Developable Metamaterials: Mass-fabricable Metamaterials by Laser-Cutting Elastic Structures

Madlaina Signer*, Alexandra Ion"*, Olga Sorkine-Hornung*
* Department of Computer Science, ETH Zurich, Zurich, Switzerland
" Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA

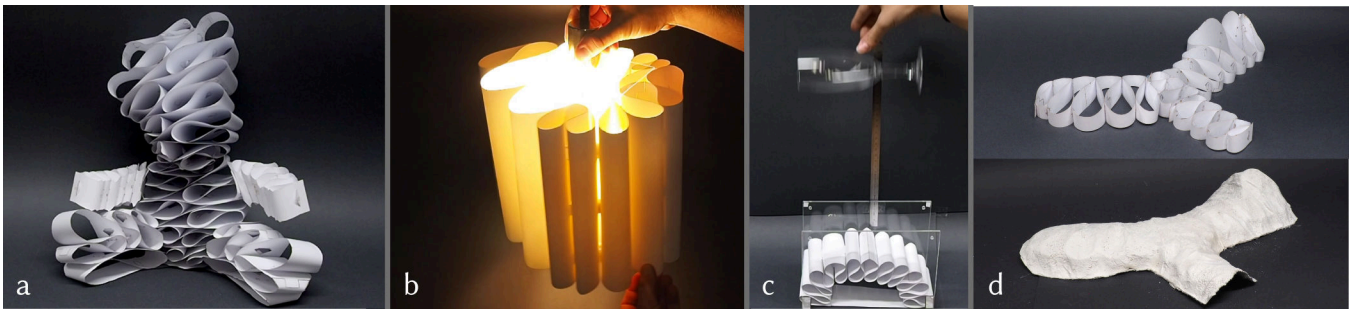

Figure 1: We propose a novel material design that can be *laser cut* with user-defined compliance. We show how such fast to produce materials allow users to create custom materials, such as (a) stuffing for toys with custom softness, (b) lamp shades with variable transparency, (c) damping materials for packaging, or (d) be used for static support (e.g., formwork for architecture).

ABSTRACT

We propose a novel design of engineered, structured materials that leverages fast fabrication technologies, pushing them towards mass-fabrication. Specifically, our metamaterial is designed to be *laser cut*, to approximate the *volumetric* shape and allow for locally *varying compliance*. Traditional mechanical metamaterials consist of intricate cells arranged on a 3-dimensional grid, limiting them to 3D printing—which is slow. Our metamaterial is designed for laser cutting, which is drastically faster. Our structures are best described as *ruffled* strips of thin sheet material, such as paper, plastics, metals, etc. Users can interactively define the ruffles' anisotropic stiffness directions and local density. Our computational design tool assists users by automatically optimizing the ruffle to fill the shape's volume, and exporting the flat ruffle design ready for cutting. We demonstrate how such ruffled metamaterials can be utilized for, e.g., custom toys with locally varying compliance, custom packaging material, or lightweight formwork for architectural shells.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Applied computing** → *Computer-aided design*; • **Computing methodologies** → *Shape modeling*; *Interactive simulation.*

KEYWORDS

Fabrication, Metamaterials, Computational design, Laser cutting, Developable surfaces, Shape modeling

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM Reference Format:
Madlaina Signer*, Alexandra Ion"*, Olga Sorkine-Hornung*. 2021. Developable Metamaterials: Mass-fabricable Metamaterials by Laser-Cutting Elastic Structures. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3411764.3445666

1 INTRODUCTION

The availability of digital fabrication machines, such as 3D printers or laser cutters, allows users to create custom physical objects of arbitrary shape. This freedom has inspired many applications, including decorative objects, interfaces for accessibility [7, 40], sturdy yet lightweight objects [8, 26], or large scale structures [23], just to name a few.

Beyond designing objects with custom shape, digital fabrication expands the design space to novel and custom materials—known as mechanical metamaterials. Metamaterials typically consist of microstructures, i.e., intricate cells arranged on a 3-dimensional grid, and exhibit functionalities differing from those of the material that they are made of [2]. Such metamaterials demonstrate advanced engineered properties, such as being ultra-lightweight [43], volume-changing [42], damping [10, 48], shape-morphing [36, 54] or enabling locally varying elasticity [28, 37, 46].

Three-dimensional metamaterials are typically limited to manufacturing by 3D printing due to the complexity of their microstructure [28, 46]. However, 3D printing is slow. Faster fabrication methods include cutting or molding, but these have only been shown to produce 2-dimensional sheet metamaterials [22, 38].

In this paper, we present *volumetric* metamaterials that are designed for *laser cutting*. They are fast to produce yet allow for custom local elasticity. With this work we push metamaterials towards mass-fabrication, with the goal of increased real-world impact. To create volumetric 3D metamaterials from laser-cut 2D

sheets, we rethink how (1) unit cells, (2) their arrangement and (3) their connections look like, which we outline in the following section.

1.1 Developable metamaterial: elasticity through ruffles

We design our metamaterials to be laser cuttable, hence we are bound to use sheet materials. Thin sheets, such as paper, have a very low bending stiffness and therefore can barely resist any loads, i.e., they are very compliant. We show in Fig. 2b how folding the same paper strip into a smooth ruffle decreases its compliance. This *ruffle* is the *unit cell* in our metamaterial design.

We can further control the compliance of our material by controlling the *arrangement* of our ruffles within the paper strip. The example in Fig. 2c shows how using the same paper strip, but folding it into multiple dense ruffles, decreases its compliance drastically. These basic concepts of our material design allow us to create volumetric approximations of 3D shapes with controllable compliance using laser cut strips.

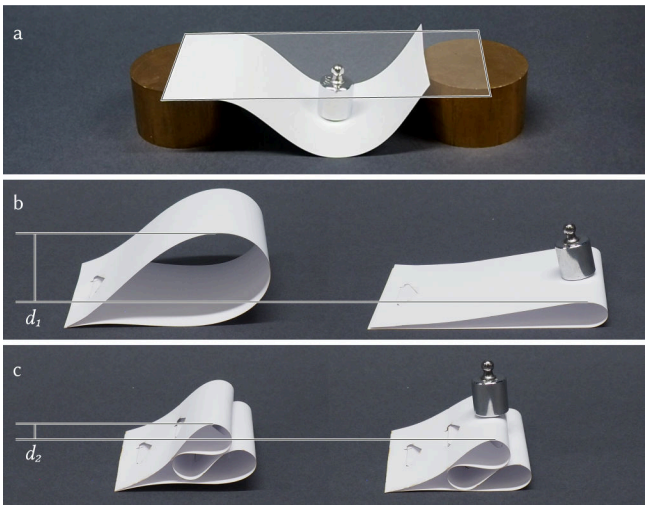

Figure 2: (a) Thin sheets, such as paper, cannot resist significant load, as they have low bending stiffness. (b) Folding the paper of the same length into a ruffle decreases its compliance, such that it deforms less under the same load. These ruffles are our *unit cells*. We can further control the compliance by changing the arrangement of the paper strip. (c) Using the same piece of paper, but folding it into multiple ruffles decreases its compliance significantly. (Paper strip width = 5 cm, length = 24 cm, load = 50 g)

The reason behind the change in compliance it that we introduce curvature [50]. The curvature introduces bending energy (thereby elastic energy), which is proportional to the mean curvature squared. The compliance of the ruffles is not uniform. Since their connection is tangential, their curvature changes. We illustrate this anisotropy in Fig. 3. We create the ruffle shape as a curve on the plane and then extrude it in the orthogonal direction. As such, the ruffle can withstand the highest load along the extrusion axis z. The compliance along the x and y axes are defined by the ruffle shape. Since our ruffles are connected such that they meet tangentially, the ruffle curve (on the plane) tends to be wider in x direction than in y direction. This aspect ratio makes y the softest direction.

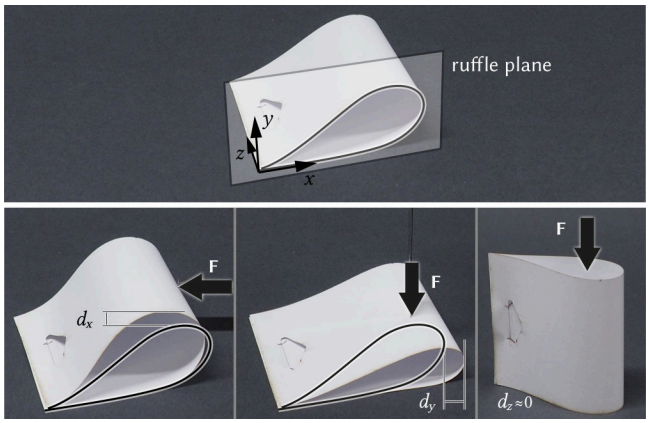

Figure 3: Such a ruffle is has different compliance directions. We can control the softer x and y directions. The z direction is the ruffle's extrusion width which is defined by the target mesh.

These different compliance directions are modeled in our computational design tool. Users place and orient the ruffle plane to define the extrusion directions. They can locally change the ruffle density to define the compliance in x and y directions. Our algorithm optimizes the ruffle according to these user specifications to fill the object's volume. The target shape is what defines the ruffle extrusion widths and thereby the local stiffness along the z axis. For more fine-grained control, users can split their object into multiple, individually controllable ruffles. We detail the interactions in Section 3 and the implementation in Section 4.

Our design tool exports the flat ruffle as an SVG file, ready for cutting. We design the fabrication with ease of assembly in mind, as we demonstrate in Fig. 4. To fold the strip into the final 3D shape, users only connect the interlocking tabs that are cut into the strip.

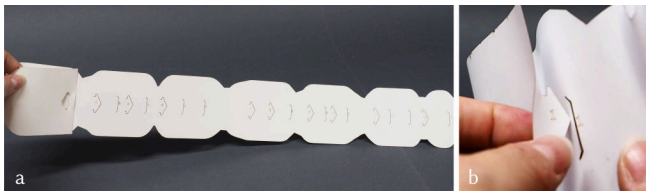

Figure 4: (a) Our design tool outputs a flat ruffle with all connector tabs, ready for laser cutting. (b) Assembling the ruffle only requires users to insert the labeled tabs into their corresponding slit.

We derive the name *developable metamaterials* from the concept of developability, where materials are not stretched or compressed but only bent. Since the 3D shape of the ruffles is created by parallel extrusion of the planar ruffle curves, our structures are inherently

developable, i.e., they can be flattened without distortion, facilitating our simple fabrication process.

1.2 Contributions

The main contribution of this work is the concept of *laser-cuttable volumetric metamaterials*. Specifically, we contribute a holistic system that consists of the following parts.

(1) *Novel metamaterial design:* We contribute the conceptual metamaterials design, which consists of the ruffles (analogous to unit cells in traditional metamaterial designs) and the arrangement on the paper strip (analogous to the tiling on a grid). This design space includes the variable density of ruffles, such that users can define 3D shapes with user-specified local compliance represented by volumetric ruffles.
(2) *Computational design tool:* Our computational design tool allows users to interactively design their material's compliance, optimizes the ruffle shapes to fill the target volume, and exports the optimized ruffle as a flat design ready for cutting.
(3) *Simple fabrication process:* We devise the design of the interlocking flap joint that enables a simple assembly process without the need for external fasteners or glue, since it is cut into the flat ruffle design.

This work is part of a higher-level goal to create *mass-fabricable metamaterials* to increase their real-world and industrial relevance. It is the first step towards this goal and focuses specifically on (laser) cutting as the fabrication technology and compliance as the adaptable material property. There is plenty of space in this area for investigating novel metamaterials targeted to other fabrication technologies, such as (injection) molding, or different material properties.

2 RELATED WORK

The work presented in this paper builds on previous work in interactive fabrication focussing on laser cutting, on shape-changing interfaces and metamaterials. We also build on existing methods in computational design and simulation of thin shells and developable surfaces.

Fabrication by laser cutting. Laser cutting is a fast fabrication technique compared to 3D printing. With the speed benefit, a restriction in dimensionality arises, which is why the approximation of 3D objects by assembly of flat parts is an active research field. To take advantage of the speed benefit of cutting, previous works have proposed methods and software systems that help users design objects that can be assembled e. g., by using slotting joints [17, 30] or finger joints [1], by extracting plates from 3D shapes [3], bending acrylic sheets [33], or welding layers of acrylic [52]. These works investigated assembling rigid plates. Laser cutting can also be used to create objects with some flexibility such as embedding bendable parts [25, 34], integrating springs to firmly hold external components [41], or directly cutting planar patterns for stretchable sheets [12] and compressible honeycomb structures with integrated sensing [55]. While reducing the need for assembly was a focus on many aforementioned works, they do require some level of assembly.

To eliminate the need for assembly, researchers investigated self-assembling objects, i. e., objects that can be fabricated flat and be activated to fold into the target shape after fabrication. To encode the folding locations, researchers have proposed different methods, e. g., layering different materials [16, 51], changing the direction of the 3D print path [13, 54], or combining pre-stretched materials with rigid structures [15].

Engineered materials & Mechanical metamaterials. Going beyond creating objects with custom shape, researchers investigated laser cut sheets that define novel material properties. The working principle behind such materials is their decomposition into cells and arrangement on a grid, known as *metamaterials*. The unit cells play together to enable the engineered material properties. Often, such 2D metamaterials are based on origami, i. e., sharp creases on a sheet of paper that govern the deformation. A famous example is the Miura fold [32]; it is auxetic and unfolds in two dimensions upon 1-dimensional actuation.

To eliminate the necessity for folding, engineers build custom material properties into *kirigami* sheets. Kirigami is a variation of origami that includes cutting the paper, which facilitates easy fabrication. Such kirigami sheets have been shown to exhibit area-changing (auxetic) properties as well [39], or to increase the stability of the sheet material after extension [38]. Beyond uniform deformation, researchers in graphics showed how such planar auxetic sheets can model 3D surfaces [22] by locally defining the amount of stretch in the cells.

Since the many cells play together to exhibit a desired property, their design is difficult. Inverse design tools for structured sheets of materials [27, 47] help users design such materials, by optimizing the parameters (e. g., edge lengths) of a structure family as to find the best-fitting material for users definition.

The utility of engineered materials is versatile and ranges from emulating the stiffness of traditional materials to make them fabricable from one type of material [4], over varying their compliance [28, 29, 37, 46], to damping materials [10, 48] and mechanically cloaking inset objects [5]. Moving towards *devices*, instead of material properties, made from metamaterials, the integration of mechanisms [18, 20], computation [18] and shape-change [19] into metamaterial structures have been shown. Note that all materials in this paragraph were *3D printed*.

In our work, we contribute a novel metamaterial design that can be laser cut and encodes localized compliance *volumetrically* in 2.5D. Our fabrication process is fast and pushes laser-cut metamaterials beyond sheets.

Thin shells & Developable surfaces. In order to provide a design tool for our developable metamaterials, we need to simulate the ruffles. Since traditional finite element methods are too slow to be used in an iterative optimizing algorithm, we build on thin shell theory. There exists a large body of work in this area. We will only discuss the closest related work and refer readers to the work of Destuynder et al. [9] for in-depth discussions. Thin shells are flexible structures, sometimes reduced to plates, that have a very small thickness compared to their width. Thin shells are curved in their resting position, while their simplified derivation of thin *plates* are flat. This applies to our problem definition and thus is the starting point for our ruffle simulation. The closest related work

is *discrete shells* [11], which we directly build on. We detail our adaptation in Section 4.

For ease of fabrication, we defined that our metamaterial shall be developable. Developability constraints define that a 3D shape may have at any point only a cylindrical deformation, i. e., the Gaussian curvature is zero. That means that the material is only bent, not stretched or compressed, which simplifies the assembly process drastically. Developable surfaces are an active research area with a long history [24] and many recent advances. Approximating arbitrary shapes with typically multiple developable surfaces is often based on iteratively deforming the shape until they achieve vanishing Gaussian curvature [49, 53]. Alternatively, effective methods decomposed the shape into developable strips [31]. The joint design for connecting the strips is left for end users to design. Zippables [45] introduced a complete system that not only optimizes developable ribbons for a given shape, but also introduces a fabrication process to add zippers for connecting the ribbons.

Our metamaterials are also developable ribbons, but they fill the volume and the connectors are embedded in the strips. Since we only consider cylindrically curved strips, we do not need to optimize for developability, as those are developable by construction. However, we see interesting opportunities to adding ruffles along 3D curves, which will require enforcing developability constraints.

3 DESIGN TOOL

To allow users to create developable metamaterials and specify the localized compliance of their objects, we contribute a computational design tool targeted at makers. In this section, we show how users specify the material properties on their target shape. We detail how our algorithm approximates the shape with optimized ruffles based on the user input in the next section.

3.1 Shape approximation with one ruffle

Users load the target shape as a common triangle mesh into our design tool. In the following, we start with a simple cuboid. Upon loading the model, our system automatically creates a default ruffle that serves as a preview. As Fig. 5 shows, this ruffle approximates the entire shape and can readily be exported.

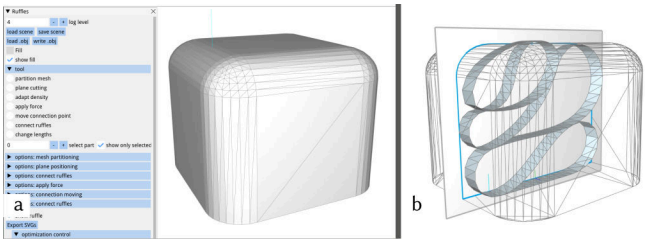

Figure 5: (a) Our editor automatically creates a (b) optimized ruffle, which serves as a preview.

Fig. 6 illustrates this process, which runs in the background. To create a ruffle, our algorithm requires a *cutting plane* to obtain the outline of the 3D mesh intersection. This outline is the *2D target shape* of our ruffle optimization. Our system initializes a default ruffle "stack" curve in 2D and optimizes the curve to fill the 2D target shape (detailed in Section 4). Our system then extrudes the optimized ruffle curve and intersects it with the *3D mesh* to obtain the 3-dimensional ruffle.

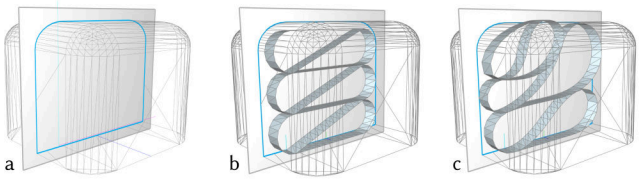

Figure 6: To create optimized volume-filling ruffles, our systems (a) places a default cutting plane and uses the outline as the target shape. (b) It initializes an approximate ruffle curve and (c) optimizes the 2D ruffle to fill the shape. All these steps are performed automatically in the background.

Our algorithm assigns a default number of ruffles in the stack, based on ruffle lengths that we determined empirically. Since the ruffle stack is uniform, the compliance of the object is uniform as well. This simple default ruffle is a good preview but might not be appropriate for users' application. Therefore, users can selectively change the density locally, as we show in Fig. 7. Increasing and decreasing the ruffle density can be done recursively on user-selected ruffles.

Changing the ruffle's density results in a new topology, i.e., introduces new connection points and changes the order of existing ones. With this new topology, we optimize the ruffle again to fill the target shape. When users are satisfied with their design, they can export the flat ruffle ready for the laser cutter. To export the flat ruffle, our system simply lays each segment flat, adds the interlocking tabs at the connection points and writes the result as an SVG file. Thanks to the cylindrical nature of the ruffle, the flattening is guaranteed to be free of self-intersections, i.e., it is always possible without any further processing or segmentation. The flat ruffle is only constrained by the cutter's bed and the material sheet size. Note that we circumvent this size restriction by manually subdividing the flattened ruffles and arranging them onto multiple sheets. The connections are planar finger joints to minimize the change in stiffness. This is visible in Fig. 7 at the top right ruffle.

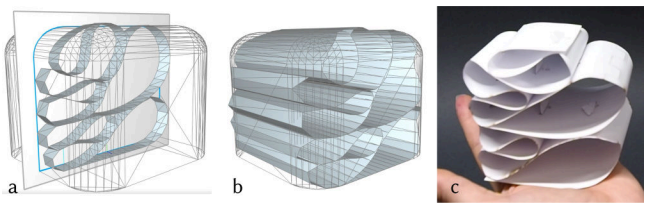

Figure 7: Users decide that their object, which is a stool, should have two different comfort zones. (a) They increase the density of the ruffles on the left side to get a more rigid sitting area. (b) Upon export, our algorithm intersects with the input 3D mesh and exports the flat ruffle. (c) A miniature version of the comfy stool.

3.2 Multiple ruffles for additional control

While changing the ruffle density alters the compliance in the xy-plane, it does not affect the stiffness orthogonal to the cutting plane. Since this direction is the stiffest dimension of the ruffle, users might want to edit the ruffle orientation for additional control over the object's local compliance. To achieve that, users can partition the mesh, which allows them to freely define one ruffle per partition. They can orient the individual cutting planes to define the stiffness directions as they wish. We illustrate in Fig. 8 how users partition the teddy into 6 parts. They orient the cutting plane for each part. Our system then creates default ruffles for each part.

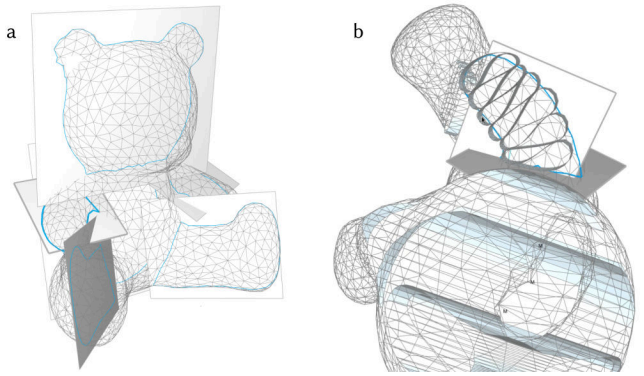

Figure 8: To control the extrusion direction individually, (a) users can create multiple cutting planes; here it's 6. (b) Default ruffles are created for each plane, we highlight the right arm in this example.

Users now have the opportunity to change the density of the individual ruffles. Fig. 9 shows how users create an elbow for the teddy. To increase the motion range of the elbow, they move the outer connection point inwards to allow for a hinging motion. Again, our algorithm automatically optimizes the ruffle to keep the volume filled. Upon exporting multi-ruffle objects, our algorithm creates additional strips that connect two model parts.

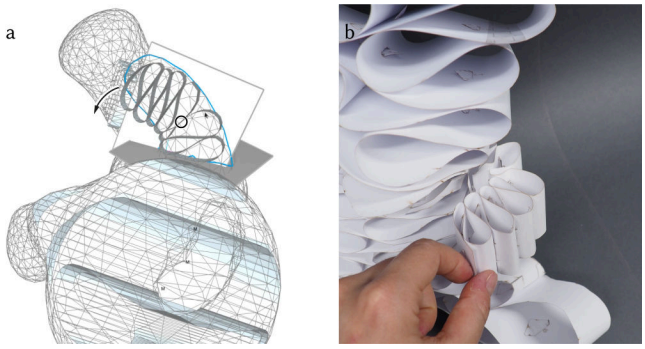

Figure 9: User can fine-tune the ruffles. (a) They move on connection point inward such that the ruffle can extend to act like an elbow, (b) which we demonstrate in the fabricated version.

4 IMPLEMENTATION

The core of our design tool is the automatic shape approximation with ruffles. We use an iterative optimization loop to compute the filling ruffle shape. In summary, our algorithm receives the outline from the user-defined plane cut, which is the 2D target shape to fill. After initializing a ruffle topology, our algorithm simulates the 2D ruffle curve to find the physically approximate shape. Then, our algorithm changes ruffle parameters while aiming to fill the target shape and simulates the altered ruffle. At every iteration, we compute the residual area, i.e., how much area is unfilled by the ruffle and how much does it protrude. When the target shape is filled, our optimization returns the 2D ruffle shape, which is extruded and intersected with the 3D mesh. We expand on the details of our algorithm in the following.

Notation. In this technical exposition, we use the following notation, which we illustrate in Fig. 10. We view the ruffle in two levels of detail. At the higher level, a ruffle consists of a set of *connection points* C and an ordered list of *sections* between them, each having a starting and an ending point, and a fixed length. The sections are laid out in the sequence in which they will appear in the final unfolded paper strip; as such, every section's ending point must be the starting point of its successor. The length of the strip is consequently the sum of the section lengths. Every connection point has at least three connecting sections, otherwise it can be collapsed. This level defines the topology of the ruffle, but not its shape.

At the lower level, every section is subdivided into *segments* of length approximately h (the resolution), with new interior *vertices* in addition to those stemming from the connection points. This results in a list of N vertices, each with a 2D position $\mathbf{x}_i$ and a width w_i. We can also view all the positions together as $\mathbf{x} = [\mathbf{x}_i]_{i=1}^{N} \in \mathbb{R}^{2N}$. Each segment i is a straight line segment between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ and has the (undeformed) length ℓ_i.

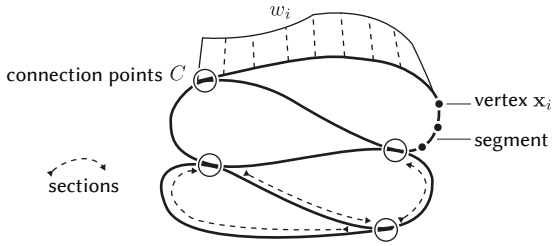

Figure 10: The ruffle notation and terminology. A ruffle has connection points, sections between them, which consist of segments. The extrusion width w_i at a vertex $\mathbf{x}_i$ is determined by intersecting with the input shape in 3D.

4.1 Ruffle simulation

A key component of our algorithm is a fast simulation of a thin-sheet, paper-like deformation. The simulation is performed iteratively to evaluate the current ruffle's fit to the user-defined target shape. General finite element analysis methods (e.g., on a tetrahedral mesh) are known for their accuracy, but are too slow to be utilized in iterative optimization methods. Especially on thin

structures such as paper, FEA would require a very large number of isotropic elements, or the use of anisotropic elements, which potentially give a low-quality result.

Instead, we make use of domain-specific information to achieve fast simulation results. Consequently, we build on thin shell theory. Specifically, we adapt the *discrete shells* model introduced in [11], which has been shown to be effective for similar problem definitions. This model simplifies the problem by applying the assumption of *thinness* (i.e., width $\gg$ thickness). Additionally, it operates on a mesh and separates membrane and flexure modes, which makes it easy to implement, computationally efficient and accurate even on coarse meshes.

In our specific application domain of creating smooth, thin ruffles, we build our simulation such that the following properties hold:

(1) The ruffle *deforms smoothly* at any given point.
(2) The ruffle *connections* meet in a smooth manner.
(3) The ruffle does not have *self-intersections*.

We detail the energies that we use in our simulation to find the vertex positions of our ruffle curve such that the aforementioned properties hold and all energies are in equilibrium.

4.1.1 Smooth ruffle deformation. We adapt the *discrete shells* model to fit a 2D line mesh, i.e., a polyline. It consists of two energies: (1) the flexure energy, which models how much the material resists bending and (2) the membrane energy, that models the stretch in the material due to the deformation.

In our implementation, the flexure energy applies at interior vertices where two segments meet. Our adapted flexure energy is described as the sum over interior vertices:

$$E_B(\mathbf{x}) = \sum_{i=2}^{N-1} (\theta_i - \pi)^2 w_i / \bar{\ell}_i,$$

where $\theta_i = \angle(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})$ is the interior angle, w_i is the strip width at $\mathbf{x}_i$, and $\bar{\ell}_i = (\ell_{i-1} + \ell_i)/2$ is the average of the adjacent segments' resting lengths.

The membrane energy applies within each ruffle segment. In our model, we only consider a single mode of intrinsic deformation, namely stretching along the strip. We omit modeling the shear stress on the tangent plane, in contrast to the original method, which is justified since our ruffles are only bent cylindrically. We simplify the membrane energy accordingly to

$$E_M(\mathbf{x}) = \sum_{i=1}^{N-1} (\|\mathbf{x}_{i+1} - \mathbf{x}_i\| - \ell_i)^2.$$

Note that while this term is originally used to model stretching materials, we weigh it with a large factor k_M. In our setting, the term more closely acts as a quadratic penalty term for the constraint $\|\mathbf{x}_{i+1} - \mathbf{x}_i\| = \ell_i$.

4.1.2 Connections along the ruffle. We model a connection point $(i, j) \in C$ (i.e. the ruffle should connect to itself at vertices i and j) by two constraints: (1) a positional constraint $\mathbf{x}_i = \mathbf{x}_j$, which we enforce by substituting $\mathbf{x}_i$ and $\mathbf{x}_j$ by a single variable, and (2) a tangential constraint $t_i = -t_j$, where t_i is the tangent direction at vertex $\mathbf{x}_i$, pointing in direction of increasing indices. The tangential constraints are pointing in different directions, since the sections run in opposite directions. We describe the tangential constraint with two additional flexural energy components, giving us the constraint energy

$$E_C(\mathbf{x}) = \sum_{(i,j)\in C} (\theta_{ij}^+ - \pi)^2 w_i / \bar{\ell}_{ij}^+ + (\theta_{ij}^- - \pi)^2 w_i / \bar{\ell}_{ij}^-,$$

where the connecting angles are given by $\theta_{ij}^{\pm} = \angle(\mathbf{x}_{i\pm1}, \mathbf{x}_i, \mathbf{x}_{j\pm1})$ and the lengths $\bar{\ell}_{ij}^{\pm}$ are computed correspondingly.

4.1.3 Preventing self-intersections. The ruffle may not interpenetrate itself. Instead, forces should be transferred across contact points. We base our collision system on *air meshes*[35], which are a triangulation of the empty space between objects. Air meshes prevent intersections by enforcing that triangle the areas stay positive, i.e., the triangles may not flip. Since rotations of objects *without collisions* can cause a twist in the air mesh, leading to very skinny triangles with a small area, the original method tests whether edge flips improve the overall triangle area. They model this as a quality score for each triangle.

While the original air mesh algorithm uses a bounding box for their triangulation, we need an *unbounded simulation area*. We achieve this by modifying the air mesh algorithm and adding a vertex at infinity, which is used for triangulation instead of the bounding box. These infinite triangles are assigned a constant quality score of q_∞ (a value of 0.05 worked well in our case) and effectively disregarded as constraints, but enable a valid triangulation of an unbounded area. We use our air mesh adaption in an energy minimization scheme for finding the resting configuration of the ruffle. Specifically, we use the following penalty method:

$$E_{AM}(\mathbf{x}) = \sum_{\triangle ABC} \max\{0, -A_{\triangle ABC}(\mathbf{x})\},$$

where $A_{\triangle ABC}(\mathbf{x})$ are the areas of the triangles created between the ruffle vertices.

4.1.4 Numerical optimization. To optimize the physically-based ruffle deformation, we combine the aforementioned energies and minimize the following final energy

$$E(\mathbf{x}) = k_M E_M(\mathbf{x}) + k_B(E_B(\mathbf{x}) + E_C(\mathbf{x})) + k_{AM} E_{AM}(\mathbf{x}).$$

This deformation energy is invariant to translations and rotations of the ruffle in the plane. To obtain a unique result for the static configuration $\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x})$ and make it physically correct, we add gravity, as well as boundary conditions. These boundary conditions include adding a floor at height $y = 0$ to make $\mathbf{x}^*$ unique up to translation in the x direction, and constraining the bottom-left vertex to a fixed position, effectively anchoring our ruffle.

To solve for $\mathbf{x}^*$, we use L-BFGS-B [6], a simple and efficient quasi-Newton method (i.e., it approximates the second derivative from instances of the gradient) that supports simple box constraints of the form $\mathbf{l} \le \mathbf{x} \le \mathbf{u}$ (where l_i and u_i can also be infinite to have open box constraints on x_i). We use these box constraints for the floor, effectively setting every second element of $\mathbf{l}$ to zero.

Note that while the above energy $E(\mathbf{x})$ is physically-based and solving the ordinary differential equation of motion $\ddot{\mathbf{x}} = -\mathbf{M}^{-1}\nabla E(\mathbf{x})$ (as in [11]) gives a realistic evolution of the ruffle (when paired with a damping term to dissipate energy), we are primarily interested in the resulting *static* resting configuration $\mathbf{x}^*$. Since in that configuration, all forces cancel out, i.e., $\nabla E(\mathbf{x}^*) = \mathbf{0}$, or similarly,

$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x})$, we reduce our simulation to a simple energy minimization.

4.2 Shape approximation

We informally view the task of shape approximation as a constrained optimization problem, where we maximize the total area within the ruffle, with the constraint of the ruffle being contained within the 2D target shape, i.e.

$$\xi^* = \arg\max_{\xi: R_\xi \subset T} A(R_\xi),$$

where R_ξ is the set of points contained within the ruffle given the parameters ξ (i.e., the section lengths) and T is the target shape. As we don't have a closed-form expression for the derivatives of the constraint function $A(R_\xi \setminus T) = 0$ or the objective function $A(R_\xi)$ with respect to the parameters ξ , conventional derivative-based optimization techniques are not applicable. Instead, we use a simple heuristic that exploits the geometric nature of the problem and gives good results in practice.

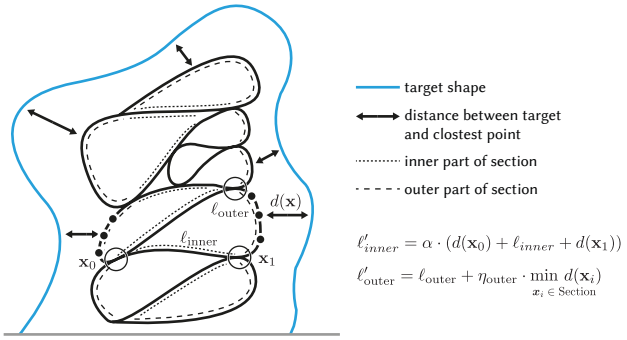

Figure 11: Overview of the notation and terminology of our heuristic shape approximation algorithm, showing instances of inner and outer sections.

We classify the sections into two classes, based on which we heuristically update the section's length:

(1) Sections that are part of the outline. We want these sections to touch the target shape boundary, as otherwise they can be lengthened, increasing the ruffle area. We define the signed distance field d to the target shape to be positive inside and negative outside, illustrated in Figure 11. The update rule for the length ℓ of an outline section is

$$\ell'_{\text{outer}} = \ell_{\text{outer}} + \eta_{\text{outer}} \cdot \min_{\mathbf{x}_i \in \text{section}} d(\mathbf{x}_i).$$

This can be considered as a step of the Newton-Raphson root-finding algorithm for the function $f = \min_{\mathbf{x}_i \in \text{section}} d(\mathbf{x}_i)$ with the assumption $f' = 1/\eta_{\text{outer}}$. Because lengthening a section by some $\Delta\ell$ can change its furthest horizontal position by at most $\frac{\Delta\ell}{2}$, we get the bound $f' < 0.5$. Consequently, values of η_{outer} below 2 are a good choice.

(2) Horizontal sections. We neither want these sections to be too long (which would place the connection points outside the target shape, making optimization of the outline sections' lengths impossible) nor too short (which would create more stress around the connection points) . Instead, we want the horizontal sections to span a fraction α of the target shape, which we approximate by using the signed distance at the section's endpoints $\mathbf{x}_0$ and $\mathbf{x}_1$:

$$\ell'_{\text{inner}} = \alpha \cdot (d(\mathbf{x}_0) + \ell_{\text{inner}} + d(\mathbf{x}_1))$$

Other sections (e.g. those created from a densify operation) are not considered by the algorithm, but their length still be manually controlled.

This iterative process usually converges after just a few steps.

4.3 Force-propagation of multiple ruffles.

For multi-part ruffles, the interactions between the different parts must be taken into account. In our system, only gravity forces can be transferred between ruffles, i.e., the forces that push up against each ground plane. To identify the correct locations to apply the transferred forces, we construct the *connectivity graph* over the components of the segmented target mesh. We then construct a forest, using components that rest on the ground as root nodes. Within each tree, forces are transferred from children to parents, starting from the leaves. The forces are applied to the closest vertex of the parent's ruffle to the separation line to the child.

4.4 System implementation.

We implemented our system in C++, using libigl [21] for geometry processing and the visual interface. Our system also utilizes Eigen [14] for general linear algebra, and CGAL [56] for the constrained Delaunay triangulations needed for the air meshes. For both packages, we use the versions delivered with libigl. We make our C++ reference implementation available at https://github.com/mad-s/ruffles-editor/. The time taken to solve the static configuration is dependent on the dissimilarity between the solution $\mathbf{x}^*$ and the initial value (being the state before the editing operation). Most operations (changing lengths, heuristic step, densify) complete in under 100 ms (measured on a 3.6 GHz AMD Ryzen 2500U Laptop with 16 GB RAM, with a ruffle consisting of around 300 vertices)

5 APPLICATIONS

We sample different application areas, where our ruffled metamaterials could have potential use. Here, we showcase example applications to illustrate the properties enabled by our structures.

Teddy toy with custom stiffness. Our ruffled, developable metamaterial allows users to create their own toys with their desired compliance features. We have previously shown the teddy bear example. For more freedom in compliance directions, users create the teddy from 6 ruffles. Additionally, we believe that such easy to fabricate metamaterials might be of interest to product designers of soft and compliant products. Such tactile qualities need to be felt early and often [19, 44], which our metamaterials support by enabling rapid prototyping. Should designers wish to create a product-version of our example teddy, we recommend a ruffled skeleton made of plastics, which can be covered with a plush hull where details such as eyes can be attached.

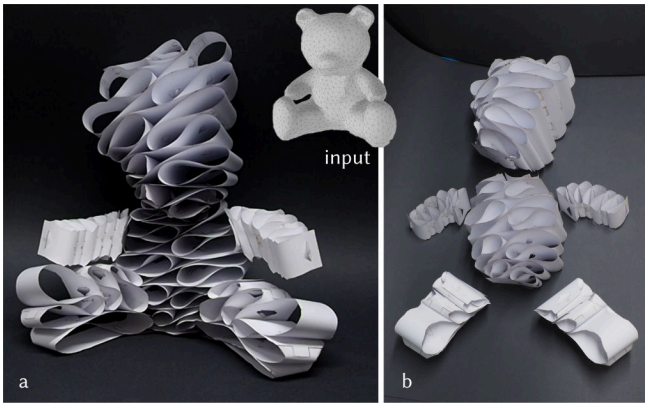

Figure 12: Our metamaterials are applicable for custom toy design, such as (a) a teddy bear. (b) For more control, it can be decomposed into multiple ruffles.

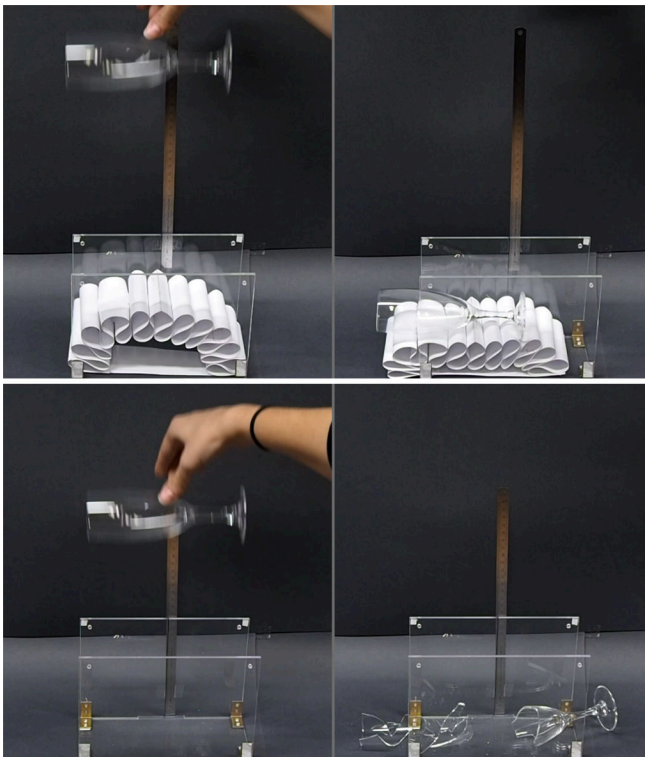

Figure 13: Our materials can be used for custom protective materials. We show their effectiveness by dropping a glass.

Damping material for packaging. Our metamaterials can also be of use when creating custom packaging which, in contrast to traditional wrinkled paper packaging materials, can be computationally tailored to the shape and mass distribution of the objects to protect, making it particularly suited to protect, e.g., art pieces. We demonstrate the potential at a simple glass drop test in Fig. 13. The glass is protected by our ruffled material, while it breaks when falling on the table. The falling height is the same.

Formwork for architectural shells. Our ruffles might also be of interest to architects. Developable surfaces are well known in architecture due to them being more efficient to form compared to doubly curved surfaces. Fig. 14 demonstrates at a miniature example how our ruffled metamaterial can be used as *formwork* for shell structures. For maximum stability, we use the extrusion direction of our ruffles to take the most load. We cover it with a plastic sheet, which could be tarp or similar in the large-scale case, and apply plaster. After hardening, we can safely remove the supporting ruffles. Note that our ruffles are most suitable for large-scale, smooth structures, rather than for sharp details. While the ruffle density can be locally increased to capture finer details, we believe that such features (e.g., stairs) would be better captured by traditional methods.

Figure 14: Our ruffles could be used as formwork for architectural shells. We show a miniature version of the (a) supporting ruffles, (b) on which we apply plaster, and (c) after removing the ruffles, we (d) obtain the original 3D shell.

Lamp shade. Lastly, we want to showcase how such ruffled materials can be used for aesthetics. We show the example of a lamp shade in Fig. 15. The density of the ruffles locally changes the permeability of the lamp.

6 TECHNICAL EVALUATION

To assess the variable compliance of our ruffle design, we sample the deformation of our ruffled material across different load cases.

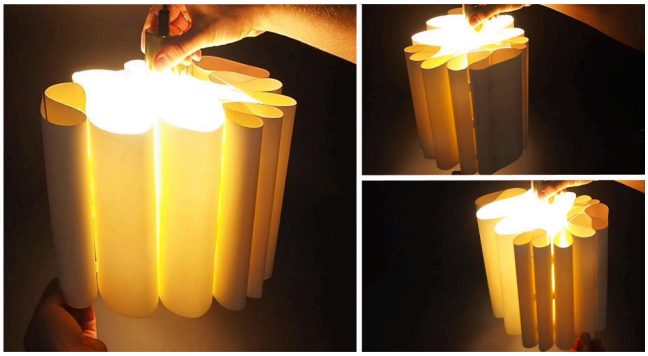

Figure 15: We create a ruffled lamp shade, with the ruffles controlling its local permeability.

We also verified the influence of the cut connector tabs on the compliance of the ruffle, detailed in the following.

6.1 Load tests

The ruffle design (i. e., the topology and section lengths) changes the compliance of the resulting materials. To assess the effect of ruffle design on material, we design our experiment using the following independent variables:

- Ruffle design: *default* ruffle stack, a *wide*, and a *dense* stack. The extrusion width of all ruffles is 40 mm.
- Material: *flipchart paper* at 80 g/m^2, *office paper* at 160 g/m^2, and thin *plastic* sheets (i.e., overhead projector transparencies).
- Load direction: we measure two stiffness directions, i. e., along the *X* and *Y* axis. We omit measuring the extrusion axis Z, since the compliance in that direction is governed entirely by the base material's strength. The max. load would be similar to e. g., a honeycomb structure with similar density.
- Load: we apply up to *7 loads*, ranging between 10 g and 200 g.

For reproducibility and better interpretability of the results, we provide the dimensions and masses of all our load test samples in Table 1.

Shape	W × H [mm]	Paper (160g)	Plastic	Paper (80g)
default	80 × 486	6g	5g	3g
wide	80 × 749	10g	9g	5g
dense	80 × 797	10g	9g	5g

Table 1: Specifications of all the load test samples.

We measure the vertical deformation, as shown in Fig. 16a, by measuring the rest length of the ruffles and its compressed height for each load condition. For each condition, we first measure the ruffle's rest height in mm, as illustrate in Fig. 16b. Since we wish to distribute load across the ruffle, we place an acrylic plate (4 mm, 114 g) on it. This acrylic plate adds a load to each ruffle, introducing a constant load offset which, for simplicity, we mark as the "0 grams" load case for each condition. We then place each load and mark the resulting ruffle height. All loads are placed subsequently without removing the acrylic plate. Lastly, we mark the ruffle height after removing the last load and the acrylic plate, which indicates plastic deformation that occurred during our experiment. We perform the same procedure for the remaining *material* and *load direction* conditions.

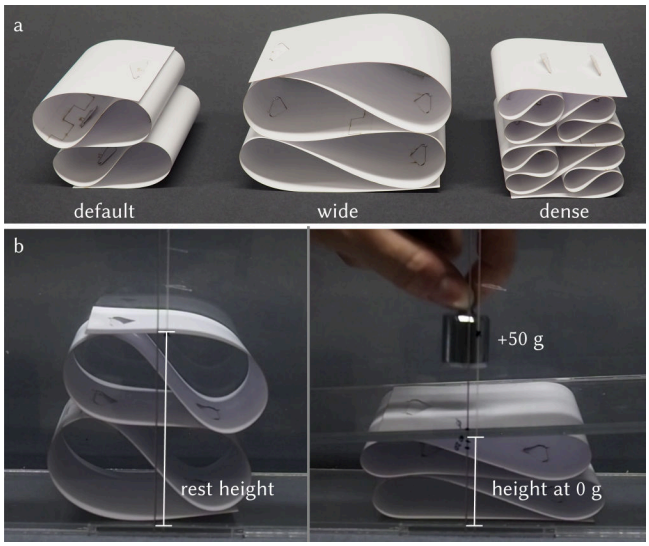

Figure 16: We use (a) three different types of ruffle stacks in our experiment. (b) The rest height of a stack is defined by its natural dimensions. (c) The height a 0 g is defined by the compression that is introduced by the acrylic plate which we place on the ruffle stack for even load distribution.

Results. We show the raw results in Fig. 17 to serve as a reference. We intend to provide these data as a reference for reproducibility. Since these measurements were not performed on an industrial-grade measurement setup, inaccuracies might occur.

We first compare the *Y* load direction across materials. For the 160 g/m^2 paper, the *default Y* and *wide Y* ruffle yield roughly similar compression at 200 g (~70% compression). Between the two types of ruffle stacks, the absolute compression is comparable across the different weights. The *dense Y* ruffle stack, on the other hand, yields only roughly 25% compression at 200 g, with a lower compression ratio. The plastic yields comparable results: at 200 g, the *default Y* and *wide Y* ruffle stacks yield large compression ratios of 65% and 52%, respectively; the *dense Y* ruffle stack yields only 35% compression. We furthermore observe a similar behavior for 80 g/m^2 paper for *dense Y*, which can also be seen in Figure 18, displaying the compression rate per gram of weight, split by the individual weights. We calculate this value by taking the current compression ratio (for each weight and material), and normalize it to the 0 g weight and the weight currently applied. This similar behavior across the three materials indicates that the density of the ruffles is the main governing factor for the compliance of the ruffles.

We observed similar, but less pronounced compression effects for our tests along the *X* axis. We believe this is due to the different connection placement, resulting in a more limited motion range of

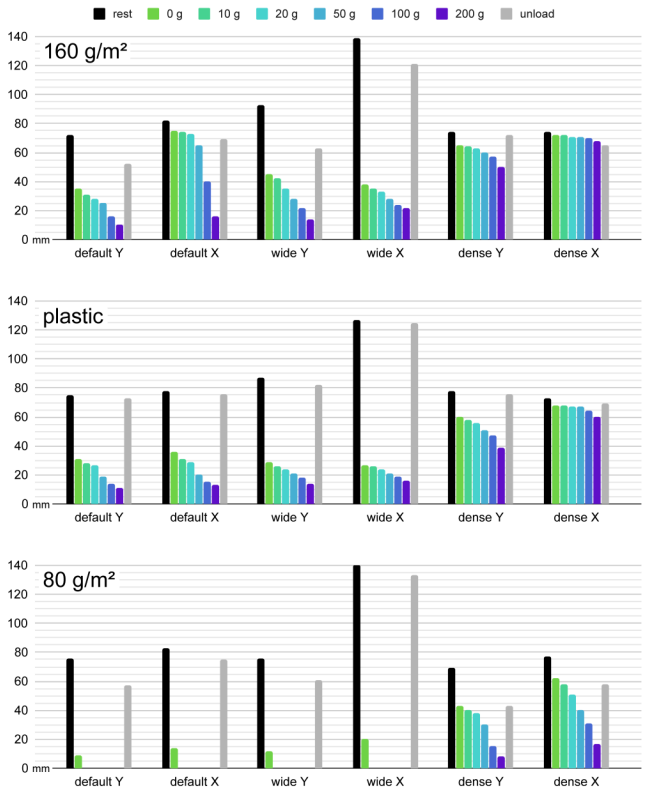

Figure 17: The raw result of our load tests. The Y axis shows the absolute height of the ruffles in their corresponding load conditions.

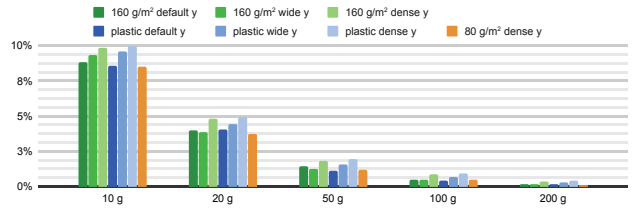

Figure 18: Compression ratio per gram weight, normalized to the 0 g compression ratio. These relative ratios are similar across different materials and ruffles, although the overall compression differs. This shows that the ruffle density is the main governing factor for compression.

the ruffles. Furthermore, due to the different load distribution, other effects such as buckling come into play. This can be seen in Fig. 19, as well as the data of *default X* ruffle from 160 g/m^2 paper. Here we see a sudden increase in compression (i. e., drop in height) between 50 g and 100 g, which can be attributed to this effect. Taking such properties into account in simulation is highly challenging and will be subject of our future work. Lastly, the data shows that for the default and wide ruffles created with the 80 g/m^2 paper, any weight larger than 10g makes the ruffle compress completely, which can be seen in Fig. 17. This emphasizes the need for careful choice of material when creating developable metamaterials.

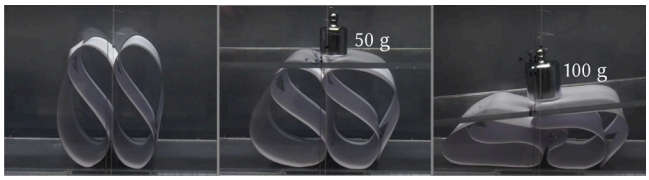

Figure 19: We show the conditions 160 g/m^2 paper, default ruffle, loading in X direction, with loads 50 g and 100 g. The data shows a large non-linear drop in this condition, which is likely due to the buckling of the long ruffles that suddenly collapse.

While we report absolute loads and sizes, note that viewing the loads relative to the ruffle samples' own weights informs about their properties more generally. As the loads range up to 314 g (200 g load including the plate), our results demonstrate how the evaluated ruffle stacks withstand up to ~60× their own mass (5 g plastic, cf. Table 1). This indicates a very good relative strength of our ruffles. Besides resistance to compression, shape retention is also an important measure for many applications. Here, the choice of the material is important, with the plastic sheet performing much better than the paper (97% vs 83% average shape retention ratio across the samples). Considering the small linear elastic region of paper (i. e., paper quickly goes into plastic deformation), we believe the respectable shape retention even with paper speaks for the usefulness of our ruffled metamaterial in this regard.

6.2 Influence of the interlocking tabs

Cutting the interlocking tabs directly into the ruffle simplifies the assembly process, yet they also reduce the stiffness of the sheet material locally. To estimate their influence on the ruffle stiffness, we compared them to a ruffle assembled with double-sided tape. We use the *default* ruffle design from our previous experiment, yet we only mark the tab locations on the strip for the tape. We use only the 160 g/m^2 paper, and perform all *load* conditions as in our aforementioned procedure.

The results of this experiment are plotted in Fig. 20, which shows that the influence of the laser cut connector is small. It seems to be a constant offset (here ~5 mm). The results also show how the tape adds mass to the ruffle as it is shorter in its rest position compared to the ruffle with the cut connectors.

6.3 Simulation

We qualitatively compare the results from our load test with the simulation. Figure 21 shows one example. The simulation achieves visually similar behavior, which we believe is sufficient for our current system. Further quantification, however, is needed in order to verify the simulation, as well as understand the impact of simulation accuracy on user behavior.

7 DISCUSSION & FUTURE WORK

Our ruffles are easy to fabricate and assemble, which is key to our goal towards mass-fabricable metamaterials. In this spirit, we made

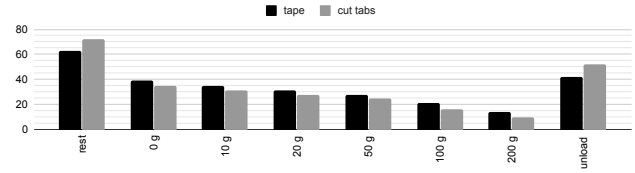

Figure 20: The laser cut connector tabs slightly weaken the ruffle, as it deforms more under load compared to a ruffle assembled with tape. The tape also adds mass, as is evident from the difference in rest height.

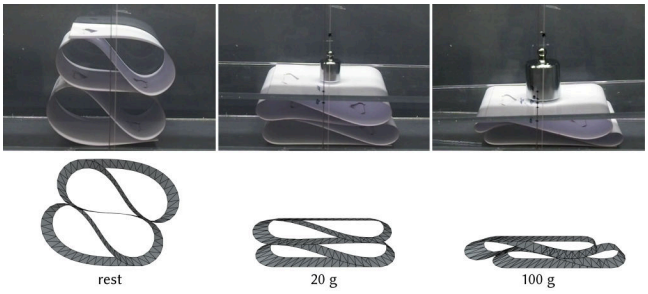

Figure 21: Qualitative comparison between the load test and our simulation, in this case with the 160 g/m^2 paper.

some design decisions with future extensions in mind, which we briefly outline in the following.

Large-scale fabrication. We kept the connections of our ruffles intentionally simple by embedding the connector in the strip. This should enable their industrial fabrication on plotters. While in our prototyping setup, our strip lengths were constrained by the laser cutter bed size, industrial plotters can cut long materials from a roll. The assembly could be achieved by two robotic arms that slide the connectors into their corresponding slit in one process. This would also enable processing materials such as metals.

Impact of material selection. While we used paper for most prototypes due to its availability and familiarity, our ruffled metamaterial structure and design tool are intended for a wide range of sheet material. Besides paper and plastic sheets, rubber sheets, felt and sheet metal are also suitable for ruffles. In general, the materials used should be elastic and not brittle, to have good shape retention. The stiffness of the material influences the size of the objects, their density and the permissible load, with softer materials allowing for higher density and more detail, whereas stiffer materials enable bigger objects and larger loads. While the simulation algorithm for thin shells is designed around *thin* materials (i. e., thickness < width/100), we expect our tool to also work for materials such as felt based on our experience (though we do acknowledge we did not perform a formal evaluation). The connectors would be modified to better fit the material, such as adjusting the embedded interlocking connector tabs to accommodate for extra thickness, or using completely different fabrication techniques, such as spot-welding for sheet metal.

Inverse design of ruffle layout. In this paper, we leave it up to the user to decide the density of their ruffle design, i.e., the ruffle topology. In the future, we plan to extend our optimization routine, such that user can specify load cases and our algorithm computes the topology that withstand the load distribution subject to the user-specified material. This would be part of a wider effort to make our software more accessible to novice users, by creating tools to manipulate the ruffles on a higher level. Furthermore, we plan investigating and simulating dynamic properties of our ruffles, such as plasticity, damping, etc.

Ruffles along 3D curves. This is the most motivating opportunity for future work. We plan to extend our system to create ruffles along spatial curves in 3D. This will allow for more control over the isotropy and enable making complex shapes (e.g., the teddy) from one strip. To achieve this, we need to enforce developability of our strips in more at every iteration. This is a non-trivial task. The design space of such spatially ruffled metamaterials require more investigation, yet are an exciting avenue.

8 CONCLUSIONS

We presented a novel metamaterial design that leverages laser cutting for fast fabrication. Our metamaterials allow users to define custom compliance across a 3D shape. We contribute a design tool that optimizes the ruffles to fill a user-defined shape and exports them flat, labeled, with embedded interlocking connector tabs ready for fabrication. The resulting ribbons do not require external materials and are thus easy to assemble. This work is part of a larger goal to make metamaterials faster, thus mass-fabricable, to increase their real world impact.

ACKNOWLEDGMENTS

We thank Philipp Herholz for his helpful advice about the simulation and David Lindlbauer for providing feedback on the paper and for his help with the figures and video. This work was in part supported by the Swiss National Science Foundation (NCCR Digital Fabrication Agreement #51NF40-182887).

REFERENCES

[1] Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen. 2019. Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300796

[2] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin van Hecke. 2017. Flexible mechanical metamaterials. *Nature Reviews Materials* 2, 11 (oct 2017), 17066. https://doi.org/10.1038/natrevmats.2017.66

[3] Dustin Beyer, Serafima Gurevich, Stefanie Mueller, Hsiang-Ting Chen, and Patrick Baudisch. 2015. Platener: Low-Fidelity Fabrication of 3D Objects by Substituting 3D Print with Laser-Cut Plates. 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, 1799–1806. https://doi.org/10.1145/2702123.2702225

[4] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and Fabrication of Materials with Desired Deformation Behavior. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 63, 10 pages. https://doi.org/10.1145/1833349.1778800

[5] Tiemo Bückmann, Michael Thiel, Muamer Kadic, Robert Schittny, and Martin Wegener. 2014. An elasto-mechanical unfeelability cloak made of pentamode metamaterials. *Nature Communications* 5, May (2014), 1–6. https://doi.org/10.1038/ncomms5130

[6] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing* 16, 5 (1995), 1190–1208.

[7] Xiang 'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2016. Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 29–39. https://doi.org/10.1145/2984511.2984512

[8] Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2018. Forte: User-Driven Generative Design. 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. https://doi.org/10.1145/3173574.3174070

[9] Philippe Destuynder. 1985. A classification of thin shell theories. *Acta Applicandae Mathematica* 4, 1 (1985), 15–63.

[10] Tobias Frenzel, Claudio Findeisen, Muamer Kadic, Peter Gumbsch, and Martin Wegener. 2016. Tailored Buckling Microlattices as Reusable Light-Weight Shock Absorbers. *Advanced Materials* 28, 28 (jul 2016), 5865–5870. https://doi.org/10.1002/adma.201600610

[11] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete Shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, 62–67.

[12] Daniel Groeger and Jürgen Steimle. 2019. LASEC: Instant Fabrication of Stretchable Circuits Using a Laser Cutter. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300929

[13] Jianzhe Gu, David E. Breen, Jenny Hu, Lifeng Zhu, Ye Tao, Tyson Van de Zande, Guanyun Wang, Yongjie Jessica Zhang, and Lining Yao. 2019. Geodesy: Self-Rising 2.5D Tiles by Printing along 2D Geodesic Closed Path. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3290605.3300267

[14] Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

[15] Ruslan Guseinov, Eder Miguel, and Bernd Bickel. 2017. CurveUps: Shaping Objects from Flat Plates with Tension-Actuated Curvature. *ACM Trans. Graph.* 36, 4, Article 64 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073709

[16] Felix Heibeck, Basheer Tome, Clark Della Silva, and Hiroshi Ishii. 2015. UniMorph: Fabricating Thin Film Composites for Shape-Changing Interfaces. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 233–242. https://doi.org/10.1145/2807442.2807472

[17] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. 2012. Crdbrd: Shape Fabrication by Sliding Planar Slices. *Comput. Graph. Forum* 31, 2pt3 (May 2012), 583–592. https://doi.org/10.1111/j.1467-8659.2012.03037.x

[18] Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. Metamaterial Mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 529–539. https://doi.org/10.1145/2984511.2984540

[19] Alexandra Ion, Robert Kovacs, Oliver S. Schneider, Pedro Lopes, and Patrick Baudisch. 2018. *Metamaterial Textures*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173910

[20] Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. 2019. Understanding Metamaterial Mechanisms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300877

[21] Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

[22] Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. 2016. Beyond Developable: Computational Design and Fabrication with Auxetic Materials. *ACM Transactions on Graphics* 35, 4 (jul 2016), 1–11. https://doi.org/10.1145/2897824.2925944

[23] Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. 2017. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2606–2616. https://doi.org/10.1145/3025453.3026016

[24] Snežana Lawrence. 2011. Developable Surfaces: Their History and Application. *Nexus Network Journal* 13, 3 (2011), 701–714. https://doi.org/10.1007/s00004-011-0087-z

[25] Danny Leen, Nadya Peek, and Raf Ramakers. 2020. LamiFold: Fabricating Objects with Integrated Mechanisms Using a Laser Cutter Lamination Workflow. 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, 304–316. https://doi.org/10.1145/3379337.3415885

[26] Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. 2014. Build-to-Last: Strength to Weight 3D Printed Objects. *ACM Trans. Graph.* 33, 4, Article 97 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601168

[27] Jonàs Martínez, Mélina Skouras, Christian Schumacher, Samuel Hornus, Sylvain Lefebvre, and Bernhard Thomaszewski. 2019. Star-Shaped Metrics for Mechanical Metamaterial Design. *ACM Trans. Graph.* 38, 4, Article 82 (July 2019), 13 pages. https://doi.org/10.1145/3306346.3322989

[28] Jonàs Martínez, Haichuan Song, Jérémie Dumas, and Sylvain Lefebvre. 2017. Orthotropic k-Nearest Foams for Additive Manufacturing. *ACM Trans. Graph.* 36, 4, Article 121 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073638

[29] Jonàs Martínez, Haichuan Song, Jérémie Dumas, and Sylvain Lefebvre. 2017. Orthotropic k-Nearest Foams for Additive Manufacturing. *ACM Trans. Graph.* 36, 4, Article 121 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073638

[30] James McCrae, Nobuyuki Umetani, and Karan Singh. 2014. FlatFitFab: Interactive Modeling with Planar Sections. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*. ACM Press, New York, New York, USA, 13–22. https://doi.org/10.1145/2642918.2647388

[31] Jun Mitani and Hiromasa Suzuki. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. 23, 3 (2004), 259–263.

[32] Koryo Miura. 1985. *Method of packaging and deployment of large membranes in space*. Technical Report. 1–9 pages. https://repository.exst.jaxa.jp/dspace/handle/a-is/7293

[33] 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. https://doi.org/10.1145/2470654.2481358

[34] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. Interactive Construction: Interactive Fabrication of Functional Mechanical Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 599–606. https://doi.org/10.1145/2380116.2380191

[35] Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air meshes for robust collision handling. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

[36] Jifei Ou, Zhao Ma, Jannik Peters, Sen Dai, Nikolaos Vlavianos, and Hiroshi Ishii. 2018. KinetiX - designing auxetic-inspired deformable material structures. *Computers & Graphics* 75 (2018), 72 – 81. https://doi.org/10.1016/j.cag.2018.06.003

[37] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Trans. Graph.* 34, 4, Article 135 (July 2015), 12 pages. https://doi.org/10.1145/2766937

[38] Ahmad Rafsanjani and Katia Bertoldi. 2017. Buckling-Induced Kirigami. *Physical Review Letters* 118, 8 (2017), 1–5. https://doi.org/10.1103/PhysRevLett.118.084301 arXiv:1702.06470

[39] Ahmad Rafsanjani and Damiano Pasini. 2016. Bistable auxetic mechanical metamaterials inspired by ancient geometric motifs. *Extreme Mechanics Letters* 9 (2016), 291–296. https://doi.org/10.1016/j.eml.2016.09.001 arXiv:1612.05988

[40] Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2016. RetroFab: A Design Tool for Retrofitting Physical Interfaces Using Actuators, Sensors and 3D Printing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 409–419. https://doi.org/10.1145/2858036.2858485

[41] Thijs Roumen, Jotaro Shigeyama, Julius Cosmo Romeo Rudolph, Felix Grzelka, and Patrick Baudisch. 2019. SpringFit: Joints and Mounts That Fabricate on Any Laser Cutter. 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, 727–738. https://doi.org/10.1145/3332165.3347930

[42] Krishna Kumar Saxena, Raj Das, and Emilio P. Calius. 2016. Three Decades of Auxetics Research – Materials with Negative Poisson's Ratio: A Review. *Advanced Engineering Materials* 18, 11 (nov 2016), 1847–1870. https://doi.org/10.1002/adem.201600053

[43] Tobias A Schaedler, Alan J Jacobsen, Anna Torrents, Adam E Sorensen, Jie Lian, Julia R Greer, Lorenzo Valdevit, and Wiliam B Carter. 2011. Ultralight metallic microlattices. *Science* 334, 6058 (2011), 962–965.

[44] Oliver Schneider, Karon MacLean, Colin Swindells, and Kellogg Booth. 2017. Haptic experience design: What hapticians do and where they need help. *International Journal of Human-Computer Studies* 107 (2017), 5–21. https://doi.org/10.1016/j.ijhcs.2017.04.004 Multisensory Human-Computer Interaction.

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

[46] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to Control Elasticity in 3D Printing. *ACM Trans. Graph.* 34, 4, Article 136 (July 2015), 13 pages. https://doi.org/10.1145/2766926

[47] Christian Schumacher, Steve Marschner, Markus Gross, and Bernhard Thomaszewski. 2018. Mechanical Characterization of Structured Sheet Materials. *ACM Trans. Graph.* 37, 4, Article 148 (July 2018), 15 pages. https://doi.org/10.1145/3197517.3201278

[48] Sicong Shan, Sung H. Kang, Jordan R. Raney, Pai Wang, Lichen Fang, Francisco Candido, Jennifer A. Lewis, and Katia Bertoldi. 2015. Multistable Architected Materials for Trapping Elastic Strain Energy. *Advanced Materials* 27, 29 (2015), 4296–4301. https://doi.org/10.1002/adma.201501708 arXiv:1207.1956

[49] Oded Stein, Eitan Grinspun, and Keenan Crane. 2018. Developability of triangle meshes. 37, 4 (2018).

[50] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. In *Proc. SIGGRAPH 1987*. 205–214.

[51] Skylar Tibbits. 2012. From Digital Materials to Self-Assembly. *Assembly Automation* 32 (2012), 216–225. http://www.nature.com/doifinder/10.1038/srep07422

[52] Udayan Umapathi, Hsiang-Ting Chen, Stefanie Mueller, Ludwig Wall, Anna Seufert, and Patrick Baudisch. 2015. LaserStacker: Fabricating 3D Objects by Laser Cutting and Welding. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 575–582. https://doi.org/10.1145/2807442.2807512

[53] Charlie CL Wang and Kai Tang. 2004. Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches. *The Visual Computer* 20, 8-9 (2004), 521–539.

[54] Guanyun Wang, Humphrey Yang, Zeyu Yan, Nurcan Gecer Ulu, Ye Tao, Jianzhe Gu, Levent Burak Kara, and Lining Yao. 2018. 4DMesh: 4D Printing Morphing Non-Developable Mesh Surfaces. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 623–635. https://doi.org/10.1145/3242587.3242625

[55] Junichi Yamaoka, Mustafa Doga Dogan, Katarina Bulovic, Kazuya Saito, Yoshihiro Kawahara, Yasuaki Kakehi, and Stefanie Mueller. 2019. FoldTronics: Creating 3D Objects with Integrated Electronics Using Foldable Honeycomb Structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300858

[56] Mariette Yvinec. 2020. 2D Triangulation. In *CGAL User and Reference Manual* (5.1 ed.). CGAL Editorial Board. https://doc.cgal.org/5.1/Manual/packages.html#PkgTriangulation2