EmbroForm: Digital Fabrication of Soft Freeform Objects with Machine Embroidered Pull-up Strings

Yu Jiang
Saarland University, Saarland
Informatics Campus
Saarbrücken, Germany
yjiang@cs.uni-saarland.de

Haonan Zhang
Saarland University
Saarland Informatics Campus
Saarbrücken, Germany
haonan.zhang@connect.ust.hk

Jürgen Steimle
Saarland University, Saarland
Informatics Campus
Saarbrücken, Germany
steimle@cs.uni-saarland.de

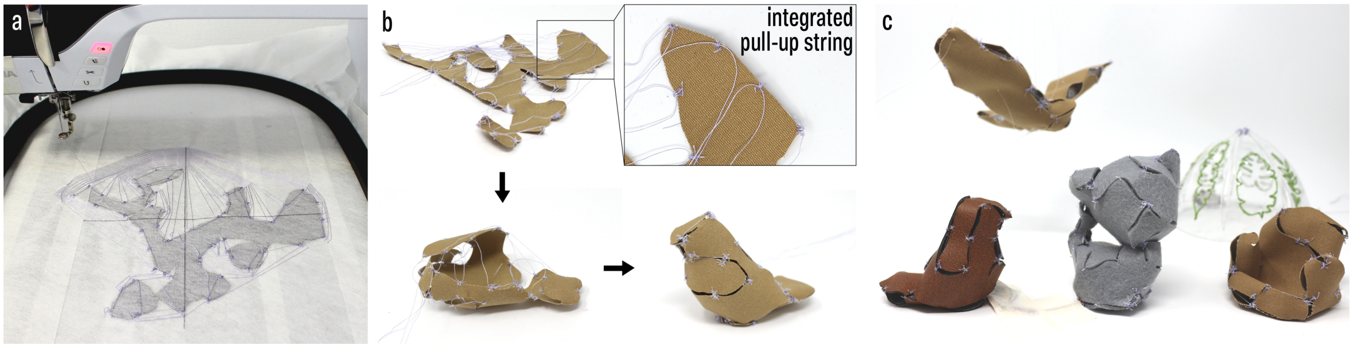


Figure 1: *EmbroForm* enables creating freeform pull-up objects that are soft, organically shaped, and digitally fabricated. (a) To achieve this, *EmbroForm* machine embroiders pull-up strings on flexible sheet materials with fabricable files generated from our pipeline. (b) The integrated string is pulled to wrap a 2D patch into the 3D target shape. (c) *EmbroForm* is applicable to different sheet materials and shapes.

Abstract

Pull-up objects form 3D shapes by pulling a string routed through a 2D material, offering low-cost 2D fabrication and reversible transformation. However, existing approaches rely on origamic folding, which creates faceted, oftentimes rigid surfaces and requires manual pull-up string routing. We introduce *EmbroForm*, a digital fabrication pipeline for fully soft pull-up objects with organic, higher-fidelity shapes. Instead of folding, *EmbroForm* forms 3D shapes by seaming the boundaries of a flexible 2D patch unwrapped from the target. To enable this, we contribute a fabrication technique that automates the routing of sliding strings on flexible sheet materials with machine embroidery, which we extend on to design zig-zag lacings for seaming the boundaries. Then we introduce an end-to-end pipeline that, given a 3D mesh, creates an optimized 2D unwrapped patch and generates pull-up string routing paths for fabrication. We provide a design tool for customization and validate our approach with technical experiments and implemented application cases.

CHI '26, Barcelona, Spain
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3790731

CCS Concepts

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

Keywords

Fabrication; Machine Embroidery; Shape-change; Computational Design.

ACM Reference Format:
Yu Jiang, Haonan Zhang, and Jürgen Steimle. 2026. EmbroForm: Digital Fabrication of Soft Freeform Objects with Machine Embroidered Pull-up Strings. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3772318.3790731

1 Introduction

Pull-up or flat-to-shape objects [23, 28, 40] use strings to actuate a 2D sheet material out of plane to form a desired 3D shape. To achieve this, the target 3D shape is unwrapped to a corresponding 2D pattern on which strings are anchored at selected locations. Pull-up objects are widely used in both research and commercial products to create customized objects [23], shape-changing effects [19], user interfaces [28], and large-scale furniture [40]. They offer unique benefits of fast and cheap 2D manufacturing, compactness during transport and storage, and simple and reversible actuation through manual pulling. Existing pull-up objects are achieved through *origamic folding* - the 2D material has softer creases sandwiched by more rigid panels that fold along the hinges when the string is pulled. Such folding-based pull-up objects [6, 28] approximate smooth 3D shapes with tessellated planar tiles, which result in faceted and oftentimes rigid surfaces.

To tackle these, we propose **a new type of fully soft pull-up objects with organic, higher-fidelity shapes that can be digitally fabricated**. This enables easy prototyping of custom deformable 3D objects that are soft and lightweight, which can be especially beneficial for wearable and intimate home contexts in which aesthetics and comfort during interaction are crucial. *EmbroForm* creates such soft freeform 3D objects by generating a flexible 2D pattern unwrapped from the 3D mesh (analogous to peeling an orange) and *seaming* the boundaries of the pattern. As the corresponding boundaries are seamed, the flexible 2D material naturally bends and morphs to recreate the 3D shape with continuously curved, organic surfaces.

The challenge in realizing this lies in the material's high deformability: different from rigid and self-supporting panels that only need to be joined at a few vertices (e.g., [28]), soft patches need to be securely joined along the boundaries to achieve shape integrity. This requires significantly more demanding pull-up string routing, whose design remains unknown and the existing manual fabrication would be labor-intensive and error-prone. To solve this, *EmbroForm* contributes a new fabrication technique that automates the routing of sliding strings on flexible sheet materials with machine embroidery and a novel algorithm that generates the unwrapped 2D pattern and pull-up string routing paths.

Our fabrication technique uses machine embroidery for automatic routing of sliding strings on flexible sheet materials. Originally used to stitch patterns on textiles, we repurposed the embroidery machine to digitally fabricate layered threads that effectively constitute tendon-like mechanisms on soft sheet materials. This includes *sliding strings* on the surface to be pulled for actuating the material and *tunnels* fixed onto the material to constrain the string's path. Based on these two principles, we design and present strategies to route zig-zag *lacings* for seaming highly irregular boundaries of the 2D patches, taking inspiration from laces in clothing, such as on corsets and shoes. We validate this technique and optimize the design parameters with characterizations of friction and material choices.

Based on this technique, *EmbroForm* further contributes an end-to-end pipeline that generates designs and machine files to digitally fabricate soft pull-up objects. We first use an existing algorithm by Zhao et al. [51] to decompose a 3D mesh into 2D developable patches. We then contribute a novel algorithm that generates the unwrapped 2D pattern by packing the decomposed developable patches in a layout optimized for embroidery, identifies points on the pattern boundary to be joined, generates the pull-up string routing, and finally transforms the designs into fabricable files for laser cutting and machine embroidery. With this, *EmbroForm* enables rapid prototyping of freeform objects that are (1) soft and deformable for pleasant and intimate user interactions, (2) organic and higher-fidelity in shape, (3) highly customizable with a digital fabrication workflow, and (4) made from a wide range of embroidery-compatible sheet materials for different physical properties.

We provide a design tool that provides an interactive front-end to designers and generates fabricable files, and we outline the fabrication steps for customizing soft freeform pull-up objects. We validate *EmbroForm* with a series of evaluations on materials, the effect of the number of boundary points to be merged, and different 3D meshes. Finally, we demonstrate *EmbroForm*'s capabilities with three implemented applications, including soft and reversible shape-changing furniture, an interactive toy with sensing capabilities, and a custom character for 3D animation.

In summary, we make the following contributions:

1. A novel automated fabrication approach for embedding sliding strings and lacings on flexible sheet materials with machine embroidery by programmatically creating layered threads.
2. A digital fabrication pipeline, along with a design tool, for creating soft pull-up freeform objects with machine-embroidered pull-up strings, by creating an optimized unwrapped 2D pattern for the target shape, generating the string routing paths, and converting the result into machine instructions for embroidery.
3. A series of technical evaluations that validate the fabrication technique and the soft pull-up shapes produced through the pipeline.
4. Three implemented example applications that demonstrate the practical feasibility of *EmbroForm*.

2 Related Work

We draw on existing works on creating 3D objects from 2D materials, 3D objects by joining 2D pieces, and machine embroidery:

2.1 3D Objects from 2D Material

Transforming 2D materials into 3D structures is becoming increasingly popular, as 2D fabrication is often more cost-effective, faster, and space-efficient than 3D fabrication methods. Researchers have applied different 2D-to-3D transformation principles to a wide range of 2D materials to realize versatile functionalities.

The most widely used principle is origami-inspired hinge-based folding, curving, and twisting. This is enabled by creating softer tunnels or grooves neighbored by more rigid panels in 2D, which can be pulled closer to fold along the hinges to form the final 3D shape. This transformation can be achieved by manual folding [21], sewing [12, 27, 37], internal material stress [18, 31], or external stimuli [3, 25, 41]. This has created a diverse set of functional interfaces, including interactive objects [29], thin-form actuators [42], circuits on complex 3D geometries [9, 44], body-fitting orthoses and furniture [43, 45], morphing food [39], and 3D textiles [6]. Among the folding-based 2D-to-3D transformations, pull-up objects [19, 23, 28] that are formed by pulling a string embedded in the 2D material have been popular. By pulling and loosening the string, these objects offer reversible and interactive shape changes that are appealing to users.

Despite their benefits, current folding-based 3D shapes and pull-up objects have constraints in shape fidelity and material stiffness. The complexity of a folded shape scales with the number of hinges. This leads to long fabrication times for more complex shapes (e.g., Demaine and Tachi [7] folded a Stanford Bunny in 10 hours), limiting current folded "3D" shapes to mostly 2.5D surface textures and simple geometries. Furthermore, folding-based approaches approximate a 3D shape with tessellated planar panels instead of organic curvatures, compromising shape fidelity. In addition to the limitation in shape, pull-up objects currently rely on rigid or highly plastic panels (e.g., plywood) that only need to be joined at very few vertices via simple, manually routed strings to create sound 3D objects. Pulling up softer material, on the other hand, would leave gaps between the merging vertices as the material bends.

To enable rapid prototyping of soft freeform objects, we adopt another 2D-to-3D principle based on seaming external boundaries that is natural to soft, flexible sheet materials and can create higher-fidelity shapes.

2.2 3D Objects by Joining 2D Pieces

Moving beyond folding, creating 3D objects by joining smaller pieces of 2D materials is also emerging. To start with, lasercut pieces [1, 8, 34] and strips [38] are manually assembled or woven together to create shapes. Layers of cut fabric are stacked together to approximate a shape [30].

Another larger and evolving class of work from the graphics community puts forward a 2D-to-3D principle that is natural to flexible 2D materials [5, 17, 50, 51]. They decompose freeform 3D objects into piecewise developable, curved surfaces that can be flattened onto a plane without stretching, compressing, or tearing [13]. The decomposed developable patches are then bent and joined at the corresponding boundaries to form the final shape, analogous to how fabric patches are "seamed" at the boundaries to create clothing. These algorithms can recreate high-fidelity, continuously curved 3D shapes with a small number of soft, flexible patches. However, as the graphics community focuses on advancing the decomposing algorithm to improve the accuracy of the recreated shape, there is currently no automated fabrication method for such recreation. Seaming the boundaries of flexible 2D materials to create soft 3D objects has been done manually by taping together paper [17, 50], sewing the boundaries of fabrics [49], and sewing zippers onto textiles [36], which is time-consuming and/or error-prone.

Thus, to enable lower-effort prototyping of 3D shapes that are soft, organic, and higher-fidelity, *EmbroForm* takes inspiration from pull-up objects and shoelaces to design pull-up lacing mechanisms that can effectively join boundaries of sheet materials and can be digitally fabricated with embroidery machines. Based on the designed lacing mechanism, *EmbroForm* contributes an end-to-end digital fabrication pipeline and algorithm that prepares an optimized 2D pattern to be seamed, and generates the lacing mechanisms needed for a smooth pull-up.

2.3 Machine Embroidery

Fabrication of soft objects has gained increasing attention in HCI, as such deformable objects allow richer and more intimate user interactions. Among these, textile-based ones have the unique benefits of being familiar to users. To this end, existing works have extended textile crafting techniques, such as cutting [30], felting [15], knitting [11, 14], smocking [37], and weaving [46] to create hand-sized 3D shapes. The fabricated soft objects, however, have limited shape fidelity and require redesigning the fabrication machine or manual fabrication. *EmbroForm* explores using off-the-shelf embroidery machines to create higher-fidelity 3D textile objects.

Machine embroidery is a digital fabrication method for creating planar patterns on textiles. It creates programmed stitches on the textile surface. Such embroidery machines have gradually made their way into everyday households thanks to the development of affordable commercial products[1], the straightforward and software-supported fabrication process, and the machine's adaptability to a wide range of sheet materials and threads.

Traditionally used to create aesthetic 2D patterns on clothing by infilling areas with colored threads, research has expanded the functionality of embroidered surfaces. In improving aesthetics, Zhenyuan et al. [52] embroidered controlled long thread segments to visually convey direction. Many works have machine-embroidered textile-based user interfaces that are soft and flexible in materiality and can seamlessly blend into clothing or our home environment. Examples include textile sensors to detect body postures [26], on-skin or on-textile touch and gesture [2, 47], pressure [32], and conductive objects [10]. Embroidered thin-form speakers have also been shown [33].

Some works also explored extending planar embroidered textile surfaces to higher dimensions. These include embroidering indents and bumps to represent UI elements such as icons, buttons, and sliders tactually [24, 35, 48], and shape-changing surface topologies folded by stretchy fabric's internal stress [18]. Closest to our work, OriStitch [6] pioneered machine embroidery with heat-shrinking threads to fold away excessive material between tessellated faces to create 3D shapes. Yet the folded objects have limitations in shape - they are faceted and have limited complexity due to the resolution of the facet, thus showing only semi-spherical prototypes, the transformation is not reversible, and they add substantial rigidity to the material due to the threads embroidered directly on the faces.

Inspired by pioneering works [4, 22] that automatically embed actuation strings into objects during 3D printing, we extend machine embroidery's capabilities and repurpose it as an automated fabrication method for routing sliding strings and lacings on textile-like flexible sheet materials to create soft and organic objects that can be pulled up.

3 Machine Embroidering Pull-up Strings

In this section, we introduce a novel fabrication method based on machine embroidery for integrating pull-up strings in flexible sheet materials to enable making soft and organic pull-up objects.

To create a pull-up object, a string should go through or be bound to the material at selected locations such that pulling the string transforms the 2D patch into a 3D geometry by joining the outer boundaries of the 2D patch. In prior work, this is done by manually routing a string through holes in the 2D material [23, 40]. We automate pull-up string routing on flexible sheet materials by repurposing a digital embroidery machine to lay out layers of functional threads on a 2D surface. Specifically, the layered threads include *sliding strings* that can slide on the surface when pulled, and

[1] https://www.bernina.com/en-US/Home-United-States

tunnels that anchor the string to the surface at selected locations, defining its path on the surface.

We start by detailing the design and fabrication principles of sliding strings and tunnels in Sections 3.1 and 3.2. Next, we extend on these basic building blocks to contribute a design of *lacings* capable of securely joining boundaries of flexible sheet materials. Contrary to previous folding-based pull-up objects that only need to join vertices of straight boundaries that are oftentimes stiff, pulling up organic shapes from flexible materials requires joining soft and highly curved boundaries. To achieve this, we take inspiration from laces in clothing (e.g., on shoes and corsets) that pull two fabric pieces closer. Similarly, we join the soft, curved boundaries with lacings that travel in a zig-zag pattern. In Section 3.3, we detail the design of the lacing and present strategies to machine embroider the lacing in 2D such that it can be smoothly pulled in 3D without knots or entanglements to join the boundaries.

3.1 Sliding Strings

The challenge in automating the routing of pull-up strings lies in that embroidery machines normally create stitches that are fixed in place. As the schematics of embroidery in Figure 2a shows, during each stitch made by the machine, the top thread above the fabric and the bottom bobbin thread below the fabric are intertwined to create a secure stitch. In traditional embroidery, such stitches and the interlocking of threads are typically made very dense (i.e., around 0.4mm - 3mm between each stitch) for creating filled patterns and to avoid threads unraveling during everyday wear and tear. In contrast, in our case, a pull-up string needs to be floating on the surface and remain free to move. To make this possible, we take advantage of machine embroidery's double-thread structure. To route a sliding string with the machine, we instead program very sparse stitches with long overhanging threads in between (i.e., several centimeters) to create the top thread as the string with a defined path that is only loosely held in-place by the bobbin thread at the stitch positions, as shown in Figure 2b.

The bobbin thread acts as a supporting structure during embroidery that can be simply cut one time anywhere and pulled out. Without the bobbin, the top thread becomes a string that entirely floats above the surface, not fixed to anything. Figure 2c shows this fabrication process. Alternatively, if highly curved string paths are needed, which require relatively dense stitches to redirect the top thread, one can use water-soluble bobbin thread[2] that can be removed more easily by washing.

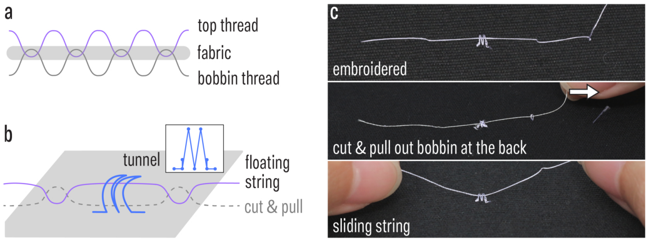


Figure 2: (a) Stitches made with machine embroidery intertwine the top and the bottom bobbin thread on the fabric. (b) We program these stitches to create sliding strings that float on the surface and fixed tunnels that define the string's path. (c) This is achieved by embroidering sparse string stitches and then the tunnels above, then removing the string bobbin thread to unravel the string stitches, making the string "slide".

3.2 Tunnels

To be able to actuate the sheet material, the created sliding string needs to be bound to the material. This is typically achieved in pull-up objects by manually traversing the string on both sides of the material. However, in machine embroidery, one thread can only travel on either the top or the bottom of the material. We therefore design tunnels that can bind the sliding string to the material surface, holding it in place but also allowing the sliding of the string through the tunnel. As shown in Figure 2b, the tunnels are made by two zig-zag stitches that go perpendicularly across the sliding string. Additional small tie-in and tie-off stitches (0.8mm apart) are added at the beginning and the end of a tunnel to fix it securely onto the material. During fabrication, the tunnel is embroidered after the string and before the supporting bobbin thread is removed. For the sliding string to smoothly travel through the tunnels, the friction needs to be minimized. This ideally can be realized by having loose stitches that create a hollow space under the tunnel; however, this cannot be achieved with typical embroidery machines, as the bobbin thread always pulls the top thread tightly. We thus control the width of the tunnel such that the sliding string have some space to move around with reduced friction under the tunnel. Through iterative prototyping, we empirically determined the width to be 3mm, which leaves sufficient space under the tunnel, but is still tight enough to avoid the string wiggling around. To guarantee a 0.8mm gap between stitches to avoid dense stitches damaging the material, the overall length of the tunnel (the longest four stitches) is 3.2mm. To reduce friction, we use silk top threads[3] (3-ply, 50 weight) for both tunnels and sliding strings.

3.3 Lacings for Joining Boundaries

To join potentially highly curved boundaries of soft sheet materials to create the pull-up objects, we contribute zig-zag lacings inspired by laces seen in real life. Figure 3 shows an example of a lacing between two boundaries to be joined. First, sets of merging points that need to be pulled closer to join are identified on the boundaries (labeled in Figure 3a). We take the points with high curvature on either boundary and their matching points on the other boundary, as such points make sharp corners that stick out if not tied to another boundary. Then additional points are added in between such that the maximum distance between any two points does not exceed a threshold. This guarantees a minimal density of points sampled on the boundary to facilitate tightly joining the boundaries. Each point set typically contains two points, but in some cases more than two when the point is shared by multiple boundaries and patches in the 3D mesh. The parameters for sampling the merging points can be

[2] https://www.quiltmania.de/Produktkategorien/Zubehoer-Co/Garne-Baender/Garne/Naehgarn-wasserloeslich/Vanish-Lite-Water-Soluble-Thread-Cone.html

[3] https://www.superiorthreads.com/thread/tiara/c/60-201

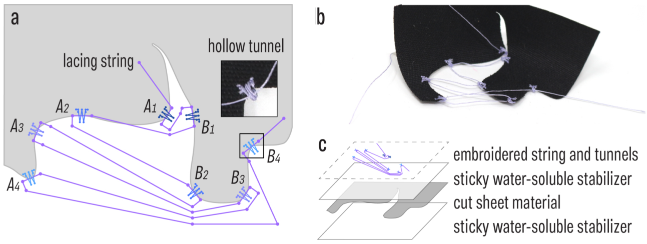

a
lacing string
hollow tunnel
A2
A1
A3
B1
B4
A4
B2
B3
b
c
embroidered string and tunnels
sticky water-soluble stabilizer
cut sheet material
sticky water-soluble stabilizer

Figure 3: (a) The lacing is embroidered to run zig-zag in hollow edge tunnels (b) to join boundaries when the string is pulled. (c) This is fabricated by sandwiching the cut sheet material with a sticky water-soluble stabilizer and embroidering the patterns on top.

programmtically defined and changed. Through empirically testing, we determine them to be points on boundaries with highest 10% curvature and a maximum distance of 2cm between neighboring points. The number of points sampled affects both the quality of the pulled-up shape and the ease of pulling up, which we evaluate in Section 6.3.

On these points, we embroider tunnels that extend side-ways beyond the boundary, as shown in Figure 3a. This creates entirely hollow tunnels and thus more volume for the sliding string to travel through to further reduce the friction. One zigzag lacing then passes through all the tunnels pair by pair, as shown in Figure 3a with highlighted stitch points. When pulled, the tunnels close up and join the boundaries; the neighboring points on the same boundary are not pulled together because the sheet material along the boundary resists pulling much more than the air gap between the boundaries. Figure 3b shows the fabricated prototype during pulling.

To fabricate the hollow tunnels, we use sticky water-soluble stabilizers[4] (i.e., tape-like sheets) to sandwich the material and embroider the tunnels, as shown in Figure 3c. The zigzag stitches of the tunnels run between the material and the stabilizer, with the tie-in and tie-off stitches on the material. After fabrication, the stabilizers are washed away to leave only the intertwined top and bobbin threads, creating a hollow tunnel extending side-ways beyond the boundary of the patch. The detailed parameters of the tunnel are characterized through technical tests, which we show in Section 6.2. Figure 4 shows example lacings for straight and curved boundaries that pull up cones.

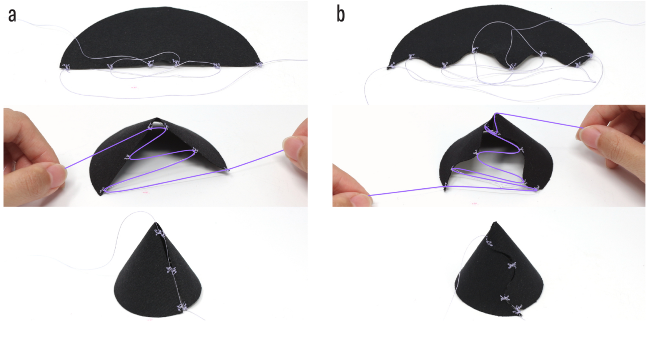


Figure 4: Using lacings to pull up a cone with (a) straight boundaries and (b) curved ones.

As shown in Figure 3a, routing the lacing in 2D with machine embroidery to enable a smooth pulling-up in 3D is non-trivial. The challenges in the routing are threefold: (1) the sliding string needs to be compatible with tunnels that will be embroidered afterwards, (2) it should route through all tunnels smoothly without creating loops or knots during pull-up, and (3) in contrast to straight boundaries in real-life laces, the lacing string travels between highly curved ones on which a naive zigzag creates entanglement. To solve these challenges, we present three steps and strategies below to route the lacing for an effective pull-up. Figure 5 shows a walk-through of the process with the same example, including skirting, connecting, and detouring, which each tackle one of the above challenges. All lacing stitches are colored in purple.

3.3.1 *Skirting the tunnel to effectively engage the lacing.*

The lacing string needs to be compatible with the tunnels, which will be embroidered after the lacing. For each tunnel, the lacing needs to travel through it exactly once. For one straight sliding string as shown in Figure 2, this is simple to achieve. However, when routing a zig-zag lacing on highly curved boundaries, the lacing might make sharp U-turns before or after going through a tunnel. Due to the resolution of the embroidery machine (~1mm) and the drifting of the sheet material, lacing stitches in these sharp U-turns embroidered too close to the tunnels can get unintentionally bound by the tunnel stitches. This could lead to the lacing making a U-turn inside and therefore escaping the tunnel, or the lacing getting bound by short and tight tie-in and tie-off stitches that create very high friction. To prevent this, we skirt each tunnel with some margin, as the bounding boxes in Figure 5a visualize, such that no lacing stitch can be placed inside the boxes. Through empirical testing, we determined this bounding box by adding a gap of 1.5mm to the border of the tunnel.

3.3.2 *Connecting the tunnels to avoid loops in the lacing.*

The tendon should travel smoothly without loops in the joined tunnels after pull-up, as shown in the inset in Figure 5b. This is determined by the direction from which the lacing enters a tunnel - if the lacing entered any of the tunnels from the other direction, a loop would be created. To achieve this, we use an important observation in 3D after pull-up to inform how to strategically connect the lacing string between the tunnels in 2D: the lacing enters all tunnels on one boundary from the same direction along the boundary, and enters all tunnels on the corresponding boundary from the other direction. This defines the lacing traversing order for all tunnels on a boundary. The midpoints of the tunnels are then connected accordingly, as shown in Figure 5b.

3.3.3 *Detouring to avoid intersecting lacings.*

After connecting all the tunnels with straight lacings while respecting the bounding boxes, the lacings can intersect, especially on highly curved boundaries, as shown in Figure 5b. During pulling up, the intersecting lacing segments risk entangling and blocking each other's movement, hindering a smooth pull-up. To solve this, we detour the intersecting lacing segments away from the patch, as shown in

[4] https://sulky.com/sulky-sticky-fabrisolvy-stabilizer-white-8-x-6-yd-roll

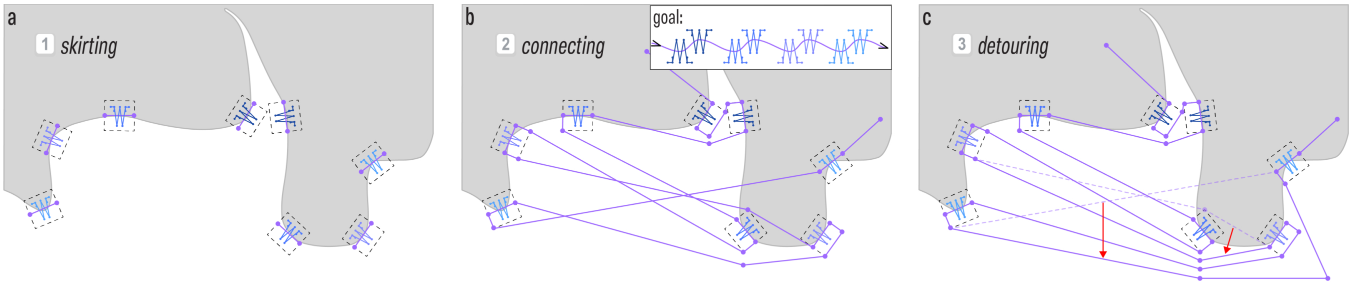


Figure 5: We present three strategies for 2D lacing routing to enable a smooth pull-up in 3D. We *skirt* each tunnel with a margin to ensure that the lacing string can freely slide, *connect* the lacing between the tunnels to avoid loops, and, where necessary, *detour* the lacing to avoid intersecting and entangling lacing segments.

Figure 5c. Algorithmically, this is achieved by scanning the connected lacing inside-out from the tunnel pairs closest to each other; if a scanned lacing segment intersects with a previous one, we detour its stitch points such that it becomes the outermost lacing segment. To realize this, we project the stitch points of the current outermost lacing onto the scanned lacing segment and find points along the projection direction 0.8mm away from the stitch points. These points, which are the current outermost points, become the stitch points of the scanned lacing segments. After the stabilizer is washed away and the detour stitches unravel, the lacing segments can be pulled tight to join the tunnels without entanglement.

4 *EmbroForm* Pipeline

To enable creating pull-up objects by joining flexible sheet materials, we contribute an algorithmic pipeline that prepares a given target 3D mesh for embroidery with our fabrication approach introduced in Section 3. In this section, we present the steps in this pipeline. With the separate 2D patches segmented from a 3D mesh, we pack the patches into one connected 2D patch optimized for embroidery that can be wrapped back to the target shape by seaming the boundaries (Section 4.1). Next, we identify sets of points on the boundaries that need to be joined (Section 4.2). Finally, we cluster these point sets into lacing areas that each can be closed up by one lacing and automatically generate the lacing routing (Section 4.3). The outcomes generated by the pipeline are converted into embroidery machine files for fabrication (Section 4.4). We make the code for the algorithm openly available[5].

4.1 Generating Unwrapped 2D Patch from 3D mesh

We first generate a 2D patch equivalent to the one obtained by unwrapping the input 3D mesh. To achieve this, we obtain a segmentation of the 3D mesh into separate 2D patches, then pack them together in an optimized connected layout.

We use a state-of-the-art evolutionary algorithm [51] to obtain the separate 2D patches. Figure 6a and b show an example of a bird mesh being segmented into seven separate 2D patches. For a mesh segmented into N patches $\mathcal{P} = \{P_i\}_{i=1}^{N}$, these patches share M boundaries $\mathcal{B} = \{B_i\}_{i=1}^{M}$ when wrapped up in 3D as shown in the highlighted blue lines in Figure 6a. Because each patch is made up of mesh triangles, each boundary B_i is composed of K_i short mesh edges $\{B_i\} = \{b_{ij}\}_{j=1}^{K_i}$.

Next, we pack the segmented 2D patches into one connected layout that can later reconstruct the target shape by seaming the boundaries. Unwrapping the 3D mesh to 2D is analogous to cutting along the shared boundaries to flatten the shape – only a subset of the shared boundaries $\mathcal{B}' \subseteq \mathcal{B}$ can be touching in 2D; and for the boundaries that still touch, they touch at a minimum of only one short edge instead of the entire boundary. We thus find an optimal unwrapping for embroidery-specific goals in two steps - we first decide the *topology* by finding $\mathcal{B}'$ that should touch in 2D, then fine-tune the *geometry* to determine the exact short edges $b^* = \{b_i^* \in B_i, \forall B_i \in \mathcal{B}'\}$ that these boundaries touch at. The packed 2D patch has one important constraint that there should be no overlap between the patches.

$$\text{Overlap}(P_i, P_j) = 0, \qquad \forall\, i \neq j,\ i, j \in \{1, \ldots, N\} \qquad \text{(C)}$$

To determine the *topology*, we model the connectivity of the 2D patches in the original 3D mesh with a graph in which a node denotes a patch and an edge denotes that the two patches share a boundary. An example of such a graph is shown in Figure 6c. The goal is to find a subset of the boundaries $\mathcal{B}' \subseteq \mathcal{B}$ with size $N-1$ that creates an acyclic connected layout (i.e., no holes) and to make the separate patches touch at long boundaries to avoid very long lacings traversing large distances in between. We first

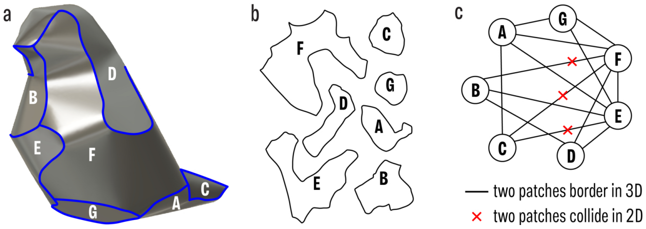


Figure 6: (a-b) We segment a mesh into flexible 2D patches with an existing algorithm [51]. (c) We then model the boundaries shared by the patches in 3D as a graph for generating a connected 2D patch.

[5] https://hci.cs.uni-saarland.de/projects/embroform/

reduce the search space by pruning edges that always create overlaps if touched – the two patches overlap no matter where on the boundary they touch. However, we can't prune all combinations of edges that create overlaps because it becomes computationally too expensive. We thus solve this by finding spanning trees on this graph that cover different combinations of the edges while prioritizing long boundaries. To satisfy this, we find 20 weighted random spanning trees [20], with the boundary length assigned as the weight of the edge. This gives us patch topologies that preserve long boundaries with high probability and avoid always keeping the same boundaries that are long but are potentially prone to collision. These patch topologies are taken as candidates for the geometry optimization.

Next, we optimize the *geometry* of the 2D patches based on the topology candidates. With a candidate subset of shared boundaries $\mathcal{B}'$, we now find the optimal touching short edges in the subset $b^* = \{b_i^* \in B_i, \forall B_i \in \mathcal{B}'\}$ to determine the packed layout. The geometry should satisfy Constraint C. To maximize the size of objects that can be fabricated, the geometry should also be optimized to fill a given embroidery hoop, which clamps the sheet material and thus defines the area that can be embroidered by the machine. The more compact the patch bounding box with a fixed aspect ratio r determined by the embroidery hoop, the more we can scale up the object. For a set of patches $\mathcal{P}$ connected by short edges b^*, we define its bounding box with fixed aspect ratio r as $\text{BBox}_r(\mathcal{P}, b^*)$. The goal is thus to minimize the bounding box size:

$$\arg\min_{b^*} size(\text{BBox}_r(b^*, \mathcal{P}))$$

We solve this by running simulated annealing (SA) (initial temperature 1, cooling rate 0.995, maximum iteration 5000) to choose the optimal b^*. Constraint C was implemented as a heavy penalty item in SA to avoid overlap. We then choose the best one from the candidates.

4.2 Generating Point Sets on the Patch Boundary

To close up sound 3D shapes with flexible 2D patches that have curved boundaries, the boundaries need to be merged at selected points. As Section 3.3 discussed, the selected points need to cover highly curved points and should be sufficient to effectively seal the boundary. We identify these points by screening the boundaries with a sliding window of 3 mesh units and computing the turning angle of the boundary segments. Figure 7 shows an example process on the packed 2D patch generated from Section 4.1. (a) We first take the segments with the largest 10% turning angle and sample their mid-points and the matching points on the corresponding boundaries as point set candidates. When a group of points is too close together on a boundary (colored in grey), which would lead to overlapping tunnels if fabricated, only the point with the highest curvature is preserved. When the points in a set is too close to each other, the set is removed. Point sets with three points are considered important points that should be joined for shape fidelity and are never removed. (b) With the remaining point sets, we add points into the gap between the points such that there is at least one point every 2cm along the boundaries.

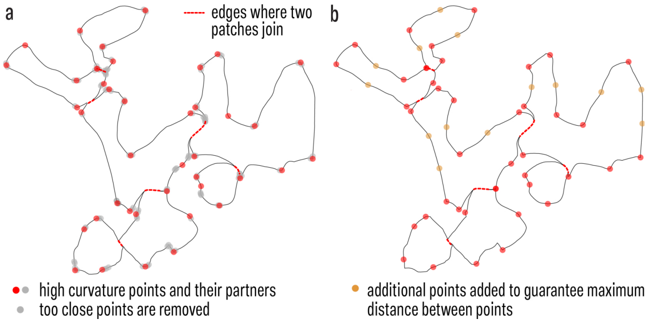

a
edges where two patches join
b
high curvature points and their partners
too close points are removed
additional points added to guarantee maximum distance between points

Figure 7: We identify sets of points to be merged by (a) finding high curvature points and their partners on the contour and (b) adding additional points to guarantee a maximum distance between adjacent points on the contour.

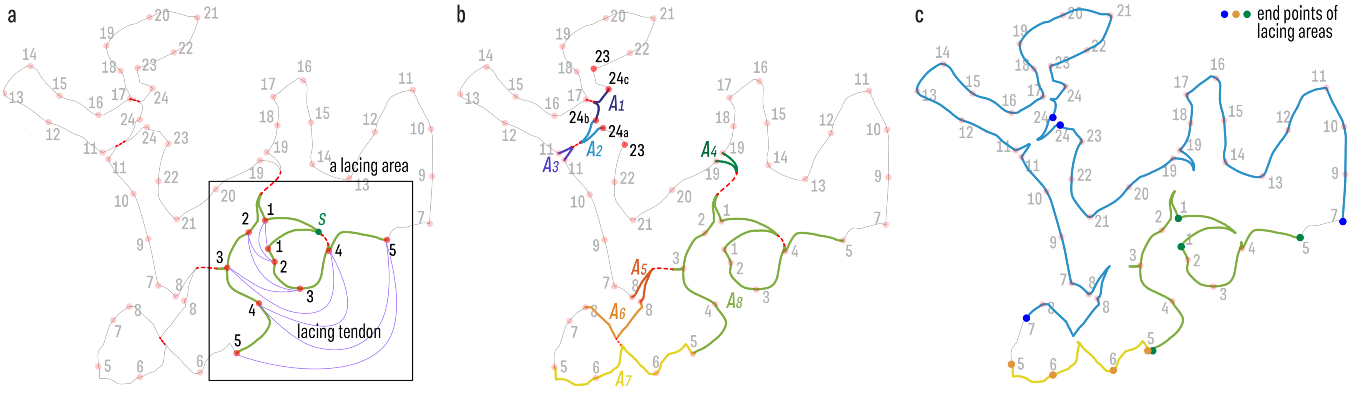
The sliding window size, the percentage of the highly curved points to keep, and the maximum distance between adjacent points were empirically iterated to effectively identify highly curved point sets while minimizing the number of sets to avoid friction from connecting a high number of points. These parameters can be changed based on applications and object sizes. An evaluation of the effect of varying numbers of point sets is carried out in Section 6.3.

4.3 Routing the Lacing

Finally, with the packed 2D patch and identified point sets to be joined, our algorithm automatically generates the lacing routing such that pulling one string effectively joins all the point sets to wrap up the 3D shape. To find a lacing routing that enables a smooth pull-up without loops or entanglements, we split the problem into manageable subproblems by clustering the point sets into *lacing areas* $\mathcal{A} = \{A_i\}_{i=1}^N$. In each lacing area, we can route the lacing as introduced in Section 3.3, which we can strategically embroider to avoid loops and entanglements.

We walk through our algorithm with the same example in Figure 8. From the previous steps, we obtain the packed 2D patch and the point sets as numbered in Figure 8a. We find lacing areas by tracing the patch contour from the edges where the separate patches join (highlighted in red dotted lines). This is analogous to zipping two matching boundaries that already touch in the middle - we zip towards both directions from the joining point to close the gap between the boundaries. Thus, starting from each endpoint of each joining edge, we trace along both directions of the contour to find consecutive matching pairs of points through which we can route a zig-zag lacing. An example is illustrated in Figure 8a. We trace from the edge endpoint S on both sides of the contour and find matching pairs of points numbered 1 to 5 which belong to one lacing area. The tracing stops after 5 because the next points on the two sides, 6 and 7, do not match. As a result, the initial tracing from the joining edges might produce disconnected lacing areas and many untraversed points, as shown in Figure 8b.

The mismatch can happen when we encounter a point set with more than two points (e.g., 5) and two or more gaps that need to be closed at the same endpoint (e.g., A_7 and A_8 should be closed and joined at point 5). After such a point set is joined, the tracing



trace from each patch joining point to identify placing areas

all lacing areas after initial tracing

iteratively merge smaller areas into big ones to traverse all points

Figure 8: (a) We trace from where the patches join (e.g., point S) to find consecutive matching sets of points that constitute a lacing area. (b) After initial tracing, the lacing areas A_1 to A_8 do not cover all merging points. We thus iteratively merge areas that share a set of merging points (e.g., A_1 can be merged into A_2) to continue tracing the boundary. (c) For this example, merging small areas with the number of point sets = 1 covers all points. We then determine how the endpoints of the areas are connected.

can continue with matching pairs of points (e.g., we can trace to 7 on two sides after 5). Therefore, in order to traverse the entire boundary to cluster all points into lacing areas, we perform local merge operations to merge smaller areas into a bigger one that shares a point set. For the merged areas, the lacing can make a detour to close the smaller lacing area before returning to the main area. We start by merging lacing areas whose number of point sets is $k = 1$. We show an example with A_1 and A_2, which both contain a pair of 24, in Figure 8b. Tracing from A_2 stops at 24_a and 24_b because the next points (23 and 24_c) do not match. To try to merge, we look at $2k - 1$ more points along the contour (24_c) - if they and the current end point (24_b) match, we merge them in the area and continue tracing. This allows merging A_1 into A_2, and the endpoints of A_1 are updated to 24_a and 24_c. The tracing can continue to 23, 22, etc., and A_3 and A_4 can be similarly merged into A_2 with $k = 1$.

Figure 8c shows the result after merging with $k = 1$. After merging, we check if all point pairs are assigned to lacing areas. For this example, this is satisfied with three lacing areas when $k = 1$. If not, we continue merging with an iteratively increasing k. Starting by merging smaller areas, which results in more lacing areas, is preferred because this avoids big lacing areas that require very long lacings to traverse. If merging creates overlapping lacing areas (i.e., some point sets can be assigned to either area), the overlapped point sets are assigned to the smaller lacing area for balance.

For each identified lacing area, how the lacing travels through the point sets within the area is determined by the lacing. We then determine how the different areas are connected by the lacing with a goal of minimizing its overall length. Specifically, the lacing can enter from one of the four endpoints $\mathcal{E}_i = \{e_i^1, e_i^2, e_i^3, e_i^4\}$ of a lacing area A_i, as highlighted in Figure 8c. The goal is to choose one entry point per lacing area $e^* = \{e_i^* \in \mathcal{E}_i\}_{i=1}^N$ and to determine the traversing order of the areas $\pi : \{1, \dots, n\} \to \{P_1, \dots, P_n\}$ that together minimizes total cost which is the lacing length connecting the lacing areas. Thus with a pairwise cost matrix $C_{i,j}[k, l]$ that gives the length of the lacing that travels between A_i with entry point k and A_j with entry point l, we run SA(initial temperature 1, cooling rate 0.99, maximum iteration 3000) to minimize:

$$\text{TotalCost}(\pi, e^*) = \sum_{i=1}^{N-1} C_{\pi(i),\pi(i+1)}\left[e^*_{\pi(i)}, e^*_{\pi(i+1)}\right]$$

After the point pair traversing order is determined, we identify the stitching pattern that can create the target routing design based on the three strategies presented in Section 3.3.

4.4 Machine File

The algorithm outputs the following files for fabrication. (1) A PDF file for laser cutting the patch contour from a sheet material and for engraving a rectangle with intersecting centerlines for embroidery calibration, (2) a SVG file that contains the generated lacing routing and tunnels. We process the SVG file into low-level embroidery machine code and export to machine-compatible EXP files using the pyembroidery library[6]. The lacing routing and tunnels are converted into corresponding stitches in two different colors. We detail the step-by-step fabrication process in Section 5.2.

5 Design Tool and Fabrication

To support customizing *EmbroForm* objects, we provide a dedicated design tool and detail the fabrication steps.

5.1 Design Tool

The visual design tool that takes in a 3D mesh from the user and prepares it for fabrication. As Figure 9 shows, the design tool shows the output of each step in the pipeline. Example steps, including (a) segmenting the imported mesh into separate 2D patches and (b) generating the lacing routing on a packed 2D patch with identified point pairs, are shown in the Figure. (c) Finally, the design

[6] https://pypi.org/project/pyembroidery/1.1.0/

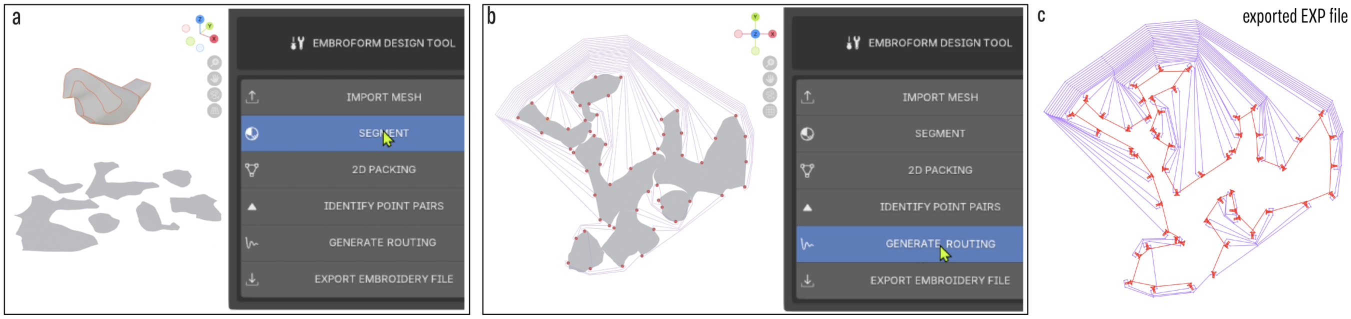


Figure 9: *EmbroForm* design tool for fabricating custom pull-up objects. Given an input 3D mesh, the design tool shows the results of each step in the pipeline. For example, (a) segmenting the mesh into 2D patches, (b) generating the lacing routing based on the packed 2D patch and identified merging points, and (c) exporting the machine file for embroidering the lacing (purple) and tunnels (red).

tool exports the necessary files for fabrication. We implemented the design tool as a plug-in in Blender.

5.2 Fabrication Steps

Figure 10 details the fabrication steps to create a pull-up soft 3D object with *EmbroForm*. The layout-optimized 2D patch in PDF generated by the algorithm is first lasercut with the chosen sheet material. Without moving or taking the cut material out of the machine, a layer of sticky water-soluble stabilizer is applied directly onto it. A grid for calibration is then engraved onto the stabilizer. To engrave on the thin stabilizer without cutting through the underlying sheet material, the laser cutter was set to 2% power, 100% speed, and 500Hz frequency. After engraving, the excess sheet material is peeled off the stabilizer, and another layer of sticky water-soluble stabilizer is put underneath the material. This sandwiches the cut 2D patch with stabilizers to secure it during embroidery. Using the embroidery file generated from the pipeline, the lacing and tunnels are then machine embroidered. A Bernina embroidery machine (model 790 PLUS) and a Jumbo embroidery hoop with an embroidery area of 260mm by 400mm[7] were used for fabrication. The machine was set to the lowest speed to reduce the drifting of the material. For the bird example, the total embroidery time needed was around 20 minutes. After embroidery, the support material, including the stabilizers and the lacing bobbin thread, is washed away. The 3D object can then be formed by pulling the lacing string.

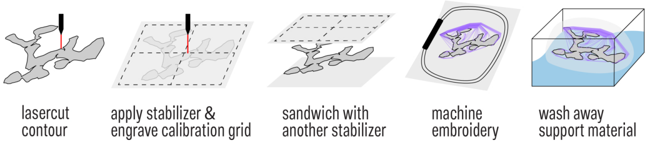

lasercut contour
apply stabilizer & engrave calibration grid
sandwich with another stabilizer
machine embroidery
wash away support material

Figure 10: The fabrication steps for creating *EmbroForm* prototypes that can be pulled up.

6 Technical Evaluation

We performed a series of tests to characterize and evaluate *EmbroForm*'s fabrication approach and pipeline. Specifically, we tested *EmbroForm* with five sheet materials, determined the parameters for the hollow tunnels through empirical tests, and evaluated fabricated *EmbroForm* prototypes with varying numbers of merging points. Finally, we evaluated the achievable shapes by running the algorithm on seven 3D meshes and show the generated designs.

6.1 *EmbroForm* with Different Materials

To be compatible with the *EmbroForm* pipeline, the used sheet materials need to be laser-cut safe, non-fraying such that the boundary hollow tunnels do not get torn away when pulled, and should be compatible with machine embroidery. We test five different types of sheet materials whose detailed specifications and fabricated prototypes are shown in Figure 11. These include a thin and light cotton fabric[8], a heavier and thicker dry waxed organic cotton[9], transparent TPU vinyl, faux leather, and polyester felt. We empirically measured the materials' bending rigidity with the cantilever (self-weight) test [16] on strips of 3cm by 5cm. The materials create sound 3D objects with thicker material, such as leather, creating stiffer objects.

6.2 Characterizing Hollow Tunnels

The hollow tunnels in our lacing design are crucial to a smooth pull-up. They should securely stay on the sheet material and should enable the lacing to travel through them with minimum friction. To fine-tune their fabrication parameters, we perform a detailed characterization of the hollow tunnels. We characterized the tunnels' width of the stitch off the patch (d_{out}) and on the patch (d_{in}), as shown in Figure 12 . We carried out the tests on all five sheet materials introduced above.

[7] https://www.bernina.com/en-US/Accessories-US/Embroidery-Accessories/Embroidery-Hoops-and-Adapters/Jumbo-Embroidery-Hoop

[8] https://swafing.de/luisa-beschichtet-buegeln-linke-warenseite-beschichtete-baumwolle/078435-000747

[9] https://stofffairliebt.de/produkt/dry-waxed-organic-cotton/

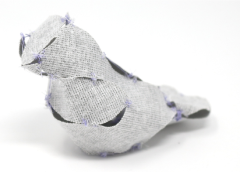
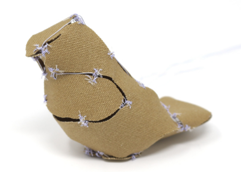
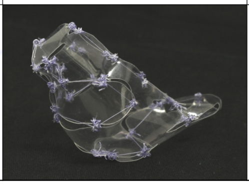
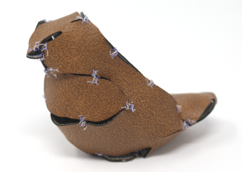
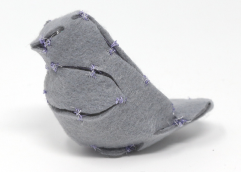
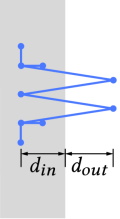
material	light acrylic-coated cotton	dry waxed organic cotton	transparent TPU vinyl	faux leather	polyester felt
thickness	0.2mm	0.3mm	0.2mm	1.5mm	1.4mm
weight	180g/m^2	227g/m^2	225g/m^2	426g/m^2	173g/m^2
bending rigidity	0.036mN•m	0.058mN•m	0.039mN•m	0.328mN•m	0.166mN•m
fabricated prototype					

Figure 11: Five flexible sheet materials tested with varying composition, thickness, weight, and bending rigidity.

For d_{out}, there is a trade-off between low friction and the boundary-merging quality. Pulling the lacing string joins the tunnels together. Tunnels with large d_{out} create large-diameter hollow tunnels which, when pulled together, still leave a gap between the material boundaries. We thus tested d_{out} from 0mm to 3mm with d_{in} set to 2mm to choose a value that balances the trade-off. To simulate the pull-up scenario with maximum friction, we closely laid out a high number (22) of tunnels on two boundaries to be joined such that the lacing between the boundaries turns with sharp angles (90 degrees) with high friction. We pulled the lacing string to join the boundaries, and the pulling forces were measured with the force gauge and shown in Figure 12. The pulling force significantly decreased for all materials when d_{out} is increased to 2mm, but stayed similar for 3mm. Therefore, to create more nicely closed boundaries, we determined d_{out} to be 2mm.

d_{in} should be minimized to reduce the width of the patch required to place a tunnel, such that it can still be put on very narrow patches. However, tunnels with too small d_{in} have tie-in and tie-off stitches embroidered very close to the boundary and can easily be torn away when the lacing string is pulled. We thus tested d_{in} from 1mm to 2mm with a fixed d_{out} of 3mm. We measured the force to tear by pulling a lacing that goes through the tunnel in the direction perpendicular to the boundary along the surface plane and recorded the force at which the tunnel was torn away from the material. A maximum of 20N pulling force was used, which is the force at which the silk lacing thread snaps. A force gauge (Baoshishan ZP-50N with 0.01 N accuracy) was used. As Figure 12 shows, none of the tunnels are torn away at $d_{in} = 2mm$ when 20N of force was applied, which is the final value we chose.



		pulling force against friction (N)				force to tear (N)	
		d_{out}				d_{in}	
		0mm	1mm	2mm	3mm	1mm	2mm
material	light fabric	2.43	1.11	0.36	0.38	1.94	>20
	heavy fabric	4.95	1.46	0.34	0.32	8.88	>20
	TPU vinyl	5.42	1.33	0.3	0.28	11.4	>20
	leather	4.66	1.11	0.33	0.39	>20	>20
	felt	7.43	5.85	0.93	0.85	11.36	>20

Figure 12: Characterization of parameters of the hollow tunnel. On the five materials, we characterize d_{out} to minimize the pulling force required against friction and d_{in} to make the tunnels tear-resistant.

6.3 Effect of Number of Merging Points

As mentioned in Section 3.3, the number of points to be merged is a parameter that can be tuned and programmatically controlled. We empirically evaluate the effect of the number of merging points with the bird example fabricated with different numbers of tunnels, as shown in Figure 13. The left one only has the 10% of points with the largest curvature. On top of this, the middle has a maximum distance of 2cm between points, and the right has a maximum distance of 1cm between points. This sampling criterion results in the prototypes having 40, 54, and 72 tunnels, respectively.

Increasing the number of points creates more precise shapes as the boundaries are better merged with denser points. As shown, higher numbers of merging points effectively seam the boundaries, while too sparse merging points result in gaps and fail to recreate details in the 3D mesh. We measured the error of the shape recreation. Compared to the input bird mesh, the shape error of the bird with piecewise developable patches generated by the segmentation algorithm [51] is 0.9%, which is the lowest possible shape error for the fabricated prototypes. We scanned the fabricated bird prototypes to compare with the input mesh. The scanning was performed with a Revpoint laser scanner with down to 0.05mm accuracy[10]. The error in the fabricated object was calculated by the two-sided Hausdorff distance normalized by the diagonal length of

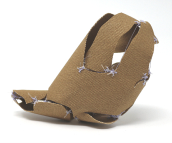
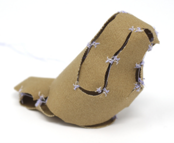
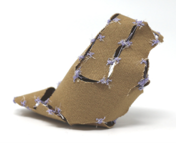
fabricated			
criteria	top 10% curvature points	top 10% curvature points max 2cm between points	top 10% curvature points max 1cm between points
number of points	40	54	72
shape error	11.2%	8.4%	5.9%
height	60.8mm	58.2mm	53.9mm
pull-up time	~2min	~4min	~8min

Figure 13: We fabricated the bird with different numbers of merging points and measured the shape accuracy, size, and pull-up time. Having more merging points lowers the shape error, reduces the achievable size, and increases the pull-up time.

[10] https://global.revopoint3d.com/en-us/products/infrared-laser-3d-scanner-inspire2

the bounding box, which measures the largest deviation between any point on one shape and the closest point on the other relative to the object size. The result shows that while all three prototypes have quite low errors and were able to reproduce the input mesh in high quality, the prototype with the largest number of merging points has the lowest error of only 5.9%.

Trade-off between accuracy and size. A higher number of merging points increases the shape accuracy, but limits the achievable size. As the number of tunnels increases, more space is needed for routing the lacing on the limited embroidery workspace. This forces the generated 2D patch and thus the size of the fabricated prototype to scale down. We report the height of the prototypes with different numbers of merging points to show this trade-off. For each prototype, we fabricated the largest possible size with the embroidery hoop that we have. As shown in Figure 13, the largest achievable height gradually scales down as the accuracy increases.

Trade-off between accuracy and pull-up effort. We found that as the accuracy improves with increasing numbers of merging points, it takes more effort and time to pull up the shape.

We evaluated the force required to pull up with the most notable factors (boundary curvature and the density of tunnels) to merge 2D boundaries of 14cm length. We varied the curvature from straight to 40mm, 80mm, and 120mm in radius and varied the number of tunnels from 6, 10, to 14. The tests showed that with the tunnel parameters optimized in Section 6.2, the pull-up forces were 0.3(±0.05)N for all tested parameters. This shows that for shorter and more regular boundaries, the pull-up can be done easily. In practical cases, more complex shapes like the bird have highly irregular and much longer boundaries, where the low-effort pull-up in the evaluation becomes only locally applicable at shorter and regular boundary segments. We empirically found out that more complex shapes need to be pulled up segment-by-segment to deal with the long string length and the high friction at places where the boundary is highly irregular (e.g., makes a sharp turn). Since the pull-up effort in such cases is highly dependent on the exact geometry and the continuous shape change throughout the pulling process, as well as how the person pulls, we empirically quantified it with the end-to-end overall duration to pull up the three bird prototypes. As shown in Figure 13, the least accurate prototype with the lowest number of merging points requires the shortest time to pull up, while the most accurate prototype takes longer to pull due to the additional length and complexity of the lacing routing.

The *EmbroForm* pipeline is developed with flexibility in mind to allow customization with respect to different application needs and available machinery. Thus, designers can make their own choices between accuracy, size, and pull-up effort depending on the context by varying the number of merging points. For the prototypes implemented in this paper, informed by the above findings, we chose the sampling criteria of the top 10% curvature points with a maximum of 2cm between points. For our prototypes, which are around 10cm in scale and the maximum that can be achieved with our embroidery setup, this is a nice middle ground that gives us high-quality shapes, hand-sized objects, and relatively easy pull-ups that can be done within 5 minutes for complex shapes. We speculate that for larger-scale objects, the criteria will be similar, as sparse merging points would leave the material unbent and create gaps, yet recommend to empirically find an optimal threshold for objects of a substantially different scale.

6.4 Evaluating the Shape Space

We successfully generated the unwrapped 2D patch and lacing routing path for seven 3D meshes. Other than the bird example shown in the paper, we show the other six in Figure 14. For each mesh, we show the segmented mesh, the packed unwrapped 2D patch with highlighted identified merging points, and the final

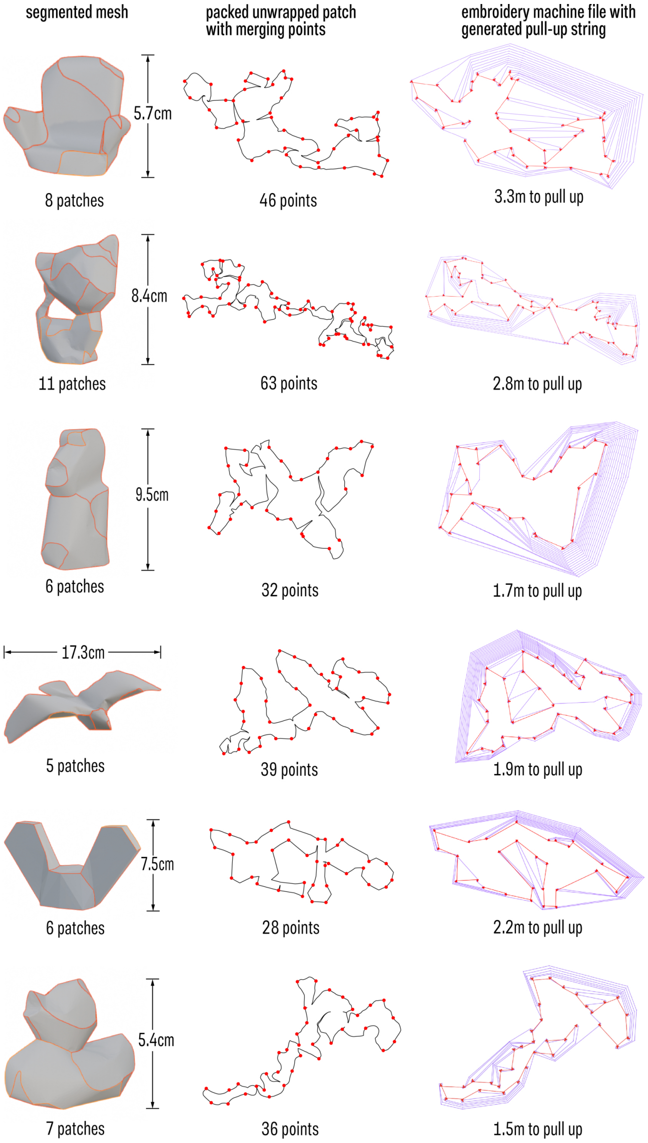


Figure 14: Segmented mesh, packed unwrapped 2D patch, identified merging points, and generated routing path for six 3D meshes. The maximum size that can be fabricated with our setup, the number of merging points, and the distance to pull up are labeled.

color-coded embroidery machine file including the generated lacing routing. We labeled the maximum dimension of the object that can be produced with our embroidery machine and hoop, the number of merging points identified, and the distance required to pull up the shape.

7 Applications

To further evaluate the practical feasibility and usefulness of *EmbroForm*'s, we designed and fabricated three example applications:

7.1 Blanket-lamp on Armrest

With a small number of tunnels and simple lacing routing designs, *EmbroForm* can achieve easily reversible pull-up for shape-changing objects that can be actuated by simple user interactions. We demonstrate this with a blanket on a sofa armrest that can be transformed into a dome-shaped lamp and can be reset effortlessly. As shown in Figure 15, we machine embroidered a petal-shaped blanket with transparent TPU vinyl with hollow tunnels at the end of each petal and a straight lacing that routes through all tunnels. We sew LED sequins for lighting the lamp and a small coin-cell battery for powering. (b) The blanket is made out of soft material, thus safe to press into or comfortably lean on. Custom aesthetic patterns can be embroidered on top, which can be easily integrated into *EmbroForm*'s fabrication process. (c-d) When the user wants a small, focused light for tasks like reading, the string can be pulled to transform the planar blanket into a dome-shaped lamp that stands on the armrest. (e) When finished using the lamp, it can be easily reverted by pushing out the petals, which then collapse under gravity to transform back to the blanket.

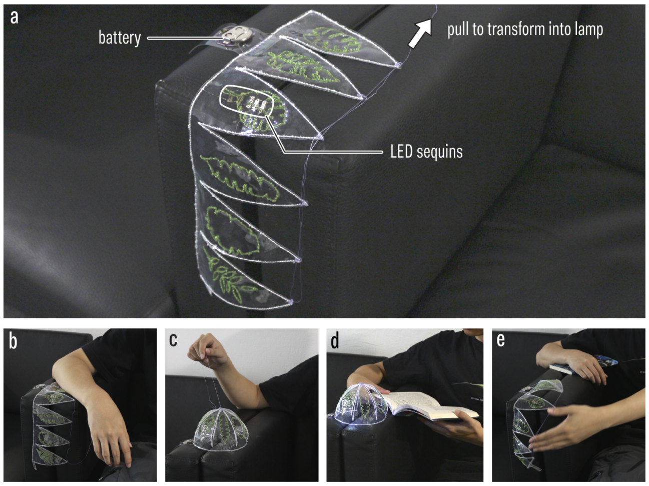


Figure 15: (a) We created a blanket on a sofa armrest that can be reversibly converted into an ambient lamp. The blanket is (b) soft and unobtrusive and (c-d) can be turned into a dome-shaped lamp for reading by pulling the string. (d) The lamp resets with a simple push.

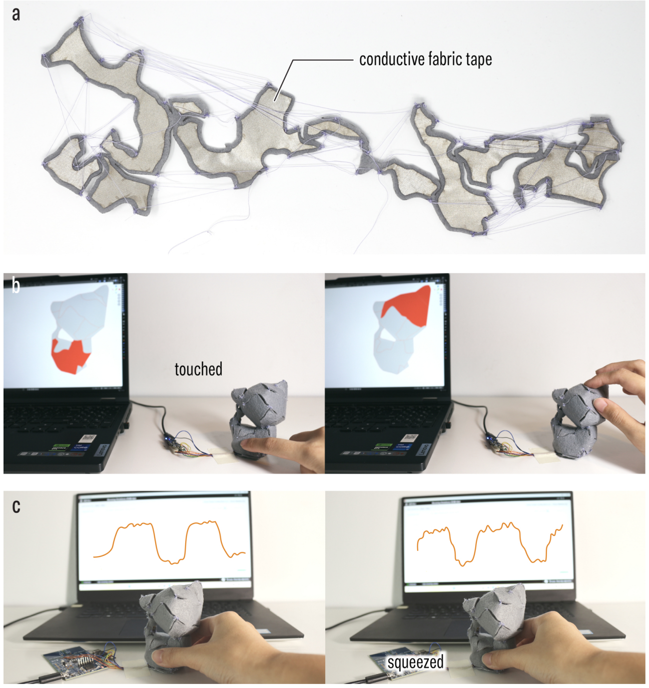


Figure 16: *EmbroForm* enables easy prototyping of soft 3D objects with integrated sensing by (a) applying conductive fabric tapes onto the 2D patch. This enables both (b) touch and (c) squeeze sensing by measuring capacitance.

7.2 Touch-sensitive Kitten

EmbroForm 2D-to-3D transformation enables easy integration of functional electronic components into freeform 3D objects: they can be added when the object is in its 2D state. We show an example of integrating sensing capabilities into a soft kitten in Figure 16. (a) We lasercut conductive fabric tapes for the original segmented 2D patches and applied them to the material after washing away the support. The conductive fabric is put on the interior side, thus does not affect the pulled-up object's surface texture and aesthetics. (b) After pulling up, this can realize patch-based capacitive touch sensing, which detects which patch(s) the user is touching. To process capacitive touch contact, we used an ESP32 microcontroller.

(c) Thanks to the soft and deformable nature of *EmbroForm* objects, they afford a richer set of input and output capabilities that can be sensed and actuated by the added components. For example, contrary to solid 3D objects, users can easily squeeze the kitten. We sense squeezing by measuring the mutual capacitance between the front and back patches on the kitten's body. We used a FDC2214EVM evaluation board for sensing mutual capacitance, and the difference in signal is shown in Figure 16c.

7.3 Animating Custom Characters

The freeform objects made by *EmbroForm* with flexible sheet materials are deformable. This allows not only making custom 3D objects, but also ones that can be manipulated through user interactions for applications like toys for children, deformable tangible

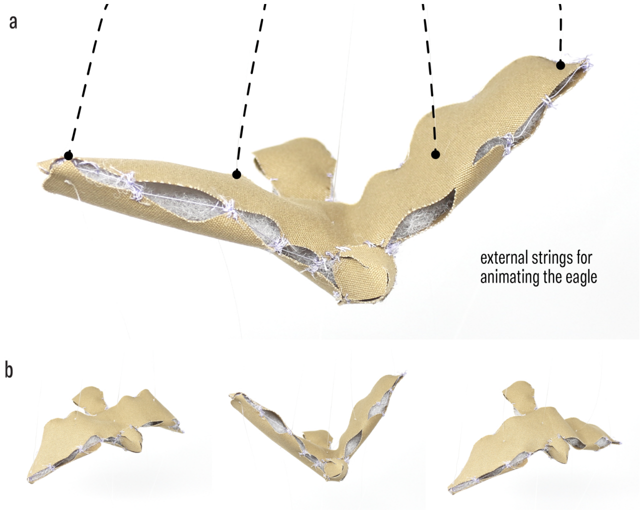


Figure 17: *EmbroForm* creates custom deformable 3D characters that can be used for animation. (a)We attach external strings onto a fabricated eagle to (b) control and animate its wings.

interfaces, authoring animation of custom characters, etc. We show an example of animating custom characters in Figure 17. (a) We marionette an eagle by attaching external strings after fabrication. Alternatively, these additional strings can also be embroidered by manually adding stitches into the generated machine file. (b) The user can then control the additional strings to deform and animate the eagle to transform between different poses. For this specific eagle mesh, we infilled the pull-up character with cotton to enhance the shape fidelity. This will be further discussed in Section 8.

8 Discussion

We discuss limitations and lay out potential future work directions on *EmbroForm*'s achievable shapes and scales, easy and interactive pull-up and resetting, and extending the aesthetics and functionalities of the current prototypes.

8.1 Achievable Shapes and Scales

Failing geometries. While *EmbroForm* works for demanding geometries like the kitten, it has limits. The space of geometries is primarily limited by the segmentation of the input mesh, which depends on the state-of-the-art segmentation algorithm used. For some geometries and their segmentation, a collision-free topology for packing the 2D segmented patches does not exist. For example, when a segmentation creates U-shaped or highly concave patches and another patch that can only be placed inside the concave, collision is unavoidable, no matter where on the merging boundary they touch. We empirically found that this typically happens when the input mesh contains very thin volumes that connect to a larger body, for example, the ears of a Stanford Bunny. The connecting boundary is then very short and of a concave shape, while the thin volume flattens into a larger patch. This makes overlaps difficult to avoid. These "unpackable" geometries are beyond the scope of this paper, as we are using a readily available segmentation method from existing research.

There are multiple potential pathways to solve "unpackable" geometries with overlapping patches. To start with, the *EmbroForm* pipeline is compatible with any other piecewise developable segmentation method. Currently, *EmbroForm* uses a state-of-the-art algorithm [51] that has very high shape accuracy and a low number of patches. This can be flexibly replaced with other existing and future segmentation algorithms that can extend the achievable geometries at some cost. For example, Baharami et al. [5] segments a 3D shape into patches with more regular and less curved boundaries, which would make collision-free pacing easier, but with compromised shape accuracy. Collisions can also be managed in a brute force manner by further segmenting the overlapping patches along the intersecting boundary. This would increase the number of patches and thus the complexity of string routing. Alternatively, our fabrication method based on machine embroidery has the potential to create a pull-up object from separate unwrapped 2D patches. Thanks to the sandwiching stabilizers, we can have separate patches during embroidery that can later be joined by closing the gap. This reduces the constraint on no overlaps by allowing overlapping patches to be disconnected during embroidery. Yet, this would require different lacing routing strategies and might sacrifice the boundary quality, which is then not rigidly connected.

Failing string routing. The *EmbroForm* algorithm strategically routed the pull-up strings to avoid loops and entanglements, as presented in Section 3.3 and Section 4.3. The success of the string routing, however, is not always guaranteed due to spatial constraints. We found that the routed string could still entangle when two lacing areas are closely facing each other and there is not enough space between the areas to accommodate the detoured strings. Because the lacing string in each area is detoured away from the patch into the openings, the string of one area could be in conflict with a tunnel from the other lacing area and thus entangle with the string. To solve this, the space between such lacing areas needs to increase by scaling up the 2D unwrapped pattern.

Volumetric vs. slender shapes. Through empirical prototyping, we found that more volumetric rather than slender shapes work better for *EmbroForm*, especially when softer sheet materials are used. For example, the eagle that we fabricated in Section 7.3 has flat wings with two pieces of fabric very close to each other. Due to the softness of the material, the upper and bottom sides of the wings risk collapsing into each other instead of creating a defined, stable shape. This can be mitigated by putting infill into flatter areas to support the outer soft "shell", such as the infilling cotton we have used to create a plushie in Figure 17. Alternatively, additional threads can be embroidered selectively on these areas to change the local stiffness of the sheet material, as demonstrated by prior works (e.g., [18]). This avoids the slender parts of the shape collapsing. To realize this, future work can extend the algorithm to analytically determine failing regions based on the input geometry and the sheet material used to add additional embroidery patterns.

Scalability. To fabricate the prototypes presented in the paper, we used one of the biggest embroidery hoops that is commercially available and is compatible with our machine. With this hoop, the fabricated prototypes were palm-sized to hand-sized, depending on the number of segmented patches and the efficiency of 2D packing.

This is in line with the scale of fabricated textile 3D structures in existing works [11, 14, 30], which has always been confined by the size of machines available in usual maker spaces. To further scale up the object, one can consider using multi-hooping on the embroidery machine, which allows embroidering a design larger than the hoop by re-hooping to reposition the fabric. However, the lacing string cannot be cut and needs to be embroidered continuously throughout the entire design, which might require frequent re-hooping or optimizing the lacing design such that it can be routed area by area on the fabric. Alternatively, industry-level embroidery machines have much larger working areas that can significantly scale up the pull-up objects.

The limiting factor of scaling down *EmbroForm* objects is the size of the hollow tunnel. We performed technical characterization (Section 6.2) to minimize its size while ensuring mechanical strength against tearing and low friction. During prototyping, we found that due to the need for alignment (i.e., align stabilizers with lasercut pieces) and calibration (i.e., calibrate the material for the machine) in the pipeline, drifts and errors occur. During embroidery, the stitches might also pull the material and create drifts. Due to these reasons, we found that the error on the final embroidered patch can go up to 2mm, which makes robust scaling down more challenging. Thus, if precision through fabrication steps can be further improved, scaling down the hollow tunnel and thereafter the pull-up object can be achieved.

8.2 Enabling Easy and Interactive Pull-up and Resetting

We chose to route only one string, instead of using several strings for different areas of the object, to minimize the post-processing effort required from the user. In consequence, the user may need to pull segment-by-segment during a pull-up or resetting (shown in Section 6.3), and the pull-up distance can be quite long (shown in Figure 14). Other than the long pulling distance, we anecdotally report that this is due to the string being locally stuck in one of the tunnels during pulling, which happens by chance. As the hollow tunnels contain a short segment of the sheet material (i.e., d_{in} in Figure 12), the string could be stuck in-between the sheet material and the tunnel stitches, resulting in local high friction. This happens by chance when the boundary and the string make a very sharp turn. The very thin silk string *EmbroForm* uses could also twist or temporarily self-loop during pulling, making a small bump that might get stuck when it passes through a tunnel. In either case, a larger pulling force is temporarily needed to free the string such that it can move smoothly again.

There are potential pathways to solve these issues to enable interactive and real-time shape-change even for complex geometries. One promising avenue to facilitate easier pull-ups is to split the pull-up string into multiple segments that the user can pull together. For example, the string could be split between the identified lacing areas. This shortens the overall pulling distance and reduces the sharp turns usually between the lacing areas to make pull-up easier. With this, the pull-up can even be automatically actuated (e.g., with DC motors) without requiring manual adjustment. The pulling distances for the split segments can be calculated and aligned to allow synchronous actuation. While our lacing design for the pull-up string effectively informs the routing to avoid entanglement and minimizes the overall string length, entirely new string routing designs and algorithms that prioritize other factors can be developed. For example, one can minimize the overall redirection of the routing to facilitate smoother string movement.

To enable easier resetting, worthwhile directions for future work include reducing the friction by using glossier thread or sheet material, employing an elastic string that bounces back automatically when released, or integrating another resetting string on the opposite side of the material that passes through the lacings, such that they can be elongated to reset the shape when the resetting string is pulled.

8.3 Extending Aesthetics and Functionality

The prototypes fabricated in this paper mostly used only the sheet materials themselves. Contrasting colors were used for the tunnels and the sheet materials to make the tunnels visible. The aesthetics and the functionalities of the prototypes, however, can be easily extended. The colors of the materials can be flexibly changed, e.g., keeping the colors consistent such that the thread-based tunnels can blend into the fabric. The sheet fabric can be manipulated both in its 2D and 3D pulled-up states. *EmbroForm* objects only have tunnels and lacings embroidered on the boundaries, leaving most of the sheet material untouched and available for additional add-ons. Our applications show examples of transforming the passive pull-up objects into interactive interfaces by embroidering aesthetic patterns, embedding fabric-compatible electronics, and attaching additional strings for actuation.

Many additional opportunities exist, such as drawing, processing the 2D material such that the pull-up object can have localized visual or haptic properties, or integrating thin-form actuators like SMA for shape changes, etc., to further extend the aesthetics and the functions of pull-up objects.

9 Conclusions

We introduced *EmbroForm*, a digital fabrication pipeline for prototyping fully soft pull-up objects with organic, higher-fidelity shapes. *EmbroForm* achieves this by seaming the boundaries of a flexible 2D patch unwrapped from a target geometry. To realize this, we contribute a novel fabrication technique that automates the routing of sliding strings on flexible sheet material using machine embroidery. We further design lacings that can be embroidered to join highly curved soft boundaries in 3D when the lacing is pulled. Based on this fabrication technique, we contribute an end-to-end pipeline that, based on an input 3D mesh, generates an optimized 2D unwrapped patch and the lacing routing paths for fabrication. We validated EmbroForm through technical studies and demonstrated its versatility with applications in shape-changing furniture, interactive toys, and custom animated characters.

References

[1] Muhammad Abdullah, Martin Taraz, Yannis Kommana, Shohei Katakura, Robert Kovacs, Jotaro Shigeyama, Thijs Roumen, and Patrick Baudisch. 2021. FastForce: Real-Time Reinforcement of Laser-Cut Structures. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 673, 12 pages. doi:10.1145/3411764.3445466

[2] Roland Aigner, Andreas Pointner, Thomas Preindl, Rainer Danner, and Michael Haller. 2021. TexYZ: Embroidering Enameled Wires for Three Degree-of-Freedom Mutual Capacitive Sensing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 499, 12 pages. doi:10.1145/3411764.3445479

[3] Byoungkwon An, Ye Tao, Jianzhe Gu, Tingyu Cheng, Xiang 'Anthony' Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, Shigeo Takahashi, Hsiang-Yun Wu, Teng Zhang, and Lining Yao. 2018. Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3173834

[4] Daniel Ashbrook, Wei-Ju Lin, Nicholas Bentley, Diana Soponar, Zeyu Yan, Valkyrie Savage, Lung-Pan Cheng, Huaishu Peng, and Hyunyoung Kim. 2024. Rhapso: Automatically Embedding Fiber Materials into 3D Prints for Enhanced Interactivity. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 120, 20 pages. doi:10.1145/3654777.3676468

[5] Hassan Baharami, Michal Piovarci, Marco Tarini, Bernd Bickel, and Nico Pietroni. 2025. Fabricable Discretized Ruled Surfaces. *ACM Trans. Graph.* 44, 3, Article 30 (June 2025), 15 pages. doi:10.1145/3734519

[6] Zekun Chang, Yuta Noma, Shuo Feng, Xinyi Yang, Kazuhiro Shinoda, Tung D. Ta, Koji Yatani, Tomoyuki Yokota, Takao Someya, Yoshihiro Kawahara, Koya Narumi, Francois Guimbretiere, and Thijs Roumen. 2024. OriStitch: A Machine Embroidery Workflow to Turn Existing Fabrics into Self-Folding 3D Textiles. arXiv:2412.02891 [cs.HC] https://arxiv.org/abs/2412.02891

[7] Erik D. Demaine and Tomohiro Tachi. 2017. Origamizer: A Practical Algorithm for Folding Any Polyhedron. In *33rd International Symposium on Computational Geometry (SoCG 2017) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 77)*, Boris Aronov and Matthew J. Katz (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 34:1–34:16. doi:10.4230/LIPIcs.SoCG.2017.34

[8] Shuyue Feng, Cheng Yao, Weijia Lin, Jiayu Yao, Chao Zhang, Zhongyu Jia, Lijuan Liu, Masulani Bokola, Hangyue Chen, Fangtian Ying, and Guanyun Wang. 2023. MechCircuit: Augmenting Laser-Cut Objects with Integrated Electronics, Mechanical Structures and Magnets. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 735, 15 pages. doi:10.1145/3544548.3581002

[9] Marco Freire, Manas Bhargava, Camille Schreck, Pierre-Alexandre Hugron, Bernd Bickel, and Sylvain Lefebvre. 2023. PCBend: Light Up Your 3D Shapes With Foldable Circuit Boards. *ACM Trans. Graph.* 42, 4, Article 142 (July 2023), 16 pages. doi:10.1145/3592411

[10] Jun Gong, Yu Wu, Lei Yan, Teddy Seyed, and Xing-Dong Yang. 2019. Tessutivo: Contextual Interactions on Interactive Fabrics with Inductive Sensing. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 29–41. doi:10.1145/3332165.3347897

[11] François Guimbretière, Victor F Guimbretière, Amritansh Kwatra, and Scott E Hudson. 2025. Using an Array of Needles to Create Solid Knitted Shapes. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 100, 11 pages. doi:10.1145/3746059.3747759

[12] Alice C Haynes and Jürgen Steimle. 2024. Flextiles: Designing Customisable Shape-Change in Textiles with SMA-Actuated Smocking Patterns. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 517, 17 pages. doi:10.1145/3613904.3642848

[13] David Hilbert and S. Cohn-Vossen. 1999. *Geometry and the Imagination* (2nd ed.). AMS Chelsea Publishing.

[14] Yuichi Hirose, Mark Gillespie, Angelica M. Bonilla Fominaya, and James McCann. 2024. Solid Knitting. *ACM Trans. Graph.* 43, 4, Article 88 (July 2024), 15 pages. doi:10.1145/3658123

[15] Scott E. Hudson. 2014. Printing teddy bears: a technique for 3D printing of soft interactive objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 459–468. doi:10.1145/2556288.2557338

[16] International Organization for Standardization. 2011. ISO 4604:2011 - Reinforcement fabrics — Determination of conventional flexural stiffness — Fixed-angle flexometer method. https://www.iso.org/standard/59252.html Accessed: 2025-09-04.

[17] Alexandra Ion, Michael Rabinovich, Philipp Herholz, and Olga Sorkine-Hornung. 2020. Shape approximation by developable wrapping. *ACM Trans. Graph.* 39, 6, Article 200 (Nov. 2020), 12 pages. doi:10.1145/3414685.3417835

[18] Yu Jiang, Alice C Haynes, Narjes Pourjafarian, Jan Borchers, and Jürgen Steimle. 2024. Embrogami: Shape-Changing Textiles with Machine Embroidery. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 63, 15 pages. doi:10.1145/3654777.3676431

[19] Martin Kilian, Aron Monszpart, and Niloy J. Mitra. 2017. String Actuated Curved Folded Surfaces. *ACM Trans. Graph.* 36, 3, Article 25 (May 2017), 13 pages. doi:10.1145/3015460

[20] Vidyadhar G. Kulkarni. 1990. Generating random combinatorial objects. *Journal of Algorithms* 11, 2 (1990), 185–207.

[21] Honghua Li, Ruizhen Hu, Ibraheem Alhashim, and Hao Zhang. 2015. Foldabilizing furniture. *ACM Trans. Graph.* 34, 4, Article 90 (July 2015), 12 pages. doi:10.1145/2766912

[22] Jiaji Li, Shuyue Feng, Maxine Perroni-Scharf, Yujia Liu, Emily Guan, Guanyun Wang, and Stefanie Mueller. 2025. Xstrings: 3D Printing Cable-Driven Mechanism for Actuation, Deformation, and Manipulation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 6, 17 pages. doi:10.1145/3706598.3714282

[23] E Meenan and Briony Thomas. 2008. Pull-up Patterned Polyhedra: Platonic Solids for the Classroom.

[24] Sara Mlakar and Michael Haller. 2020. Design Investigation of Embroidered Interactive Elements on Non-Wearable Textile Interfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/3313831.3376692

[25] Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. 2013. LaserOrigami: laser-cutting 3D objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2585–2592. doi:10.1145/2470654.2481358

[26] Sara Nabil, Jan Kučera, Nikoletta Karastathi, David S. Kirk, and Peter Wright. 2019. Seamless Seams: Crafting Techniques for Embedding Fabrics with Interactive Actuation. In *Proceedings of the 2019 on Designing Interactive Systems Conference* (San Diego, CA, USA) *(DIS '19)*. Association for Computing Machinery, New York, NY, USA, 987–999. doi:10.1145/3322276.3322369

[27] Uyen Nguyen. 2020. Folding Fabric: Fashion from Origami. In *Proceedings of Bridges 2020: Mathematics, Art, Music, Architecture, Education, Culture*, Carolyn Yackel, Robert Bosch, Eve Torrence, and Kristóf Fenyvesi (Eds.). Tessellations Publishing, Phoenix, Arizona, 93–102. http://archive.bridgesmathart.org/2020/bridges2020-93.html

[28] Lauren Niu, Xinyi Yang, Martin Nisser, and Stefanie Mueller. 2023. PullupStructs: Digital Fabrication for Folding Structures via Pull-up Nets. In *Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction* (Warsaw, Poland) *(TEI '23)*. Association for Computing Machinery, New York, NY, USA, Article 52, 6 pages. doi:10.1145/3569009.3573123

[29] Simon Olberding, Sergio Soto Ortega, Klaus Hildebrandt, and Jürgen Steimle. 2015. Foldio: Digital Fabrication of Interactive and Shape-Changing Objects With Foldable Printed Electronics. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 223–232. doi:10.1145/2807442.2807494

[30] Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. 2015. A Layered Fabric 3D Printer for Soft Interactive Objects. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1789–1798. doi:10.1145/2702123.2702327

[31] Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski. 2017. Computational design and automated fabrication of kirchhoff-plateau surfaces. *ACM Trans. Graph.* 36, 4, Article 62 (July 2017), 12 pages. doi:10.1145/3072959.3073695

[32] Andreas Pointner, Thomas Preindl, Mira A. Haberfellner, Nitzan Cohen, Niko Münzenrieder, and Michael Haller. 2025. Embroidering Resonant Circuits for Inductive Pressure Sensing. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 2, 7 pages. doi:10.1145/3746059.3747733

[33] Thomas Preindl, Cedric Honnet, Andreas Pointner, Roland Aigner, Joseph A. Paradiso, and Michael Haller. 2020. Sonoflex: Embroidered Speakers Without Permanent Magnets. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 675–685. doi:10.1145/3379337.3415888

[34] Thijs Roumen, Yannis Kommana, Ingo Apel, Conrad Lempert, Markus Brand, Erik Brendel, Laurenz Seidel, Lukas Rambold, Carl Goedecken, Pascal Crenzin, Ben Hurdelhey, Muhammad Abdullah, and Patrick Baudisch. 2021. Assembler3: 3D Reconstruction of Laser-Cut Models. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 672, 11 pages. doi:10.1145/3411764.3445453

[35] René Schäfer, Oliver Nowak, Lovis Bero Suchmann, Sören Schröder, and Jan Borchers. 2023. What's That Shape? Investigating Eyes-Free Recognition of Textile Icons. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 580, 12 pages. doi:10.1145/3544548.3580920

[36] Christian Schüller, Roi Poranne, and Olga Sorkine-Hornung. 2018. Shape representation by zippables. *ACM Trans. Graph.* 37, 4, Article 78 (July 2018), 13 pages. doi:10.1145/3197517.3201347

[37] Aviv Segall, Jing Ren, Amir Vaxman, and Olga Sorkine-Hornung. 2024. Fabric Tessellation: Realizing Freeform Surfaces by Smocking. *ACM Trans. Graph.* 43, 4, Article 89 (July 2024), 20 pages. doi:10.1145/3658151

[38] Masahito Takezawa, Takuma Imai, Kentaro Shida, and Takashi Maekawa. 2016. Fabrication of freeform objects by principal strips. *ACM Trans. Graph.* 35, 6, Article 225 (Dec. 2016), 12 pages. doi:10.1145/2980179.2982406

[39] Ye Tao, Youngwook Do, Humphrey Yang, Yi-Chin Lee, Guanyun Wang, Catherine Mondoa, Jianxun Cui, Wen Wang, and Lining Yao. 2019. Morphlour: Personalized Flour-based Morphing Food Induced by Dehydration or Hydration Method. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 329–340. doi:10.1145/3332165.3347949

[40] Sapna Tayal, Lea Albaugh, James McCann, and Scott E Hudson. 2025. Creating Furniture-Scale Deployable Objects with a Computer-Controlled Sewing Machine. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 438, 15 pages. doi:10.1145/3706598.3713735

[41] Guanyun Wang, Chuang Chen, Xiao Jin, Yulu Chen, Yangweizhe Zheng, Qianzi Zhen, Yang Zhang, Jiaji Li, Yue Yang, Ye Tao, Shijian Luo, and Lingyun Sun. 2025. TH-Wood: Developing Thermo-Hygro-Coordinating Driven Wood Actuators to Enhance Human-Nature Interaction. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 745, 19 pages. doi:10.1145/3706598.3714304

[42] Guanyun Wang, Tingyu Cheng, Youngwook Do, Humphrey Yang, Ye Tao, Jianzhe Gu, Byoungkwon An, and Lining Yao. 2018. Printed Paper Actuator: A Low-cost Reversible Actuation and Sensing Method for Shape Changing Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3174143

[43] Guanyun Wang, Tingyu Cheng, Youngwook Do, Humphrey Yang, Ye Tao, Jianzhe Gu, Byoungkwon An, and Lining Yao. 2018. Printed Paper Actuator: A Low-cost Reversible Actuation and Sensing Method for Shape Changing Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3174143

[44] Guanyun Wang, Fang Qin, Haolin Liu, Ye Tao, Yang Zhang, Yongjie Jessica Zhang, and Lining Yao. 2020. MorphingCircuit: An Integrated Design, Simulation, and Fabrication Workflow for Self-morphing Electronics. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 157 (Dec. 2020), 26 pages. doi:10.1145/3432232

[45] Guanyun Wang, Yue Yang, Mengyan Guo, Kuangqi Zhu, Zihan Yan, Qiang Cui, Zihong Zhou, Junzhe Ji, Jiaji Li, Danli Luo, Deying Pan, Yitao Fan, Teng Han, Ye Tao, and Lingyun Sun. 2023. ThermoFit: Thermoforming Smart Orthoses via Metamaterial Structures for Body-Fitting and Component-Adjusting. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 31 (March 2023), 27 pages. doi:10.1145/3580806

[46] Rundong Wu, Claire Harvey, Joy Xiaoji Zhang, Sean Kroszner, Brooks Hagan, and Steve Marschner. 2020. Automatic structure synthesis for 3D woven relief. *ACM Trans. Graph.* 39, 4, Article 102 (Aug. 2020), 10 pages. doi:10.1145/3386569.3392449

[47] Tianhong Catherine Yu, Nancy Wang, Sarah Ellenbogen, and Cindy Hsin-Liu Kao. 2023. Skinergy: Machine-Embroidered Silicone-Textile Composites as On-Skin Self-Powered Input Sensors. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 33, 15 pages. doi:10.1145/3586183.3606729

[48] Clint Zeagler, Scott Gilliland, Halley Profita, and Thad Starner. 2012. Textile Interfaces: Embroidered Jog-Wheel, Beaded Tilt Sensor, Twisted Pair Ribbon, and Sound Sequins. In *2012 16th International Symposium on Wearable Computers*. 60–63. doi:10.1109/ISWC.2012.29

[49] Xiaoting Zhang, Guoxin Fang, Melina Skouras, Gwenda Gieseler, Charlie C. L. Wang, and Emily Whiting. 2019. Computational design of fabric formwork. *ACM Trans. Graph.* 38, 4, Article 109 (July 2019), 13 pages. doi:10.1145/3306346.3322988

[50] Zheng-Yu Zhao, Qing Fang, Wenqing Ouyang, Zheng Zhang, Ligang Liu, and Xiao-Ming Fu. 2022. Developability-driven piecewise approximations for triangular meshes. *ACM Trans. Graph.* 41, 4, Article 43 (July 2022), 13 pages. doi:10.1145/3528223.3530117

[51] Zheng-Yu Zhao, Mo Li, Zheng Zhang, Qing Fang, Ligang Liu, and Xiao-Ming Fu. 2023. Evolutionary Piecewise Developable Approximations. *ACM Trans. Graph.* 42, 4, Article 120 (July 2023), 14 pages. doi:10.1145/3592140

[52] Liu Zhenyuan, Michal Piovarči, Christian Hafner, Raphaël Charrondière, and Bernd Bickel. 2023. Directionality-Aware Design of Embroidery Patterns. *Computer Graphics Forum* 42, 2 (2023), 397–409. doi:10.1111/cgf.14770 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14770