototype an interactive interface like an animated character that can react to the user's voice input, turn on the light using a pressing motion, and change the light intensity by bending it.

### Exploration of Different Size and Design for the Module

While we only investigated a single size and design, differing module sizes and designs can be used for diverse applications (e.g., smaller sizes for constructible robots like Topobo [29] and larger sizes for wearable devices like exoskeleton suits). Differing form factors (e.g., tentacles and sheets) can also allow for a variety of shape-changing capabilities. While our approach does not depend on a specific size and design, additional design and technical challenges may arise when working with differing scales and form factors.

### Sophisticated Control Device and Fabrication Process

We notice that pneumatic control systems can become more complex as the number of modules increases. To control many modules, it is suitable to use a more advanced tool kit such as Pneuduino [23]. Moreover, the fabrication of our conductive sponge sensor was an entirely manual process, which is tedious and introduces individual differences. In the future, we anticipate an advanced fabrication process using a next generation 3D printer that can print conductive elastomer to simultaneously create both an outer actuation structure and an internal porous structure.

## CONCLUSION

This paper introduces MorphIO, a soft sensing and actuation module for programming by demonstration of soft robots and shape-changing interfaces. Our main contribution is the integrated and entirely soft input and output capability of the MorphIO unit. We describe the design and fabrication method of a novel conductive elastic sponge sensor and an integrated pneumatic actuator. By leveraging this capability, MorphIO allows the user to interactively program physical behaviors through tangible interaction. Our user study results confirm the benefits of MorphIO, compared to the existing programming approach. Finally, we envision a future where people can interactively explore various behaviors and dynamics of soft objects in the physical world, just like sculpting a shape with clay.

## ACKNOWLEDGEMENTS

We thank Jun Kato for his helpful feedback and Layne Hubbard for proofreading the paper. This research was supported by the JST ERATO Grant Number JPMJER1501 and the Nakajima Foundation.

## REFERENCES

[1] Alan T Asbeck, Stefano MM De Rossi, Kenneth G Holt, and Conor J Walsh. 2015. A biologically inspired soft exosuit for walking assistance. *The International Journal of Robotics Research* 34, 6 (2015), 744–762.

[2] Stuart Biggar and Wei Yao. 2016. Design and evaluation of a soft and wearable robotic glove for hand rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24, 10 (2016), 1071–1080.

[3] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. 2008. Robot programming by

demonstration. In *Springer handbook of robotics*. Springer, 1371–1394.

[4] Allen Cypher and Daniel Conrad Halbert. 1993. *Watch what I do: programming by demonstration*. MIT press.

[5] Alexandra Delazio, Ken Nakagaki, Roberta L Klatzky, Scott E Hudson, Jill Fain Lehman, and Alanson P Sample. 2018. Force Jacket: Pneumatically-Actuated Jacket for Embodied Haptic Experiences. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 320.

[6] Krishna Manaswi Digumarti, Andrew T Conn, and Jonathan Rossiter. 2018. EuMoBot: replicating euglenoid movement in a soft robot. *Journal of the Royal Society Interface* 15, 148 (2018), 20180301.

[7] Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion* 6, 3-4 (1992), 169–200.

[8] 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*. ACM, 519–528.

[9] Phil Frei, Victor Su, Bakhtiar Mikhak, and Hiroshi Ishii. 2000. Curlybot: designing a new class of computational toys. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 129–136.

[10] Juri Fujii, Takuya Matsunobu, and Yasuaki Kakehi. 2018. COLORISE: Shape-and Color-Changing Pixels with Inflatable Elastomers and Interactions. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 199–204.

[11] Tim Giffney, Mengying Xie, Aaron Yong, Andrew Wong, Philippe Mousset, Andrew McDaid, and Kean Aw. 2016. Soft pneumatic bending actuator with integrated carbon nanotube displacement sensor. *Robotics* 5, 1 (2016), 7.

[12] Liang He, Gierad Laput, Eric Brockmeyer, and Jon E Froehlich. 2017. SqueezaPulse: Adding Interactive Input to Fabricated Objects Using Corrugated Tubes and Air Pulses. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 341–350.

[13] 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*. ACM, D113.

[14] Hideki Kozima, Marek P Michalowski, and Cocoro Nakagawa. 2009. Keepon. *International Journal of Social Robotics* 1, 1 (2009), 3–18.

[15] Rebecca K Kramer, Carmel Majidi, and Robert J Wood. 2013. Masked Deposition of Gallium-Indium Alloys for Liquid-Embedded Elastomer Conductors. *Advanced Functional Materials* 23, 42 (2013), 5292–5296.

[16] Christina Larson, B Peele, S Li, S Robinson, M Totaro, L Beccai, B Mazzolai, and R Shepherd. 2016. Highly stretchable electroluminescent skin for optical signaling and tactile sensing. *Science* 351, 6277 (2016), 1071–1074.

[17] Henry Lieberman. 2001. *Your wish is my command: Programming by example*. Morgan Kaufmann.

[18] Benjamin C Mac Murray, Xintong An, Sanlin S Robinson, Ilse M van Meerbeek, Kevin W O'Brien, Huichan Zhao, and Robert F Shepherd. 2015. Poroelastic foams for simple fabrication of complex soft robots. *Advanced Materials* 27, 41 (2015), 6334–6340.

[19] C Majidi, R Kramer, and RJ Wood. 2011. A non-differential elastomer curvature sensor for softer-than-skin electronics. *Smart Materials and Structures* 20, 10 (2011), 105017.

[20] Satoshi Nakamaru, Ryosuke Nakayama, Ryuma Niiyama, and Yasuaki Kakehi. 2017. FoamSense: Design of Three Dimensional Soft Sensors with Porous Materials. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 437–447.

[21] Ryuma Niiyama, Xu Sun, Lining Yao, Hiroshi Ishii, Daniela Rus, and Sangbae Kim. 2015. Sticky actuator: Free-form planar actuators for animated objects. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 77–84.

[22] Donald A Norman. 1986. Cognitive engineering. *User centered system design* 31 (1986), 61.

[23] Jifei Ou, Felix Heibeck, and Hiroshi Ishii. 2016a. TEI 2016 Studio: Inflated Curiosity. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 766–769.

[24] Jifei Ou, Mélina Skouras, Nikolaos Vlavianos, Felix Heibeck, Chin-Yi Cheng, Jannik Peters, and Hiroshi Ishii. 2016b. aeroMorph-heat-sealing inflatable shape-change materials for interaction design. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 121–132.

[25] 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*. ACM, 65–72.

[26] Selim Ozel, Erik H Skorina, Ming Luo, Weijia Tao, Fuchen Chen, Yixiao Pan, and Cagdas D Onal. 2016. A composite soft bending actuation module with integrated curvature sensing. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 4963–4968.

[27] Yong-Lae Park, Bor-rong Chen, Néstor O Pérez-Arancibia, Diana Young, Leia Stirling, Robert J Wood, Eugene C Goldfield, and Radhika Nagpal. 2014. Design and control of a bio-inspired soft wearable robotic device for ankle–foot rehabilitation. *Bioinspiration & biomimetics* 9, 1 (2014), 016007.

[28] 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*. ACM, 189–198.

[29] 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*. ACM, 647–654.

[30] Sanlin S Robinson, Kevin W OâĂŹBrien, Huichan Zhao, Bryan N Peele, Chris M Larson, Benjamin C Mac Murray, Ilse M Van Meerbeek, Simon N Dunham, and Robert F Shepherd. 2015. Integrated soft sensors and elastomeric actuators for tactile machines with kinesthetic sense. *Extreme Mechanics Letters* 5 (2015), 47–53.

[31] Daniela Rus and Michael T Tolley. 2015. Design, fabrication and control of soft robots. *Nature* 521, 7553 (2015), 467.

[32] 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*. ACM, 3669–3680.

[33] Ronit Slyper, Guy Hoffman, and Ariel Shamir. 2015. Mirror puppeteering: Animating toy robots in front of a webcam. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 241–248.

[34] Yuta Sugiura, Calista Lee, Masayasu Ogata, Anusha Withana, Yasutoshi Makino, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. 2012. PINOKY: a ring that animates your plush toys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 725–734.

[35] Ryo Suzuki, Jun Kato, Mark D Gross, and Tom Yeh. 2018. Reactile: Programming Swarm User Interfaces through Direct Physical Manipulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 199.

[36] Haodan Tan, John Tiab, Selma Šabanović, and Kasper Hornbæk. 2016. Happy Moves, Sad Grooves: Using Theories of Biological Motion and Affect to Design Shape-Changing Interfaces. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 1282–1293.

[37] Shan-Yuan Teng, Tzu-Sheng Kuo, Chi Wang, Chi-huan Chiang, Da-Yuan Huang, Liwei Chan, and Bing-Yu Chen. 2018. PuPoP: Pop-up Prop on Palm for Virtual Reality. In *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 5–17.

[38] Frank Thomas, Ollie Johnston, and Frank Thomas. 1995. *The illusion of life: Disney animation*. Hyperion New York.

[39] Daniel Vogt, Yong-Lae Park, and Robert J Wood. 2012. A soft multi-axis force sensor. In *Sensors, 2012 IEEE*. IEEE, 1–4.

[40] 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*. ACM, 13–22.

[41] Huichan Zhao, Kevin O'Brien, Shuo Li, and Robert F Shepherd. 2016. Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides. *Science Robotics* 1, 1 (2016), eaai7529.