4DMesh: 4D Printing Morphing Non-Developable Mesh Surfaces

Guanyun Wang[1], Humphrey Yang[2], Zeyu Yan[3], Nurcan Gecer Ulu[3], Ye Tao[1,4], Jianzhe Gu[1], Levent Burak Kara[3], Lining Yao[1]

[1]HCI Institute, Carnegie Mellon University {guanyunw, ytao2, jianzheg, liningy}@andrew.cmu.edu

[2]School of Architecture, Carnegie Mellon University hanliny@andrew.cmu.edu

[3]College of Engineering, Carnegie Mellon University {zeyuy, ngecerul, ikara}@andrew.cmu.edu

[4]Zhejiang University Hangzhou, China taoye@zju.edu.cn

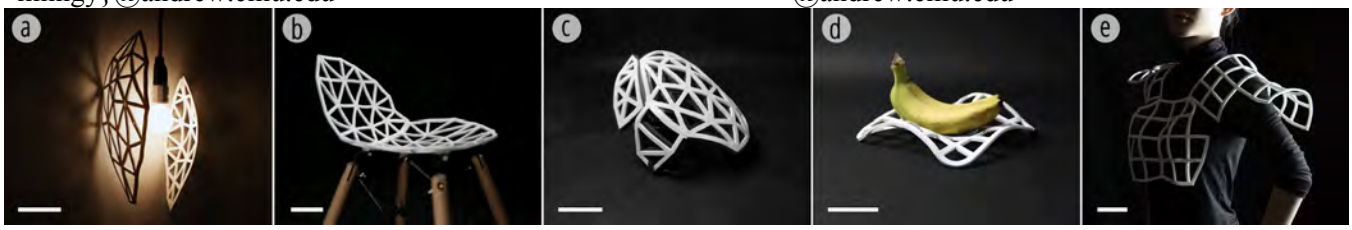

Figure 1. 4DMesh applications: (a) Lampshade; (b) Chair; (c) Helmet; (d) Fruit plate; (e) Costume. (Scale bar: 10 cm)

ABSTRACT

We present 4DMesh, a method of combining shrinking and bending thermoplastic actuators with customized geometric algorithms to 4D print and morph centimeter- to meter-sized functional non-developable surfaces. We will share two end-to-end inverse design algorithms. With our tools, users can input CAD models of target surfaces and produce respective printable files. The flat sheet printed can morph into target surfaces when triggered by heat. This system saves shipping and packaging costs, in addition to enabling customizability for the design of relatively large non-developable structures. We designed a few functional artifacts to leverage the advantage of non-developable surfaces for their unique functionalities in aesthetics, mechanical strength, geometric ergonomics and other functionalities. In addition, we demonstrated how this technique can potentially be adapted to customize molds for industrial parts (e.g., car, boat, etc.) in the future.

Author Keywords

4D printing; 3D printing; shape changing interfaces; non-developable surface; mesh surface; morphing; self-folding; self-assembly.

CCS Concepts

• Human-centered computing~Human computer interaction (HCI)

UIST '18, October 14–17, 2018, Berlin, Germany

ACM ISBN 978-1-4503-5948-1/18/10…$15.00
https://doi.org/10.1145/3242587.3242625

INTRODUCTION

Researchers have envisioned a 4D printing morphing system whose structures can enable a new way of manufacturing [30]. 4D printing can save assembly effort [50], printing material, time [4, 7] and shipping/packaging costs by manufacturing flat artifacts that self-assemble into 3D shapes on site [54]. To push the practical uses of 4D printed self-deployable structures further, we have to make efforts on the following aspects: material composition design, manufacturing procedure, and design tools. While researchers in engineering have been making great progress developing materials and manufacturing procedures [4, 13, 29, 30], designing practical and suitable design tools to augment the design potential requires the effort of HCI.

In this paper, we focus on design tools and workflows. We will share two *end-to-end inverse design* approaches to 4D print morphing mesh surfaces that are *non-developable*. With our tools, users can input CAD models of target surfaces and produce respective printable files.

We focus on *non-developable* surfaces because they are favored in both artificial and natural systems. Non-developable surfaces are often associated with mechanical performances (e.g., domes and egg shells), aerodynamics (e.g., shapes of fast swimming fish), ergonomics (e.g., chairs with body-fitting shapes) and unique aesthetics (e.g., organic and biomimetic architectures). We believe the ease of workflow for such surfaces will enrich the toolbox of 4D printing, or the fabrication of shape changing materials and interfaces in general. Nonetheless, these shapes are hard or time-consuming to make with other manufacturing methods including laser cutting and CNC milling. In 3D printing, these surfaces often require extensive supporting structures that render the fabrication process inefficient. Compared to flatly packed surfaces (e.g., certain IKEA furniture), non-developable structures take up more spaces for packaging

and shipping. With our design tools and workflow, we can effectively tackle some of these challenges. The main contributions of this work are as follows:

- **Techniques**. Two end-to-end pipelines to flatten a given non-developable geometry into printable G-code. One pipeline is based on bending actuators adapted to the Chebyshev Net algorithm; the other is based on shrinking actuators adapted to a conformal mapping algorithm. Table 1 situated 4DMesh among all the design tools for material-driven morphing structures.
- **Design**. Applications emphasizing the flat-packing, morphing, and customizability of non-developable and organic shapes in the context of home furniture, bespoke wearables, and customizable composite molds.
- **Performance Quantification**. Empirical analysis and characterization of guidelines and challenges of thermoplastic-based 4D printing from centimeter to meter in scale.

		Dimension				
		nm	μm	mm	cm	m
Non-developable	Inverse Design				4DMesh	
	Forward Simulation			4D biomimetic printing [13]	Active printing [44]	
Developable	Inverse Design	DNA and RNA origami [16]		Uniform heating [2] Thermorph [4]		
	Forward Simulation	DNA and RNA origami [16]		Thermorph [4]	Self-folding polymer [26]	

Table 1. 4DMesh contributes an inverse design workflow that covers centimeter to meter scale non-developable surfaces.

BACKGROUND

Residual Stress in Thermoplastic Processing

Residual stress in polymer refers to the stress remaining in a part that has been chilled quickly during or after molding, extrusion, or forming [14]. Directly relevant to 4DMesh, FDM 3D printing causes flow induced residual stress, which has been utilized for 4D printing repeatedly [4, 29]. When melted in the heated extruder, the polymer molecules are at an unstressed, equilibrium, and random coil state. During the extrusion process, the molecules are aligned to the flow direction as it is sheared and elongated. When the polymer hits the printing bed, the solidification occurs before the polymer molecules can be fully relaxed to their state of equilibrium. The molecular orientation is locked within the formed part, which causes residual stress. The stress can be released if the polymer is re-heated above its own glass transition temperature and softened.

Non-developable Surface

Shell structures can be categorized into single or double curvature structures. Single curvature shell, or developable surface, is curved on one linear axis and is a part of a cylinder or cone; double curvature shell, or *non-developable* surface, is either part of a sphere or a hyperboloid of revolution [25]. With our technique, we can self-deploy both developable and non-developable surfaces in mesh. As developable surfaces are relatively easier to flatten and have design tools discussed previously [2, 4], we will mainly focus on non-developable surfaces in the following sections.

RELATED WORK

Personal Fabrication and Customization Design

Personal fabrication devices such as laser cutters and 3D printers allow users to create physical objects economically, rapidly and efficiently. To lower the prerequisites and to democratize these tools, researchers developed methods to streamline design and fabrication in sculpting [62], laser cutting [32], modeling and prototyping [56], and knitting [8]. Meanwhile, to achieve customization design, researchers have advanced the boundaries of 3D printing techniques. For example, printing explorations have been conducted with soft fabrics [18, 39, 45]; *3D Printed Hair* [23] and *Cilllia* [36] investigated methods and applications for printing hair-like artifacts; artifacts are produced with tailored mechanical functions [19, 20]; *ExoSkin* [12] inquired into on-body fabrication; and *ReForm* [55] and *Reprise* [5] probed the concept of iterative design in 3D printing context. 4DMesh investigates and expands the boundary of personal fabrication and personalized design by proposing a method to 3D print artifacts with 2D sheets, allowing for rapid customization and fabrication of object.

Shape Changing Interfaces and 4D Printing

Recently, researchers approached shape changing interfaces in various ways. For instance, *PneUI* [58], *Sketching in circuits* [43], and *Organic Primitives* [21] explored this topic using different materials; *inForm* [11], *LineForm* [34], and *Printflatables* [46] approached from a function point of view; *Jamming user interfaces* [10], *JamSheets* [38], and *aeroMorph* [37] investigated using different fabrication techniques.

In addition to 3D printing, 4D printing encodes an extra dimension of transformation over time in the artifact [30, 44]. The fabricated objects can be triggered using hot water or other forms of energy. 4D printing has recently become a popular research interest in HCI. Using biological materials to encode transformation and humidity to trigger, *bioLogic* [59] achieved reversible actuators. *xPrint* [53] developed a modular printing system for 4D printing various materials. *Transformative Appetite* [54] proposed making 3D foods as flat pieces, saving space and shipping cost for manufacturers. Using hot water to trigger, our work harnesses these advantages of making artifacts as flat pieces, and offers a method to design customized 4D artifacts.

Shape Memory Thermoplastics in HCI

Thermoplastics are widely-used in daily life, construction, engineering, and rapid prototyping. Other than injection modeling and FDM based 3D printing, thermoplastic has been introduced to HCI in various studies. *ShrinkyCircuits* [28] and *Foldio* [35] used electronics along with thermoplastics to design shape-changing interfaces, *Thermorph* [4] and *Printed Paper Actuator* [52] used pre-

strained composite materials to produce 4D artifacts. Beyond HCI, shape memory thermoplastics have been used to make non-reversible interfaces [2, 51] and shrinkage-based shape memory interfaces [29]. Researchers also embedded resistive heating components to selectively heat regions to achieve sequential morphing [9].

In particular, 4DMesh is different from *Thermorph* in three aspects: 1) Scale. The improvement in scale is one of the critical achievements of 4DMesh, covering many challenges such as cubically growing gravitational effects, new global mesh structures, optimized inner structure and toolpath to decrease the weight and the printing time, alternative actuator types for controllable behavior at this scale, and altered triggering conditions. 2) Developability. *Thermorph* re-meshes non-developable surfaces into foldable shapes with flat faces, often requiring cuts into the geometry which weaken their mechanical strengths; in contrast, 4DMesh can flatten the geometry as a smooth surface without any cuts. 3) Material Usage. Instead of using two different materials in actuation mechanism, our approach introduces morphing process along the beam using the print direction relative to the longitudinal axis to determine the out-of-plane bending direction, which can be achieved with a singular material.

Building Objects via Mesh Structure

Using mesh-like structures, we can approximate 3D shapes with elements of lower dimension. *WirePrint* [31] and *On-The-Fly Print* [40] replicated mesh edges to give a preview of the object itself. Wu et al. [57] introduced a pipeline to print 3D wireframes from meshes using 5DOF printers. *WireDraw* [60] prototyped models with a doodle pen. In manual assembling fabrication, Cignoni et al. [6] introduced a method to produce complex objects with mesh structures. A series of studies aimed to manufacture artifacts with mesh structures across various scales. *WeaveMesh* [49] for making wearables and gadgets; *Protopiper* [1] for making furniture-sized artifacts; *TrussFab* [22] for making room-sized installations. Our research takes inspiration from meshes to produce objects scaling from wearables to furniture.

Surface Flattening for Fabrication

Fabricating 3D objects with 2D patterns requires geometric flattening and/or segmentation methods adapted for specific material properties. Researches in HCI and computer graphics provided us with various pipelines intended for different fabrication methods and materials. *CardBoardiZer* [61] processed 3D meshes into 2D foldable pieces; *CurveUps* [15] unrolled 3D surfaces into flat, tension-actuated plates with cuts; Designing inflatable structures [47] divided inflatable shapes into flat pieces; Jesús Pérez's team [41] developed a method to fabricate Kirchhoff-Plateau surfaces with fabrics and rods. In addition to these solutions, 4DMesh proposes two methods to flatten mesh edges and to produce printable files for off-the-shelf FDM 3D printers.

4DMESH OVERVIEW

4DMesh introduces two inverse design methods to produce morphing non-developable surfaces. These surfaces can be either a part of a sphere or a hyperboloid of revolution. The two methods utilize either shrinking or bending actuators that we designed and characterized. In our applications, we leverage the inherent advantages of non-developable geometries by probing ergonomic and aerodynamic designs and investigate practical uses of customization and labor-saving manufacturing (Figure 2). In particular, we quantify their structural strengths with both physical experiments and digital simulations. The inverse design pipeline of 4DMesh produces artifacts by meshing and flattening the input surface, then generate the respective printing pattern and G-code (Figure 3a) according to the function assignment of each element (Figure 3b). The artifacts fabricated with the G-code and an FDM printer have the configuration of a mesh and can morph into an approximation of the target surface.

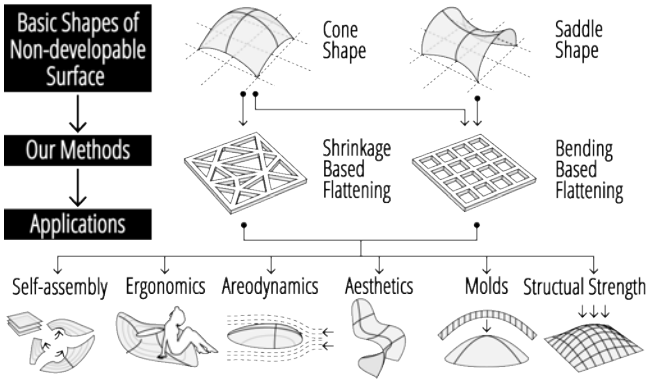

Figure 2. Design space of 4DMesh.

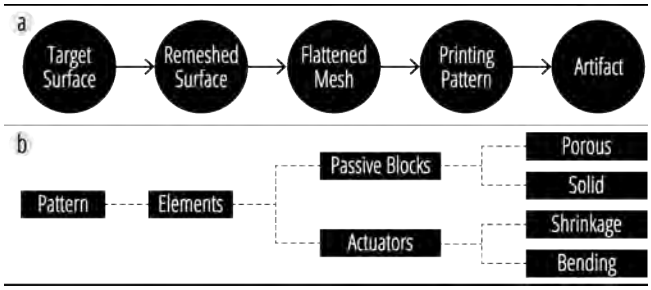

Figure 3. Workflow preview. (a) Flattening procedure; (b) Hierarchy of pattern and elements.

We use commercially available PLA filaments (polymaker PolyMax PLA) and an FDM 3D printer (Stacker S4) with a 0.8 mm extrusion nozzle that works within 520x320x625 mm cubic space. The large-diameter nozzle accelerates the prototyping process while achieving the resolution required to program materials. To print artifacts faster and to ensure the print quality, we choose 5000 (mm/min) as our printing speed and 0.2 mm as layer height base on the results from our experiments.

METHOD ONE: SHRINKAGE-BASED FLATTENING

Material Mechanism - Shrinkage Actuator

Shrinkage elements are designed as a combination of two solid passive blocks at both ends with an actuator block at the center. In a layer of the shrinkage element, the printing

path of the actuator block is parallel to the shrinking direction while the passive blocks have toolpaths that are perpendicular to it (Figure 4a). We vertically repeat the layers to achieve the desired height in the fabricated object. With this design, the overall shrinkage within an element can be controlled through modulation of the actuator length, or equivalently the actuator ratio such that longer the actuator, higher the shrinkage.

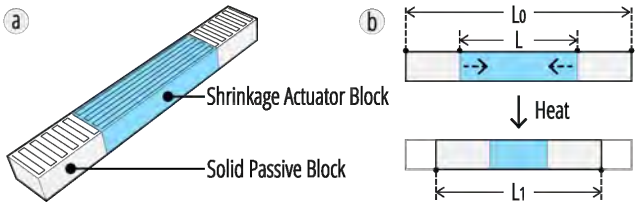

Figure 4. Shrinkage actuator design. (a) Block assignments and printing toolpaths; (b) Actuation.

To characterize the performance with different printing parameters, we conduct a quantitative analysis of the shrinkage actuators. Specifically, we investigate the shrinkage ratio with respect to layer thickness as shown in Figure 5. The shrinkage ratio here is the ratio between the final and initial lengths, where 1 represents no shrinkage and a smaller ratio indicates more shrinkage. The samples are printed 6 cm in length, 0.73 cm in width, and 0.4 cm in height. We observe that the shrinkage performance increases with decreasing layer thickness. In repetitive tests, we print these samples three times each and obtain consistent results on the same printer, but there may be differences depending on the machines used. In our work, we select the parameter with the high performance, *i.e.* small shrinkage ratio, to broaden the design space of achievable surfaces with the shrinkage mechanism.

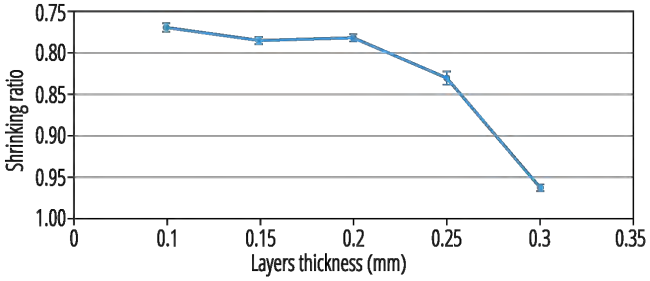

Figure 5. Plot of shrinkage versus actuator layer thickness.

Algorithm

This algorithm takes a target surface ***S*** as input and produces a flat pattern containing shrinkage actuators as G-code. The framework consists of four main steps: surface pre-processing, flattening, shrinkage actuator segmentation and G-code generation (Figure 6). In the process of meshing the surface, Quadric Edge Collapse Decimation (QECD) is used to achieve a reasonable mesh density such that the wireframe object can be printed with conventional 3D printers. For flattening, we use conformal mapping. This flattening approach mimics the shrinking process by distorting lengths while preserving angles. Although conformal mapping could result in both extension and shrinkage, we scale the flattened mesh to ensure that each element is extended when flattened and thus achieve the target shape, only shrinking when triggered. Next, we segment the mesh and place shrinkage actuators that satisfy the shrinkage requirement for each element. Finally, printing paths are created in the form of G-codes by connecting raster lines on the segmented pattern blocks and stacking them vertically.

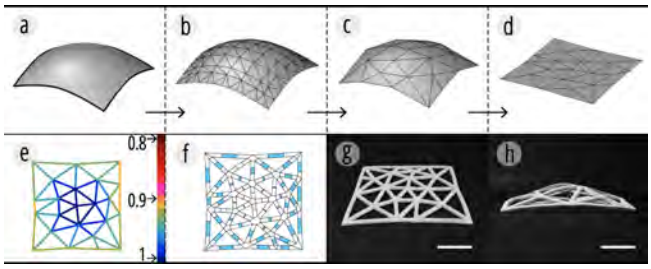

Figure 6. Shrinkage-based flattening pipeline overview. (a) Target surface *S*; (b) Meshed *S*; (c) Simplified mesh *M*; (d) flattened Pattern M_f; (e) length distortion; (f) actuator assignments; (g) fabricated artifact; (h) triggered artifact. (Scale bar: 5 cm)

Step 1: Pre-processing Input Surface

Given ***S*** that is represented as a geometric model, we triangulate it as a mesh ***M = (V, F)*** where vertex ***v*** is in the set ***V***, a face ***(v1, v2, v3)*** is a triangle with 3 vertices in the face set F. Since we fabricate wireframe objects, it is important to create a mesh where patterns on each edge can be printed with conventional 3D printers. For this reason, we coarsen the mesh using Quadric Edge Collapse Decimation.

Step 2: Flattening

In this work, an important consideration is to find a flat configuration M_f' of the input mesh ***M*** that allows the realization of the desired 3D shape with shrinkage actuators. It is often impossible to flatten a non-developable surface without distortions, *i.e.* either angles or lengths should be altered. Conformal mapping provides an effective tool for flattening a given surface in an angle preserving manner and distorts the edge lengths.

We use Least Squares Conformal Maps [24] with the eigenvalue approach as presented in Spectral Conformal Parameterization (SCP) [33]. With these maps, length alterations could be both extensions and shrinkages. However, the latter is not achievable due to the material mechanism. Hence, all edges in M_f' should be longer than their counterparts in ***M***. For this reason, we scale up the initial flattened mesh M_f' to obtain the adopted flattened mesh, M_f. In this process, for each edge in M_f', we calculate the ratio of its flattened length versus the original length. The global maximum ratio is used as the overall scaling factor.

Step 3: Shrinkage Actuator Segmentation

In this step, all edges in the flattened mesh are segmented to achieve the desired shrinkage ratio. In addition to the passive and actuator blocks, we introduce joint blocks as shown in

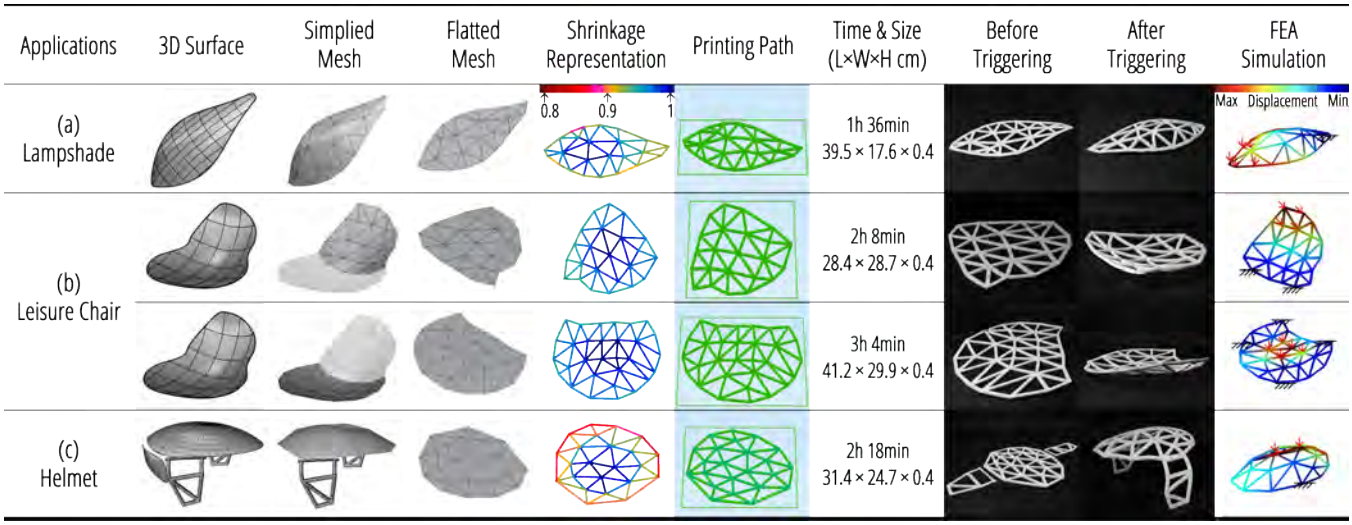

Figure 7. Shrinkage-based method applications.

Figure 8b. Joint blocks are placed at vertices where more than two edges converge. Actuators, joints and passive blocks are all represented as $\boldsymbol{B} = \boldsymbol{(b, flag)}$, where in each block $\boldsymbol{B}$, $\boldsymbol{b}$ contains all the vertices in this block in the counterclockwise order, and ***flag*** labels the function of the block as actuator, joint, or solid passive unit. Shrinkage ratio of an edge is controlled by modulating the actuator length. Length of actuators on each edge can be calculated as:

$$l = \frac{l_0 - l_1}{1 - \alpha}$$

where $\boldsymbol{l}$ is the length of the actuator to be printed (before morphing); $\boldsymbol{l_0}$ is the length of the edge in mesh $\boldsymbol{M_f}$ and $\boldsymbol{l_1}$ is the length of the corresponding edge in mesh $\boldsymbol{M}$ (Figure 6c, d); α is the shrinkage ratio determined by the printing parameters as explained in the Material Mechanism Section. Each edge is also given a width $\boldsymbol{W}$. In our examples, we use a constant width for all edges of the mesh. Note that the width parameter does not affect the shrinking performance significantly as long as it is reasonably set. Our design tool can segment a mesh automatically to meet print size limits. However, manual intervention may be required for complicated segmentation, such as in the helmet example, in which different pieces use different folding actuators.

Step 4: Printing path and G-code generation

We generate toolpaths as G-codes by rasterizing each segmented block in the flat pattern. To enable out-of-plane rotations during transformation, we print joint blocks as porous units with 50% infill rate. Adjacent blocks are fused together by slightly overlapping toolpaths at the interface.

Limited by the size of printer work area, to manufacture large surfaces, we divide the mesh into smaller pieces that fit the print bed when flattened. Utilizing the intrinsic feature of conformal mapping, contiguous meshes will have identical interface outlines after transformation. Hence, we can simply split the connection elements by the center line (Figure 8a) and join them together after trigger.

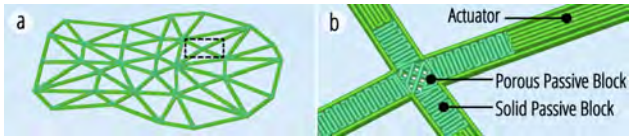

Figure 8. (a) flattened pattern of bar stool; (b) Close up of the area selected in (a).

Application Examples

Contemporary furniture is mass produced, fabricated following an invariant instruction set. We envision our method to enable and democratize modular, personalized, and custom-made designs. By digitally sketching a surface and processing it with our algorithm, the artifacts can be manufactured easily and rapidly with an FDM 3D printer (Figure 7). Fabricated and transported as flat sheets, our method also saves molding and shipping expenses.

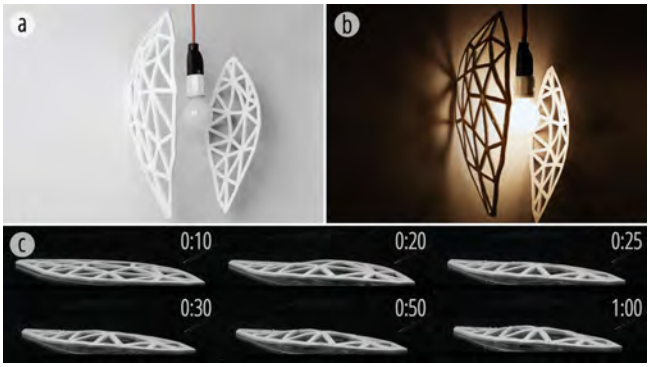

Figure 9. Lampshade (a) without light, (b) with lights on, and (c) triggered transformation over time.

Lampshade

The sparse structure generated by our method made it ideal for designing lampshades. Using our tool, we design an organic and leaf-like shape. By modulating the mesh density, the artifact render different atmospheres into the environment. These objects are triggered in 175°F hot water for 2.5 minutes.

Leisure Chair

Being larger than the printer work area, this design was divided into seat and back support for generating print files. With the division, the overall shrinkage rate was also reduced in both pieces. The connection edges matched well after triggering and was joined together using epoxy and left to cure for 24 hours. Using PLA for production, defects on the artifact can be easily repaired with doodle pens without replacement. The structure can soundly withstand the weight of an adult seated and leaned on the back.

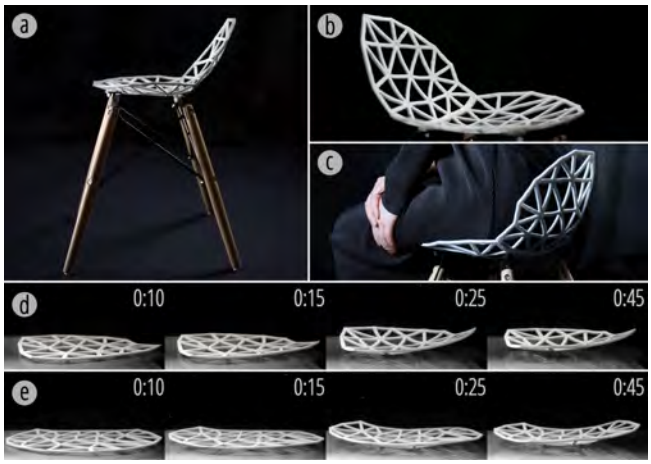

Figure 10. (a, b, c) The triggered artifacts and serial transformation of (d) back support and (e) seat.

Helmet

Using both of our flattening methods, we designed an aerodynamic helmet. The shell was generated with shrinkage-based method and the face shields with bending-based flattening. The design was simultaneously streamline and permeable.

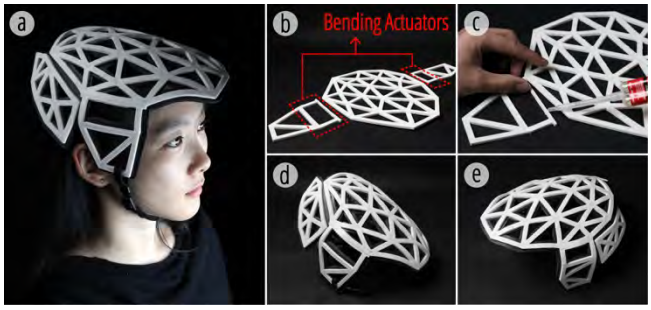

Figure 11 Aerodynamic helmet. (a, d, e) The triggered and assembled helmet; (b) The printed flat pieces; (c) The pre-assembling process.

METHOD TWO: BENDING-BASED FLATTENING

Material Mechanism - Bending Actuator

The design of bending elements is identical to the design of shrinkage elements, where the actuator block is placed in between two passive blocks. Yet, in the case of bending elements, we substitute certain layers of the actuator, either on top or at the bottom, with perpendicular constraint blocks creating a bi-component structure (Figure 12). This bi-component structure causes differential length changes between layers and thus results in arc-like bending of the elements. Like the shrinkage elements, we control the overall bending of each element by modulating the length of the bi-component structure such that longer it is, the larger the bending angle.

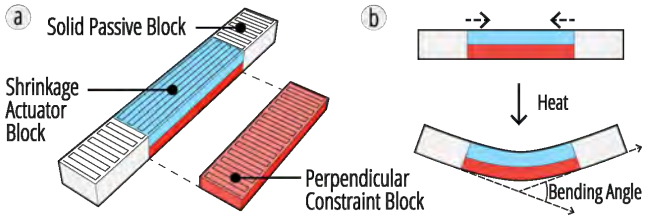

Figure 12. Bending actuator design. (a) Block assignments and printing toolpaths; (b) Actuation.

We conduct a quantitative analysis for bending actuators to characterize the relationship between printing/design parameters and the resulting bending performances. The samples are printed with the same dimensions as those used for the shrinkage actuators. As shown in Figure 13, we examine the correlation between the number of actuator layers and the resulting performance, which is measured in bending angles per unit length of the bi-component structure. Here, bending angle of an element refers to the angular difference between the tangent vectors at its two ends on the plane of the arc. Expectedly, as the number of actuator layers increases, bending performance improves. For a given surface mesh, we identify the highest bending angle within the mesh and select the number of actuator layers that can accommodate this value. Keeping the number of actuator layers constant for the mesh, we adjust the length of the bi-component structures to control the bending angle throughout the surface.

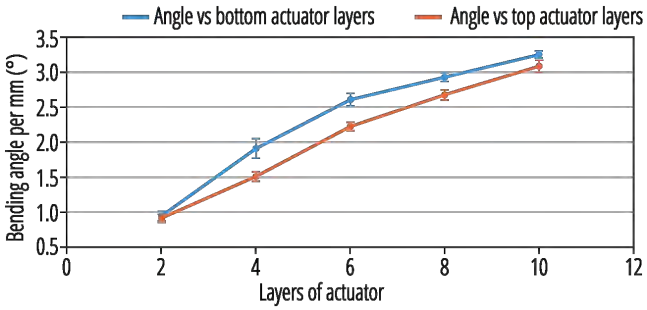

Figure 13. Plot of bending performance (angle/mm) versus actuator layers.

Algorithm

This algorithm takes a target surface $\boldsymbol{S}$ as input and produces a flat pattern containing bending actuators as G-code. The pipeline consists of four main steps: surface pre-processing, flattening, actuator segmentation, and G-code generation. We formulate this algorithm based on the Chebyshev Net Algorithm (CNA) [42] to convert and flatten NURBS surfaces into quad meshes. CNA is commonly adopted in models where torsion is absent and element lengths are

invariant. To use CNA as a mesh flattening algorithm, we modify it to preserve surface outlines and to accommodate for edges of different lengths. The first feature requires pre-processing of the surface and the second requires CNA to take variable edge length as input when applied on plane.

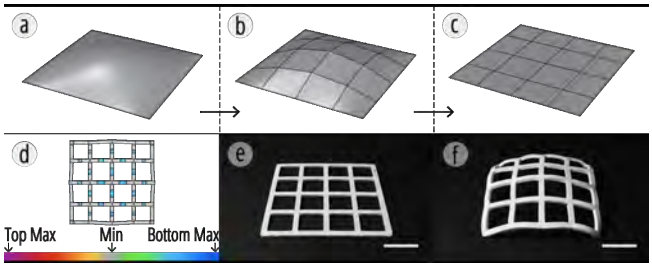

Figure 14. Bending-based flattening pipeline overview. (a) Target surface S; (b) CNA meshed M; (c) Flattened mesh M_f; (d) Actuator assignments; (e) Fabricated artifact; (f) Triggered artifact. (Scale bar: 5 cm)

Step 1: Pre-processing Input Surface

The CNA produces an approximation mesh that is smaller or equal to the original input S in size, whereas in product design the consistency of geometry outline is desired. To tackle this, we preprocess S by extending the four edges of the surface to derive intermediate extended surface S' (Figure 15a). We then subject S' to CNA to generate M' that fully covers the area of S. For edges in M' and their corresponding two ends, we process them with following rules to replicate and approximate the outlines of S:

- If both ends of the edge lands on the surface, both vertices are preserved without modification.
- If either end of the edge lands outside of the surface, the vertex is moved along the edge to the position closest to the surface outline.
- If none of the ends lands on the surface, the vertices are deleted.

With the modified vertices, we regenerate edges of M. The result is a quad mesh of equidistant edges at the center with variable length edges around the outline (Figure 15b).

To better approximate the target surface S and to translate the edges into bending actuators, we replace the straight edges in M with arcs (Figure 15c). These arcs are defined by three points: start point P_{start} and end point P_{end} of the original edge, and an additional midpoint P_{mid}. P_{mid} can be derived by either (1) evaluating S with the UV midpoint of P_{start} and P_{end} if the surface has regular UV frame intervals, or by (2) projecting the midpoint of the straight edge onto S along the averaged normal vector at P_{start} and P_{end}. Concavity-convexity of each arc is solved by checking its circle center against S to identify which side it is on.

Step 2: Flattening

To obtain a flattened Mesh M_f, we apply a modified version of CNA on the XY plane that takes variable distances during edge generation. The varying lengths of edges correspond to the arc lengths on M. The result of this step is a flat quad mesh with non-equidistant edges (Figure 15d).

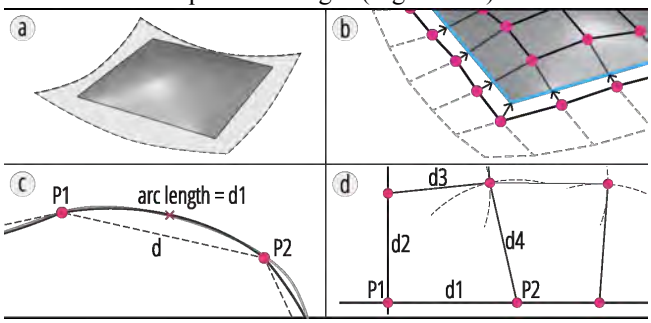

Figure 15. Modifications to Chebyshev Net Algorithm. (a) Surface extension, (b) edge trimming, (c) line-to-arc edge conversions (edge length changes from d to d1) in step 1; (d) Variable distance meshing in step 2.

Step 3: Bending Actuator segmentation

This step is similar to its equivalent in the shrinkage-based method, but the ***flag*** in $B = (b, flag)$ now indicates an actuator, constraint, solid or porous passive unit assignment. Some edges on M_f, are assigned as porous passive units to save printing time and material. We control the bending angle of each actuator by modulating its length. The length for each bending actuator is calculated as followed:

$$l = \frac{\theta_b}{\theta}$$

Where l is the required actuator length, θ_b indicates the overall bending angle for the edge, and θ is the bending angle per unit length of our actuator design.

Step 4: Printing Path and G-code Generation

In addition to the methods mentioned in shrinkage-based flattening method, we include a feature to control the concavity-convexity of the elements. We encode the bending directions by placing the constraint blocks either on the top or at the bottom. When the constraint blocks are placed on top, the local actuator bends forming a convex hull and vice versa for concave hulls.

Toolpaths for porous passive units are like the toolpaths for joints on shrinkage-based flattening. These blocks do not morph as bending actuators but ought to be as flexible as possible to absorb torsion and shearing, hence interfering less with the bending motions during global transformation.

To divide a large print into smaller pieces, we simply split the pattern into segments with flattened mesh edges (e.g., the element center lines), generate G-code, fabricate them separately, and join them together before triggering.

Application Examples

Fruit Plate and Chairs

Comparing to shrinkage-based flattening, this method allows for larger curvatures on the target and concurrent programming of concave and convex. We exploit this feature to design a fruit plate and two chair models (Figure 17).

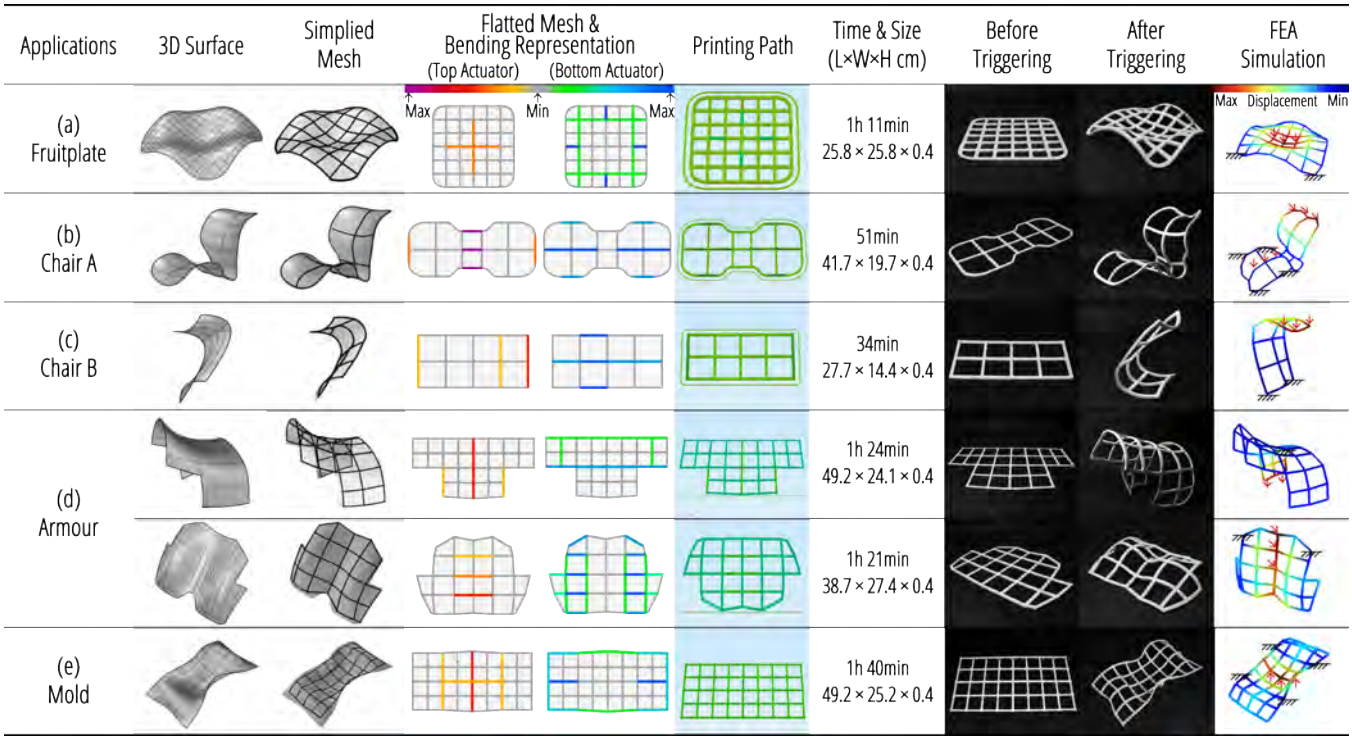

Figure 16. Bending-based method applications.

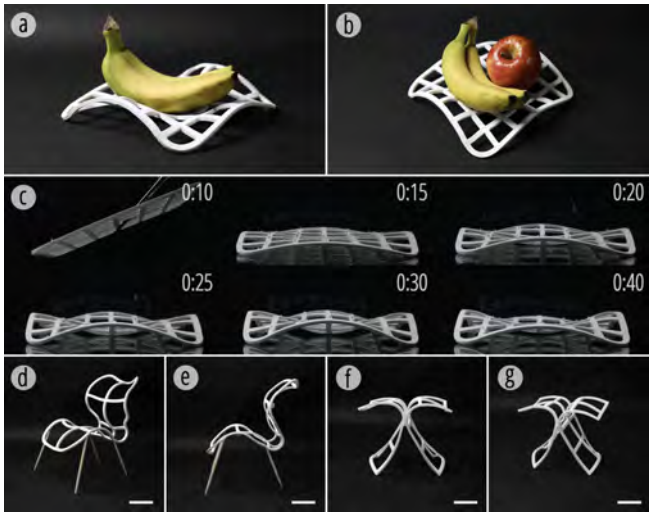

Figure 17. (a, b) Triggered fruit plate and (c) its sequential transformation over time; (d, e) A scaled chair model; (f, g) A chair made of two segments. (Scale bar: 5 cm)

Costume Design

Costume design usually requires certain degree of personalization. Body curvatures are non-developable by nature, and are difficult and uneconomic to replicate with conventional 3D printing approaches. Utilizing this method, we can design bespoke props that fit the stature of the wearer (Figure 18). Here we present an armor set made to fit with bending-based flattening.

Figure 18. (a, b) An armor set comprising two shoulder armors and one chest piece; (c) Sequential transformation of the shoulder armor.

Mesh for Composite Molding

Molding is a major challenge in producing large scale artifacts [48]. Researches have been conducted to produce reconfigurable formworks [3]. Our method can be used to fabricate the scaffolds for forming. Using the bending-based flattening method, we print the molds as flat sheets and trigger it before forming (Figure 19). This method scales better with size comparing to conventional CNC milled molds, where the volume and molding time rise cubically while our design increases in quadratic space, drastically diminishing the effort required for mold making and accelerating iterative design process. Here we experiment resin-fiberglass composite molding with 4DMesh.

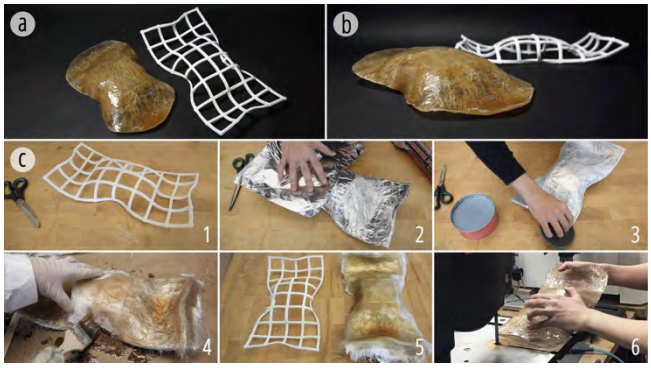

Figure 19. (a, b) Resin composite prototype; (c) Workflow of resin composite molding using 4DMesh.

DESIGN TOOL

We implemented our algorithms in Rhinoceros 6 with Grasshopper and Human UI. Users can calibrate the system to their own printer and thermoplastics following our actuator characterization tests, and set parameters including beam width, height, standard length, and scaling to process their input geometry with either flattening method. The system estimates the printing time and outputs G-code files.

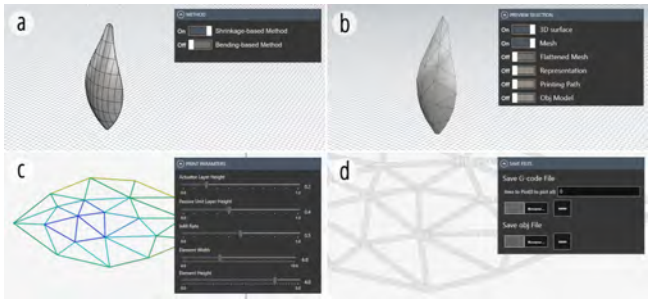

Figure 20. Interface of the design tool. (a) Input surface; (b) re-meshing options; (c) flattened mesh and shrinkage preview; (d) toolpath preview.

TRIGGERING METHOD

To trigger 4DMeshes, we submerge them into hot water. The medium is heated up to 175°F, which is much higher than 149°F, the glass transition temperature of PLA, to 1) minimize convection for a uniform trigger; 2) allow sufficient transformation time before the temperature drops to 140°F, the re-solidify temperature of PLA. The artifacts are submerged and stay still underwater until the temperature reaches 140°F. For quantitative experiments on the actuator performances and triggering large pieces, we added sugar into the water to increase its density to around 1.25 g/mm^3- the density of PLA- to compensate for the effect of gravity.

MECHANICAL PERFORMANCE VERIFICATION

Most applications we envisioned (chairs, helmets, etc.) and their respective contexts require specific structural strengths. We conducted both physical experiments and simulations to evaluate the mechanical performance of the structures.

To verify the structural integrity of triggered meshes, we conducted mechanical experiments on a set of six cone-shape samples including three different mesh resolutions generated by both methods (Figure 21b, c). Before actuation, these samples were 17 cm by 17 cm in size, 0.4 cm in thickness, and have 0.7 cm beam width. During the tests, each sample was fixed at four corners and subjected to a compressive load anchored at the surface center, with a metal pedal to distribute the force (Figure 21a). The compression load was progressively increased 5 lb. at a time until either the sample yielded, or the load exceeded 65 lb. Figure 21d shows the resulting vertical deformation of the samples with the applied vertical load. We observe that in this set, the maximum load they can hold before fracture vary from a minimum of around 23 lb. (Figure 21c-B1) to a maximum of 108.027 lb. (Figure 21b-S3). Fractures usually occurred at the contact interfaces of blocks, where the seams caused by FDM are structural weak points. Ruptures also occurred occasionally at porous joints. As expected, samples generated by both methods exhibit positive correlation between mesh resolution and structural strength. Qualitatively speaking, meshes generated by shrinkage-based method can withstand more load.

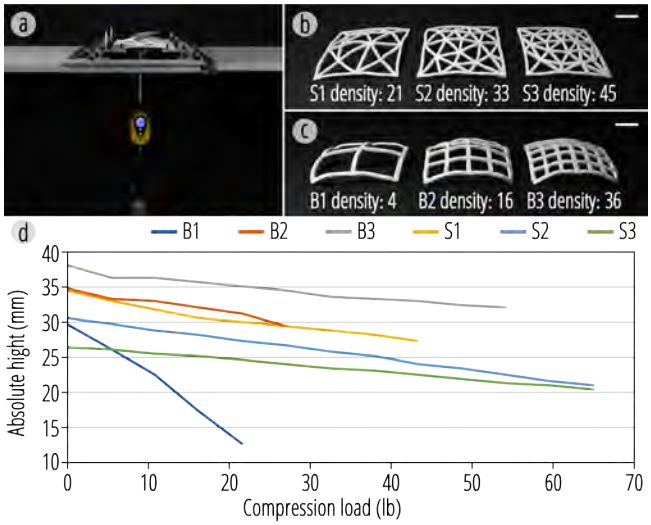

Figure 21. (a) Apparatus; (b) Shrinkage-based sample set and (c) bending-based sample set (scale bar: 5 cm); (d) Plot of sample height versus subjected load.

Structural Simulation

For structural simulations, we use space frame elements [27] in which edges are represented by one or multiple beams, namely finite elements. Each finite element is represented with two endpoints (position and angles) and cross section (width and height) information. Using simple beam elements, we represent our wireframe structures efficiently and enable fast simulations during the design process.

In the simulations, we used PLA material parameters provided by the manufacturer which were 3.5GPa for Young's Modulus and 0.36 for Poisson's ratio. We use standard PLA parameters as a conservative approach since the thermal treatment is shown to produce favorable compressive residual stress and improve mechanical performance in glassy polymers [17]. Note that the simulations presented here aim to support geometric design processes under given loads. These simulations ensure

mechanical soundness in real world use of the object. However, an accurate modeling of the thermal morphing process remains a challenge for future studies.

DISCUSSIONS AND LIMITATIONS

Comparison of Both Methods and Design Choices

Comparing two different flattening methods - shrinkage-based method (A) and bending-based method (B), pros and cons are exhibited in each method.

Surfaces

B can take on both developable and non-developable surfaces while A is limited to the former. Deploying conformal mapping on a developable surface is geometrically feasible but will not give shrinkage ratio locally. For A, while a conformal map can be found for any surface, the length distortions may fall beyond an achievable range. To solve this, we introduce segmentation to the surface to keep length distortions within achievable range. B can achieve larger curvatures but ignores torsion within elements and assumes that all joints are capable of free rotation. In reality, the deviations caused by these factors are absorbed by the plasticity of the material. Currently, B can achieve simultaneous concave and convex (e.g., a saddle shape), while A has a relatively arbitrary bending direction.

Maximum Curvature

The maximum *Principal Curvature* achievable with B is 0.022/mm. For A, the maximum curvature varies depending on the geometrical position: the surface center can be curved more than the edges due to curvature accumulation. In a cone, the center allows 0.010/mm maximum principal curvature. Our design tool can identify beams not processible with the maximum shrinkage ratio (78%) in Figure 5. For instance, the dramatic curves of the armor (Figure 18) require B, whereas A better preserves the outline of the chair in Figure 10.

Mesh Density

Our earlier experiment in Figure 20 shows that in A, increasing the mesh density will cause the accuracy to decrease, while B exhibits an opposite trend. Prior to testing, S1, S2, and S3 had displacements of 7.5, 11.5, and 15.5 mm, while B1, B2, and B3 had displacements of 12.2, 7.4, and 3.9 mm. For both methods, the higher the mesh density the more resembling the approximation is. B used arcs and gives a smoother surface but rebuilds the outline of the input. This method works well with surfaces of rectangular outlines and arbitrary curvatures. On contrary, A preserves the outline but requires a higher mesh density to approximate a surface well.

Printing Speed

A requires higher resolution of material programming and more time to manufacture. To produce a cone-shaped mesh of identical weights, A (Figure 21b-S2) and B (Figure 21c-B3) takes 166 and 137 minutes respectively.

Beam Design Parameters

We aim on functional objects for daily use and therefore based the beam thickness (4 mm) on typical plastic chairs (3-5 mm) and the width (7.3 mm) on the nozzle diameter (0.8 mm) and resulting print quality of our printer. While thinner beams minimize the size of joints (interfering less with transformation), the morphed artifacts cannot withstand as much load. Additionally, higher mesh density requires more printing time and more joints. Our tool provides print time estimates for users to balance between structural and fabrication efficiencies. Overall, our beam width, thickness and density suggestions are based on our actual tests and FEA simulation results and serve as good starting points.

Transformation Accuracy and Reproducibility

Our methods exhibit good reproducibility in general. Figure 22 shows the same shoulder armor printed three times. However, as the geometry scales up, the volume and the gravitational effect increase cubically, leading to controlling difficulty, transformational inaccuracy, and more mismatch between the actual transformation and simulation.

Figure 22. Shoulder armors printed thrice. (Scale bar: 6 cm)

Other Limitations

There are many technical challenges faced in this research, including the limit of geometry, the effect of gravity, and the transformation precision. Our current tools can only process surfaces that form no enclosure, have no periodic frame, and have no interior holes. To account for the precision of transformation and the effect of gravity, a rapid simulation tool is required. While high resolution finite element analysis does provide accurate results, the computational cost to compute large scale morphing structures is colossal. A fast simulation tool will also enable real-time iterative design with transformative materials. With increasing size, fabrication time may also become an issue.

CONCLUSION

In this paper, we present a pipeline to fabricate non-developable surfaces as 2D sheets across scales, using off-the-shelf 3D printers and materials. Compared to previous 4D printing studies, our method (1) uses only one material, (2) requires no manual assembly, and (3) produces structurally sound objects that can withstand reasonable loads. We hope 4DMesh enriches the design toolbox of material-driven and 4D printed structures, or in a larger scope, shape changing materials and interfaces.

ACKNOWLEDGEMENTS

This research was supported by the Carnegie Mellon University Manufacturing Futures Initiative, which was made possible by the Richard King Mellon Foundation. The authors thank Wayne Dudding for 3D Printing support, Professor Daniel Cardoso Llach and CMU School of Architecture for supporting the resin composite experiments.

REFERENCES

[1] Agrawal, H., Umapathi, U., Kovacs, R., Frohnhofen, J., Chen, H.-T., Mueller, S. and Baudisch, P. 2015. Protopiper. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15* (2015).

[2] An, B., Miyashita, S., Tolley, M.T., Aukes, D.M., Meeker, L., Demaine, E.D., Demaine, M.L., Wood, R.J. and Rus, D. 2014. An end-to-end approach to making self-folded 3D surface shapes by uniform heating. *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014).

[3] Bell, B., Read, T.C., Ede, A. and Barnes, N. 2014. Casting non-repetitive Geometries with Digitally Reconfigurable Surfaces. (2014).

[4] Byoungkwon An, Ye Tao, Jianzhe Gu, Tingyu Cheng, Xiang "Anthony" Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, Shigeo Takahashi, Hsiang-Yun Wu, Teng Zhang, Lining Yao Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. *Proceedings of the 36rd Annual ACM Conference on Human Factors in Computing Systems - CHI '18* (2018).

[5] Chen, X. 'anthony', Kim, J., Mankoff, J., Grossman, T., Coros, S. and Hudson, S.E. 2016. Reprise. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16* (2016).

[6] Cignoni, P., Pietroni, N., Malomo, L. and Scopigno, R. 2014. Field-aligned mesh joinery. *ACM transactions on graphics*. 33, 1 (2014), 1–12.

[7] Ding, Z., Weeger, O., Qi, H.J. and Dunn, M.L. 2018. 4D rods: 3D structures via programmable 1D composite rods. *Materials & design*. 137, (Jan. 2018), 256–265.

[8] Efrat, T.A., Mizrahi, M. and Zoran, A. 2016. The Hybrid Bricolage. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016).

[9] Felton, S., Tolley, M., Demaine, E., Rus, D. and Wood, R. 2014. Applied origami. A method for building self-folding machines. *Science*. 345, 6197 (Aug. 2014), 644–646.

[10] Follmer, S., Leithinger, D., Olwal, A., Cheng, N. and Ishii, H. 2012. Jamming user interfaces. *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12* (2012).

[11] Follmer, S., Leithinger, D., Olwal, A., Hogge, A. and Ishii, H. 2013. inFORM. *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13* (2013).

[12] Gannon, M., Grossman, T. and Fitzmaurice, G. 2016. ExoSkin. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016).

[13] Gladman, A.S., Matsumoto, E.A., Nuzzo, R.G., Mahadevan, L. and Lewis, J.A. 2016. Biomimetic 4D printing. *Nature materials*. 15, 4 (Apr. 2016), 413–418.

[14] Gooch, J.W. 2010. *Encyclopedic Dictionary of Polymers*. Springer Science & Business Media.

[15] Guseinov, R., Miguel, E. and Bickel, B. 2017. CurveUps. *ACM transactions on graphics*. 36, 4 (2017), 1–12.

[16] Han, D., Qi, X., Myhrvold, C., Wang, B., Dai, M., Jiang, S., Bates, M., Liu, Y., An, B., Zhang, F., Yan, H. and Yin, P. 2017. Single-stranded DNA and RNA origami. *Science*. 358, 6369 (Dec. 2017). DOI:https://doi.org/10.1126/science.aao2648.

[17] Hornberger, L.E. and Devries, K.L. 1987. The effects of residual stress on the mechanical properties of glassy polymers. *Polymer Engineering and Science*. 27, 19 (1987), 1473–1478.

[18] Hudson, S.E. 2014. Printing teddy bears. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014).

[19] Ion, A., Frohnhofen, J., Wall, L., Kovacs, R., Alistar, M., Lindsay, J., Lopes, P., Chen, H.-T. and Baudisch, P. 2016. Metamaterial Mechanisms. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16* (2016).

[20] Ion, A., Wall, L., Kovacs, R. and Baudisch, P. 2017. Digital Mechanical Metamaterials. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

[21] Kan, V., Vargo, E., Machover, N., Ishii, H., Pan, S., Chen, W. and Kakehi, Y. 2017. Organic Primitives. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

[22] Kovacs, R., Kommana, Y., Popiak, A., Bläsius, T., Baudisch, P., Seufert, A., Wall, L., Chen, H.-T., Meinel, F., Müller, W., You, S., Brehm, M. and Striebel, J. 2017. TrussFab. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

[23] Laput, G., Chen, X. 'anthony' and Harrison, C. 2015. 3D Printed Hair. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15* (2015).

[24] Lévy, B., Petitjean, S., Ray, N. and Maillot, J. 2002. Least squares conformal maps for automatic texture atlas generation. *Proceedings of the 29th annual conference on Computer graphics and interactive techniques - SIGGRAPH '02* (2002).

[25] Li, G.-L., Wang, C.-C., Kuo, C.-H. and Lin, Y.-C. 2006. Fast Flattening Algorithm for Non-developable 3D Surfaces. *MSV* (2006), 243–248.

[26] Liu, Y., Boyles, J.K., Genzer, J. and Dickey, M.D. 2012. Self-folding of polymer sheets using local light absorption. *Soft matter*. 8, 6 (2012), 1764–1769.

[27] Logan, D. 2011. *A First Course in the Finite Element Method, SI Version*. Cengage Learning.

[28] Lo, J. and Paulos, E. 2014. ShrinkyCircuits. *Proceedings of the 27th annual ACM symposium on*

User interface software and technology - UIST '14 (2014).

[29] van Manen, T., Janbaz, S. and Zadpoor, A.A. 2017. Programming 2D/3D shape-shifting with hobbyist 3D printers. *Materials horizons*. 4, 6 (Nov. 2017), 1064–1069.

[30] Momeni, F., Hassani.N, S.M.M., Liu, X. and Ni, J. 2017. A review of 4D printing. *Materials & design*. 122, (2017), 42–79.

[31] Mueller, S., Im, S., Gurevich, S., Teibrich, A., Pfisterer, L., Guimbretière, F. and Baudisch, P. 2014. WirePrint. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14* (2014).

[32] Mueller, S., Lopes, P. and Baudisch, P. 2012. Interactive construction. *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12* (2012).

[33] Mullen, P., Tong, Y., Alliez, P. and Desbrun, M. 2008. Spectral Conformal Parameterization. *Computer graphics forum: journal of the European Association for Computer Graphics*. 27, 5 (2008), 1487–1494.

[34] Nakagaki, K., Follmer, S. and Ishii, H. 2015. LineFORM. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15* (2015).

[35] Olberding, S., Ortega, S.S., Hildebrandt, K. and Steimle, J. 2015. Foldio. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15* (2015).

[36] Ou, J., Dublon, G., Cheng, C.-Y., Heibeck, F., Willis, K. and Ishii, H. 2016. Cilllia. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016).

[37] Ou, J., Skouras, M., Vlavianos, N., Heibeck, F., Cheng, C.-Y., Peters, J. and Ishii, H. 2016. aeroMorph - Heat-sealing Inflatable Shape-change Materials for Interaction Design. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16* (2016).

[38] Ou, J., Yao, L., Tauber, D., Steimle, J., Niiyama, R. and Ishii, H. 2013. jamSheets. *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction - TEI '14* (2013).

[39] Peng, H., Mankoff, J., Hudson, S.E. and McCann, J. 2015. A Layered Fabric 3D Printer for Soft Interactive Objects. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (2015).

[40] Peng, H., Wu, R., Marschner, S. and Guimbretière, F. 2016. On-The-Fly Print. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016).

[41] Pérez, J., Otaduy, M.A. and Thomaszewski, B. 2017. Computational design and automated fabrication of kirchhoff-plateau surfaces. *ACM transactions on graphics*. 36, 4 (2017), 1–12.

[42] Popov, E.V. 2002. Geometric approach to chebyshev net generation along an arbitrary surface represented by nurbs. *International Conference Graphicon* (2002).

[43] Qi, J. and Buechley, L. 2014. Sketching in circuits. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014).

[44] Raviv, D., Zhao, W., McKnelly, C., Papadopoulou, A., Kadambi, A., Shi, B., Hirsch, S., Dikovsky, D., Zyracki, M., Olguin, C., Raskar, R. and Tibbits, S. 2014. Active printed materials for complex self-evolving deformations. *Scientific reports*. 4, (Dec. 2014), 7422.

[45] Rivera, M.L., Moukperian, M., Ashbrook, D., Mankoff, J. and Hudson, S.E. 2017. Stretching the Bounds of 3D Printing with Embedded Textiles. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

[46] Sareen, H., Umapathi, U., Shin, P., Kakehi, Y., Ou, J., Ishii, H. and Maes, P. 2017. Printflatables. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

[47] Skouras, M., Thomaszewski, B., Kaufmann, P., Garg, A., Bickel, B., Grinspun, E. and Gross, M. 2014. Designing inflatable structures. *ACM transactions on graphics*. 33, 4 (2014), 1–10.

[48] Smits, J. 2016. Fiber-Reinforced Polymer Bridge Design in the Netherlands: Architectural Challenges toward Innovative, Sustainable, and Durable Bridges. *Proceedings of the Estonian Academy of Sciences: Engineering*. 2, 4 (Dec. 2016), 518–527.

[49] Tao, Y., Wang, G., Zhang, C., Lu, N., Zhang, X., Yao, C. and Ying, F. 2017. WeaveMesh. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

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

[51] Tolley, M.T., Felton, S.M., Miyashita, S., Aukes, D., Rus, D. and Wood, R.J. 2014. Self-folding origami: shape memory composites activated by uniform heating. *Smart Materials and Structures*. 23, 9 (2014), 094006.

[52] Wang, G., Cheng, T., Do, Y; Yang, H., Tao, Y., Gu, J., An, B. and Yao, L. Printed Paper Actuator: A Low-cost Reversible Actuation and Sensing Method for Shape Changing Interfaces. *Proceedings of the 36rd Annual ACM Conference on Human Factors in Computing Systems -CHI '18* (2018).

[53] Wang, G., Yao, L., Wang, W., Ou, J., Cheng, C.-Y. and Ishii, H. 2016. xPrint. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016).

[54] Wang, W., Yao, L., Zhang, T., Cheng, C.-Y., Levine, D. and Ishii, H. 2017. Transformative Appetite. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

[55] Weichel, C., Hardy, J., Alexander, J. and Gellersen, H. 2015. ReForm. *Proceedings of the 28th Annual ACM*

Symposium on User Interface Software & Technology - UIST '15 (2015).

[56] Weichel, C., Lau, M., Kim, D., Villar, N. and Gellersen, H.W. 2014. MixFab. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014).

[57] Wu, R., Peng, H., Guimbretière, F. and Marschner, S. 2016. Printing arbitrary meshes with a 5DOF wireframe printer. *ACM transactions on graphics*. 35, 4 (2016), 1–9.

[58] Yao, L., Niiyama, R., Ou, J., Follmer, S., Della Silva, C. and Ishii, H. 2013. PneUI. *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13* (2013).

[59] Yao, L., Ou, J., Cheng, C.-Y., Steiner, H., Wang, W., Wang, G. and Ishii, H. 2015. bioLogic. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (2015).

[60] Yue, Y.-T., Zhang, X., Yang, Y., Ren, G., Choi, Y.-K. and Wang, W. 2017. WireDraw. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17* (2017).

[61] Zhang, Y., Gao, W., Paredes, L. and Ramani, K. 2016. CardBoardiZer. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016).

[62] Zoran, A. and Paradiso, J.A. 2013. FreeD. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (2013).