

Constraint-Driven Robotic Surfaces, At Human-Scale

Jesse T. Gonzalez
jtgonzal@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Sonia Prashant
sprashan@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Sapna Tayal
sapnat@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Juhi Kedia
jkedia@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Alexandra Ion
alexandraion@cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Scott E. Hudson
scott.hudson@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

Robotic surfaces, whose form and function are under computational control, offer exciting new possibilities for environments that can be customized to fit user-specific needs. When these surfaces can be reprogrammed, a once-static structure can be repurposed to serve multiple different roles over time. In this paper, we introduce such a system. This is an architectural-scale robotic surface, which is able to begin in a neutral state, assume a desired functional shape, and later return to its neutral (flat) position. The surface can then assume a completely different functional shape, all under program control.

Though designed for large-scale applications, our surface uses small, power-efficient constraints to reconfigure itself dynamically. The driving actuation force, instead of being positioned at each "joint" of the structure, is relocated to outer edges of the surface. Within the work presented here, we illustrate the design and implementation of such a surface, showcase a number of human-scale example functional forms that can be achieved (such as dynamic furniture), and present technical evaluations of the results.

KEYWORDS

Human Scale Interaction, Shape-Changing Devices

ACM Reference Format:

Jesse T. Gonzalez, Sonia Prashant, Sapna Tayal, Juhi Kedia, Alexandra Ion, and Scott E. Hudson. 2023. Constraint-Driven Robotic Surfaces, At Human-Scale. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, October 29–November 01, 2023, San Francisco, CA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3586183.3606740>

1 INTRODUCTION

Shape-changing structures enable tangible interfaces and interactive environments. Since their form, and hence function, can be established under computational control, they offer interesting new possibilities for highly adaptive objects. Fittingly, these shape-shifting surfaces have a diverse history of implementations. To name a few: 4D-printed structures, that deform in response to an

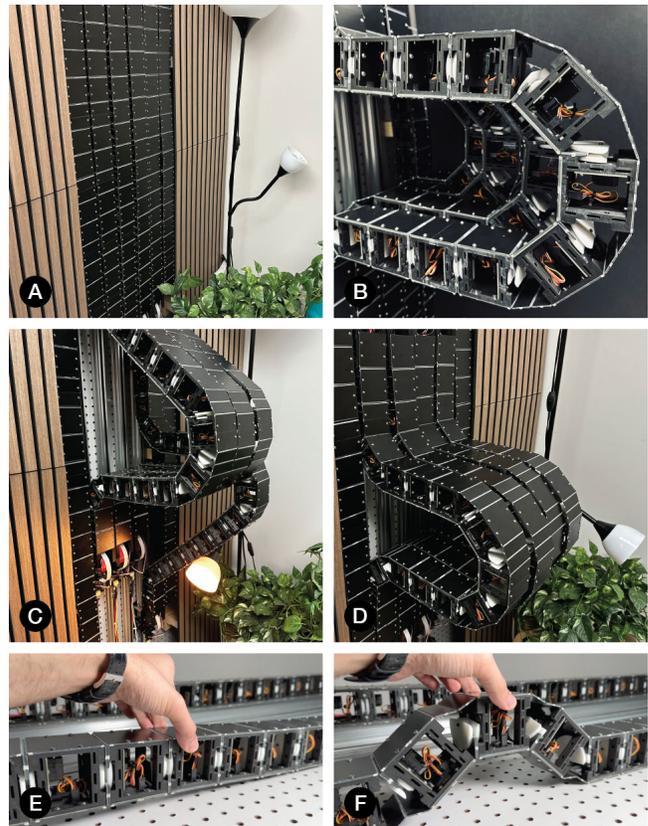


Figure 1: By constraining the cells of our robotic surface in various patterns, we are able to create robust, human-scale structures that "pop out" of a living room wall (a-d). These constraints are added and removed through the computer-controlled rotation of a rigid clamp, which adds mountain and valley folds (f) to an otherwise rigid surface (e). Once constrained, a squeezing force (located at the top and bottom edges of the structure) pushes the cell columns out-of-plane, and locks them in position. Our surface can render shapes in a few seconds or less, allowing for the possibility of real-time user interactions.



This work is licensed under a [Creative Commons Attribution-NonCommercial International 4.0 License](https://creativecommons.org/licenses/by-nc/4.0/).

UIST '23, October 29–November 01, 2023, San Francisco, CA, USA

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0132-0/23/10.

<https://doi.org/10.1145/3586183.3606740>

external trigger (e.g., heat) [38]; foldable origami-inspired materials, that can be bent into 3D shapes using embedded actuators (e.g., shape memory polymers) [31]; and robotic constructs known as shape displays [8, 26], which consist of many (perhaps hundreds of) actuators that drive the change in shape by repositioning their parts.

Such shape displays offer the highest flexibility in terms of the number of shapes that can be formed. However, this comes at a cost. Beyond requiring an actuator (e.g., motor) at each configurable bit, the actuator must be strong enough to hold its position under load. Such actuators are not only pricey, but also power-hungry, which further require expensive equipment, such as sophisticated power supplies. To mitigate the cost and implementation issues, the demonstrated capabilities are often limited to low-force and often small-scale applications (data physicalization, or manipulation of lightweight, centimeter-scale objects).

In this paper, we propose a shape-changing robotic surface that is designed for large-scale applications, but still uses small, power-efficient actuators to reconfigure itself dynamically. The key to our material is this: instead of driving the shape change at many points (i.e. at each hinge of our structure), we simply constrain its shape, and relocate the transformative actuation force to the outer edges of our surface.

1.1 Constraint-Driven Robotic Surfaces

We introduce a reconfigurable surface which is able to begin in a neutral — in this case flat — state, and then under computational control, assume a desired functional shape. The material can later be returned to its neutral state, and then assume a completely different functional shape. Our surface works at human-scale — larger than the typical hand-scale of many HCI devices, but still designed to fit alongside room-scale structures. It uses a series of actuated columns each made up of a linear collection of identical cells, with bending joints between each pair of cells. Rather than directly actuating each bend, which would require powerful, heavy, and expensive motors, cells establish constraints on the bend angle at each joint using small, lightweight, and inexpensive actuators. Once constraints have been established, bending actuation is then provided by a simple but more powerful system which presses from the ends of the columns in order to achieve and hold the final desired shape.

The work presented here details the design and implementation of our material, which we refer to as a "constraint-driven robotic surface". We include techniques that make partial use of a layered fabrication method which simplifies construction. We consider a number of example functional forms that can be achieved, and present some simple technical evaluations that test strength and stiffness of the material when deployed.

1.2 Contributions

The main contribution of this work is the concept of constraint-driven robotic surfaces. We make the following specific contributions:

- (1) A technique for surface reconfiguration through programmable constraints.
- (2) A novel structure for a bendable column, and the associated fabrication method.

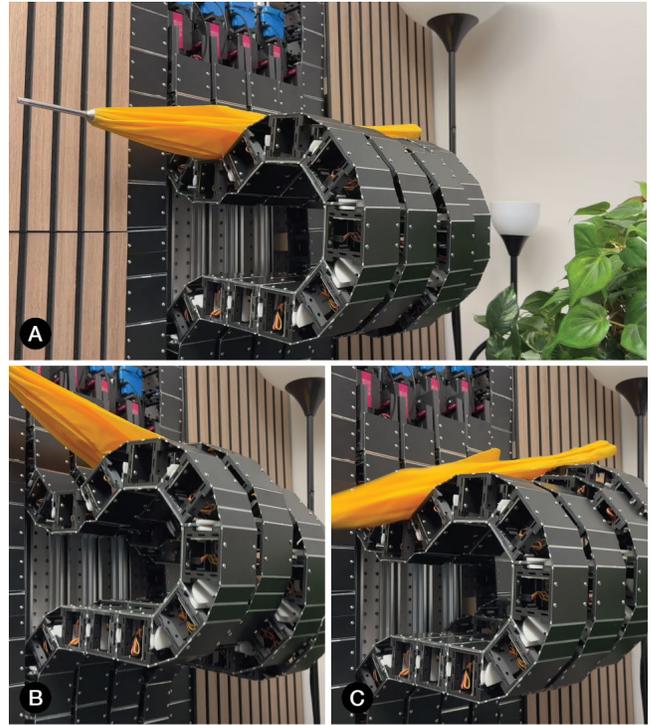


Figure 2: A physical ripple effect (resembling a stadium wave) propagates across the wall, reminding a nearby user to grab their umbrella. The makeshift shelf has a small divot to cradle the umbrella.

- (3) An exploration of the application space, within the context of interactive architecture.

2 LIVING WITH ROBOTIC SURFACES

Lee is an imagined person, living in a small apartment full of robotic surfaces. Their space is too cramped for roving robotic agents, but nevertheless, Lee's apartment is filled with functionality — neatly concealed within the walls and floors of their home. Though Lee's story is fictional, the shape-changing surfaces that they will encounter are all realizable by our system today.

Wearing a coat and carrying a bag of groceries, Lee enters their apartment and is greeted by a subtle waving gesture from the wall across the room. Lee would like to take off their coat, but they first need a spot to set down their bag of groceries — fortunately, a shelf emerges from a portion of the wall beside them (Figure 3). As Lee sets down the bag, a coat hook descends from the wall (3b). Lee hangs up their jacket, grabs the groceries again (3c), and goes into the kitchen. As Lee leaves, the coat is raised upwards, out of the way, and the shelf collapses back into the wall, its purpose served (Figure 3d-f).

A bit later, Lee returns to the living room with a meal in hand, ready to eat. A small table rises from the ground — a bit low, but Lee doesn't mind sitting on the floor (Figure 4). The furniture folds flat once it's no longer in use.

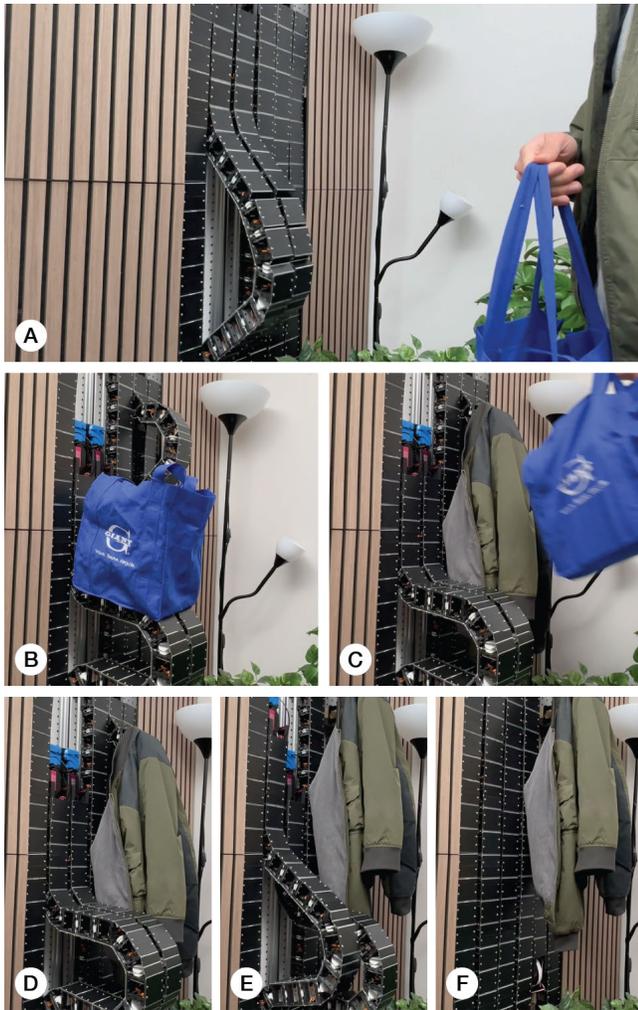


Figure 3: From the wall, a small shelf emerges (a), able to support loads such as the grocery bag in (b). This temporary holding spot allows a human user to (for example) remove their coat, and hang it on a hook that pops out of the wall (c). When the shelf is no longer needed, it folds back into the wall (d,e,f), and the coat hook rises out of the way (f).

Lee has some work to catch up on, so they turn back towards their wall and solicit a small desk for their laptop (Figure 5). Lee has been sitting for a while and would like to work while standing up, so the wall adjusts the height of the desk.

Afterwards, Lee decides to settle in for some reading. As they lean against the wall behind them, the vertical surface transforms into a reading nook (Figure 6). An "appendage" emerges as well, angling a nearby gooseneck lamp to illuminate Lee's book for the evening (6b,c).

The next day, when rain is forecast, Lee's wall reminds them to take their umbrella with them before they leave (Figure 2).

The robotic functionality in Lee's apartment is seamlessly integrated into their environment — it lies in the space between

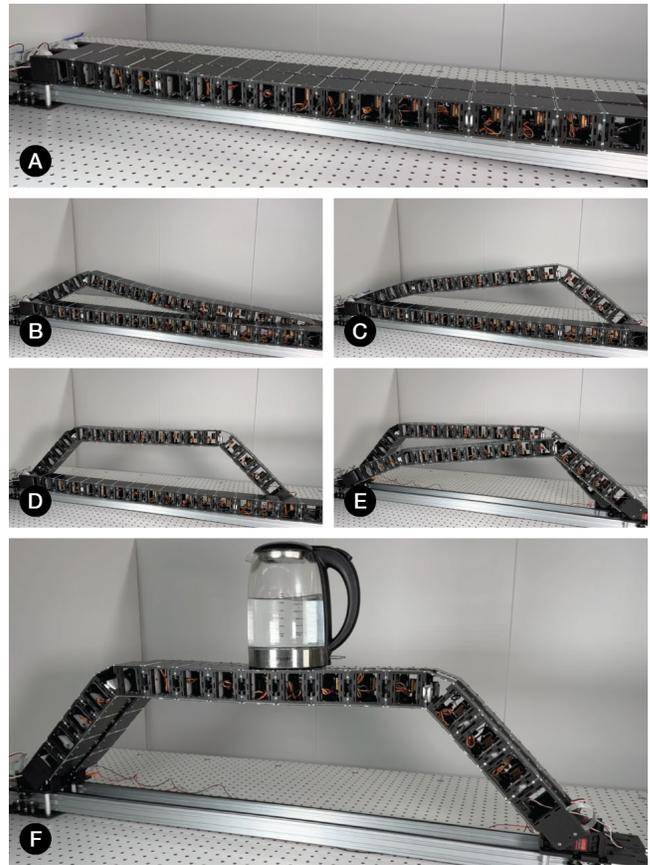


Figure 4: A table rises from the floor, shifting into place in just a few seconds (a-e). It is able to hold the 2 kg pitcher of water (f) without any visible deflection.

"material" and "machine". Such surfaces, however, are difficult to construct if we insist on directly actuating every joint. If we want something that scales as a material does, then we instead might want to examine a constraint-driven solution.

3 RELATED WORK

There is an ongoing fascination, shared by many researchers, around bringing the dynamism of software into the physical world [3, 34]. This is not a straightforward task — physical objects are much harder to reconfigure than digital ones.

3.1 Tangible, Shape-Changing Interfaces

Perhaps the most recognizable style of shape-shifting interface is the "pin display" and its variants [22]. These devices consist of individually controlled physical "bits", arranged in a two-dimensional array, that can each move up and down to alter the overall topography of a surface. The pins are typically motorized, driven by linear actuators [8], though pneumatic control schemes have been studied as well [37]. The handheld [45] and tabletop [30] demonstrations of these interfaces are compelling, and there is interest in expanding such shape-shifting surfaces to the architectural scale [16, 36].

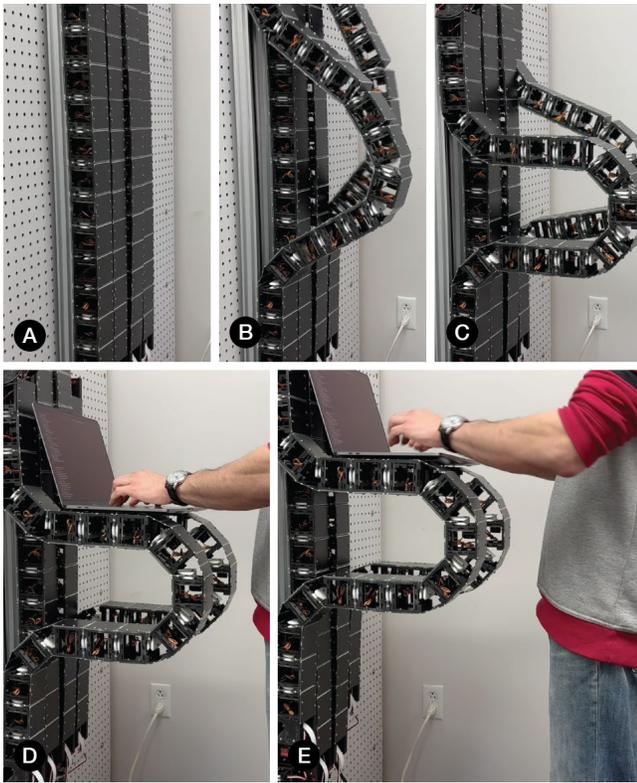


Figure 5: A lightweight desk emerges from the wall in seconds (a-c), easily able to support a laptop computer (d). In this example, the user would prefer to work while standing, so the desk raises the laptop to a comfortable height (e).

However, the construction of pin displays makes this level of scalability a challenge. In these interfaces, there is a unique actuator for each physical bit, and the shape-change capabilities of the surface are dictated by the size and strength of those actuators. Practically, this often limits us to smaller areas, and lightweight object manipulation.

At larger scales, researchers have tried to solve this problem by using latching mechanisms. Je et al. [18] demonstrate a shape-shifting floor — a single large actuator sweeps across the surface, adjusting the height of physical bits one row at a time. After each configuration step, a brake (one for each bit) locks the piece in place. The resulting surface is robust (a user can stand on it), but the transformation process takes a long time. Researchers have also explored inflatable surfaces [36, 37]. This technique has the advantage of being soft and clearly safe for human interaction, and additionally, can be collapsed into a flat state.

One constraint of pin-like shape displays (both motorized and pneumatic) is that their rendering capabilities are limited to “2.5” dimensions — overhangs and voids are impossible to achieve [34]. A different approach leverages origami-inspired techniques (bending and folding) [4]. Like pin displays, these “self-folding” surfaces also start in a flat state. But instead of actuating two-dimensional subsections (moving them orthogonally), these techniques instead

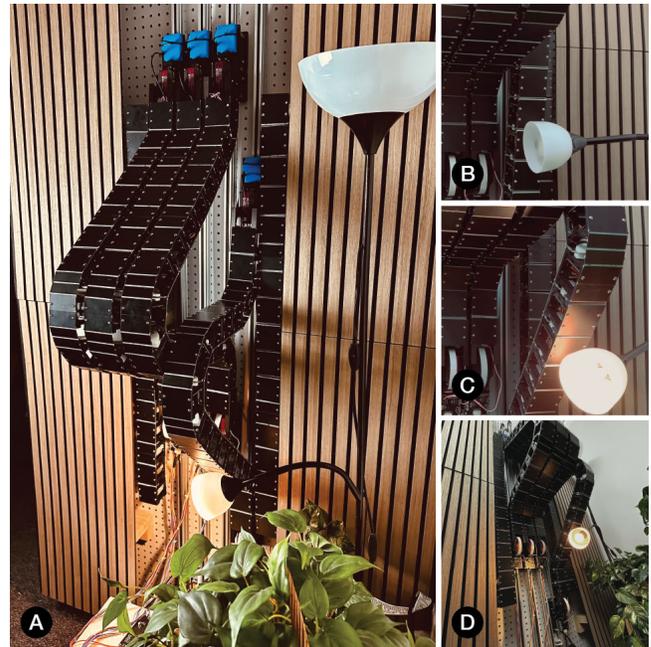


Figure 6: Our robotic surface forms a small canopy — a person can sit underneath, and use it as a cozy reading nook (a). To signal the user to engage with the space, one column of the robotic surface forms an appendage and pushes a nearby gooseneck lamp (b-d), angling it downward and illuminating the seating area.

actuate one-dimensional hinges and folds, deforming the surface. (This is closer to the approach we take in this work).

These joints can be motorized [2], but many researchers instead leverage material properties (i.e. shape memory polymers) in order to trigger a shape change [42]. These composite materials lend themselves well to miniaturization [5]. However, at larger scales, these types of actuators are often under-powered. Such challenges are familiar to roboticists; we discuss this further in the next section.

3.2 Serpentine and Continuum Robots

Many researchers look to the natural world for insights, and in the realm of shape-change, the snake is a frequent source of inspiration. The resulting serpentine robots have earned a good deal of attention, due to their ability to bend and wiggle their way through complex environments [24]. Typically, these robots are built with a motor at each joint, for maximum configurability [40, 41].

HCI researchers have adopted some of these designs for use as “one-dimensional” shape-changing interfaces such as LineFORM and ChainFORM [25, 26]. But there are challenges associated with directly driving each joint — the structure becomes heavy, and difficult to actuate at greater lengths. As a result, these interfaces are usually designed to sit flat on a table or other surface (as opposed to bending out-of-plane).

Some of these issues are addressed by continuum robots. These designs feature tendons, threaded through a tubular structure, that are pulled to alter the overall curvature of the robot. The powerful

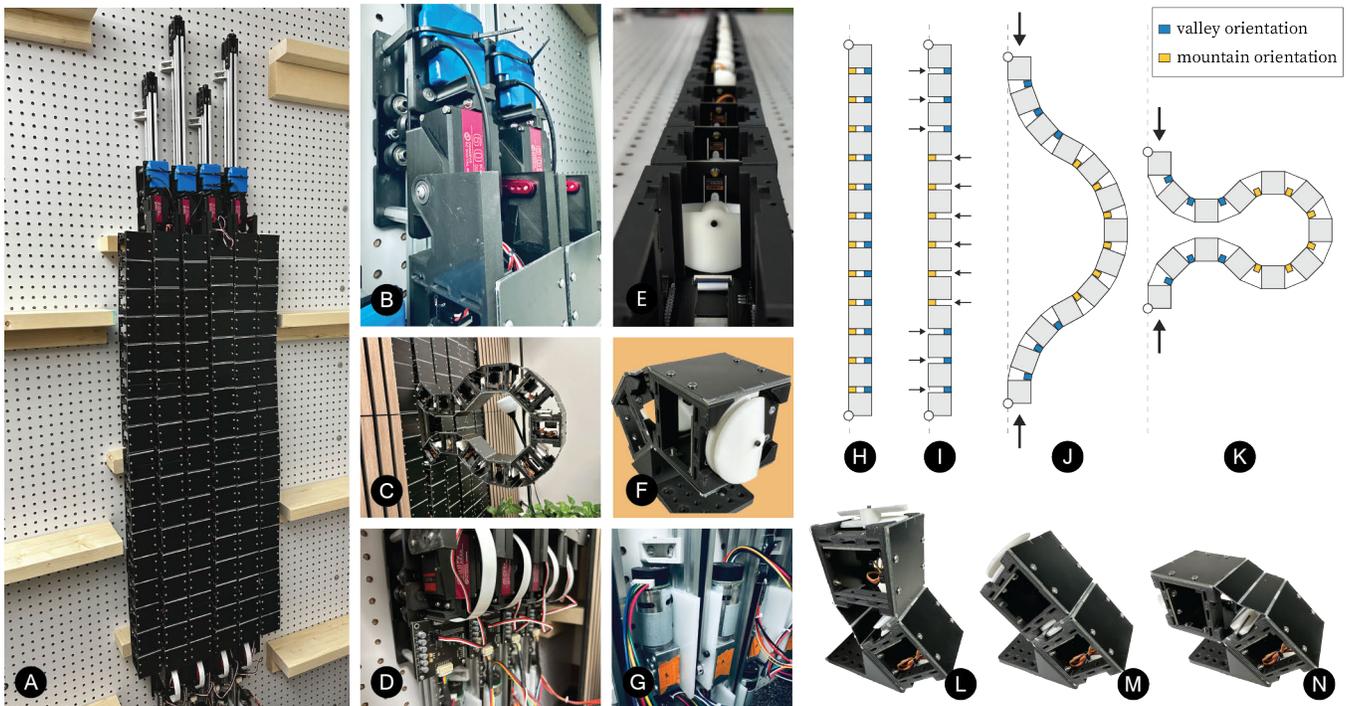


Figure 7: Our robotic surface consists of four fully-actuated columns, flanked by two decorative columns (a). Each column contains seventeen double-hinged cells, which can be constrained in either a valley (l), rigid (m), or mountain (n) orientation. These constraints are achieved by rotating a plastic clamp with a low-cost servo motor (e,f). For each cell pair, the clamp prevents either one or both sets of hinges from opening. After configuration, a belt-drive system (g) squeezes the column endpoints together, resulting in a rendered shape (c). This process is illustrated in (h-k), which models unconstrained hinges as fixed-length tendons, and models constrained hinges as ultra-stiff springs. The first and last cells are attached to servo motors (b,d), used for deployment and retraction procedures. All electronics within the column draw power from a shared 6V bus, running on flexible cables that are integrated into the cell structure. However, the servo motors attached to the topmost cell (b) are powered by a 7.4V rechargeable battery, to compensate for a small voltage drop across the column.

actuation (motors that pull the tendons) is moved *outside* of the main structure.

In traditional implementations, multiple curves require multiple sets of tendons. However, some roboticists instead approach this by using one set of tendons, but then altering the stiffness of the robot (i.e. constraining it) at various points along their primary axis [7, 20, 43]. Once these continuum robots are appropriately stiffened, the actuating tendon then pulls the robot into a newly-defined shape. This constraint-driven technique is what we adopt and expand upon in our work.

3.3 Actuated Architecture

While not traditionally thought of as a "user interface", there has been recent interest in using the built environment (i.e. architecture) as an interactive medium [13]. Shape-shifting walls, floors, and furniture can greatly impact how we experience environments [12], collaborate with others [11], adjust our posture [9], and move through physical space [27].

Walls, in particular, are ubiquitous in our daily life, and have the potential to support rich, engaging interactions – transforming into multi-use structures [16]. But perhaps due to their expensive

construction process, these kinetic, "pin-display-esque" walls exist primarily as art installations today. Within HCI, research in actuated architecture has mostly centered around controlling the in-room position of free-standing walls. When attached to motorized platforms, these roving structures can support automatic room partitioning [28], or form "just-in-time" surfaces for virtual reality experiences [35, 44]. In these instances, however, shape-change is limited to adjusting the overall height of the wall.

Non-rectangular surfaces, when built from a combination of active and passive truss members, can change form as well[21]. These mesh-based approaches are typically accompanied by an inverse design pipeline[14] – users specify multiple "keyframes", and a software system generates construction plans for a structure that can alternate between these pre-specified states.

To achieve greater shape flexibility in the "post-construction" phase, some researchers have embraced a fully-actuated, modular design for reconfigurable structures [15, 33]. Within this line of work, a popular demonstration is self-assembling furniture – chairs and tables that are composed, for instance, of multiple spherical robots. The assembly process is fascinating to watch, but there

are limitations regarding power and speed, particularly for larger structures.

An apparent conclusion is that the “actuators-everywhere” approach is difficult to implement at the architectural scale — at least when the actuators are responsible for driving the shape-change. Our strategy is to instead adopt the constraint-driven tactic seen in some serpentine robots, moving the powerful actuators to the edges of the structure, as opposed to the joints.

4 BUILDING ROBOTIC SURFACES AT HUMAN-SCALE

It is rare to see a slim robotic arm that can contend with human-scale forces. In most designs, since the end effector is relatively free to move (connected only to the last link of the kinematic chain), power-hungry motors are required at each joint to hold the assembly in position. As the number of junctions increases, the torque demands become greater and greater, adding weight and expense to the machine. Though this is often acceptable in an industrial setting, it is impractical to build a shape-changing surface in this manner — particularly if we’d like such surfaces to become ubiquitous in the built environment.

A more lightweight and scalable option is to take advantage of joint constraints (often seen in flexible, continuum robots). Under this paradigm, the heavy, powerful actuators are brought to the edges of the surface, leaving only lightweight “constraining” actuators at the joints. This is one of the key ideas in our implementation, which allows us to create robotic materials that can be integrated into the surfaces around us.

4.1 System Overview

Our prototype surface is a semi-modular material — it consists of multiple independent columns (in our prototype we provide four, but additional copies can be added), each comprising seventeen double-jointed cells. By rotating a rigid clamp with a low force servo motor, each cell pair can be constrained in one of three orientations: a mountain position, a valley position, and a coplanar (flat) position. When the endpoints of each column are squeezed together, the structure buckles out of plane, and the column forms a curve that is governed by the cell constraints (Figure 7). Each column functions as a “cross-section” of a larger three-dimensional surface.

The cells themselves are designed with a planar manufacturing process in mind, initially intended to be folded and assembled from a single flat sheet (Figure 8). This sheet is a laminated, FR4-PET composite: the rigid outer layers serve as lightweight skeleton, and the flexible inner layer works as an integrated hinge. (Similar rigid-flex constructions can be seen in [19, 29]). For ease of prototyping, we opted to replace the “pillars” of the cell with 3D-printed PLA components, allowing for faster disassembly and reassembly during development (Figure 7f). (These “hybrid” cells are the ones that we feature in all our application scenarios, mechanical tests, and supplementary video.) A single cell is 65 mm on each side, and weighs 140 grams (including all internal components).

4.2 Cell Fabrication and Assembly

Because we are interested in shape-changing surfaces at architectural scales, mass-manufacturability was front-of-mind from

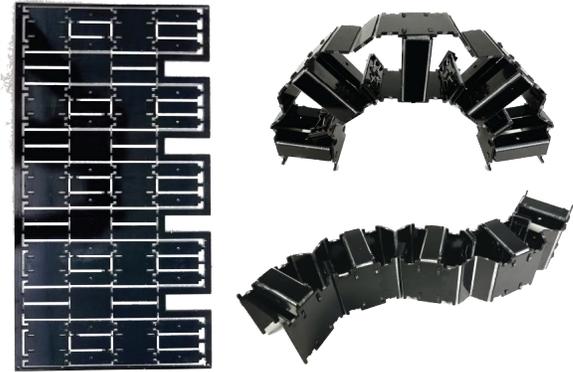


Figure 8: A flat, pre-cut sheet (left) can be folded to form a column of double-jointed cells (right). While ultimately, we used fasteners and 3D-printed pillars in our prototype columns (for quick repairability during the research phase), our design is intended to support this style of foldable construction.

the earliest stages of this research. Planar fabrication processes, such as those used in circuit board production (i.e. laser cutting, layup/bonding) are highly scalable, and our procedure is designed to be achievable within the context of this already-mature manufacturing pipeline. Both FR4 and PET are common materials in this process, which led us to choose an FR4-PET composite as our base material. (The high strength-to-weight ratio is also desirable.)

Cells were fabricated in groups of three, as shown in Figure 9. To begin, two sheets of a heat-activated adhesive, each with a paper backing, were bonded to the front and back sides of a metalized PET film¹. Bonding was achieved with an off-the-shelf heat press², at 120°C for 3 seconds. After cooling, the paper backing was removed, and the cell pattern was laser cut³ into the PET-adhesive composite. This layer was then sandwiched between two externally-machined FR4 sheets⁴, along with six flexible flat (FFC) cables⁵, and bonded at 180°C for 20 seconds. (To ensure consistent alignment, we found it helpful to add a small amount of masking tape around the edges of the still-to-be-laminated layers.)

Following the bonding process, cell segments were manually cut from their parent sheets, and the exposed FFC cables were attached to individual circuit boards, joining the cells electrically (Figure 9d,e). Note that each group of three cells contains at least one “half-piece” at its end (Figure 9e), which is connected via a “jumper” board to another group of three cells (mating with a corresponding “half-piece”). At scale, these extra jumper boards may not be necessary, as a rigid-flex PCB manufacturing process could ensure connectivity without the use of external components [17]. Cells in a column all share power connections for the motor-driven clamps, and each cell

¹0.05 mm, McMaster-Carr 7538T11

²CREWORKS 15"x15" Heat Press

³Rabbit Laser RL-80-1290

⁴0.8mm, JLCPCB

⁵JUSHUO JS05A-30P-100-4-8

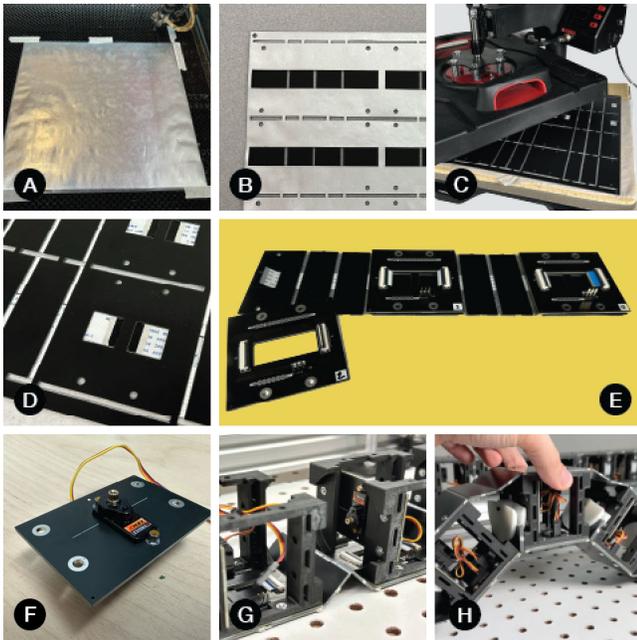


Figure 9: Fabrication process for our double-jointed, configurable columns. PET-adhesive composites are laminated then and laser-cut (a). The patterned sheet is then sandwiched between two FR4 plates (b), alongside several FFC cables, and laminated (c). The completed FR4-PET composite is shown in (d). The exposed FFC cables are attached to a connecting PCB (e). A small servo motor is attached to a mounting plate (f). This mounting plate is attached to two PLA 3D-printed frames, which are fastened to the composite (g). SLA 3D-printed clamps are added, and the structure is finalized (h).

has a unique data line for receiving servo commands. (We discuss a more extensible, daisy-chaining scheme in Section 7.)

By rotating a passive, 3D-printed clamp (Figure 7e,f), we are able to constrain each cell pair in either a mountain, valley, or rigid orientation. This clamp is driven by low-cost servo motor⁶, which we mount on a small FR4 plate (Figure 9f). This motor plate is attached to two 3D-printed frames, and the entire assembly is then fastened to the PET-FR4 laminates (Figure 9g,h).

4.3 Endpoint Actuation

The endpoints of each column (the first and last cell) can be moved axially, squeezing and stretching the chain of cells that sit between. A pair of belt-drive assemblies control the endpoint positions, pulling them along a two-meter aluminum rail. These belt drives are where our more powerful gear motors⁷ appear. An integral worm gear at the output shaft prevents back driving, so after the endpoints are moved into position, the device consumes no power. (This is important for surfaces that may be used as furniture, and will be static for long periods of time.)

⁶EMAX ES08MA II

⁷E-S Motor 5840WG-555PM-17-EN

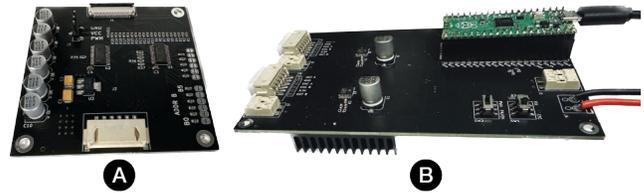


Figure 10: Circuit boards for driving small servo motors (a) and larger gearmotors (b), one pair per column. The gearmotor control firmware runs on a Raspberry Pi Pico.

A standard PID controller has some difficulty steering the column endpoints along the rail. Because the shape of the column is constantly changing, there are many non-linearities in the system – the same force from the gearmotor can have different effects on the endpoint movement, depending on both the column configuration and the deployment status. For instance, in Figure 11b, the column starts in a "stiffer" state, closer to the vertical than in Figure 11c. As a consequence, the endpoint in (b) does not move as far as the endpoint in (c), even when they are subjected to the same control signal. To compensate for this, our lower-level controller runs two feedback loops: an inner loop that ramps a PWM signal to maintain a constant velocity, and an outer loop that generates a target velocity as a function of position error. This allows for smooth-running control in any column configuration. (We also get the added benefit of speed control, which can allow for more "animated" or expressive movements.)

In addition to axial movement, we can also control the angular position of the endpoint cells. Servo motors⁸ at the ends of each column are sufficient to raise the structure out of plane in a controlled fashion (Figure 7), and are used primarily for the deployment and retraction procedures described in Section 5.

4.4 Power, Modularity, and Mounting

Each column of our surface can be constructed and controlled independently, which allows us to extend the surface by adding additional identical copies. We built two dedicated driver boards for each column – one to control the smaller motors that configure the column hinges (Figure 10a), and one to control the larger gearmotors that move the column endpoints (Figure 10b).

The driver board for the servo motors is attached to the bottom cell of the column as shown in Figure 7d, it and travels with the cell as it moves along the linear rail. It receives instructions over I2C from a master computer, and sends PWM signals to each of servo motors in the column. These servo motors run off a 6V supply that is routed the integrated flex cables in the FR4-PET composite.

The driver board for the gearmotors is stationary, located at the bottom of the rail near the gearmotors themselves (Figure 7e). An onboard Raspberry Pi Pico runs a low-level position controller, and receives high-level position commands from a master computer over a USB connection.

The linear rail is a two-meter v-slot aluminum extrusion, with a 20mm x 40mm cross section. Belt drives for both the top and

⁸DS5160, 60 kg-cm

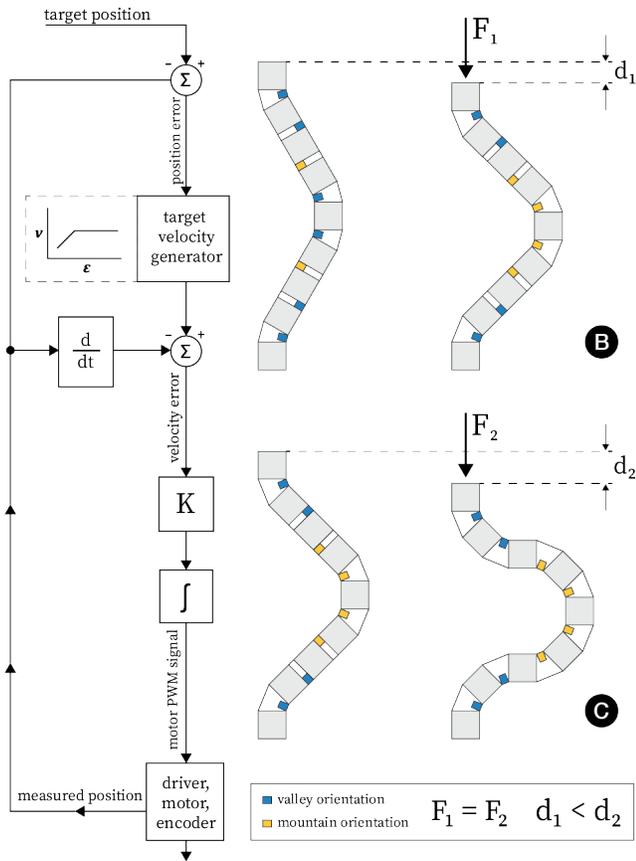


Figure 11: Since a column can be configured in many different ways, and because additional shape-change happens during the deployment process, there are many nonlinearities in our system response. The same force, applied at the endpoints of a column, may result in a different displacement depending on the initial configuration (b,c). We solve this by adding an inner loop to our position controller (left), that drives the column endpoints at a consistent velocity irrespective of the configuration. The target velocity is function of the real-time position error (it is a linear function, with a configurable limit). The proportional and integral gains transform the real-time velocity error into a PWM signal that maintains the target velocity.

bottom cell run on this rail. Two 3D-printed brackets, at the ends of each rail, are used to bolt the assemblies to a free-standing wall or horizontal base.

4.5 Software Pipeline

To demonstrate the actuation capabilities of our robotic surface, we built a lightweight software system, used to reconfigure columns, pre-program shapes and initiate transformations. It consists of (1) a custom, high-level Python API, running on a Raspberry Pi 3B, and (2) a custom, low-level motor control API, running on each of the

four Raspberry Pi Picos shown in Figure 10b (one microcontroller per column).

Through a remote Python REPL, we can use the high-level API to alter individual joint constraints (mountain/valley/rigid); control the position, velocity, and angle of column endpoints; and trigger scripted movement sequences. These scripted movement sequences (such as our deployment/retraction procedures, or pre-defined joint configurations for common shapes) are themselves written with the high-level API. During development, we would often construct shapes by SSH-ing into the Pi 3B and using the Python REPL to adjust joints and move endpoints.

Our low-level API is used for driving the column endpoints with a DRV8874 IC (see Figure 11 for the nested feedback loops that make up this motor controller). Our firmware accepts simple, "G-code-like" commands over a USB serial port, and can act as either a velocity or a position controller. (During normal operation, we do not send these commands directly; rather, this is done via the Python API).

5 HEURISTICS FOR CONTROL

By design, the mechanics of our shape-change procedure are relatively straightforward (simply configure a column, then squeeze the endpoints together). Still, there are some subtleties that bear discussing, as the nonlinear behavior of our robotic columns is not always intuitive.

When considering how to control a novel structure, it's helpful to construct a rough model of the system dynamics. For our surface, we can describe each column as a collection of square-shaped rigid bodies, connected by pairs of ultra-stiff springs. (See Figure 7, where the blue and yellow markers represent clamping forces at the valley and mountain hinges respectively.) This small amount of compliance is explicitly used in our deployment and retraction procedures; so representing the hinges as simpler, revolute joints would not be entirely sufficient.

Conceptually, configuring a column is akin to removing one (or zero) clamping springs from each cell pair. Each newly unclamped hinge can be represented as a fixed-length cable, which will be pulled taut when the corresponding cell pair is bent at 45 degrees (Figure 7).

It's also useful to draw a distinction between our system of winged, double-jointed cells, and a more "textbook" kinematic chain. Importantly, while the latter chain remains at a constant length regardless of the joint configuration, our robotic column *grows* in length as the tucked-in wings unfold from between the cells (Figure 12). This has implications for our control strategies (endpoint positioning) as well as the space of shapes that we can render.

5.1 Deploying Shapes

When a column is in its fully-retracted (flat) state, squeezing it from the edges will not trigger any transformation. The sequence of mountain and valley constraints has no effect here — the cells will simply push against each other until the axial motors stall. Similarly, if the "crest" of the column lies below the line-of-action imposed by the cell endpoints (Figure 13), then a squeezing motion will only bring the column back to a neutral position (as opposed to popping it out-of-plane).

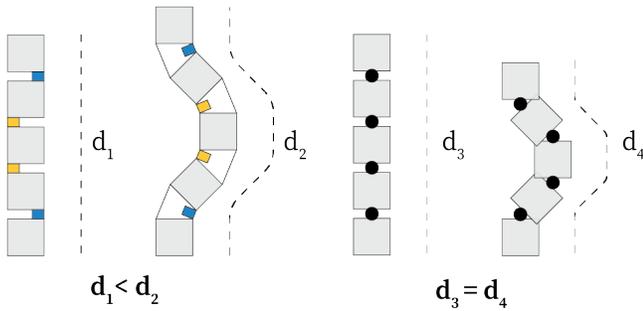


Figure 12: When the joints of a typical snake-like robot are articulated (right), the overall length of the robot does not change. However, in our double-hinged construction (left), a column will grow length as its joints rotate. This change in length has implications for our control techniques, as well as the space of shapes we can achieve.

To successfully deploy a column, we must first move it from a “retractable” state (crest below the line-of-action) to a “deployable” state (crest above the line-of-action). However, since the length of the column *increases* as the unclamped wings unfold, there is a region between these two states where cells are squeezed together – this forms a small mechanical energy barrier that we must overcome (Figure 13c). Put another way, the column exhibits a degree of bistability. Near this energy barrier, spring forces in the hinges will push the column into either the deployable or retractable state. (Anecdotally, colleagues who have physically interacted with the wall have described this feeling as “magnetic”.)

While a user can transverse this energy barrier by physically pushing or pulling on a column, an automated control strategy is often more desirable. To allow for this, the first and last cells in our columns are attached to servo motors (Figure 7), which can tilt these endpoints out-of-plane. Importantly, since the pivot is located on the side of the cell closest to the wall (the opposite of a typical valley fold), this rotation also creates additional space for the newly-lengthened column to pass through. Once the crest is sufficiently out of plane, we can return the endpoint to its original flat state.

From here, deployment becomes more intuitive. Moving the column endpoints closer together will cause the configured structure to bend, and the desired shape is achieved.

5.2 Retracting Shapes

The bistability that we encounter during deployment adds some complexity to the retraction process as well. To retract a shape, we begin by driving the endpoints of the deployed column back to their original position. At this point, however, we are still on the “wrong side” of the energy barrier – and unlike the deployment procedure, we can not rely on the endpoint servos to push the column over this boundary. (Flush with the rest of the wall, these first and last cells are already at or near their minimum angular position.) Instead, we drive the endpoints even further apart, stretching the column as far as the configuration allows. This quick motion triggers an oscillation within the column, and the crest bobs above and below

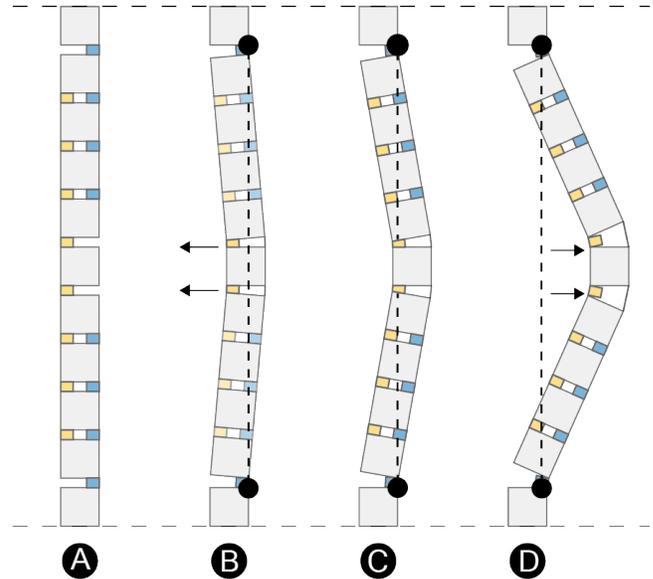


Figure 13: A configured column (a) may exist in a bistable state. The mechanical singularity is depicted in (c) – though the endpoints are fixed, the column has slightly grown in length (see Figure 12), and the folded hinges act as internal springs, pushing the cells apart. From this state, the column will snap to either (b) or (d), depending the external disturbance. In state (b), the crest of the column is below the line of action imposed by the endpoints, so squeezing the column will return it to state (a). In state (d), the crest of the column is above the line of action, so squeezing the column will deploy the shape.

the endpoint line-of-action. With precise timing, we can “catch” the crest at the right moment, squeeze the endpoints together, and return to the neutral (flat) state. The entire sequence occurs within a one-second timeframe – the squeeze is initiated approximately 300 ms after the stretch. (Once empirically determined, we used this timing for all application examples without issue).

Even though this is an open-loop routine, we are able to reliably retract the columns using this “stretch-oscillate-squeeze” approach. Still, for a more robust procedure, it may be beneficial to add an accelerometer to our FR4 cells, and leverage orientation data from the structure (see section 6 for a brief discussion).

5.3 Axial Movement

In the procedures so far, we’ve only considered manipulating the position of *one* endpoint; the other has remained fixed. However, in our columns, both the first *and* last cells can be driven independently along a linear rail. This allows us to alter the deployment trajectory as well as the resultant shape. In Figure 6 for example, we squeeze our columns from the bottom rather than the top, arriving at an alcove-like structure that a person can sit underneath. In Figure 7, we squeeze both top and bottom endpoints together simultaneously, so that during deployment, the crest of the column travels perpendicular to the surface (as opposed to an upward or

downward angle). This latter type of motion is particularly relevant for robotic manipulation tasks, for which we require position control of an end effector at multiple points along a trajectory.

In addition, we can axially reposition the deployed columns, by moving both endpoints in the same direction. (The position control loop described in Section 4.3 helps keep these endpoints in sync – without it, we are at risk of damaging the column by over-stretching or over-squeezing.) This is how we achieve the adjustable-height desk in Figure 5, and the “wave” greeting in Figure 2.

6 MECHANICAL CHARACTERIZATION

One of the main contributions of our work is a lightweight robotic material capable of withstanding human-scale forces. During the development of our application scenarios, we found that the surface could handle common household objects (full pitchers of water, laptop computers, cast-iron pans) without any visible trouble. Specifically, we tested the two-column shelf configuration (shown in Figure 5d) at loads up to 5 kg (11 lb), and our system was able to move these loads vertically at 12 cm/s. This load capacity is shape-dependent. A surface configuration made with “triangular supports” (see the right-most column in Figure 6a) is most robust, and when constructed with two columns, this “reinforced shelf” was able to handle loads of 7 kg before noticeable fatigue. (Note that for the above examples, we did not test to failure; see the next section for these characteristics).

For a more precise characterization, we measured the forces on our cells in the three configurations of clamping and under tension and compression. We performed three tests, as outlined in the following subsections. We used a Mark-10 ESM303 Motorized Tension/Compression Test Stand with a 1.5 kN load cell for all tests.

Compression - both hinges clamped. Bending forces are generally of most concern to our robotic column. To evaluate these, we mounted two cells onto a custom made jig that positions them in a 45° angle. In this setup, we clamped both hinges to produce our rigid connection and performed a compression test, i.e., pushed the load cell down onto the cell. We show the test setup and the results in Figure 14a. Our test shows that the cells can withstand up to 302 N.

Compression - bottom hinge clamped. Next, we were interested in how much force the cells can withstand if the only the bottom hinge is clamped, while the top hinge can extend. We kept the same test setup and only altered the clamping configuration, as shown in Figure 14b. The force that it can withstand is higher at 486 N, where it started to fail. The increase in strength is contributed by the top hinge being in tension, which is more favorable for our construction.

Tension - top hinge clamped. Lastly, we tested how our cells perform under tension. As the test setup in Figure 14c shows, we clamp the top hinge and pull on the cell. This is the weakest configuration of our robotic column and withstands 126 N before the SLA-printed clamp breaks.

When we test the cell pairs to failure, the failure reliably occurs at the SLA 3D-printed clamp, as shown in Figure 15d. A different material choice for the clamp could postpone this breakage – but

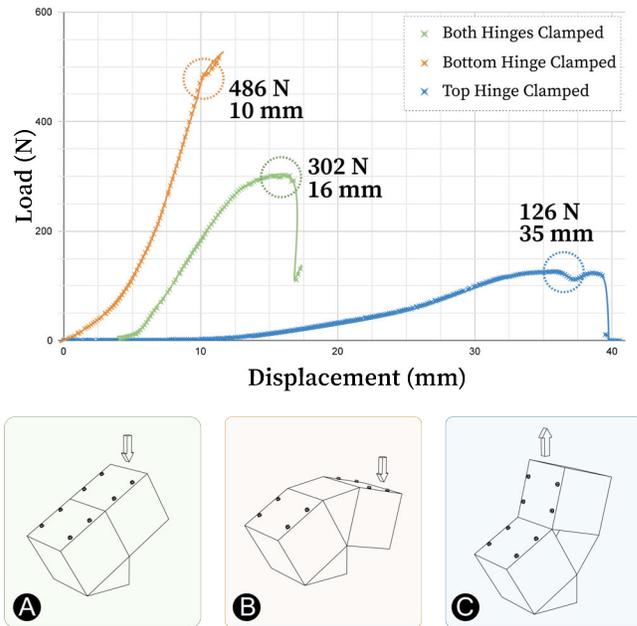


Figure 14: Mechanical tests for cell pairs in multiple configurations. (a) Compression test on cells with both hinges clamped mounted in a 45° degree angle. They can withstand 302 N of force. (b) Compression test with only the bottom hinge clamped yields 486 N of strength. (c) Tensile test on cells with the top hinge clamped shows 126 N of strength.

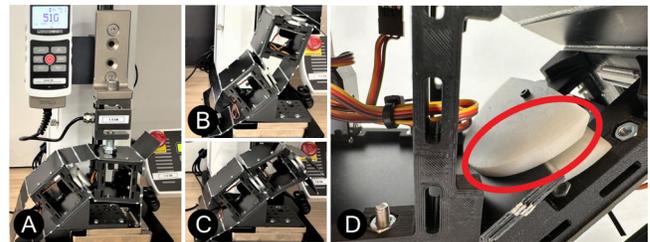


Figure 15: Test configuration for cell pairs (a-c). Failure occurs at the SLA 3D-printed clamp (d) circled in red.

we still prefer for the clamp to be the first component to fail, since it is the easiest component to replace.

7 DISCUSSION, FUTURE WORK, AND CONCLUSION

Typically, programmable surfaces make use of a separate actuator for each configurable bit, with the actuators themselves responsible for driving the shape change in their own local areas. This technique is difficult to implement at large scales, because a high-powered actuator is required for each individual cell. Our constraint-driven approach circumvents this problem. By decoupling the “per-cell” actuation from the “driving” actuation (moving powerful motors to the surface boundaries) we are able to achieve human-scale shape change without the use of large motors for each cell.

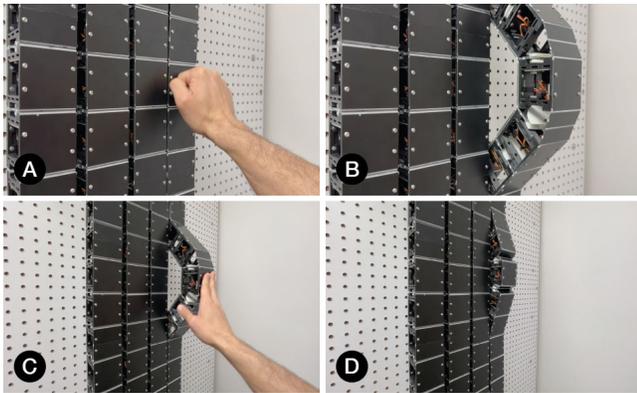


Figure 16: A mock interaction with our robotic surface. A user knocks on a section of the wall (a) and the corresponding piece pops out of plane (b). A user could potentially use this as a handle to pull on the column, and shape the surface into a desirable object. To return the handle to a neutral state, the user might nudge the protrusion (c), and the surface would retract this piece (d).

Constraints in our system can be quickly added and removed, and the deployment process takes a few seconds or less. This speedy reconfiguration means that our surface can support real-time user interactions, which we plan to explore more extensively in future work. In Figure 16, we show a mock example — a user knocks on a portion of the wall, and the wall “responds” by popping that small portion out-of-plane. The user could then use this newly emerged section as a “handle” to pull and sculpt the column into a desired shape. If, on the other hand, they decide that they do not need this handle, they could give it a nudge (Figure 16c), and watch it retract back into the wall. (While we consider this to be a relatively safe interaction, users would have to be mindful of a pinching force that can occur between cells, particularly as a column folds into the flat state. A fully-featured, user-facing robotic surface might operate similarly to an automatic doorway, reopening upon detection of an obstruction).

The interactions that we describe require input sensing, to perceive both intentional and unintentional user contact. Such sensors (i.e. for detecting acceleration, orientation, or capacitance) could be added to our FR4 column with little or no change to the overall structure (the integrated flex cables already provide power and data connections to each cell).

Beyond detecting human touch, adding these sensors would also allow for a tighter control loop. Although we already have closed-loop control over the column endpoints and clamp position, the column can still fall into an “unknown” state if it is in an underconstrained configuration (multiple angles under 45 degrees). While it’s good to know that we can successfully deploy and retract shapes with an open-loop approach, additional orientation data would allow for robust, in-situ reconfigurations.

Going forward, we may also wish to alter our cell-to-cell communication scheme. Currently, each cell in a column receives a PWM signal from a per-column control board (Figure 10a), which rotates the associated clamp. However, a separate data line for each cell

will not scale to longer columns — for better extensibility, we may wish to adopt a daisy-chained control scheme [10].

In our current system, cells are 65mm on each side. If we wish to increase our resolution (decrease the cell size), we may also want to explore alternate clamping mechanisms. Some possibilities include stiffness control through shape-memory polymers [7], pneumatically-actuated pressurized joints [32], and electrostatics [23].

Finally, future work should go towards making these surface more “expressive”. Especially for non-anthropomorphic robots, exaggerated or animated movements can be useful for communicating intent, and are generally more engaging for human users [1, 39]. Some users also enjoy the perceived companionship they get from objects such as these, particularly as an add-on to an otherwise functional device [46]. Subtle gestures from these robotic objects can even mediate human-human communications, while still existing on the periphery of our attention [6]. Architecture can shape human interactions, and *robotic* architecture expands this space even further.

Our robotic surface provides a platform for exploring topics such as these. By adopting a constraint-based configuration method (as opposed to directly driving the joints), we are able to build shape-changing surfaces at human-scale, allowing for rich new interactions between people and the built environment. Our fabrication technique, which makes use of rigid-flex composites, is not only accessible to researchers, but can be readily industrialized using planar manufacturing processes. Our application examples demonstrate the validity of this approach, as we are able to render multiple architectural objects (shelves, desks, and alcoves), all from a single surface.

REFERENCES

- [1] Lucy Anderson-Bashan, Benny Megidish, Hadas Erel, Iddo Wald, Guy Hoffman, Oren Zuckerman, and Andrey Grishko. 2018. The greeting machine: an abstract robotic object for opening encounters. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 595–602.
- [2] Christoph H Belke and Jamie Paik. 2017. Mori: a modular origami robot. *IEEE/ASME Transactions on Mechatronics* 22, 5 (2017), 2153–2164.
- [3] Alberto Boem and Giovanni Maria Troiano. 2019. Non-rigid HCI: A review of deformable interfaces and input. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 885–906.
- [4] Sebastien JP Callens and Amir A Zadpoor. 2018. From flat sheets to curved geometries: Origami and kirigami approaches. *Materials Today* 21, 3 (2018), 241–264.
- [5] Shanshan Chen, Jianfeng Chen, Xiangdong Zhang, Zhi-Yuan Li, and Jiafang Li. 2020. Kirigami/origami: unfolding the new regime of advanced 3D microfabrication/nanofabrication with “folding”. *Light: Science & Applications* 9, 1 (2020), 75.
- [6] Hadas Erel, Denis Trayman, Chen Levy, Adi Manor, Mario Mikulincer, and Oren Zuckerman. 2022. Enhancing emotional support: The effect of a robotic object on human-human support quality. *International Journal of Social Robotics* 14, 1 (2022), 257–276.
- [7] Amir Firouzeh, Marco Salerno, and Jamie Paik. 2017. Stiffness control with shape memory polymer in underactuated robotic origamis. *IEEE Transactions on Robotics* 33, 4 (2017), 765–777.
- [8] 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.
- [9] Kazuyuki Fujita, Aoi Suzuki, Kazuki Takashima, Kaori Ikematsu, and Yoshifumi Kitamura. 2021. TiltChair: manipulative posture guidance by actively inclining the seat of an office chair. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [10] Jesse T Gonzalez and Scott E Hudson. 2022. Layer by Layer, Patterned Valves Enable Programmable Soft Surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–25.

- [11] Jens Emil Grønbaek, Henrik Korsgaard, Marianne Graves Petersen, Morten Henriksen Birk, and Peter Gall Krogh. 2017. Proxemic transitions: designing shape-changing furniture for informal meetings. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 7029–7041.
- [12] Erik Grönvall, Sofie Kinch, Marianne Graves Petersen, and Majken K Rasmussen. 2014. Causing commotion with a shape-changing bench: experiencing shape-changing interfaces in use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2559–2568.
- [13] Mark D Gross and Keith Evan Green. 2012. Architectural robotics, inevitably. *Interactions* 19, 1 (2012), 28–33.
- [14] Jianzhe Gu, Yuyu Lin, Qiang Cui, Xiaoqian Li, Jiayi Li, Lingyun Sun, Cheng Yao, Fangtian Ying, Guanyun Wang, and Lining Yao. 2022. PneuMesh: Pneumatic-driven truss-based shape changing system. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [15] Simon Hauser, Mehmet Mutlu, P-A Léziart, Hala Khodr, Alexandre Bernardino, and Auke Jan Ijspeert. 2020. Roombots extended: Challenges in the next generation of self-reconfigurable modular robots and their application in adaptive and assistive furniture. *Robotics and Autonomous Systems* 127 (2020), 103467.
- [16] Jiayi Hong, Xiyao Wang, Aluna Everitt, and Anne Roudaut. 2023. Interacting with actuated walls: Exploring applications and input types. *International Journal of Human-Computer Studies* 172 (2023), 102986.
- [17] John Isaac. 2007. Rigid-flex technology: mainstream use but more complex designs. *CIRCUITREE-CAMPBELL*- 20, 10 (2007), 30.
- [18] Seungwoo Je, Hyunseung Lim, Gongpyung Moon, Shan-Yuan Teng, Jas Brooks, Pedro Lopes, and Andrea Bianchi. 2021. Elevate: A walkable pin-array for large shape-changing terrains. In *Proceedings of the 2021 CHI Conference on human Factors in Computing Systems*. 1–11.
- [19] Jaakko T Karras, Christine L Fuller, Kalind C Carpenter, Alessandro Buscicchio, Dale McKeely, Christopher J Norman, Carolyn E Parcheta, Ivan Davydychev, and Ronald S Fearing. 2017. Pop-up mars rover with textile-enhanced rigid-flex PCB body. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5459–5466.
- [20] Joshua Kaufmann, Priyanka Bhovad, and Suyi Li. 2022. Harnessing the multi-stability of kresling origami for reconfigurable articulation in soft robotic arms. *Soft Robotics* 9, 2 (2022), 212–223.
- [21] Robert Kovacs, Alexandra Ion, Pedro Lopes, Tim Oesterreich, Johannes Filter, Philipp Otto, Tobias Arndt, Nico Ring, Melvin Witte, Anton Synytsia, et al. 2018. TrussFormer: 3D printing large kinetic structures. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 113–125.
- [22] 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.
- [23] David J Levine, Kevin T Turner, and James H Pikul. 2021. Materials with electro-programmable stiffness. *Advanced Materials* 33, 35 (2021), 2007952.
- [24] Jindong Liu, Yuchuang Tong, and Jinguo Liu. 2021. Review of snake robots in constrained environments. *Robotics and Autonomous Systems* 141 (2021), 103785.
- [25] 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.
- [26] 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.
- [27] Binh Vinh Duc Nguyen, Jihae Han, and ANDREW VANDE MOERE. 2022. Towards Responsive Architecture that Mediates Place: Recommendations on How and When an Autonomously Moving Robotic Wall Should Adapt a Spatial Layout. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–27.
- [28] Yuki Onishi, Kazuki Takashima, Shoi Higashiyama, Kazuyuki Fujita, and Yoshifumi Kitamura. 2022. WaddleWalls: Room-scale Interactive Partitioning System using a Swarm of Robotic Partitions. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–15.
- [29] Daniela Rus and Michael T Tolley. 2018. Design, fabrication and control of origami robots. *Nature Reviews Materials* 3, 6 (2018), 101–112.
- [30] Alexa F Siu, Eric J Gonzalez, Shenli Yuan, Jason B Ginsberg, and Sean Follmer. 2018. Shapeshift: 2D spatial manipulation and self-actuation of tabletop shape displays for tangible and haptic interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [31] Hyegyo Son, Yunha Park, Youngjin Na, and Changkyu Yoon. 2022. 4D Multiscale Origami Soft Robots: A Review. *Polymers* 14, 19 (2022), 4235.
- [32] Canberk Sozer, Linda Paternò, Giuseppe Tortora, and Arianna Menciassi. 2022. A novel pressure-controlled revolute joint with variable stiffness. *Soft Robotics* 9, 4 (2022), 723–733.
- [33] Alexander Sproewitz, Aude Billard, Pierre Dillenbourg, and Auke Jan Ijspeert. 2009. Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture. In *2009 IEEE international conference on robotics and automation*. IEEE, 4259–4264.
- [34] Miriam Sturdee and Jason Alexander. 2018. Analysis and classification of shape-changing interfaces for design and application-based research. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 1–32.
- [35] Ryo Suzuki, Hooman Hedayati, Clement Zheng, James L Bohn, Daniel Szafir, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2020. Roomshift: Room-scale dynamic haptics for vr with furniture-moving swarm robots. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–11.
- [36] Ryo Suzuki, Ryosuke Nakayama, Dan Liu, Yasuaki Kakehi, Mark D Gross, and Daniel Leithinger. 2020. LiftTiles: constructive building blocks for prototyping room-scale shape-changing interfaces. In *Proceedings of the fourteenth international conference on tangible, embedded, and embodied interaction*. 143–151.
- [37] Shan-Yuan Teng, Cheng-Lung Lin, Chi-huan Chiang, Tzu-Sheng Kuo, Liwei Chan, Da-Yuan Huang, and Bing-Yu Chen. 2019. Tilepop: Tile-type pop-up prop for virtual reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 639–649.
- [38] 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.
- [39] Steve Whittaker, Yvonne Rogers, Elena Petrovskaya, and Hongbin Zhuang. 2021. Designing personas for expressive robots: Personality in the new breed of moving, speaking, and colorful social home robots. *ACM Transactions on Human-Robot Interaction (THRI)* 10, 1 (2021), 1–25.
- [40] Cornell Wright, Austin Buchan, Ben Brown, Jason Geist, Michael Schwerin, David Rollinson, Matthew Tesch, and Howie Choset. 2012. Design and architecture of the unified modular snake robot. In *2012 IEEE international conference on robotics and automation*. IEEE, 4347–4354.
- [41] Cornell Wright, Aaron Johnson, Aaron Peck, Zachary McCord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-Rivero, Ross Hatton, and Howie Choset. 2007. Design of a modular snake robot. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2609–2614.
- [42] Zheng Yan, Fan Zhang, Jiechen Wang, Fei Liu, Xuelin Guo, Kewang Nan, Qing Lin, Mingye Gao, Dongqing Xiao, Yan Shi, et al. 2016. Controlled mechanical buckling for origami-inspired construction of 3D microstructures in advanced materials. *Advanced functional materials* 26, 16 (2016), 2629–2639.
- [43] Yang Yang, Yonghua Chen, Yingtian Li, and Michael Zhiqiang Chen. 2016. 3D printing of variable stiffness hyper-redundant robotic arm. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3871–3877.
- [44] Yan Yixian, Kazuki Takashima, Anthony Tang, Takayuki Tanno, Kazuyuki Fujita, and Yoshifumi Kitamura. 2020. Zoomwalls: Dynamic walls that simulate haptic infrastructure for room-scale vr world. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 223–235.
- [45] Shigeo Yoshida, Yuqian Sun, and Hideaki Kuzuoka. 2020. Pocopo: Handheld pin-based shape display for haptic rendering in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [46] Oren Zuckerman, Dina Walker, Andrey Grishko, Tal Moran, Chen Levy, Barak Lisak, Iddo Yehoshua Wald, and Hadas Erel. 2020. Companionship is not a function: The effect of a novel robotic object on healthy older adults' feelings of "being-seen". In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.