FlexMaps: Computational Design of Flat Flexible Shells for Shaping 3D Objects

LUIGI MALOMO*, ISTI - CNR
JESÚS PÉREZ*, IST Austria
EMMANUEL IARUSSI, CONICET and IST Austria
NICO PIETRONI, University of Technology Sidney and ISTI - CNR
EDER MIGUEL, Universidad Rey Juan Carlos
PAOLO CIGNONI, ISTI - CNR
BERND BICKEL, IST Austria

(a) (b) (c)

Fig. 1. Given an input quad mesh, the FlexMaps framework first generates a coarse parametrized patch layout (a). Then, an optimization is performed to adjust the mechanical properties of a series of embedded spiral microstructures. Once the patches are assembled, they assume the specified target shape (b). The designed spiral layout can be fabricated flat by laser cutting and easily assembled using a specifically crafted set of connectors (c).

We propose FlexMaps, a novel framework for fabricating smooth shapes out of flat, flexible panels with tailored mechanical properties. We start by mapping the 3D surface onto a 2D domain as in traditional UV mapping to design a set of deformable flat panels called *FlexMaps*. For these panels, we design and obtain specific mechanical properties such that, once they are assembled, the static equilibrium configuration matches the desired 3D shape. FlexMaps can be fabricated from an almost rigid material, such as wood or plastic, and are made flexible in a controlled way by using computationally designed spiraling microstructures.

CCS Concepts: • **Computing methodologies** → *Shape modeling*;

Additional Key Words and Phrases: computational fabrication, shape abstraction

ACM Reference Format:
Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. 2018. FlexMaps: Computational Design of Flat Flexible Shells for Shaping 3D Objects. *ACM Trans. Graph.* 37, 6, Article 241 (November 2018), 14 pages. https://doi.org/10.1145/3272127.3275076

1 INTRODUCTION

A wide range of intriguing 3D objects can be designed and composed of planar sheets of materials. Common examples are paper craft models [Mitani and Suzuki 2004], clothing [Bartle et al. 2016], or complex developable shapes in architecture, such as iconic buildings by Frank Gehry [Pottmann et al. 2015]. In all these cases, a non-trivial relationship exists between the 2D configuration and the resulting 3D shape. A major challenge in computational fabrication

---

*L. Malomo and J. Pérez are joint first authors.

Authors' addresses: Luigi Malomo, ISTI - CNR, luigi.malomo@isti.cnr.it; Jesús Pérez, IST Austria; Emmanuel Iarussi, CONICET, IST Austria; Nico Pietroni, University of Technology Sidney, ISTI - CNR; Eder Miguel, Universidad Rey Juan Carlos; Paolo Cignoni, ISTI - CNR; Bernd Bickel, IST Austria.

0730-0301/2018/11-ART241 $15.00
https://doi.org/10.1145/3272127.3275076

is modeling this relationship and providing intuitive tools for design. Although significant progress has been achieved in important classes of materials and fabrication techniques, such as designing folding patterns, developable surfaces, and garment patterning, designs with elastic sheets that can sustain significant stretching and bending have received little attention. This class of materials offers interesting possibilities because, thanks to their flexibility, these materials can go beyond the restrictions of developability and can be used to assemble self-supporting 3D shapes.

In this paper, we propose a novel framework to build 3D shapes out of flat, flexible panels. Taking inspiration from UV mapping, we map a 3D surface onto a 2D domain to design a set of panels with tailored mechanical properties. Once these panels are assembled, their static equilibrium configuration matches the desired 3D shape. We refer to these flexible 2D panels as *FlexMaps*. FlexMaps can be fabricated with almost rigid materials, such as wood or plastic, and are made flexible by engraving spiraling microstructures in them.

Inspired by recent works on volumetric microstructures, we control elasticity locally by continuously varying the shape of spiral structures to obtain different bending and stretching properties. Given some input target shape, we formulate an optimization that efficiently determines the heterogeneous distribution of spirals by interpolating patterns from a pre-computed database. As the technical core of our method, we introduce a data-driven two-scale model that can approximate the essential physical behavior of our structures. At the fine level, we parametrically generate an accurate geometric model of a single spiral structure and simulate its behavior using a non-linear FEM with thousands of degrees of freedom. At the coarse level, we use these simulation data to parametrize a simple rod mesh model that captures the main deformation modes and can be efficiently integrated into a sensitivity analysis for global shape optimization.

As demonstrated by our results, our approach allows the design of intricate shapes such as the Stanford bunny. Our 2D microstructures can be easily fabricated by additive or subtractive techniques through 3D printing, milling, or laser cutting. We show shapes manufactured from various materials and sizes, and we demonstrate that large objects can be economically fabricated because the production cost scales linearly with respect to the surface area of the object.

Our method allows for an efficient design, fabrication and assembly of doubly curved surfaces. It significantly differs from other recent works that depend on pre-stress [Guseinov et al. 2017; Pérez et al. 2017], inflation or gravitational loading [Konaković et al. 2018], or plasticity and physical reference models [Konaković et al. 2016]. Our structures lay flat in rest configuration and their final shape is just the result of internal elastic forces that arise when connecting multiple patches together. We believe FlexMaps' simplicity opens the possibility for its direct use in a wide range of application areas, such as industrial or architectural design.

2 RELATED WORK

Fabrication-aware shape approximation. Fabrication-aware computational design allows the transformation of virtual geometric shapes into tangible objects. Our work falls into the category of *stylized fabrication*, with the aim of reproducing a stylized approximation of a given input shape. A comprehensive overview of such methods can be found in the works of Bermano et al. [2017] and Bickel et al. [2017]. Aesthetically, our resulting shapes resemble ornamental curve networks. However, the primary goal of our method is not aesthetic control over individual curves [Zehnder et al. 2016], but achieving the required mechanical properties such that the initially flat patches approximate the desired shape once assembled. In the following, we will first focus on methods for reproducing shapes from initially flat primitives.

Based on the restriction that paper is (almost) inextensible but can be easily folded, bent, and/or cut, several methods have been developed for approximating shapes with papercraft [Kilian et al. 2008, 2017; Massarwi et al. 2007; Mitani and Suzuki 2004; Shatz et al. 2006; Takezawa et al. 2016]. Similarly to our approach, these methods require a good parametrization; however, in contrast to our stretchable patches, paper is almost inextensible and therefore requires low or even no distortion in the mapping process. Because of the discrete nature of folds and cuts, the resulting objects commonly exhibit artifacts in the case of smooth target surfaces. Dudte et al. [2016] introduced a set of geometric construction rules and a constrained optimization algorithm to determine spatially modulated origami patterns that drape complex surfaces. The authors also investigate the accuracy at which the pattern conforms to the target surface, as well as the effort associated with creating finer folds. Not allowing stretch but shear, Garg et al. [2014] model the behavior of interwoven, inextensible flat wire sheets using Chebyshev nets.

Relaxing the inextensibility constraint, Konaković et al. [2016] rationalized surfaces via auxetic materials, i.e., flat materials that can isotropically stretch up to a certain extent, by introducing cuts. Inspired by this approach, we extend the concept of using cuts to control physical behavior, such as stretching, bending, and twisting. Our method differs in several fundamental concepts. Most importantly, while theirs is purely geometric, our considers the constitutive relation between deformation and forces. Furthermore, to realize the shape, their method relies on a physical 3D reference model for forming, as well as material plasticity, once the reference model is removed. In contrast, our shapes are automatically expressed by the computationally designed elastic properties of the surface, and the desired shape is reached just by static equilibrium. Another class of methods assembles shapes out of interlocking planar elements. Similar to our method, many of them rely on a fabrication process that can be easily performed using laser cutting. While some of the methods in this class focus on how to distribute planar elements within the object's volume, thus neglecting the external surface [Cignoni et al. 2014; Hildebrand et al. 2012; McCrae et al. 2011; Schwartzburg and Pauly 2013], others approximate the surface using planar interconnected elements [Chen et al. 2013; Skouras et al. 2015]. Alternatively, Miguel et al. [2016] employ in-plane bent wires as elements which are interlocked in a configuration of static equilibrium. These methods are used to approximate a limited set of geometries only, and unlike our method, the resulting objects cannot exhibit smooth features.

Parametrization methods. Field-aligned parametrization techniques [Alliez et al. 2003; Bommes et al. 2013, 2009; Kälberer et al. 2007] use

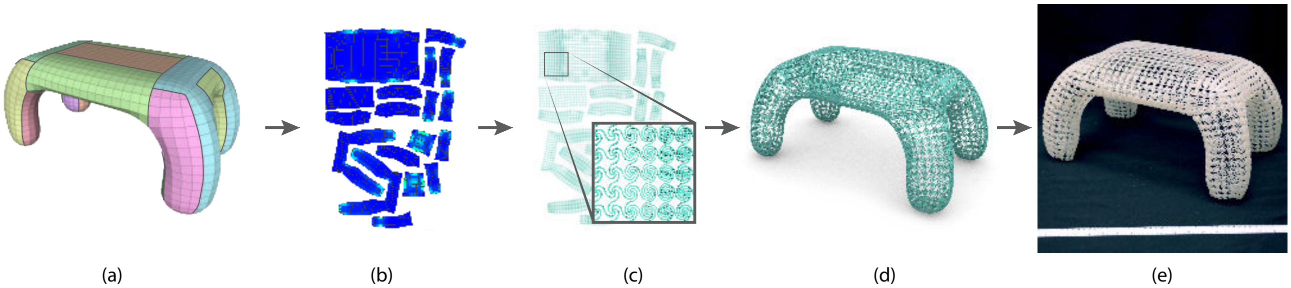


Fig. 2. An overview of our modeling pipeline. (a) An input mesh is split into multiple patches composed by quads. (b) The patches are mapped on a 2D domain. (c) We embed special spiral-shaped microstructures for each quad. (d) The parameters defining the shape of spiral microstructures are optimized to improve accuracy in the final representation. (e) Flat panels are fabricated and assembled to reproduce the input mesh.

a cross field to orient triangles in the parametric domain. A proper quadrangulation is a byproduct of a globally smooth parametrization and can be obtained by sampling the parametric domain at the nodes of an integer grid. While these methods can provide high-quality quadrangulations, the global structure of the quadrangulation may be very complex (singular points are not aligned and quads cannot be grouped directly into larger patches), so they are not directly suitable for our purpose. A different class of methods designs coarse quad patch layouts. This class of parametrization domain is more suitable for our application field because it directly matches with our 2D patch layout. The approaches in [Bommes et al. 2011; Tarini et al. 2011] simplify an existing quadrangulation and group the elements into rectangular patches. The methods proposed in [Campen et al. 2012; Pietroni et al. 2016; Razafindrazaka et al. 2015; Usai et al. 2015; Zhang et al. 2016] design quad patch layouts by solving a global energy minimization. Other methods, such as those of [Campen and Kobbelt 2014; Ji et al. 2010; Marcias et al. 2015; Takayama et al. 2013] allow the user to interactively draw a quad patch layout by using a simple graphical user interface. For a more complete description of quadrangulation and parametrization methods, we refer to [Bommes et al. 2012].

Designing custom elastic properties. Using small-scale material structures is a powerful approach to tune material behavior at a larger scale. The elasticity of an object can be controlled by tiling of pre-computed structures [Bickel et al. 2010], obtained either by topology optimization [Schumacher et al. 2015] or from a parametrized design space of elastic textures [Panetta et al. 2017, 2015]. Martinez et al. [2016] investigated an efficient method for generating procedural microstructures inspired by Voronoi open-cell foams. A key underlying concept of these methods is that there is a relationship between local geometric characteristics and local averaged material behavior. This approximated relationship is described by the theory of homogenization, which assumes a sufficiently large repetition of a constant local geometric structure to describe an averaged material behavior [Allaire 2012; Kharevych et al. 2009]. This assumption fails in cases in which individual structures are combined with properties varying with high frequency, as may happen in our structures. In contrast, we approximate the behavior of individual elements with a reduced, tailored approach based on a rod model, similar to that recently investigated by Perez et al. [2015]. Given a deformable surface and a set of deformed poses as input, they automatically compute a printable rod mesh that, once manufactured, matches the input poses under the same boundary conditions. However, their optimization scheme (a) works on an exact representation of the geometry, (b) directly adjusts the cross-sectional profiles of the rods and their rest centerline, (c) is an offline approach, and (d) can adjust stretching within a very limited range. Because of the resulting computational cost, this limits the approach to a sparse rod network. In contrast, our method works (a) with a reduced geometry, (b) operates in a higher-level parameter space of microstructures, (c) is interactive, and (d) can handle a much larger range of stretching. Furthermore, while their meshes need to be 3D printed because of varying cross-section profiles, ours can be more easily fabricated from a sheet with uniform thickness.

Subspace simulation provides a framework to compute the deformation of complex deformable objects efficiently, and it has been applied for interactive material design [Xu et al. 2015]. However, standard model reduction techniques are not suitable for our problem. First, they require costly precomputation that must be recalculated if the material or geometric parameters change. Recent approaches reduce the computational cost of this operation via an incremental update strategy [Mukherjee et al. 2016], but are limited to small mesh changes. Second, the application to a multi-domain problem such as ours, where each microstructure would represent a domain, requires specific methods to seamlessly couple the different deformable domains [Wu et al. 2015], thus creating additional computational cost.

The topic of self-actuated material and structure design is also closely connected to the present work. These types of structures are usually composed of an actuation mechanism and a deformation-limiting mechanism that, when coupled together, produce the desired deformed shape. Recent examples include planar rod networks embedded in a pre-stretched flat fabric that deploy into complex, three-dimensional shapes [Pérez et al. 2017], or *CurveUps* [Guseinov et al. 2017], which consist of an arrangement of small rigid tiles that limit the deformation of the pre-stretched elastic sheet to which they are glued. These methods rely on two independent mechanisms for actuation and deformation-limiting. In contrast, our approach simplifies the fabrication by using a single layer responsible for both.

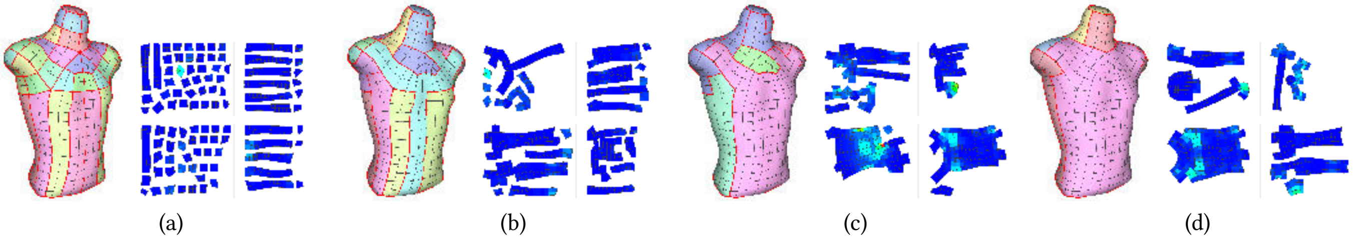


Fig. 3. An example of automatic merge sequence involving a final step of user intervention: (a) the initial patch layout; (b) an intermediate step; (c) the final result of the procedure; (d) the patch layout after user editing. The entire automatic merge sequence took 59 seconds to complete.

3 OVERVIEW

Our goal is to physically approximate a given 3D smooth shape by using a low number of fabricated flat, flexible panels and solely by snapping together connectors on their boundaries. We achieve this goal by flattening the 3D shape and then carefully tuning the elastic parameters of the panels by introducing a procedural cutting pattern so that, once assembled, it will deform to closely approximate the input shape. Figure 2 shows an overview of our fabrication pipeline. In the following, we introduce our system and discuss the motivation for important design choices.

Structure of FlexMaps. A FlexMap $\mathcal{F}$ consists of a collection of n quadrilateral elements Q_i grouped in m patches P_j separated by cuts and defined in a 2D domain $\mathcal{D}$, and a microstructure S_i enclosed by each quad. In addition, given some input shape $\mathcal{T}$ described by a set of n quads $\mathcal{F}$, there exists a bijective parametrization $f : \mathcal{D} \rightarrow \mathcal{T}$. The physical creation of the shape is obtained by fabricating the microstructure patches (by laser cutting or 3D printing), and joining them along the cut lines.

Several factors contribute to the appearance of the final shape once assembled: the topology of the patches, defined by the cut lines, the position of the structures in the 2D domain, the distortion introduced by f, and the geometric shape of the spirals S_i. The 3D shape is governed by the requirement that it must be in equilibrium; i.e., all forces are in balance.

Our system. Our workflow starts by mapping the input surface shape into a planar domain and creating a quadrangulation with minimal distortion (Section 4). We then embed a structure in each quad, drawn from the parametric space of the microstructures (Section 5). Our model, tailored for these microstructures, in combination with a data-driven approach for estimating model parameters, allows us to predict the resulting deformed shape in 3D at interactive rates, as well as to automatically adjust and optimize microstructures for improving the quality of the shape approximation (Sections 6 and 7).

To reduce the number of required patches, we employ an automatic merging process. It greedily merges adjacent patches trying to minimize the maximum distortion. Additionally, if desired, the user can guide the process by performing merge operations along patch-separating cuts through an interactive system.

4 COMPUTING A 2D MAPPING

As we previously introduced, producing a FlexMap requires the initial form to be parametrized onto a 2D domain. To make the fabrication physically plausible, the parametrization must be bijective.

To obtain an appropriate representation of a surface once the FlexMaps are assembled, our parametric domain should exhibit as low distortion as possible. In our setup, keeping a low distortion has a main practical advantage: low distortion parametrizations produce meshings with regular, uniformly sized quads. Our reduced model for a physically based simulation is built on the assumption that microstructures are embedded in regularly shaped quads. Hence, the closer the quads are to their ideal shape, the more accurate the physical simulation. We refer to Section 8 for a more detailed discussion on the consequences of this assumption.

A good strategy for deriving a bijective parametrization with low distortion is to design a coarse rectangular quad layout following the topology of a curvature-aligned cross field defined on the surface of the object. Because quads are aligned with the main curvature directions, they are nearly developable. Thus, they can be mapped to a 2D space introducing as little distortion as possible.

We consider as input mesh a field-aligned quadrangulated model produced by the automatic method proposed in [Pietroni et al. 2016]. This technique also outputs a patch decomposition, which makes it perfectly suitable for our framework. Alternatively, we could also use manually modeled quad meshes, in which case we would trace parametric lines from irregular points of the quadrangulation and split the mesh into rectangular patches. All quad meshes are further optimized using the regularization method proposed by Pietroni et al. [2015] to improve the shape of the quads. Our framework automatically provides an initial parametrization by mapping each patch into a portion of the parametric domain by using the as-rigid-as possible parametrization derived from Liu et al. [2008]. This method offers a good compromise between efficiency and the capacity to preserve angles and areas during the mapping. We do not enforce continuity constraints across seams as this would significantly reduce the degrees of freedom during the parametrization process and increase the distortion. This step often results in a low distortion parametrization with a high number of patches. Even if length across adjacent edges does not match, the practical assembly is guaranteed by the elasticity of the Flexmaps. Notice that, since our base quadrangulations are globally consistent across seams, the assembled model will not have T-junctions.

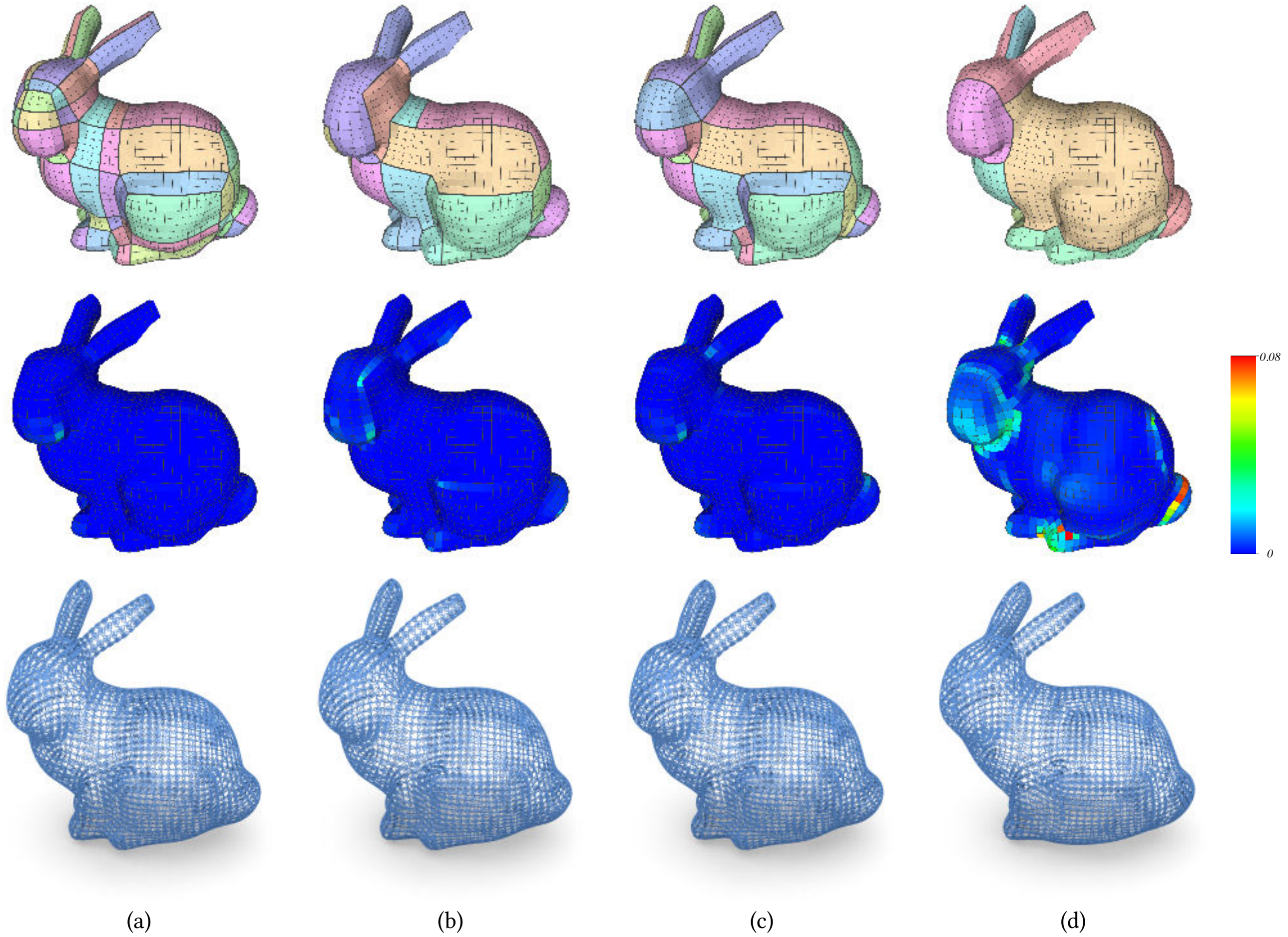


Fig. 4. (top) Three different cut layouts: (a) base, (b) automatically computed and (c) manually edited; (middle) the introduced distortion measured as per-quad residual of ARAP energy; (bottom) the corresponding flexmaps results. (d) Merging patches exceeding our distortion threshold will result in visible artifacts in the resulting model.

To facilitate the assembly of the fabricated model, our framework offers the possibility of merging adjacent patches to reduce the complexity of the parametric domain. Within the space of all possible layouts with low distortion, the final assembled configuration is not sensitive to the shape and number of patches (see Figure 4). Consequently, the design of the patch layout is mainly an aesthetic design choice that determines the seams along which connector elements are placed.

In this light, we give the user the possibility to define a custom layout by using a simple interactive interface or rely on an automatic merging procedure. Our automatic merging algorithm iteratively merges adjacent partitions, prioritizing operations based on the lowest generated distortions. As in [Poranne et al. 2017], operations that induce a distortion exceeding a certain threshold are forbidden (maximum ARAP-energy of 0.035). Similarly, we disable operations that produce long thin or highly irregular patches because those will be difficult to assemble (i.e., minimum area-square-root to perimeter ratio of 5). The merge process proceeds in a greedy fashion until no possible merge operation remains. Both manual and automatic merging can be seamlessly alternated.

5 MICROSTRUCTURE DESIGN

Our framework can be applied to any simple pattern parametrized by a small set of scalar values. We seek a microstructure design that satisfies the following conditions: (a) it fits within the boundaries of a regular quad, (b) it can be tiled to cover the input surface and create a unique connected structure, (c) it can change its elastic properties depending on its parameters, (d) it has sufficient flexibility to assume a doubly curved shape, and (e) it is easily fabricable by carving a rigid material.

One possible option would be to use planar no-sag springs; this alternative was initially considered and then rejected due to a sub-optimal use of quad space which affects the final aesthetics and deformation range. In the end we found that the spiral-shaped geometry shown in Figure 5 (left) fits perfectly with our applicative scenario.

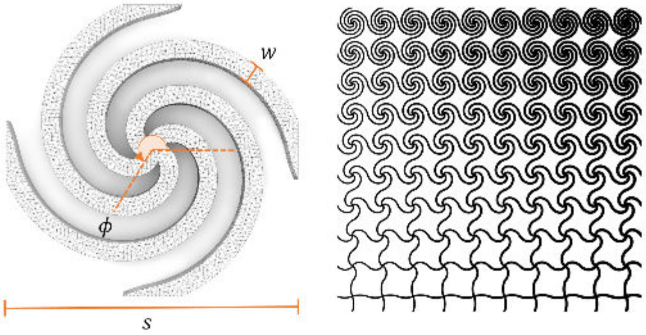


Fig. 5. Left: descriptors determining the geometry of our spiral-shaped microstructrures. Right: assemblies made out of spirals can smoothly variate the twist (vertical axis) and width (horizontal axis) of the spirals. This allows us to specify heterogeneous elastic properties throughout the parametrized surfaces.

Our structure is composed by four Archimedean spirals having the same origin and constant separation distance. Thanks to the lengths of their arms, our spirals are capable of efficiently distributing the internal stress due to large deformations without breaking apart. Hence, as noticed by [Zarrinmehr et al. 2017], this design is capable of showing high flexibility, changing surface area, and approximating double curvature. In addition, carving spiral patterns from rigid sheets has been previously explored in the design community as an alternative to kerfing and living hinge modeling. Finally, spirals are easily parametrizable which allows us to explore the deformation space of each design.

Our spirals are geometrically determined by three parameters:

1. **Scale**, s, the diameter of the circumscribed circle.
2. **Twist**, ϕ, an angle that determines the number of windings. It defines the deviation at the origin of the spiral between the twisted arm and the base cross-shaped configuration.
3. **Width**, w, which determines the shape of the rectangular cross-section. We keep the height h fixed so that the structure can be easily carved from a planar sheet.

The 2D curve corresponding to a single spiral arm can be computed using the following parametric formula: for an arc-length parameter $t \in [0, 1]$, $x(t) = r(t)\cos(t\phi)$ and $y(t) = r(t)\sin(t\phi)$, where $r(t) = 0.5\,s(1 - t)$. The 3 remaining spirals are computed by rotating the first one by 90°, 180° and 270° respectively. The final 3D surface is obtained by extruding the rectangular cross-section $w \times h$ along the path defined by the spirals.

We found that these three parameters are a good compromise between expressiveness and efficiency in the exploration of the deformation space. When tiled within a quad mesh, each microstructure is connected to its neighbors at the mid-point of their shared edge. While the scale of each spiral is locally determined by the size of the parametrized quad, the width and twist can still be adjusted. Both parameters produce different nonlinear effects on the deformation behavior of the spiral and keeping them separated allows for larger strain ranges. As shown in Figure 5 (right), the spiral assembly can continuously variate these parameters. The variation of each spiral geometry implies a change on the elastic properties of the overall structure, which allows us to locally control the physical properties of the FlexMap to better approximate target shapes.

In the following, we refer to these geometric magnitudes as *spiral descriptors*, $\mathbf{d}_i = \{s, w, \phi\}$. Given a target mesh $\mathcal{T}$, defined by a set of quads $\mathcal{F}$, $|\mathcal{F}| = n$, a fabricable FlexMap design is then completely determined by a 2D parametrization of the mesh Q together with the vector of spiral descriptors $\mathbf{d} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$.

6 SPIRAL STRUCTURE SIMULATION

Our method relies on physical simulation to predict the deformation behavior of FlexMaps. One straightforward solution would be to use a volumetric FEM model, where the geometry of the spiral structure is explicitly represented. Although that approach would provide a very accurate approximation, the high computational cost makes it unsuitable for an interactive design application.

Instead, we propose a reduced mechanical model based on the rod coupling formulation in [Pérez et al. 2015] that approximates the range of deformations of the much more complex spiral structure. Following a data-driven approach, we characterize the elastic properties of the reduced model as a function of the spiral shape. To this end, we first sample regularly the space of the spiral descriptors. For each individual spiral, we then find the parameters of the reduced model that best match the behavior of an accurate FEM simulation. This results in a discrete map from spiral descriptors to reduced model parameters, which is finally interpolated using a radial basis function network (RBFN). In the following sections, we will describe each step in detail.

6.1 Spiral reduced model

We represent each spiral using a rigid body attached to a series of edges with adapted frames. Each edge connects the tip of an arm of the spiral to the central point. Frames allow us to effectively capture the deviation of each arm with respect to the rigidly rotated spiral, independently considering stretch, bending, and twist deformation (see Figure 6).

Spiral elastic energy.

We describe the kinematic state of a 4-arm spiral using one central node, $\mathbf{c} \in \mathbb{R}^3$, 4 extremal nodes, $\mathbf{r}_i \in \mathbb{R}^3$, 4 orthonormal frames, $\mathbf{m}_i = \{\mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i\} \in SO(3)$, and three Euler angles, $\gamma \in \mathbb{R}^3$, which represent the rigid rotation of the spiral $\mathbf{R}(\gamma)$. Note that the frames are adapted to each corresponding edge, and thus, the following condition must hold: $\mathbf{t}_i = \mathbf{e}_i / \|\mathbf{e}_i\|$, for $\mathbf{e}_i = (\mathbf{r}_i - \mathbf{c})$. To avoid using constraints, we take the curve-angle frame representation from [Bergou et al. 2010] and express each deformed frame $\mathbf{m}_i$ through a rotation roll θ_i around the edge of an adapted reference frame $\underline{\mathbf{m}}_i$, $\mathbf{m}_i = \mathbf{R}(\mathbf{e}_i, \theta_i)\,\underline{\mathbf{m}}_i$. Reference frames are initialized at the rest configuration equal to the rest frame $\bar{\mathbf{m}}_i$ and kinematically determined through parallel transport at any deformed configuration. For a typical spiral with four arms, this discretization results in a state vector of 22 DoF, $\mathbf{q} = \{\mathbf{c}, \mathbf{r}_1, \ldots, \mathbf{r}_4, \theta_1, \ldots, \theta_4, \gamma\}$.

Given this discretization, we compute the elastic potential of a single spiral as the sum per edge $\mathbf{e}_i$ of stretch V_s, bending V_b, and

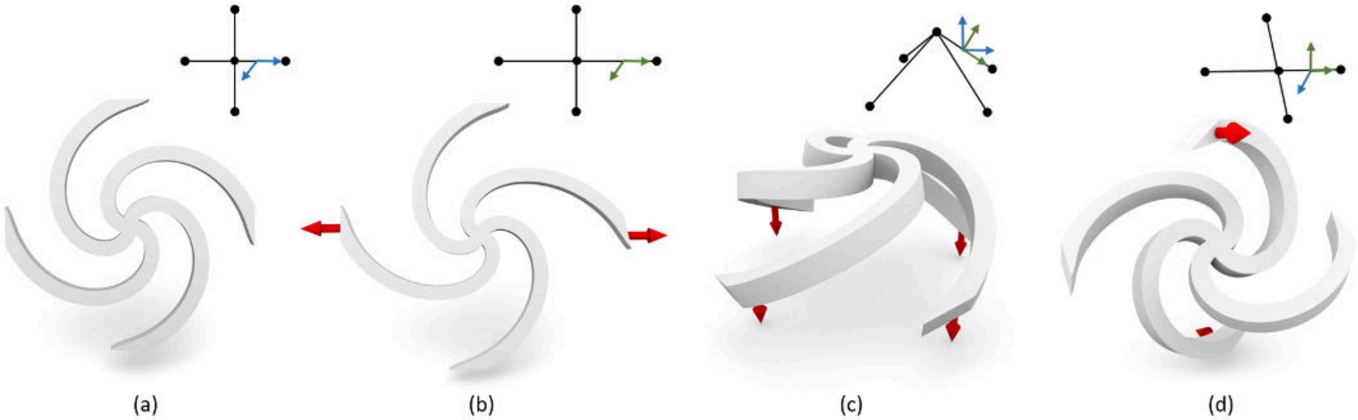


Fig. 6. A single spiral (a) undeformed, (b) stretched, (c) bent, and (d) twisted. For each individual arm, our reduced model captures the length difference, as well as the deviation of the deformed frame (green) with respect to the rigidly rotated rest frame (blue). Although for bending deformations the normal and the tangent of the arm might deviate from the rigid transformation, in purely twist deformations the tangent remains undeformed.

twist V_t energy terms:

$$V = \sum_{i=1}^{4} V_s(\mathbf{e}_i) + V_b(\mathbf{m}_i, \mathbf{m}_i^R) + V_t(\mathbf{m}_i, \mathbf{m}_i^R, \theta_i). \tag{1}$$

Here, $\mathbf{m}_i = \mathbf{R}(\mathbf{e}_i, \theta_i)\,\underline{\mathbf{m}}_i$ is the i-th deformed frame, and $\mathbf{m}_i^R = \mathbf{R}(\gamma)\,\bar{\mathbf{m}}_i$ is the i-th rigidly-rotated rest frame. The energy terms are defined as follows:

(1) **Stretch energy**. Stretch strain depends on the deformed-to-rest edge length ratio, $s = \|\mathbf{e}\|/\|\bar{\mathbf{e}}\|$, where the bar symbol $\bar{\mathbf{e}}_i$ refers to magnitudes evaluated at the rest configuration. For stretch stiffness k_s and integration domain $D = \|\bar{\mathbf{e}}\|$, the stretch energy is

$$V_s = \frac{1}{2}\, k_s\, D\, (s-1)^2\,. \tag{2}$$

(2) **Bending energy**. Bending strain measures the deviation between two orthonormal frames $\mathbf{m}_1, \mathbf{m}_2$, and is expressed in terms of the normal and binormal curvatures, $\kappa_n = 0.5\kappa\mathbf{b}\cdot(\mathbf{n}_1+\mathbf{n}_2)$ and $\kappa_b = -0.5\kappa\mathbf{b}\cdot(\mathbf{b}_1+\mathbf{b}_2)$, respectively. Here, $\kappa\mathbf{b}$ is the curvature binormal as defined in [Bergou et al. 2008]. For the bending stiffnesses k_{b0}, k_{b1} and integration domain $D = \|\bar{\mathbf{e}}\|$, the bending energy is

$$V_b = \frac{1}{2}\frac{1}{D}\left(k_{b0}\,\kappa_n^2 + k_{b1}\,\kappa_b^2\right). \tag{3}$$

(3) **Twist energy**. Twist strain is directly determined by the roll angle that defines the edge frame, plus a reference twist $\psi(\mathbf{m}_i, \mathbf{m}_i^R)$ introduced when parallel transporting the reference frame (see [Bergou et al. 2010]). For twist stiffness k_t and integration domain $D = \|\bar{\mathbf{e}}\|$, the twist energy is

$$V_t = \frac{1}{2}\frac{1}{D}\, k_t\, (\theta+\psi)^2. \tag{4}$$

Note that the bending and twist energies depend on all DoF. Twist forces acting on the edge $\mathbf{e}_i$ and Euler angles γ are non-zero due to their influence on the parallel transport and the reference twist ψ_i. Similarly, bending forces acting on roll angles θ_i and Euler angles γ are non-zero due to their effect on the deformed and rigidly rotated frames. The mechanical behavior of a single spiral is then completely determined by four model parameters $\mathbf{p} = \{k_s, k_{b0}, k_{b1}, k_t\}$

Spiral assembly structure.

We model the spiral assembly as a quadrangular mesh $\mathcal{S}$ that we call the *spiral* mesh. The topology of the mesh is defined as a set of vertices $\mathcal{V}$, $|\mathcal{V}| = n$, which represents the center of each spiral, and a set of edges $\mathcal{E}$, $|\mathcal{E}| = m$, which models the interaction between each pair of neighboring spirals. Note that this mesh can be easily computed as the dual of the target mesh, $\mathcal{T}$. Each spiral shares with its neighbors the DoF corresponding to its arms, i.e., roll angles θ_i and edges $\mathbf{e}_i$, including central and extremal nodes. Only Euler angles γ representing the rigid rotation remain unique for each individual spiral. This discretization results in a global state vector $\mathbf{q} = \{\gamma_1, \ldots, \gamma_n, \mathbf{c}_1, \ldots, \mathbf{c}_n, \theta_1, \ldots, \theta_m\} \in \mathbb{R}^{6n+m}$.

Given this discretization, the overall elastic potential can be computed as the sum of the potentials of each individual spiral. As spiral arms are shared, the integration domain D in Equations 2, 3, and 4 must be modified to account for the overlap. Note that two neighboring spirals might not be of identical sizes and thus the corresponding portion of the shared edge is not equal.

Using this model, the elastic behavior of the spiral structure is completely determined by a vector, $\mathbf{p} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \in \mathbb{R}^{4n}$, with $\mathbf{p}_i$ the model parameters of the i-th spiral. In the next section, we describe how these parameters can be fitted to match the elastic behavior of a high-resolution FEM model.

6.2 Model parameter fitting

Equipped with the reduced model, we next characterize model parameters to match the elastic behavior of a high-resolution FEM simulation. Following a data-driven approach, we populate a database with volumetric models of individual spirals that encompass the space of all possible spiral geometric shapes. To this end, we regularly sample spiral descriptors $\mathbf{d}$ within the fabricable range, i.e., $s \in [0.01, 0.03]m$, $w \in [0.001, 0.002]m$, and $\phi \in [0, 360]$ degrees. The

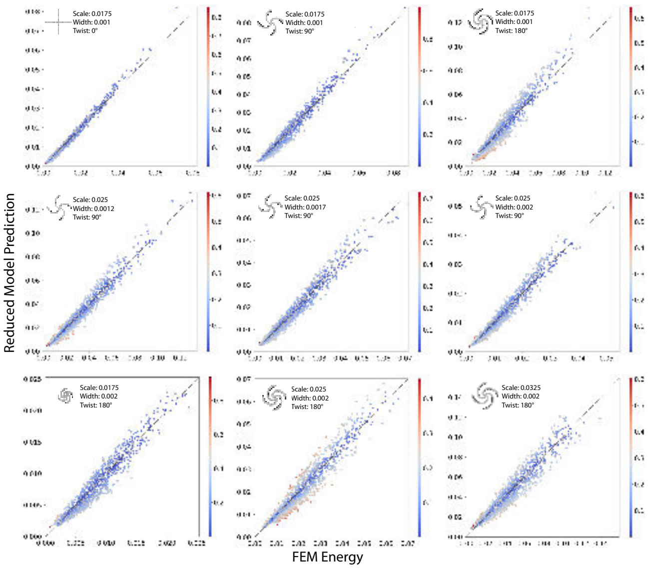


Fig. 7. Elastic energies for nine different spirals in the dataset. Each dot represents a sampled deformation. X axis indicates the amount of elastic potential computed using our FEM simulation and Y axis shows the predicted energy using our fitted reduced model. Samples are colored according to the corresponding relative error.

step size in each case is adjusted leading to a total of five discrete sampling values per parameter.

For each of the resulting 125 models, we perform $r = 1000$ offline FEM simulations that compute the static equilibrium of the volumetric model for random boundary conditions. We attach five orthonormal frames **F**: one to the tip of each arm and one at the center of the spiral. We also track their positions **P**, to capture stretch, bending, and twist deformations. Note that given these features we can geometrically reconstruct the DoF vector **q** that defines the kinematic state of our reduced rod model.

We use a St.VK material model and linear tetrahedral discretizations ranging from 4K to 160K elements, for the smallest (simplest) and largest (most complex) spirals, respectively. We refer to Section 8 for a more elaborated discussion on the motivation for this choice. In all the performed simulations, we apply forces on three of the four spiral arm extremities and keep the remaining one fixed. The direction of the forces is generated by uniformly sampling the unit sphere. The magnitude is randomly sampled in a chosen range $[1, 5]N$ that produces a diverse range of deformations. Printing material characterization is provided by the manufacturer resulting in Young modulus $E = 2600MPa$, Poisson's ratio $\mu = 0.3$, and density $\rho = 1.185g/cm^3$. All simulations are performed with the *VegaFEM* [Sin et al. 2013] built-in quasi-static solver. The average data generation time for each spiral ranges from 1 to 8 hours depending on the complexity of the model.

Inspired by previous works using energy-based parameter fitting (e.g., [Chen et al. 2015]), we formulate a nonlinear least-squares problem to find the model parameters **p** that minimize the relative difference in elastic potential between the reduced model and the FEM simulation:

$$min_{\mathbf{p}} \sum_{i=1}^{r} \left(\frac{V(\mathbf{p}, \mathbf{q}_i)}{U_i^F} - 1 \right)^2 . \tag{5}$$

Here, U_i^F refers to the elastic potential at static equilibrium of the FEM simulation, and $\mathbf{q}_i$ is geometrically determined from the tracked features $(\mathbf{F}_i, \mathbf{P}_i)$. We ensure an accurate approximation of the forces is obtained by densely sampling the source energy functions. Furthermore, as the reduced model is based on a moderate number of parameters, overfitting effects (oscillations) are prevented. We solve this optimization problem using a standard L-BFGS algorithm, as provided by the *Knitro* optimization package [2006]. We repeat this process for all the 125 sampled spirals resulting in a set of mapped pairs $M = \{\mathbf{d}_i, \mathbf{p}_i\}, i = 1, \ldots, 125$.

Figure 7 shows the fitting results for some of the spirals in the dataset. The error is uniformly distributed throughout all samples and fitted values provide a consistent generalization of spirals elastic behavior. As it can be seen in Section 8, our reduced model is capable of approximating with sufficient accuracy the deformation behavior of the spiral structures for our intended application scenarios.

6.3 Spiral descriptors mapping

The fitting procedure described in the previous section results in a discrete map for each generated sample, from spiral descriptors to the corresponding rod model parameters. In Section 7, we describe a spiral structure optimization scheme that relies heavily on an accurate simulation of our reduced model. Thus, we require a continuous and differentiable function, $\mathbf{p} : \mathbb{R}^3 \to \mathbb{R}^4$, $\mathbf{p} = \mathbf{p}(\mathbf{d})$, which would allow us to compute the gradient of the objective function with respect to spiral descriptors.

For this purpose, we seek a method capable of producing a continuous interpolation of any arbitrary set of mapped samples. We find that RBFNs perform better than other alternatives (e.g., polyharmonic splines or inverse distance weighting), when predicting the energy values from test data. Thus, we employ *MATLAB*'s built-in RBFN implementation, and train the network using the set of discretely mapped descriptors M. To avoid scale dependent artifacts, we preprocess the data to be in the unit range for the inputs and outputs of the mapping function. Note that for simplicity of notation and without loss of generality, we will ignore this normalization step in our shape optimization formulation in the following section. The configuration space of the RBFN is a single scalar that determines the "spread" (i.e. the smoothness of the resulting approximation function). To avoid overfitting, we uniformly sample this variable in the range $[0.2, 0.8]$ with 20 samples and took the most general result (i.e. less number of neurons). In this process, we reject any solution for which the training RMS error is higher than the threshold $\epsilon = 0.001$.

7 SPIRAL STRUCTURE OPTIMIZATION

FlexMaps allow us to produce elastic surfaces with heterogeneous material properties. However, it is highly nontrivial to infer what combination of local spiral descriptors will result in a specific global shape. Intuitively, regions of high curvature of the mesh should be populated with characteristically compliant spirals (i.e. low width

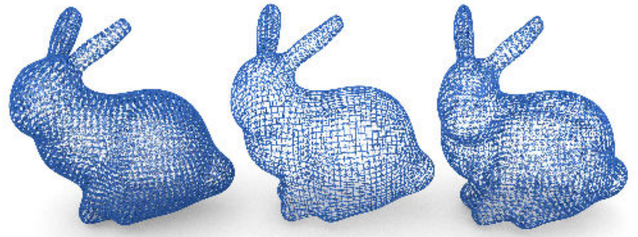


Fig. 8. Our shape optimization formulation noticeably improves the approximation of the input target mesh: (left) unoptimized bunny with homogeneous spirals; (center) unoptimized bunny with curvature-adjusted spirals; (right) optimized bunny using curvature-adjusted spirals as initialization.

and high twist), and vice versa. Consequently we provide an initial solution where the width and twist of the i-th spiral, $w_i \in [w_{min}, w_{max}]$ and $\phi_i \in [\phi_{min}, \phi_{max}]$ respectively, are linearly interpolated between the extremal values, considering the relative RMS curvature $c_i = C_i / C_{max}$ as a parameter, i.e. $w_i = w_{max} - c_i(w_{max} - w_{min})$ and $\phi_{min} + c_i(\phi_{max} - \phi_{min})$.

However, this straightforward solution does not often provide satisfactory enough results (see Figure 8). To address this problem, we further refine the curvature-adjusted solution using a shape optimization that automatically finds the spiral descriptors $\mathbf{d}$ which minimize a position-based distance to the target quadrangular mesh, $\mathcal{T}$. As mentioned in Section 5, the scale of each spiral is locally determined by the size of each parametrized quad, but the width and twist can still be adjusted. For notation simplicity, in the following we denote the optimization parameters with $\mathbf{d}$, although only two of the three spiral descriptors are considered.

We formulate the problem to minimize the difference in Laplacian coordinates between the deformed configuration of the spiral's centers in static equilibrium, $\mathbf{c} \in \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$, and the centroids of the quads in the target shape $\mathcal{T}$, $\mathbf{t} \in \{\mathbf{t}_1, \ldots, \mathbf{t}_n\}$. This results in the following constrained nonlinear optimization problem:

$$\begin{aligned} \min_{\mathbf{q},\mathbf{d}} g &= \frac{1}{2}\left(\mathbf{L}\left(\mathbf{C}\,\mathbf{q} - \mathbf{t}\right)\right)^2 \\ \text{s.t.}\ &\mathbf{f}(\mathbf{q}, \mathbf{p}(\mathbf{d})) = \mathbf{0} \\ &\mathbf{d}_{min} \leq \mathbf{d} \leq \mathbf{d}_{max}, \end{aligned} \tag{6}$$

where $\mathbf{C}$ is a selection matrix of the spiral's centers, $\mathbf{c} = \mathbf{C}\,\mathbf{q}$, and $\mathbf{L}$ is the uniform Laplacian matrix of the spiral mesh, $\mathcal{S}$. We include the constraint $\mathbf{f} = \mathbf{0}$ to enforce the result is in static equilibrium, as well as bound constraints to guarantee the solution is within a fabricable range, $[\mathbf{d}_{min}, \mathbf{d}_{max}]$. Our optimization objective uses Laplacian coordinates to preserve local details. Error distribution in Cartesian coordinates favors global shape optimization leading to over-smoothed results. Instead, the Laplacian error provides a gradient toward matching the local curvature of the target shape. We experimented with Cartesian coordinates and obtained results that were worse in comparable optimization times (see Figure 9).

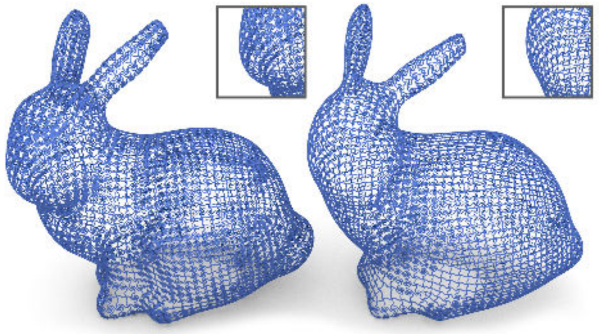


Fig. 9. A comparison between optimization using Laplacian coordinates (left) and Cartesian coordinates (right). Cartesian resulting solutions are over-smoothed leading to the lost of shape features (e.g., neck and tail of the bunny).

We aim to keep the optimization times within an acceptable range for user interaction, and to provide a fast solution that can be incrementally updated. For this purpose, we reformulate the problem to enforce nonlinear constraints implicitly and iteratively explore the static equilibrium manifold. Inspired by other shape design works, we compute equilibrium shape derivatives with respect to our design parameters. For a given deformed configuration $\mathbf{q}$ with spiral descriptors $\mathbf{d}$, such that $\mathbf{f}(\mathbf{q}, \mathbf{p}(\mathbf{d})) = \mathbf{0}$, we assume any incremental change $\Delta\mathbf{d}$ must leave the equilibrium constraints satisfied to the first order. Applying the chain rule,

$$\Delta\mathbf{f} = \frac{\partial\mathbf{f}}{\partial\mathbf{q}}\frac{\partial\mathbf{q}}{\partial\mathbf{p}}\frac{\partial\mathbf{p}}{\partial\mathbf{d}}\Delta\mathbf{d} + \frac{\partial\mathbf{f}}{\partial\mathbf{p}}\frac{\partial\mathbf{p}}{\partial\mathbf{d}}\Delta\mathbf{d} = \mathbf{0}. \tag{7}$$

Notice that $\nabla_{\mathbf{q}}\mathbf{f}$ and $\nabla_{\mathbf{p}}\mathbf{f}$ can be analytically computed from the energy formulation in Equation (1), and $\nabla_{\mathbf{d}}\mathbf{p}$ is the Jacobian of the RBFN map from the spiral descriptors to model parameters introduced in Section 6.3. Solving for $\nabla_{\mathbf{d}}\mathbf{q}$ from the resulting equation leads to the well-known *sensitivity matrix*

$$\mathrm{S} = \frac{\partial\mathbf{q}}{\partial\mathbf{p}}\frac{\partial\mathbf{p}}{\partial\mathbf{d}} = -\left(\frac{\partial\mathbf{f}}{\partial\mathbf{q}}\right)^{-1}\frac{\partial\mathbf{f}}{\partial\mathbf{p}}\frac{\partial\mathbf{p}}{\partial\mathbf{d}}. \tag{8}$$

The sensitivity matrix linearly approximates the change in the static equilibrium configuration $\mathbf{q}$ with respect to spiral descriptors $\mathbf{d}$. This allows us to reformulate the problem in Equation (6) in terms of an implicit function $\mathbf{q} = \mathbf{q}(\mathbf{d})$ as follows:

$$\begin{aligned} \min_{\mathbf{d}} h &= \frac{1}{2}\left(\mathbf{L}\left(\mathbf{C}\,\mathbf{q}(\mathbf{d}) - \mathbf{t}\right)\right)^2 \\ \text{s.t.}\ &\mathbf{d}_{min} \leq \mathbf{d} \leq \mathbf{d}_{max}, \end{aligned} \tag{9}$$

where the gradient can be computed as:

$$\nabla_{\mathbf{d}} h = \left(\mathbf{L}\,\mathbf{C}\,\mathbf{q}(\mathbf{d}) - \mathbf{L}\,\mathbf{t}\right)^T \mathbf{LCS}. \tag{10}$$

We solve the resulting bound-constrained optimization using an L-BFGS algorithm with line-search, as provided by the *Knitro* optimization package [2006]. In practical terms, we avoid the computation of the sensitivity matrix using the adjoint method, for directly calculating $\nabla_{\mathbf{p}}\mathbf{h}$. Note that this formulation requires equilibrium constraints to hold at each step. At the initialization and each time there is a new candidate solution, a static equilibrium is solved using our own

Table 1. Statistics of our models, respectively: number of spirals and patches, optimization time, assembly time and RMS error in Laplacian coordinates. Timings correspond to the optimization performed with a 3.1GHz quad-core Intel Core i7 processor (4770S).

	#Spiral	#Patch	Opt.(s)	Ass.(h)	RMSE(mm)
Sphere	137	1	1.41	0.50	0.259
3Holes	937	17	21.23	5.50	0.185
Table	1856	18	78.91	5.00	0.233
Bunny	2412	52	179.00	7.50	0.610
Torso	664	9	9.14	1.50	0.605
Shell	1296	4	28.79	0.25	0.107
Botijo	2150	68	210.28	–	0.489
Hand	2250	43	141.06	–	0.800
Horse	902	28	87.15	–	0.325

Sequential Quadratic Programming (SQP) algorithm (Newton step with back-tracking line search). These computations take around 50% of the total time. Note that, for our particular application, the stiffness-to-mass ratio of the material used to fabricate the examples allowed us to neglect the effect of gravity. However, extending our method to more general materials would only require considering the gravity force in the simulation and hence approximating the mass and its derivatives w.r.t. spiral descriptors. Overall, this optimization scheme allows us to keep computation times within the user interaction range (see Table 1).

8 RESULTS

Design workflow. We first tested our design system on a simple sphere, as shown in Figure 10. We mapped the sphere on a 2D domain and the resulting patch decomposition was easily merged to a single component. Although the presence of singularities in the quadrangulation may induce some distortion, overall, the fabricated sphere maintains its smooth shape. In general, a field-aligned parametrization tends to concentrate distortion around singularities. In our case, because of the tiling of the shape, that portion of the surface is empty. In general, for our setup, having the spiral mesh connectivity that is the dual of the target quad mesh represents a great advantage.

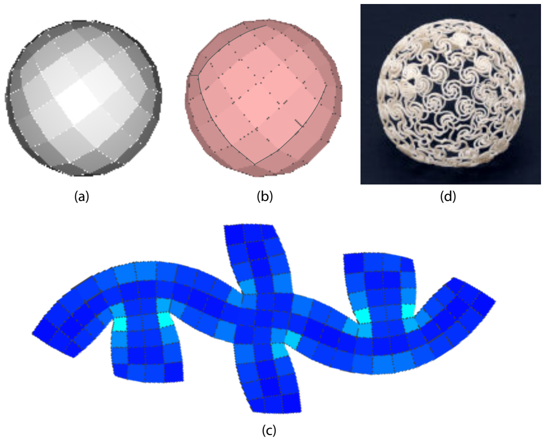


Fig. 10. A simple sphere as testing case. Starting from a target quadrangular mesh (a), a decomposition (b) and parametrization (c) are computed. Once the parametrization is populated with spirals, shape optimization leads to the fabricated result (d) that closely match the input model geometry.

We also created a number of challenging examples, shown in Figures 11 and 14. The derivation of the initial parametrization is performed as an automated step. Afterwards, the resulting patch layout is refined with some manual intervention. However, as the quad mesh is already organized into big patches, this task is simple and fast. As mentioned in Section 4, our framework prevents the user from merging patches that may introduce high distortion and thereby limits the degrees of freedom. For some models, we were able to easily and quickly design patch layouts entirely manually. Typically, a manual merging session takes around 2 minutes for simple models. For our most complex manually designed model (the bunny in Figure 14), this step required 30 minutes. In any case, for all the models, the design time was considerably faster than the fabrication. Statistics can be found in Table 1.

Thanks to the fact that we rely on field-aligned patch decompositions, the fabricated objects appear regular, aesthetically pleasant, and sometimes also symmetric. Plus, thanks to the underlying optimization, the user can focus on the final visual impact produced by the FlexMaps, trying to place the seams along non-visible regions or to place them in a way that preserves objects symmetry.

Our method can be applied in a variety of potential application fields, ranging from robotics to furniture design. As a proof-of-concept, we tested our method with an architectural model as shown in Figure 12. The model is composed by only 4 patches and, thanks to the variance on the spiral pattern, we can achieve complex double curvatures once some support is fixed to the ground applying compressive forces.

Fabrication and Assembly. FlexMaps can be fabricated using both subtractive and additive manufacturing technologies. We used a Stratasys J750 3D printer with materials from the Vero family. 3D printing of FlexMaps requires no support and is fast because of the small sheet thickness. All examples shown in the paper are $2mm$ thick. We printed special dovetail connections at every seam and used color coding to derive a proper matching during the assembly process (see Figure 13). A specific tolerance has been taken into account to ensure a proper grip between adjacent connectors. The largest object we manufactured (the bunny shown in Figure 11) is $40cm$ tall and $50cm$ deep, but our technique could easily scale to larger models.

Laser cutting is another valid option to manufacture FlexMaps. Thanks to the size of the workspace, this technology fits well our workflow and can be easily used for the production of larger objects. For the laser cut example (Table model), we used a $2mm$ material sheet made of Delrin, a thermoplastic with high stiffness, low friction, and excellent dimensional stability, that showed a similar elastic behavior (Young's modulus) as the VeroMaterial in our tests.

Finally, the assembly procedure greatly benefits from the intuitive decomposition of the domain and the color coding of the seams. However, visually comparing the partially assembled model with

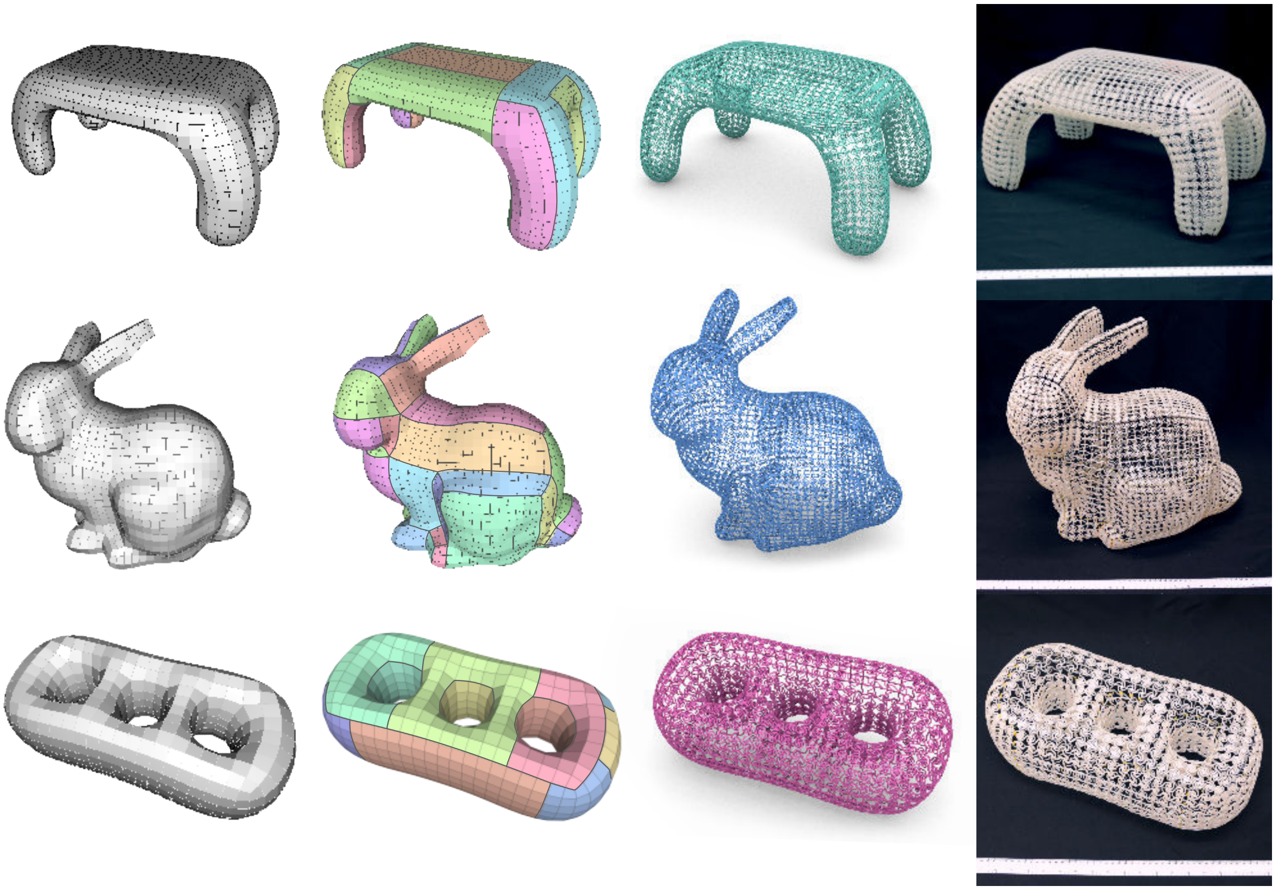


Fig. 11. Some of our fabricated examples: input target mesh, patch decomposition, optimized model, and fabricated result.

the digital model shown on a laptop helps during the assembly process.

Validation. We measured the accuracy of our simulation by laser scanning one of the fabricated models and computing the distance from each simulated spiral center to the scanned surface. For this purpose we selected our only open surface (the architectural model), as these are the more challenging examples in terms of simulation accuracy. We obtained errors relative to the size of the model in the range $[0.008, 4.092]\%$, with an RMS error of 0.971%. The resulting scanned surface can be seen in Figure 12. We believe that this experiment shows that the amount of approximation error introduced by our reduced model is within an acceptable range for our target applications.

Effect of quad regularity. To estimate the dependency of our framework on the regularity of the quadrangulation, we produced several simple models with increasing quad distortion and visually compared the resulting deformation with our simulations. We found that quad regularity slightly affects the deformation locally, but the global effect is barely noticeable and this effect is compensated when quad distortion is evenly distributed. As we use state of the art quadrangulation methods, the deformation of the parametrized quads with respect to a perfect square is generally low. For the Shell (which is one of the examples with the most distorted quads), the deformation measured in ARAP-energy is below 0.238 with median value 0.019; yet we quantified through scanning the quality of the approximation obtaining an RMS error below 1%, relative to the size of the model.

Adequacy of the material model. Our selection of material model and discretization order are motivated by the range of deformations affecting the spirals. Note that, while the quads in the target shape might suffer pretty severe deformations with respect to the planar configuration, the local strain in the spiral geometry is generally low. On an average deformed spiral, 99% of the tetrahedra have strain ϵ in the interval $[0.995, 1.005]$, where $\epsilon = tr(\mathbf{C})/3$ and $\mathbf{C}$ is the Cauchy-Green strain tensor. Under such conditions, we considered adequate to use the St.VK hyperelastic model (quadratic strain measure $\mathbf{C}$ and linear strain-stress relationship). We experimentally validated our choice through fabrication from the earliest prototypes to the final presented results. In all cases, we found that our predictions were almost visually indistinguishable from the fabricated object. Yet, our pipeline could be easily adapted to use more complex material models if needed.

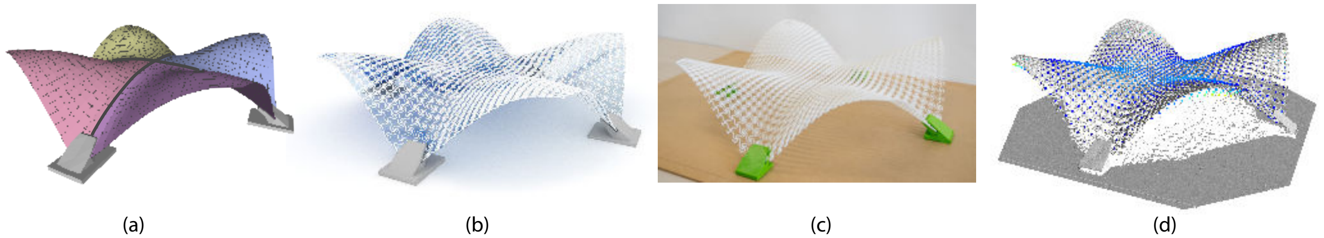


Fig. 12. Architectural shell featuring a doubly curved open surface fixed to the ground by four lateral supports. The figure shows (a) the target mesh, (b) the optimized spiral mesh, (c) the resulting fabricated example and (d) a comparison between the laser-scanned surface (gray point cloud) and the expected deformation (colored spheres). Sphere colors show the distribution of the relative error, which is in the range $[0.008, 4.092]\%$, with an RMS error of 0.971%.

9 CONCLUSION

We have presented an interactive framework for designing and building smooth surfaces out of flat, flexible panels with tailored mechanical properties. Our approach maps a 3D surface onto a 2D domain as in traditional UV mapping to design a set of deformable flat panels. We embed these panels with flexible microstructures and optimize their geometry such that, when the panels are assembled, the equilibrium configuration matches the desired 3D shape. To make this design problem tractable, we have introduced a reduced model of our microstructures and used a data-driven approach to characterize its elastic behavior based on FEM simulations. This allows us to efficiently simulate objects made out of thousands of such structures, and optimize their heterogeneous distribution to control the global deformation behavior of our FlexMaps. Our results show that the proposed pipeline is a very efficient tool for the rapid design and fabrication of a variety of smooth surfaces.

Limitations and future work. Because of physical limitations on the robustness of the material and fabrication tolerances, our microstructures require to be manufactured above a minimal size threshold of $1cm$. This implies a minimal size of our physical objects depending on the number of microstructures they consist of.

While our design interface shows the optimization result for a given input shape at interactive rates, currently there is no intuitive indication on the limits of achievable shapes. Figure 15 shows two challenging examples specifically selected to highlight the foremost limitations of our geometric approximation. On one hand, modeling sharp features and high frequency curvature details is limited by the resolution of our current fabrication technology. On the other hand, multistable configurations are not explicitly considered. This might lead to some regions of the deformation space remaining unexplored during the shape optimization.

Fig. 13. A closeup of the connectors used to secure FlexMaps across seams.

For future work, it would be interesting to further analyze these limitations in order to provide a modeling interface for exploring only achievable shapes. In addition, we do not optimize the quad layout taking the physical effects of the microstructures into account. It would be interesting to consider these degrees of freedom during the shape optimization. Finally, adding internal connections could help in creating shapes with sharper creases and concave features.

On the practical assembly side, we remark that the dovetail connectors usually work well offering a robust grip. However, in high curvature regions when the internal stress is very high, the assembly process can become more difficult. This can be solved by simply reducing the overall thickness of the FlexMap to further lower the stiffness of the pattern so that it is simpler to bend and assemble.

Although the material we used in our process has favorable mechanical properties, an interesting avenue for future work would be to explicitly take the structural strength of structures and connectors into account, similar as in the work of Panetta et al. [2017], and extend our database to a wider range of microstructures.

ACKNOWLEDGMENTS

The authors thank Francesco Banterle and Thomas Alderighi for their help and especially Marco Callieri for helping with models assembling and 3D scanning. The bunny model is courtesy of the *Stanford 3D Scanning Repository.* The hand, the botijo, the 3holes, and the drill models are courtesy of the *AIM@SHAPE Shape Repository.* The skull is courtesy of *Turbo Squid.* The research was partially funded by the EU H2020 Programme *EMOTIVE: EMOTIve Virtual cultural Experiences through personalized storytelling* H2020-SC6-CULT-COOP-08-2016 (grant no. 727188), by the European Research Council (ERC) *MATERIALIZABLE: Intelligent fabrication-oriented Computational Design and Modeling* (grant no. 715767), by the European H2020 Programme *SOMA: Soft Manipulation* (grant no. 645599), by the EU H2020 programme (Marie Sklodowska-Curie grant no. 754411) and by the Italian PRIN project *DSURF* (grant no. 2015B8TRFM). Part of the hardware resources used for this research were donated by the NVIDIA Corporation.

REFERENCES

Grégoire Allaire. 2012. *Shape optimization by the homogenization method.* Vol. 146. Springer Science & Business Media.

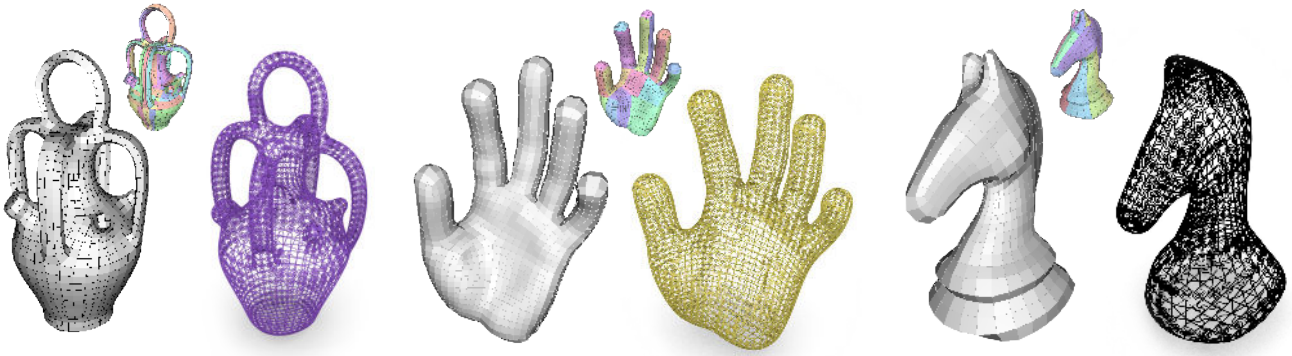
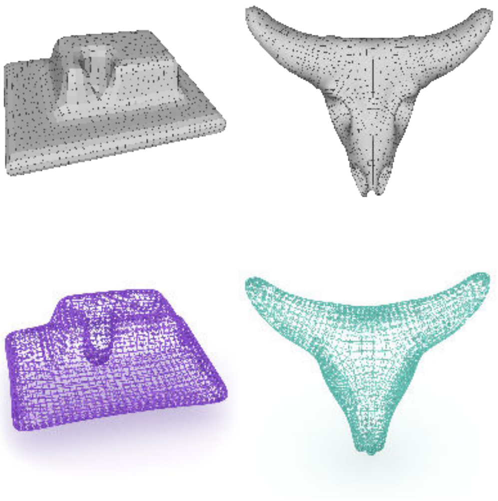


Fig. 14. Some additional challenging examples showing the input target mesh, the patch decomposition, and the optimized model.



(a) (b)

Fig. 15. Two examples of failure cases showing a remarkable difference between the target mesh (top) and the optimized FlexMap (bottom). Note that some sharp features in the left example and high frequency details in the right example are oversmoothed in our solution. Additionally, in some particularly symmetric bistable areas (e.g., cow skull eyes in (b)), the optimization gets stuck in the indefinite configuration and is unable to progress to a better solution.

P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun. 2003. Anisotropic polygonal remeshing. *ACM Trans. Graph.* 22, 3 (2003), 485–493.

Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-driven pattern adjustment for direct 3D garment editing. *ACM Transactions on Graphics* 35, 4 (jul 2016), 1–11. `DOI`:http://dx.doi.org/10.1145/2897824.2925896

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Trans. Graph.* 29, 4, Article 116 (July 2010), 10 pages. `DOI`:http://dx.doi.org/10.1145/1778765.1778853

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. *ACM Trans. Graph.* 27, 3, Article 63 (Aug. 2008), 12 pages. `DOI`:http://dx.doi.org/10.1145/1360612.1360662

Amit H. Bermano, Thomas Funkhouser, and Szymon Rusinkiewicz. 2017. State of the Art in Methods and Representations for Fabrication-Aware Design. *Computer Graphics Forum* 36, 2 (2017), 509–535. `DOI`:http://dx.doi.org/10.1111/cgf.13146

Bernd Bickel, Moritz BÃďcher, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics* 29, 4 (jul 2010), 1. `DOI`:http://dx.doi.org/10.1145/1778765.1778800

Bernd Bickel, Paolo Cignoni, Luigi Malomo, and Nico Pietroni. 2017. State of the Art on Stylized Fabrication. *Computer Graphics Forum* (2017). `DOI`:http://dx.doi.org/10.1111/cgf.13327

David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, Leif Kobbelt, and others. 2013. Integer-Grid Maps for Reliable Quad Meshing. *ACM Trans. Graph.* 32, 4 (2013).

David Bommes, Timm Lempfer, and Leif Kobbelt. 2011. Global Structure Optimization of Quadrilateral Meshes. *Comput. Graph. Forum* 30, 2 (2011), 375–384.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2012. Quad Meshing, Marie-Paule Cani and Fabio Ganovelli (Eds.). Eurographics Association, Cagliari, Sardinia, Italy, 159–182.

D. Bommes, H. Zimmer, and L. Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), 77.

Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. 2006. Knitro: An Integrated Package for Nonlinear Optimization.

Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual loops meshing: quality quad layouts on manifolds. *ACM Trans. Graph.* 31, 4 (2012), 110.

Marcel Campen and Leif Kobbelt. 2014. Dual Strip Weaving: Interactive Design of Quad Layouts Using Elastica Strips. *ACM Trans. Graph.* 33, 6, Article 183 (2014), 10 pages.

Desai Chen, David I. W. Levin, Shinjiro Sueda, and Wojciech Matusik. 2015. Data-driven Finite Elements for Geometry and Material Design. *ACM Trans. Graph.* 34, 4, Article 74 (July 2015), 10 pages. `DOI`:http://dx.doi.org/10.1145/2766889

Desai Chen, Pitchaya Sitthi-amorn, Justin T. Lan, and Wojciech Matusik. 2013. Computing and Fabricating Multiplanar Models. *Computer Graphics Forum* 32, 2pt3 (2013), 305–315. `DOI`:http://dx.doi.org/10.1111/cgf.12050

Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. 2014. Field-aligned Mesh Joinery. *ACM Trans. Graph.* 33, 1, Article 11 (Feb. 2014), 12 pages. `DOI`:http://dx.doi.org/10.1145/2537852

Levi H. Dudte, Etienne Vouga, Tomohiro Tachi, and L. Mahadevan. 2016. Programming curvature using origami tessellations. *Nature Materials* 15, 5 (jan 2016), 583–588. `DOI`:http://dx.doi.org/10.1038/nmat4540

Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire Mesh Design. *ACM Trans. Graph.* 33, 4, Article 66 (July 2014), 12 pages. `DOI`:http://dx.doi.org/10.1145/2601097.2601106

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. `DOI`:http://dx.doi.org/10.1145/3072959.3073709

Kristian Hildebrand, Bernd Bickel, and Marc Alexa. 2012. crdbrd: Shape Fabrication by Sliding Planar Slices. *Computer Graphics Forum* 31, 2pt3 (2012), 583–592. `DOI`:http://dx.doi.org/10.1111/j.1467-8659.2012.03037.x

Zhongping Ji, Ligang Liu, and Yigang Wang. 2010. B-Mesh: A Modeling System for Base Meshes of 3D Articulated Shapes. *Computer Graphics Forum (Proceedings of Pacific Graphics)* 29, 7 (2010), 2169–2178.

F. Kälberer, M. Nieser, and K. Polthier. 2007. QuadCover: Surface Parameterization using Branched Coverings. *Comput. Graph. Forum* 26, 3 (2007), 375–384.

Lily Kharevych, Patrick Mullen, Houman Owhadi, and Mathieu Desbrun. 2009. Numerical coarsening of inhomogeneous elastic materials. *ACM Transactions on Graphics* 28, 3 (jul 2009), 1. `DOI`:http://dx.doi.org/10.1145/1531326.1531357

Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann. 2008. Curved Folding. *ACM Trans. Graph.* 27, 3, Article 75 (Aug. 2008), 9 pages. DOI:http://dx.doi.org/10.1145/1360612.1360674

Martin Kilian, Aron Monszpart, and Niloy J. Mitra. 2017. String Actuated Curved Folded Surfaces. *ACM Transactions on Graphics* 36, 3 (may 2017), 1–13. DOI:http://dx.doi.org/10.1145/3015460

Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. 2016. Beyond developable. *ACM Transactions on Graphics* 35, 4 (jul 2016), 1–11. DOI:http://dx.doi.org/10.1145/2897824.2925944

Mina Konaković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid Deployment of Curved Surfaces via Programmable Auxetics. (2018).

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. 2008. A Local/Global Approach to Mesh Parameterization. *Computer Graphics Forum* 27, 5 (July 2008), 1495–1504.

Giorgio Marcias, Kenshi Takayama, Nico Pietroni, Daniele Panozzo, Olga Sorkine, Enrico Puppo, and Paolo Cignoni. 2015. Data-Driven Interactive Quadrangulation. *ACM Trans. Graph. (Siggraph 2015)* 34, 65 (2015).

Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. 2016. Procedural voronoi foams for additive manufacturing. *ACM Transactions on Graphics* 35, 4 (jul 2016), 1–12. DOI:http://dx.doi.org/10.1145/2897824.2925922

Fady Massarwi, Craig Gotsman, and Gershon Elber. 2007. Papercraft Models using Generalized Cylinders. In *Proc. of the 15th Pacific Conf. on Computer Graphics and Applications*. IEEE Computer Society, Washington, DC, USA, 148–157. DOI:http://dx.doi.org/10.1109/PG.2007.16

James McCrae, Karan Singh, and Niloy J. Mitra. 2011. Slices: A Shape-proxy Based on Planar Sections. *ACM Trans. Graph.* 30, 6, Article 168 (Dec. 2011), 12 pages. DOI:http://dx.doi.org/10.1145/2070781.2024202

Eder Miguel, Mathias Lepoutre, and Bernd Bickel. 2016. Computational Design of Stable Planar-rod Structures. *ACM Trans. Graph.* 35, 4, Article 86 (July 2016), 11 pages. DOI:http://dx.doi.org/10.1145/2897824.2925978

Jun Mitani and Hiromasa Suzuki. 2004. Making Papercraft Toys from Meshes Using Strip-based Approximate Unfolding. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 259–263. DOI:http://dx.doi.org/10.1145/1015706.1015711

R. Mukherjee, X. Wu, and H. Wang. 2016. Incremental Deformation Subspace Reconstruction. *Computer Graphics Forum* 35, 7 (oct 2016), 169–178. DOI:http://dx.doi.org/10.1111/cgf.13014

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case stress relief for microstructures. *ACM Transactions on Graphics* 36, 4 (jul 2017), 1–16. DOI:http://dx.doi.org/10.1145/3072959.3073649

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. *ACM Transactions on Graphics* 34, 4 (jul 2015), 135:1–135:12. DOI:http://dx.doi.org/10.1145/2766937

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:http://dx.doi.org/10.1145/3072959.3073695

Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and fabrication of flexible rod meshes. *ACM Transactions on Graphics* 34, 4 (jul 2015), 138:1–138:12. DOI:http://dx.doi.org/10.1145/2766998

Nico Pietroni, Enrico Puppo, Giorgio Marcias, Roberto Scopigno, and Paolo Cignoni. 2016. Tracing Field-Coherent Quad Layouts. *Computer Graphics Forum (Proceedings of Pacific Graphics 2016)* (2016). In Press.

Nico Pietroni, Davide Tonelli, Enrico Puppo, Maurizio Froli, Roberto Scopigno, and Paolo Cignoni. 2015. Statics Aware Grid Shells. *Comput. Graph. Forum* 34, 2 (2015), 627–641.

Roi Poranne, Marco Tarini, Sandro Huber, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Autocuts: Simultaneous Distortion and Cut Optimization for UV Mapping. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)* 36, 6 (2017).

Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner. 2015. Architectural geometry. *Computers & graphics* 47 (2015), 145–164.

Faniry H. Razafindrazaka, Ulrich Reitebuch, and Konrad Polthier. 2015. Perfect Matching Quad Layouts for Manifold Meshes. *Comput. Graph. Forum* (2015). DOI:http://dx.doi.org/10.1111/cgf.12710

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics* 34, 4 (jul 2015), 136:1–136:13. DOI:http://dx.doi.org/10.1145/2766926

Yuliy Schwartzburg and Mark Pauly. 2013. Fabrication-aware Design with Intersecting Planar Pieces. *Computer Graphics Forum* 32, 2pt3 (2013), 317–326. DOI:http://dx.doi.org/10.1111/cgf.12051

Idan Shatz, Ayellet Tal, and George Leifman. 2006. Paper craft models from meshes. *The Visual Computer* 22, 9-11 (2006), 825–834. DOI:http://dx.doi.org/10.1007/s00371-006-0067-6

F. S. Sin, D. Schroeder, and J. Barbič. 2013. Vega: Non-Linear FEM Deformable Object Simulator. *Computer Graphics Forum* 32, 1 (2013), 36–48. DOI:http://dx.doi.org/10.1111/j.1467-8659.2012.03230.x

Mélina Skouras, Stelian Coros, Eitan Grinspun, and Bernhard Thomaszewski. 2015. Interactive Surface Design with Interlocking Elements. *ACM Trans. Graph.* 34, 6, Article 224 (Oct. 2015), 7 pages. DOI:http://dx.doi.org/10.1145/2816795.2818128

Kenshi Takayama, Daniele Panozzo, Alexander Sorkine-Hornung, and Olga Sorkine-Hornung. 2013. Sketch-based Generation and Editing of Quad Meshes. *ACM Trans. Graph.* 32, 4, Article 97 (2013), 8 pages.

Masahito Takezawa, Takuma Imai, Kentaro Shida, and Takashi Maekawa. 2016. Fabrication of Freeform Objects by Principal Strips. *ACM Trans. Graph.* 35, 6, Article 225 (Nov. 2016), 12 pages. DOI:http://dx.doi.org/10.1145/2980179.2982406

Marco Tarini, Enrico Puppo, Daniele Panozzo, Nico Pietroni, and Paolo Cignoni. 2011. Simple quad domains for field aligned mesh parametrization. In *ACM Trans. Graph.*, Vol. 30. 142:1–142:12.

Francesco Usai, Marco Livesu, Enrico Puppo, Marco Tarini, and Riccardo Scateni. 2015. Extraction of the Quad Layout of a Triangle Mesh Guided by Its Curve Skeleton. *ACM Trans. Graph.* 35, 1, Article 6 (Dec. 2015), 13 pages.

Xiaofeng Wu, Rajaditya Mukherjee, and Huamin Wang. 2015. A unified approach for subspace simulation of deformable bodies in multiple domains. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 241.

Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbivč. 2015. Interactive Material Design Using Model Reduction. *ACM Transactions on Graphics* 34, 2 (mar 2015), 1–14. DOI:http://dx.doi.org/10.1145/2699648

Saeid Zarrinmehr, Mahmood Ettehad, Negar Kalantar, Alireza Borhani, Shinjiro Sueda, and Ergun Akleman. 2017. Interlocked Archimedean Spirals for Conversion of Planar Rigid Panels into Locally Flexible Panels with Stiffness Control. 66 (06 2017).

Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2016. Designing Structurally-sound Ornamental Curve Networks. *ACM Trans. Graph.* 35, 4, Article 99 (July 2016), 10 pages. DOI:http://dx.doi.org/10.1145/2897824.2925888

Sen Zhang, Hui Zhang, and Jun-Hai Yong. 2016. Automatic Quad Patch Layout Extraction for Quadrilateral Meshes. *Computer-Aided Design and Applications* 13, 3 (2016), 409–416.