Differentiable Stripe Patterns for Inverse Design of Structured Surfaces

JUAN MONTES MAESTRE, ETH Zürich, Switzerland YINWEI DU, ETH Zürich, Switzerland RONAN HINCHET, ETH Zürich, Switzerland STELIAN COROS, ETH Zürich, Switzerland BERNHARD THOMASZEWSKI, ETH Zürich, Switzerland



Fig. 1. Optimization of a compliant gripper using Differentiable Stripe Patterns. An initial design with parallel stripes (*a*) and an optimized design (*b*) deforms out of plane under actuation, but the initial design does not produce sufficient closure (*c*). Our optimized design (*d*) produces significantly larger lateral deflection for the same actuation in simulation, as shown in the rendered insets. The corresponding physical prototype confirms this prediction and is able to enclose and lift a small object.

Stripe patterns are ubiquitous in nature and everyday life. While the synthesis of these patterns has been thoroughly studied in the literature, their potential to control the mechanics of structured materials remains largely unexplored. In this work, we introduce Differentiable Stripe Patterns-a computational approach for automated design of physical surfaces structured with stripe-shaped bi-material distributions. Our method builds on the work by Knöppel and colleagues [2015] for generating globally-continuous and equally-spaced stripe patterns. To unlock the full potential of this design space, we propose a gradient-based optimization tool to automatically compute stripe patterns that best approximate macromechanical performance goals. Specifically, we propose a computational model that combines solid shell finite elements with XFEM for accurate and fully-differentiable modeling of elastic bi-material surfaces. To resolve non-uniqueness problems in the original method, we furthermore propose a robust formulation that yields unique and differentiable stripe patterns. We combine these components with equilibrium state derivatives into an end-to-end differentiable pipeline

Authors' addresses: Juan Montes Maestre, ETH Zürich, Switzerland, jmontes@inf.ethz. ch; Yinwei Du, ETH Zürich, Switzerland, yinwei.du@inf.ethz.ch; Ronan Hinchet, ETH Zürich, Switzerland, ronanhinchet@gmail.com; Stelian Coros, ETH Zürich, Switzerland, stelian@inf.ethz.ch; Bernhard Thomaszewski, ETH Zürich, Switzerland, bthomasz@ ethz.ch.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2023/8-ART1 \$15.00

https://doi.org/10.1145/3592114

that enables inverse design of mechanical stripe patterns. We demonstrate our method on a diverse set of examples that illustrate the potential of stripe patterns as a design space for structured materials. Our simulation results are experimentally validated on physical prototypes.

CCS Concepts: • Applied computing \rightarrow Computer-aided manufacturing; • Computing methodologies \rightarrow Modeling and simulation.

ACM Reference Format:

Juan Montes Maestre, Yinwei Du, Ronan Hinchet, Stelian Coros, and Bernhard Thomaszewski. 2023. Differentiable Stripe Patterns for Inverse Design of Structured Surfaces. *ACM Trans. Graph.* 42, 4, Article 1 (August 2023), 14 pages. https://doi.org/10.1145/3592114

1 INTRODUCTION

From soft to stiff, from isotropic to anisotropic, and from homogeneous to functionally-graded—designing materials with tailored mechanical properties is a central problem in many fields of science and engineering. Structured materials are particularly interesting in this context since their macromechanical behavior can be controlled through their microscale geometry. Here we consider a particular class of structured surfaces that are quasi-inextensible in a given direction while being compliant in the orthogonal direction. By varying the principal directions across the surface, these materials can achieve a broad range of macromechanical effects, making them interesting for applications in, e.g., sportswear, orthotics, and robotics (see Fig. 1). Designing structured surfaces that lead to desired mechanical behavior, however, is a challenging problem.

In this work, we propose Differentiable Stripe Patterns—a computational approach to performance-oriented design of structured surfaces. We draw inspiration from the work of Knöppel et al. [2015] who proposed a method for generating equally-spaced, globallycontinuous stripe patterns on arbitrary surfaces. Our key observation is that, when interpreted as bi-material distributions, stripe patterns form an ideal design space for structured surfaces with high stiffness contrast. To unlock the full potential of this material space, we envision an inverse design tool that automatically computes stripe patterns that lead to an optimal approximation of high-level performance goals.

To implement this vision, we must overcome a number of challenges. First, stripe patterns are generated from tangent vector fields by solving a complex eigenvalue problem (EVP). Due to intrinsic symmetries, eigenvalues of this EVP always occur in pairs. This nontrivial multiplicity means that eigenvectors are not unique-they form an eigenplane-and derivatives do not exist. Second, predicting the mechanics of bi-material distributions with macroscopic stripe patterns requires accurate modeling of material interfaces. While conforming discretizations are an obvious choice for static interfaces, finite changes in stripe patterns require remeshing to maintain valid tesselations. Such discrete changes are highly problematic for gradient-based optimization. Third, to obtain valid stripe patterns, eigenvectors must be further processed with vector normalizations and other nonsmooth operations. Derivatives of these operations are discontinuous or diverge at singularities, which is again highly problematic for optimization-based design.

Our Differentiable Stripe Patterns integrate dedicated solutions to each of these problems:

- We achieve robust evolution of eigenvectors by decomposing the problem of finding the *best* vector from an eigenplane into the much simpler sub-problems of finding *a*) *some* reference vector from the eigenplane and *b*) the *optimal* in-plane rotation with respect to this vector. This unique and differentiable parameterization of the eigenplane yields well-defined stripe pattern derivatives.
- To allow for continuous motion of material interfaces without remeshing, we propose a computational model that combines solid shells with the extended finite element method (XFEM). Solid shells are made of prismatic elements that derive their response to bending from differential stretching through the thickness. Unlike discrete bending models common in graphics, solid shells follow standard finite element theory and thus integrate seamlessly with XFEM.
- We remove and replace non-smooth operations and introduce a design-space regularizer that prevents numerical singularities with logarithmic barrier functions.

Collectively, these contributions combine into a powerful and fully differentiable model for stripe pattern materials. In combination with adjoint sensitivity-analysis, Differentiable Stripe Patterns enable gradient-based optimization of high-level design objectives with opportunities for broad applications. We illustrate our method on a set of inverse design examples, including fabrics with 3Dprinted reinforcements, structural textures for thin-walled 3D-prints, a compliant shell gripper, and personalized insoles. We validate our examples through physical prototypes and observe good agreement between simulation and real-world behavior.

2 RELATED WORK

Structured Materials. Designing structured materials with desired macromechanical behaviors is an active field of research in material science and engineering [Bertoldi et al. 2017]. With the widespread availability of 3D printing, the graphics community has likewise started to investigate the creation of metamaterials with lattice- [Gongora et al. 2022; Panetta et al. 2017, 2015], voxel- [Bickel et al. 2010; Schumacher et al. 2015; Zhu et al. 2017], and foam-like [Martínez et al. 2016, 2017] microstructures. A particular line of research focuses on two-dimensional, sheet-like materials [Leimer and Musialski 2020; Malomo et al. 2020]. Our work also targets sheet and thin shell materials, but focuses on global design optimization instead of periodically tileable material cells.

Closest to our setting in the context of material design is arguably the work by Tricard et al. [2020] for creating microstructures with freely-orientable channels. These thin-walled tubes are generated such as to follow a user-provided orientation field using a stochastic process. The resulting structures are extremely anisotropic, showing high stiffness along channels but compliance in the orthogonal plane. Our differentiable stripe patterns similarly target materials with high stiffness contrast along locally controllable directions. However, while Tricard et al. describe a forward geometry design process that is not informed by simulation, we propose an inverse material design approach in which stripe patterns are generated automatically such as to obtain desired mechanical behaviors.

Although not a primary focus of our work, we show through examples that Differentiable Stripe Patterns can be used to modulate the mechanical properties of conventional textiles using a print-onfabric process similar to [Jourdan et al. 2021; Pérez et al. 2017]. Instead of using 3D-printing, an alternative strategy would be to use embroidery for reinforcement, as demonstrated by Moore et al. [2018] and Sati et al. [2021].

Eigenvector Optimization. Since stripe patterns emerge as the solution to a generalized eigenvalue problem, optimizing for eigenvectors is at the core of our method. Designing mechanical systems with desired spectral properties is a problem that has occurred frequently in graphics literature. Applications include metallophones that produce desired sounds [Bharaj et al. 2015; Musialski et al. 2016; Umetani et al. 2010], coarse-level simulations [Chen et al. 2017, 2019] and differential geometry operators [Chen et al. 2020; Liu et al. 2019] that preserve spectral properties, as well as mechanical assemblies that are robust to perturbations [Liu et al. 2022; Thomaszewski et al. 2014]. A specific challenge in our setting is that all eigenvalues have nontrivial geometric multiplicity, making eigenvectors non-unique and derivatives undefined. By identifying the origins of this inherent multiplicity, we develop a strategy to obtain unique reference vectors in arbitrary eigenplanes and express eigenvectors in this space through an additional rotation parameter.

Designing Physical Surfaces. As opposed to purely geometric design, physical surfaces are subject to equilibrium constraints. Applications include inflatable membranes [Panetta et al. 2021; Skouras et al. 2014], surfaces made from elastic curves [Neveu et al. 2022; Panetta et al. 2019; Pillwein and Musialski 2021; Zehnder et al. 2016], as well as auxetic [Chen et al. 2021; Jiang et al. 2022; Konaković-Luković et al. 2018] and tension-actuated [Guseinov et al. 2017; Jourdan et al. 2022b; Pérez et al. 2017] deployable structures. Adding to this line of work, we identify stripe patterns as a powerful paradigm for physical surfaces and present a differentiable pipeline for inverse design in that space.

Material Interfaces and XFEM. Our method uses stripe patterns to define bi-material distributions. This leads to the question of how to best model material interfaces that, in general, do not align with element boundaries in the simulation mesh. XFEM is a technique for modeling material interfaces with displacement or strain discontinuities without mesh adaptation [Moës et al. 2003]. In the graphics literature, XFEM has been used, e.g., for cutting shells [Kaufmann et al. 2009] and deformable solids [Koschier et al. 2017], metamaterial design [Zehnder et al. 2017], and for streamlined simulation of CAD models [Hafner et al. 2019]. To unlock stripe patterns as a design space for structured surfaces, we seek an efficient shell model that is amenable to XFEM. While the method by Kaufmann et al. [2009] is an option in principle, the high polynomial order of the bi-cubic patches together with the texture-based enrichment approach and discontinuous Galerkin formulation translate into excessive complexity for inverse design. While simpler discrete shell models from the graphics community [Bridson et al. 2005; Chen et al. 2018; Garg et al. 2007; Grinspun et al. 2003] offer favorable trade-offs between accuracy and computation cost, their discrete nature cannot be combined with standard XFEM approaches. We therefore propose a solid shell model [Hauptmann and Schweizerhof 1998; Ko and Lee 2017] that uses bi-linear triangular prism elements which, as we show in our analysis, offer good accuracyperformance trade-offs for the range of problems that we consider while integrating seamlessly with XFEM.

Stripe Patterns in Material Design. Stripe patterns have recently seen increasing attention for material design and structural optimization. For example, Boddeti et al. [2020] use stripe patterns to guide the design of continuous reinforcement fields within a fiberin-matrix approach. Panetta et al. [2021] proposed an approach for designing inflatables that deploy into desired shapes upon pressurization. Stripe patterns are used for initializing air channels, which are then optimized such as to best approximate a given target shape at equilibrium. Jourdan et al. [2022b] likewise use stripe patterns for initialization when designing reinforcement curves for self-deploying fabric models. We share the excitement of these works for stripe patterns as a material design space. Instead of mere initialization, however, we propose a fully differentiable end-to-end pipeline enabling gradient-based design optimization within the space of stripe patterns.

3 OVERVIEW

The goal of our method is to automatically compute stripe patterns that, when used as bi-material distributions, lead to desired mechanical behavior. A visual summary of our method is shown in Fig. 2. To evaluate the performance of a design, we must compute its deformed equilibrium state given applied loads and boundary conditions using simulation. Determining the design parameters that best approximate a desired behavior requires inverting the entire design pipeline from input vector field to high-level mechanical function. This inversion notably includes derivatives of stripe patterns with respect to input vector fields, and derivatives of the equilibrium states with respect to material interfaces.

We start with a minimal description of Stripe Patterns [Knöppel et al. 2015] in Sec. 4, including an analysis of eigenvalue multiplicity. To use Stripe Patterns for inverse design of bi-material distributions, we propose a solid shell model with extended finite elements to allow for moving material interfaces (Sec. 5). Building on this computational model, we introduce Differentiable Stripe Patterns in Sec. 6, including eigenvector derivatives and design objectives. We present design examples and further analysis in Sec. 7.

4 STRIPE PATTERNS

The method by Knöppel et al. [2015] generates globally continuous, equally-spaced stripe patterns on arbitrary triangle meshes. It accepts as input a triangle mesh with *n* vertices $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ as well as an initial vector field $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)$ with each $\mathbf{z}_i \in \mathbb{R}^3$ indicating the vector field evaluated at vertex \mathbf{x}_i . The goal of Stripe Patterns is to find an angle α_i per vertex that indicates its phase, i.e., the location in a periodic function that determines which material will be assigned to each point. To make this angle function globally periodic and smooth, angles are represented as complex numbers $\Psi_i \in \mathbb{C}$ with $\alpha_i = \arg \Psi_i$. Stripes should be locally orthogonal to the input vector field, i.e., the gradient of the angle function should be collinear to the input vector field; $\nabla_{(u,v)}\alpha(u,v) = \mathbf{z}(u,v)$ where (u, v) are local coordinates. In the discrete setting, changes in angle are measured along edges \mathbf{e}_{ij} of the triangulation by linear interpolation and integration as

$$\omega_{ij} = \frac{1}{2} (\mathbf{e}_{ij}^T \mathbf{z}_i + \mathbf{e}_{ij}^T \mathbf{z}_j) \ . \tag{1}$$

In general, the change in angle cannot perfectly agree with the vector field, unless z is integrable. For this reason, the mismatch between target change and actual change along all edges is minimized as

$$E_{\Psi} = \sum_{ij \in \mathcal{E}} w_{ij} |\Psi_j - e^{\iota \omega_{ij}} \Psi_i|^2, \qquad (2)$$

where \mathcal{E} denotes the edge index set and w_{ij} is a mesh-dependent weight. To arrive at an algorithm with only real arithmetics, pervertex phases are expanded into real and imaginary parts as $\Psi_i = a_i + \iota b_i \rightarrow (a_i, b_i)$ which we store in a real-valued vector $\mathbf{v} \in \mathbb{R}^{2n}$. Furthermore, since the above energy is quadratic in phases, the realvalued analogue can be expressed as $\frac{1}{2}\mathbf{v}^T \mathbf{A}\mathbf{v}$ with a matrix \mathbf{A} that depends on the input mesh and vector field. To eliminate the trivial minimizer $\mathbf{v} = 0$, Knöppel et al. pose the constrained optimization problem

$$\min_{\mathbf{v}} \frac{1}{2} \mathbf{v}^T \mathbf{A} \mathbf{v} \quad \text{s.t.} \quad \frac{1}{2} \mathbf{v}^T \mathbf{B} \mathbf{v} = 1 , \qquad (3)$$

1:4 • Montes et al.



Fig. 2. Overview of our differentiable stripe pattern pipeline. Starting from an initial input vector field, we compute the corresponding stripe pattern by solving a generalized eigenvalue problem (Sec. 4). To accurately model the mechanics of bi-material surfaces structured with such patterns, we combine solid shell finite elements with XFEM (Sec. 5). To enable gradient-based minimization of high-level design objectives, we must compute derivatives of eigenvectors (Sec. 6), material interface, and equilibrium states with respect to vector field parameters **p**. These design gradients are then used to automatically compute stripe patterns that best approximate macromechanical performance goals.

where \mathbf{B} is a lumped mass matrix whose per-vertex entries are computed by summing up areas of incident triangles. Considering the first part of the first-order optimality conditions

$$\mathbf{A}\mathbf{v} - \lambda \mathbf{B}\mathbf{v} = \mathbf{0} , \qquad (4)$$

it is clear that solutions to this constrained optimization problem are generalized eigenvectors of **A**. Consequently, eigenvectors corresponding to the smallest eigenvalue will minimize the mismatch energy across all eigenvectors of **A**. This vector is then further processed through component-wise normalization, nonlinear pertriangle interpolation, and other filtering operations to yield pervertex *texture coordinates* that ultimately determine the *color*—or, in our case, material identity—of each point on the mesh. We refer to this process as the *forward stripe problem*. For our method, we aim to compute stripe patterns that lead to desired mechanical performance. Solving this *inverse stripe problem* involves computing derivatives of eigenvectors of **A** with respect to the input vector field, which is made challenging due to the following observation.

THEOREM 4.1. All generalized eigenvalues of A have geometric multiplicity two, i.e., all eigenvalues are duplicate and the dimension of the associated eigenspace is two.

PROOF. To see this, we observe that Eq. (2) measures only differences between phases along edges—rotating all per-vertex phases by the same angle will not change the energy. Consider a given eigenvector **v** and its corresponding phase vector Ψ . Let Ψ^{\perp} denote the phase vector defined as $\Psi_j^{\perp} = \iota \Psi_j$, i.e., with per-vertex phases rotated by $\pi/2$. The corresponding real-valued vector is \mathbf{v}^{\perp} with $\mathbf{v}_j^{\perp} = (-b_j, a_j)$. Since \mathbf{v}^{\perp} has the same norm and energy as **v**, it must also be an eigenvector for the same eigenvalue.

The consequence of this observation is that eigenvectors are not unique and derivatives not well-defined. Fig. 3 illustrate the space of eigenvectors for a given eigenplane; see also the accompanying video for an animation. It can be seen that motion in this eigenspace corresponds to globally coordinated phase changes, yielding waves travelling across the surface. To compute derivatives of stripe patterns with respect to the input vector field, we must make eigenvectors unique and their derivatives well-defined. Before we describe our solution to this problem in Sec. 6, we first introduce our computational model. FF



Fig. 3. Sequence of stripe patterns corresponding to eigenvectors from a two-dimensional eigenspace. Linear motion in this eigenspace corresponds to coordinated phase changes that yield travelling waves.

5 SOLID SHELLS AND XFEM

Stripe patterns give rise to piece-wise linear material interfaces which, in general, will cross through elements of the underlying triangulation. To predict the mechanical behavior of a given stripe pattern, interfaces between soft and rigid materials must be modeled accurately. Since we are interested in bi-material distributions with high stiffness ratio, simple interpolation of material parameters is not an option [Kharevych et al. 2009]. Conforming meshes would avoid material interpolation but changing interfaces require remeshing, which is not differentiable. For this reason, we opt for an extended finite element method (XFEM) that models material interfaces with additional degrees of freedom [Moës et al. 2003]. As its central benefit, this approach allows for continuous evolution of material interfaces without changes to the underlying mesh. The additional degrees of freedom, so called enrichment coordinates, allow for strain discontinuities across the interface, which is precisely what is needed in our context.

5.1 Solid Shells

To model structured surfaces that can stretch and bend, we combine XFEM with solid shell finite elements—volumetric elements in the shape of triangular prisms [Ko and Lee 2017]. Given a triangle mesh as input, we start by creating two offset surfaces by extruding midsurface vertices along their vertex normals. The resulting extruded surfaces model the top and bottom layers of the structured surface and are used to define a set of six-node triangular prism elements as shown in Fig. 4. We use a standard isoparametric approach and define the deformed geometry for each element as

$$\mathbf{x}(\mathbf{u}) = \sum_{i=1}^{6} \mathbf{x}_i N_i(\mathbf{u}) , \qquad (5)$$

and analogously for the rest state geometry X. In these expressions, $\mathbf{u} = (u, v, t)$ are local coordinates with (u, v) parametrizing the midsurface triangle and the third coordinate $t \in [-h/2, h/2]$ runs along the normal direction of the shell, with *h* indicating its thickness. The six bi-linear basis functions N_i are uniquely defined by the interpolation property $N_i(\mathbf{u}_j) = \delta_{ij}$. It is worth noting that, for any fixed *t*, the in-plane coordinates define a linear triangle element with constant strain. Even though deformation is constant for any given thickness value, it varies linearly through the thickness. It is this linear variation that, as we will see, allows solid shells to perform significantly better for bending deformation than constant-strain linear tetrahedra.



Fig. 4. Solid Shell Elements. *Left*: a bi-linear prism element is generated by extruding a triangle of the input mesh in the direction of its vertex normals. *Right*: level set values ϕ_i define the location of a material interface and additional enrichment DOFs $\hat{\mathbf{x}}_i$ allow for different strains on the two sides.

Using the continuous approximations of element geometry, all other kinematic quantities are obtained in the standard form, i.e.,

$$\mathbf{F}(\mathbf{u}) = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}|_{(u,v,t)} \quad \mathbf{C}(\mathbf{u}) = \mathbf{F}(\mathbf{u})^T \mathbf{F}(\mathbf{u}), \qquad (6)$$

where we made explicit the fact that, unlike for linear elements, the deformation gradient **F** and the right Cauchy-Green tensor **C** vary across the element.

Strain Energy. We use a standard Neo-Hookean constitutive law as the basis for both soft and stiff materials. The energy density of this material is defined as

$$\Psi_{\rm Iso} = \frac{1}{2} \left[\mu({\rm tr}({\rm C}) - 3) - 2\mu \ln J + \lambda (\ln J)^2 \right] , \qquad (7)$$

where λ and μ are the Lamé parameters. We use different applicationdependent material parameters for soft and stiff materials, see Sec. 7. If orthotropic behavior is desired, the isotropic base material can be augmented with stiffening fibers as

$$\Psi_{\text{Ortho}} = \Psi_{\text{Iso}} + \frac{1}{2}\beta_f \mathbf{n}_f^T \mathbf{C} \mathbf{n}_f , \qquad (8)$$

where \mathbf{n}_f is the unit vector indicating the fiber orientation and β_f is a material parameter that reflects the fiber stiffness and density; see, e.g., [Holzapfel and Gasser 2001].

We then obtain the elastic energy of a given solid shell element by integrating the strain energy density across the element. In practice, we approximate this integral using numerical quadrature,

$$U_e = \int_{\Omega} \Psi(\mathbf{C}(\mathbf{u})) \, dV \approx \sum_j w_j \Psi(\mathbf{C}(\mathbf{q}_j)) \left| \frac{\partial \mathbf{X}}{\partial \mathbf{q}} \right| \,, \tag{9}$$

where $\mathbf{q}_j = (u_j, v_j, t_j)$ are quadrature points in generic coordinates, and w_j are corresponding quadrature weights. For elements with a single material, we use a six-point numerical quadrature scheme with three points per thickness value. Elements crossed by a material interface require more elaborate treatment, as explained next.

5.2 Material Interfaces & XFEM

Level Sets for Material Interfaces. To model the bi-material distributions induced by stripe patterns, we must determine the location of the material interfaces. We model these interfaces using a level set ϕ defined through a set nodal values $\phi_i \in \mathbb{R}$. The nodal values represent the signed distance to the interface, with the sign indicating whether the node is on the soft or stiff side. To convert a given stripe pattern to its corresponding level set, we first retrieve the per-vertex angle $\alpha_i = \arg(\mathbf{v}_i)$ from the corresponding eigenvector component \mathbf{v}_i . We then compute level set values using a smoothed triangle wave as transfer function,

$$\phi_i = 1 - \frac{2\arccos[(1 - a_1)\sin(\alpha_i - \frac{\pi}{2})]}{\pi} - a_2, \tag{10}$$

where a_1 is a smoothing term and a_2 determines the ratio between stiff and soft material by vertically translating the cut-off value, see Fig. 5.



Fig. 5. Quasi-linear transfer function for mapping per-vertex angles α_i into level set values ϕ_i . Smooth junctions between linear segments ensure differentiability. The material interface is located at $\phi = 0$, $\phi > 0$ corresponds to stiff materials, whereas $\phi < 0$ indicates soft material. The ratio between stiff and soft material can be controlled by translating this curve vertically.

XFEM. Elements whose nodes have level set values ϕ_i with different signs are crossed by a material interface. To accurately model the behavior of these bi-material elements we resort to an extended finite element approach. While the location of the interface is given by the level set values ϕ_i , allowing for different strains on opposite sides of the interface requires additional degrees of freedom, the so called *enrichment coordinates* $\hat{\mathbf{x}}$. Enrichment coordinates $\hat{\mathbf{x}}_i$ are collocated with the standard nodal DOFs \mathbf{x}_i but have special basis functions.

We choose the ridge function [Moës et al. 2003], which was designed for material interfaces that are continuous in displacements but discontinuous in strain (C^0 but not C^1). For simplicity, we only allow one interface per element, although generalizations are possible [Zehnder et al. 2017]. This can be achieved by adjusting the



Fig. 6. XFEM discretization of stripe patterns with regions of stiff (*dark gray*) and soft (*light gray*) material as indicated. Elements crossed by an interface (*black*) are split into three sub-elements (*red*) to integrate their elastic energy.

frequency of the stripe pattern for a given input mesh, and viceversa. The ridge enrichment function is given by

$$\psi = \sum_{i} |N_i \phi_i| - |\sum_{i} N_i \phi_i|, \qquad (11)$$

with the corresponding interpolation function for the nodal positions _____

$$\mathbf{x} = \sum_{i} N_i(\mathbf{u}) \mathbf{x}_i + \psi \sum_{i} N_i(\mathbf{u}) \hat{\mathbf{x}}_i .$$
 (12)

We refer to Zehnder et al. [2017] for a visual construction of these functions. The strain in the enriched elements varies throughout the element, with a discontinuity located at the interface. To properly integrate over these enriched elements, we subdivide them into three sub-prisms and evaluate the sub-integrals numerically using an 11-point quadrature rule (Fig. 6).

5.3 Evaluation

Solid shells have, to the best of our knowledge, not been explored in graphics literature before. We therefore perform a series of experiments intended to test the accuracy and convergence behavior of solid shells in comparison to other models. In particular, we compare to conforming discretizations with four-node linear tetrahedron elements and ten-node quadratic tetrahedron elements, as well as conforming bi-linear prisms and a discrete shell model [Gingold et al. 2004]. We use two simple experiments in which we impose different periodic boundary conditions—cylindrical bending (radius r = 10cm) and uni-axial loading (stretch $\varepsilon = 10\%$)—onto a square plate with dimensions $7cm \times 7cm$.



Fig. 7. Problem setups for shell comparison. A square plate is endowed with a disc-shaped region of stiff material (*dark blue*) subject to periodic boundary conditions that impose cylindrical curvature (*left*) and uni-axial stretching (*right*) as indicated.

To investigate the impact of material interfaces, we use a discshaped region with stiffer material at the center of the plate. See

ACM Trans. Graph., Vol. 42, No. 4, Article 1. Publication date: August 2023.

also Fig. 7. We solve these problems numerically using different types of elements, different mesh resolutions, and plates of different thickness. We then compare the elastic energy and per-vertex displacement norms at equilibrium.

The results for the bending test are summarized in Tab. 1. Quadratic tetrahedra perform best and converge very rapidly under refinement. Because differences are very small, we only list energy and compute time for the coarsest discretization and use them as reference values. It can further be noted that the discrete shell model [Gingold et al. 2004] performs extraordinarily well, offering the best compromise between accuracy and computation time. The picture is very different for linear tetrahedra, which perform poorly and worst by far. Solid shell elements fare much more favorably in comparison. Especially the moderate resolution version seems to offer an interesting trade-off between accuracy and performance. Tab. 1 also shows that the performance of solid shell elements and linear tetrahedra degrades as the shell thickness decreases from 0.6mm to 0.15mm. While these observations imply that solid shells are not competitive for extremely thin shells, we are primarily interested in problems with thickness values above 1mm.

The results for uni-axial stretching underscore these initial impressions. As illustrated in Fig. 7, the stiff disk prevents homogeneous stetching which, in combination with a nonzero Poisson ratio, leads to spatially varying compressive stresses in the orthogonal direction that induce buckling. Fig. 8 shows results for this experiment in the form of normal displacement plots. Using again quadratic tetrahedra as ground truth, it can be seen from the plots that results for the discrete shell model are quite close to the reference solution, although convergence under refinement is somewhat unclear. Solid shells do not produce satisfying results for the lowest resolution, but converge to qualitatively acceptable solutions under refinement. In stark contrast, linear tetrahedra fail to produce out of plane motion altogether, which we attribute to the well-documented phenomenon of volumetric locking.



Fig. 8. Normal displacement plots for uni-axial loading. Almost all models produce out-of-plane buckling given sufficient resolution, whereas linear tetrahedra completely fail to capture this behavior.

Table 1. Comparison between different shell models on a cylindrical bending example.

Method	DOFs	Energy (0.6mm)	Energy (0.15mm)	Timing [s]
Quadratic Tets	73764	0.0596	0.0009	10.86
Linear Tets	12381	0.3542	0.0483	0.74
	47914	0.2072	0.0146	3.61
	244285	0.0951	0.0070	35.36
Solid Shells	11884	0.1215	0.0164	0.70
	46346	0.0751	0.0048	3.50
	183022	0.0635	0.0032	17.96
XFEM	12126	0.1210	0.0161	0.83
	46864	0.0752	0.0048	3.90
	184438	0.0635	0.0032	19.84
Discrete Shells	5942	0.0598	0.0009	0.56
	23173	0.0597	0.0009	2.60

Discussion. Our analysis shows that the discrete shell model offers very good accuracy already for comparatively coarse discretizationson par with quadratic tetrahedra, but at a much lower computational costs. While this makes them an ideal choice for static bi-material distributions, we were unable to find a robust formulation for moving interfaces with conforming meshes. Our experiments with different options invariably led to numerical instabilities: directly moving interface nodes along with the stripe pattern produces ill-shaped elements and diverging derivatives. Moving non-interface nodes using, e.g., a Laplacian regularizer [Montes et al. 2020; Pérez et al. 2017] delays these problems without solving them. While remeshing can avoid instabilities in principle, the corresponding discontinuities would prevent gradient-based optimizers from converging. These observations are in line with expectations as they are among the primary reasons for using XFEM instead of conforming discretizations in inverse problems with material interfaces [Kaufmann et al. 2009; Moës et al. 2003; Zehnder et al. 2017]. While discrete shells and XFEM would appear to be a winning combination, we could not find a way to reconcile these disparate concepts: extended finite elements rely on element geometry and strain being defined in terms of basis functions and their derivatives, but the curvature computation of discrete shell elements does not fit this framework. Ultimately, solid shells with XFEM emerge as the best trade-off for our problem setting. Unlike linear tetrahedra, they provide acceptable accuracy already for moderate resolution. Compared to quadratic tetrahedra with XFEM, solid shells are considerably simpler and, for the level of accuracy and mesh resolution that we require, the more efficient choice.

6 DIFFERENTIABLE STRIPE PATTERNS

Inverse material design based on stripe patterns leads to the central challenge of determining the change in stripes induced by a given change in input vector field \mathbf{z} . Since we only care about the direction of per-vertex vectors \mathbf{z}_i and not their magnitude, we parameterize the input vector field as $\mathbf{z} = \mathbf{z}(\mathbf{p})$ where $\mathbf{p} = (p_1, \dots, p_n)$ is the vector

of design parameters with per-vertex angles p_i . The relation between design parameters and output material distribution is then given implicitly through the generalized eigenvalue problem (4). However, since eigenvalues for this problem have geometric multiplicity two, the corresponding eigenvectors are not unique. We first show that eigenvector derivatives are not well defined before we describe our solution to this problem.

6.1 Eigenvector Derivatives

We start by expressing optimization problem (3) through its Lagrangian

$$\mathcal{L}(\mathbf{v},\lambda) = \frac{1}{2}\mathbf{v}^T \mathbf{A}\mathbf{v} - \frac{\lambda}{2} \left(\mathbf{v}^T \mathbf{B}\mathbf{v} - 1\right) .$$
(13)

Any solution $\mathbf{s} = (\mathbf{v}, \lambda)$ to this problem must satisfy the first-order optimality conditions $\mathbf{g}(\mathbf{v}, \lambda) := \nabla \mathcal{L} = \mathbf{0}$ where

$$\mathbf{g}_{v} = (\mathbf{A} - \lambda \mathbf{B})\mathbf{v} , \qquad (14)$$

$$\mathbf{g}_{\lambda} = -\frac{1}{2} \left(\mathbf{v}^T \mathbf{B} \mathbf{v} - 1 \right) \ . \tag{15}$$

Requiring that these conditions be satisfied for all admissible parameter changes,

$$\frac{d\mathbf{g}}{d\mathbf{p}} = \frac{\partial \mathbf{g}}{\partial \mathbf{p}} + \frac{\partial \mathbf{g}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{p}} = \mathbf{0}, \qquad (16)$$

leads to the saddle-point system

$$\begin{bmatrix} \mathbf{A} - \lambda \mathbf{B} & -\mathbf{B} \mathbf{v} \\ -(\mathbf{B} \mathbf{v})^T & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial v}{\partial p} \\ \frac{\partial \lambda}{\partial p} \end{bmatrix} = \begin{bmatrix} -\frac{\partial g_v}{\partial p} \\ 0 \end{bmatrix} .$$
(17)

Since λ has geometric multiplicity two, the system is singular and, consequently, the derivative of **v** is not well-defined. This is easily verified by observing that the vector (**v**_{\perp}, 0) is in the nullspace of $\frac{\partial g}{\partial s}$, where **v**_{\perp} is the second eigenvector for λ with **v**^{*T*}**v**_{\perp} = 0.

To resolve eigenvector ambiguity, we recall from Sec. 4 that vectors in the two-dimensional eigenspace correspond to synchronous rotations of per-vertex phase vectors. Using this insight, we can define unique eigenvectors by specifying the phase for an arbitrary vertex k. To this end, we simply set $\mathbf{v}_k = (a_k, 0)$ which eliminates one degree of freedom. This leads to a modified optimization problem with an extra constraint on b_k ,

$$\mathcal{L}(\mathbf{p}, \mathbf{v}, \lambda, \mu) = \frac{1}{2} \mathbf{v}^T \mathbf{A} \mathbf{v} - \frac{1}{2} \lambda (\mathbf{v}^T \mathbf{B} \mathbf{v} - 1) - \mu b_k, \qquad (18)$$

whose corresponding first-order optimality conditions are

$$\mathbf{g}_{\mathbf{v}} = \mathbf{A}\mathbf{v} - \lambda \mathbf{B}\mathbf{v} - \mu \mathbf{e}_{b_k} = 0 \tag{19}$$

$$\mathbf{g}_{\lambda} = -\frac{1}{2} (\mathbf{v}^T \mathbf{B} \mathbf{v} - 1) = 0 \tag{20}$$

$$g_{\mu} = -b_k = 0$$
. (21)

A vector **v** that satisfies these optimality conditions is also an eigenvector of $\mathbf{A} - \lambda \mathbf{B}$, since we can always find a vector in the eigenplane that satisfies the condition $b_k = 0$. Consequently, μ is always equal to 0. Requiring Eqs (19–21) to be satisfied for any change in parameters yields the modified saddle-point system

$$\begin{bmatrix} \mathbf{A} - \lambda \mathbf{B} & -\mathbf{B} \mathbf{v} & \mathbf{e}_{b_k} \\ -(\mathbf{B} \mathbf{v})^T & 0 & 0 \\ \mathbf{e}_{b_k}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{v}}{\partial \mathbf{p}} \\ \frac{\partial \lambda}{\partial \mathbf{p}} \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{g}_{\mathbf{v}}}{\partial \mathbf{p}} \\ 0 \\ 0 \end{bmatrix}, \quad (22)$$

where \mathbf{e}_{b_k} is the unit vector corresponding to the index b_k . Intuitively, there exists no vector from the eigenplane that is both orthogonal to \mathbf{v} and \mathbf{e}_{b_k} . The above system is therefore non-singular and can be solved for the unknown eigenvector derivatives $\frac{\partial \mathbf{v}}{\partial \mathbf{p}}$.

Eigenplane Parameterization. It should be noted that, while constraining a given phase variable leads to unique eigenvectors, any specific choice of k and b_k will introduce bias—another vector from the eigenplane might be better suited for decreasing the design objective. To avoid arbitrary choices, we introduce an additional parameter θ that represents the current eigenvector as a rotation in the eigenplane relative to the unique reference eigenvector defined through the choice of k and b_k .

6.2 Regularization and Parametrization

Vanishing Phase Values. The idea of stripe patterns is to express phase as the argument of complex number since this representation can offer global continuity. Once eigenvectors are computed, pervertex phases (complex numbers) have to be normalized to retrieve their argument (i.e., the real-numbered angle value). However, as phase vector tend towards zero, this operation becomes numerically unstable and, eventually, diverges. This is typically not much of a concern for the forward stripe pattern problem, since even numerically infinitesimal vectors can still be normalized. However, the sensitivities of angles diverge as per-vertex phases approach zero, which is very problematic for the inverse problem. Our solution to this problem is to simply disallow phase vectors from becoming arbitrarily small. We achieve this goal by imposing smoothly-clamped logarithmic barrier functions [Li et al. 2020] on per-vertex phases as,

$$R_{\text{sing}}(d, -\hat{d}) = \begin{cases} -(d - \hat{d})^2 \ln(\frac{d}{\hat{d}}) & 0 < d \le \hat{d} \\ 0 & \text{otherwise}, \end{cases}$$
(23)

where $d = |\mathbf{v}_i|$ and $\hat{d} = 0.1$ is a cut-off value. By preventing zero phases through a design objective, we ensure that eigenvectors still have the same meaning as before—they represent ideal approximations to potentially non-integrable vector fields. However, the design parameters are now strongly repelled from values that lead to eigenvectors with zero per-vertex phases.

An interesting question in this context is *what do we lose when preventing zero per-vertex phases*? As described by Knöppel et al. [2015], scaling per-vertex phases down to zero permits the introduction of singularities in the phase field that would otherwise lead to non-integrability. Our experiments indicate that the expressiveness of stripe patterns generated without such non-integrable components does not suffer. Indeed, while almost-zero phase vectors can be normalized without problems, they often lead to randomly oriented phases with significant curl. These high-curl regions translate into quasi-amorphous material distributions with isolated patches rather than clean stripes. We argue that such phase fields are neither useful nor desirable from a material design perspective.

Smoothness. While the regularizer introduced above prevents singularities in the phase field, the input vector field can still exhibit high-frequency components that translate into artifacts in the stripe patterns as shown in Fig. 9. To avoid such artifacts, we introduce

ACM Trans. Graph., Vol. 42, No. 4, Article 1. Publication date: August 2023.

an additional smoothness regularizer based on the formulation by Crane et al. [2010],

$$R_{\rm sm}(\mathbf{p}) = \sum_{i,j\in\mathcal{E}} w_{ij} \left(\left[\cos(p_i) - \cos(p_j) \right]^2 + \left[\sin(p_i) - \sin(p_j) \right]^2 \right) ,$$

where \mathcal{E} denotes the edge index set, w_{ij} is cotangent weight, and p_i , p_j are design parameters (i.e., angles) expressed in a common reference frame. As can be seen from Fig. 9 (*right*), this simple regularizer leads to substantially cleaner stripe patterns. It furthermore reduces singularities even before the log barrier penalty becomes active.



Fig. 9. Smoothness regularizer. By promoting smooth vector fields during optimization, this regularizer removes high-frequency noise (*left*), resulting in clean and aesthetically pleasing stripe patterns (*right*).

6.3 Design Objectives

Our differentiable formulation for stripe patterns can be used for solving a large array of design problems, see Sec. 7. In the following, we summarize the objective functions that we use for our examples.

Target Deformation. A basic task is to find stripe patterns such that the design best approximates a given target shape $\tilde{\mathbf{x}}$ in equilibrium. In its simplest form, the corresponding objective function is

$$T_{\text{match}} = |\mathbf{x}(\mathbf{p}) - \tilde{\mathbf{x}}|^2 .$$
(24)

Macromechanical Properties. Stripe patterns are ideal for creating materials with high stiffness contrast, e.g., by modulating the macromechanical properties of a soft (fabric) substrate with stripe patterns made from stiffer material. In this context, designers are typically interested in averaged, high-level behavior of the reinforced material, rather than in local deformations. In order to find stripe patterns that best approximate a given macromechanical behavior, we define an objective function that penalizes deviations between actual and target directional stiffness as

$$T_{\text{Mat}} = \sum_{i} |k(\theta_i, \mathbf{x}(\mathbf{p})) - \hat{k}_i|^2, \qquad (25)$$

where $\theta_i \in [0, \pi]$ are sample locations in polar coordinates with corresponding directional stiffness targets $\hat{k_i}$ given as generalized Young's moduli. To compute the directional stiffness for given design parameters, we simulate a unit-patch of the corresponding structure subject to periodic boundary conditions that enforce a given uniaxial target deformation. We obtain the corresponding stiffness $k(\theta_i, \mathbf{x}(\mathbf{p}))$ by evaluating the stress on the boundary of



Fig. 10. Controlling Macromechanical Properties. Starting from an initial concentric pattern (*a*), we optimize for patterns that lead to isotropic (*b*), orthotropic (*c*), and tetragonal (*d*) macromechanical behavior.

the unit cell and dividing by the strain magnitude as described by Schumacher et al. [2018].

Generalized Stiffness. Another basic design goal that we use in our examples is to achieve a desired stiffness with respect to given loads. When defining generalized stiffness as displacement norm divided by applied force magnitude, this objective becomes a special case of Eq. (24).

6.4 Optimization Algorithm

Our formulation allows us to compute derivatives of objective functions with respect to the design parameters. These gradients can then be used to drive first-order descent or quasi-Newton methods for minimization. We experimented with steepest descent and L-BFGS, but ultimately found the Globally-Convergent Method of Moving Asymptotes (GC-MMA) [Svanberg 1995] to be most efficient for our problem setting. We express the objective gradient with respect to the parameters as

$$\frac{dT}{d\mathbf{p}} = \frac{\partial T}{\partial \mathbf{p}} + \left(\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{v}} \frac{\partial \mathbf{v}}{\partial \mathbf{p}}\right)^T \left(\frac{\partial T}{\partial \boldsymbol{\phi}} + \frac{\partial \mathbf{x}}{\partial \boldsymbol{\phi}}^T \frac{\partial T}{\partial \mathbf{x}}\right), \qquad (26)$$

where $\frac{\partial \phi}{\partial v}$ is the derivative of the level set with respect to the eigenvector (Eq. 10), $\frac{\partial v}{\partial p}$ denotes the sensitivity of the eigenvector with respect to the parameters as illustrated in section 6.1 and $\frac{\partial x}{\partial \phi}$ is the sensitivity of the nodal positions with respect to the level set. Instead of fully evaluating both sensitivity matrices, we use the adjoint method for both terms sequentially, requiring just two linear solves per gradient evaluation.

Simulation Sensitivity. The map of the nodal positions as a function of the level set $\mathbf{x}(\boldsymbol{\phi})$ is given by the force equilibrium constraint of the forward simulation, $f = -\frac{dU}{d\mathbf{x}} = 0$. As a result, changes in $\boldsymbol{\phi}$ induce changes in the equilibrium configuration \mathbf{x} . Since $\frac{df}{d\boldsymbol{\phi}} = 0$ for every equilibrium configuration, we have

$$\frac{df}{d\phi} = \frac{\partial f}{\partial \phi} + \frac{\partial \mathbf{x}}{\partial \phi}^T \frac{\partial f}{\partial \mathbf{x}} = 0, \qquad (27)$$

from which $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\phi}}$ is extracted.

Evaluating Candidate Parameters. Whenever a candidate parameter update is evaluated during the optimization process, we must recompute eigenvectors at the new parameter location. To this end, we compute a pair of orthogonal eigenvectors $(\mathbf{v}_1, \mathbf{v}_2)$ as a basis for the eigenplane with minimal eigenvalue. We obtain the reference eigenvector by finding θ that satisfies the condition $b_k = 0$ in the equation

$$\mathbf{v} = \mathbf{v}_1 \cos(\theta) + \mathbf{v}_2 \sin(\theta) . \tag{28}$$

The current eigenvector is computed by rotating the reference vector by the new angle $\theta + \Delta \theta$ as given by the search direction. Having computed the new eigenvector in this way, the material interfaces are updated accordingly, and new equilibrium positions are computed using forward simulation.

7 RESULTS

We evaluate our method on a range of examples that demonstrate its potential to solve general inverse design problems for elastic surfaces structured with bi-material distributions. Statistics for all experiments are listed in Tab. 2.

Controlling Macromechanical Properties. In our first example, we explore stripe patterns as a design space for modulating the mechanical properties of an isotropic base material. To this end, we initialize our method with a radial vector field that generates concentric stripes on a square patch. We impose periodic boundary conditions and use homogenization to compute directional stiffness profiles based on the formulation by Schumacher et al. [2018]. As can bee seen from Fig. 10(a), despite its apparent symmetry, the tiling of this pattern is not rotationally symmetric and its stiffness is thus not isotropic. Given an isotropic stiffness profile as target, however, our method finds a concentric but slightly more rectangular pattern that leads to the desired behavior (Fig. 10(b)). As shown in Fig. 10(c) and (d), our method is likewise able to find modified patterns that yield orthotropic and tetragonal behavior, respectively.

Variable Stiffness Materials. In a second example, we use our method to design variable stiffness materials. We consider a rectangular patch clamped at two opposite boundaries as shown in Fig. 11. As design objective, we ask that displacements should vary linearly when applying a constant horizontal force density such as to encourage a stiff to soft distribution from the top to the bottom

boundary. We start with a initial material distribution of vertical stripes, for which the stiffness—measured as applied force magnitude divided by resulting displacement norm—is the same for both top and bottom boundaries. As can be seen from Fig. 11 (*b*, *right*), our method produces a pattern that yields the desired stiffness gradient. Visually, stripes have changed direction on the stiffer boundary (*top*) into an almost horizontal arrangement, whereas they remain largely vertical on the softer end (*bottom*). This non-trivial transition is enabled through a sequence of turning and branching points that, despite rather substantial changes, lead to an overall smooth and continuous pattern.

Compliant Gripper. While the previous examples have focused on the control of in-plane mechanical properties, Differentiable Stripe Patterns can likewise be used to drive large out-of-plane deformations upon actuation. To explore this application, we study an example inspired by Kirigami thin-shell grippers [Yang et al. 2021]. Starting from the parallel stripes design shown in Fig. 1 (*a*), we aim to optimize the pattern such that, when pulled along its horizontal axis, the wings of the sheet fold such as to minimize the distance between their tips. Actuating the initial design with parallel stripes does not result in sufficient lateral deflection (Fig. 1 (*c*)). For our optimized design, however, the same actuation produces the desired large out-of-plane motion that is able to lift a small 3D-printed model; see Fig. 1(*d*) and the accompanying video.

Soft Pneumatic Actuator. Our method can also be combined with other means of actuation to produce large controlled deformations. We demonstrate such a use case by designing a pneumatic actuatora basic building block for soft robotics applications. The actuator is made from a rectangular piece of textile structured with stiff stripe patterns. The reinforced design is connected along its long boundary and sealed at the far ends. We simulate inflation by applying constant pressure forces [Montes et al. 2020] and enforce periodic boundary conditions on both displacement and stripe patterns. Our design objective is to reach a target shape corresponding to a constantcurvature bend of 180°. As shown in Fig. 12, our method produces a smooth stripe pattern that closely approximates the target shape in simulation. One can distinguish two main mechanisms that enable this behavior: ribs running along the radial direction such as to allow axial stretch on the top surface while preventing an increase in radius; and a concentration of stiff material on the bottom surface that leads to differential axial stiffness and, consequently, a preferred bending direction. Our physical prototype confirms the feasibility of this design and achieves the targeted change in end-effector orientation, albeit with a somewhat larger lateral contraction.

Optimization of Insoles. Stripe patterns are ideal for designing materials with targeted directional deformations, and all examples shown so far relied on this unique capability. To analyze the robustness of our method with respect to design goals outside this space, we investigate an example that does not inherently require material anisotropy. To this end, we consider the design of a compliant shoe sole that exhibits low stiffness to normal loading in the heel and forefoot regions but offers higher stiffness in the midfoot region for increased stability. As can be seen in Fig. 13, our method finds a pattern layout that achieves these stiffness goals in simulation by

concentrating stiff material in the midfoot region. To evaluate this result quantitatively, the initial design shows average displacements in the normal direction of -3.66mm for heel and forefoot regions and -6.31mm for the midfoot. After optimization, these values have changed to -4.67mm and -2.52mm, respectively, indicating that the desired stiffness distribution has been achieved. It can further be observed that the quality of the stripes deteriorates as the optimization tries to increase the amount of stiff materials. Our smoothness and singularity regularizers put a bound to this trend and largely succeed in maintaining stripe integrity.

Garment Design. Fitted sportswear and medical garments rely on the ability to locally control stretch and stiffness. To investigate the potential of Differentiable Stripe Patterns in this context, we study the problem of designing an elastic sports shirt structured with stripe-shaped reinforcements. The shirt is fitted onto a torso model and, compared to its rest shape, must increase in area by about 10% to conform to the body. Stretch is therefore inevitable and we aim to route stripes such as to minimize the elastic energy of the shirt. The initial design with parallel stripes shown in Fig. 14(*left*) exhibits a high strain energy value since stiff reinforcements experience high stress with such a layout. Our optimization method finds a stripe pattern that decreases the energy by more than 25% with stripes that branch and meander such as to avoid long tension lines and alignment with principal stress directions. See Fig. 14(*right*).

Structural Optimization of Thin Shells. While we primarily designed our method to operate on bi-material distributions, Differentiable Stripe Patterns can also be used to generate geometric modifications for single material designs. We investigate this geometry mode on a structural optimization problem where we seek to increase the stiffness for a simple vase model subject to vertical loading as shown in Fig. 16. Instead of defining a bi-material distribution, we use stripe patterns with sinusoidal cross sections to generate normal displacements. Structuring surfaces with so called beads and groves to increase stiffness is a common strategy-also referred to as topography optimization-in engineering design. As shown Fig. 16 (3-4), the optimized design found by our method yields a structure with significantly improved axial stiffness that shows no sign of buckling even as we increase load by a factor of two (Fig. 16). As a visual interpretation, adding stripe-shaped normal displacements means that quasi-isometric bending modes for the plain design would induce substantial in-plane deformation in the optimized design. Thanks to its fully-differentiable nature, our method is able to exploit this effect, leading to structurally efficient and aesthetically pleasing patterns.

Choice of Initial Guess. As with any nonlinear, nonconvex optimization problem, there can be many local minima in the solution landscape. As a result, the pattern found depends on the choice of initial guess. We explore the effect of different initial guesses on an example that aims to modulate the directional stiffness of a reinforced fabric into a tetragonal profile. Starting from unidirectional and radial vector fields, we optimize for the same target and compare their results. As can be seen in Fig. 15, the symmetry of the concentric circles allows the pattern to comfortably reach the

Differentiable Stripe Patterns for Inverse Design of Structured Surfaces • 1:11



Fig. 11. Variable Stiffness Design. Comparison between an initial design with constant stiffness (*a*, *left*) and our optimized design with linearly varying stiffness (*a*, *right*). When applying the same force density, our optimized design (*b*, *right*) exhibits the desired stiffness gradient whereas the initial design shows constant stiffness as expected.



Fig. 12. Soft Actuator. Using a stretchable textile as substrate, we optimize for stripe-shaped reinforcements (*a*) such that, once sewn and sealed (*b*, *top*), the soft actuator deforms into a desired target shape upon pressurization (*c*). The prediction from our simulation model (*b*, *bottom*) is in good agreement with the physical actuator.



Fig. 13. Heterogeneous stiffness optimization. We optimize for stripe patterns such as to produce a shoe insole with a soft response to vertical loading in the heel and forefoot region while offering larger stiffness in the midfoot region. A bi-material stripe pattern with soft TPU (*green*) and stiff PET (*blue*) is placed on top of a soft infill (*a*). The stripe pattern is optimized to yield a soft response to vertical loading in the heel and forefoot region while offering larger stiffness in the midfoot region (*b*) such as to increase comfort when worn (*c*).

target with very few changes. In contrast, the unidirectional pattern, which exhibits an extreme initial stiffness profile, is forced to break its symmetry to better fit the target, without achieving the same level of success as the concentric circles.

8 CONCLUSION

We presented Differentiable Stripe Patterns, a computational approach that unlocks Stripe Patterns as a design space for structured surfaces. In order to invert the stripe pattern pipeline due to Knöppel et al. [2015], we have addressed several key challenges. First,

Example	# dofs	# iterations	Time per iter. [s]	Initial objective	Final objective	Figure
Isotropic	10481	55	21.41	391.81	38.04	Fig. 10(b)
Orthotropic	10481	310	21.22	7324.53	60.06	Fig. 10(c)
Tetragonal	10481	267	21.50	1721.02	21.92	Fig. 10(d)
Cloth	8138	66	193.97	776.17	642.59	Fig. 14
Vase	14026	592	25.68	2344.09	1957.59	Fig. 16
Actuator	11689	384	34.27	3574.64	901.40	Fig. 12
Gripper	6718	14	389.31	1486.14	311.59	Fig. 1
Var. Stiffness	11592	139	130.13	1031.89	12.49	Fig. 11
Insole	6681	357	126.10	209.17	103.76	Fig. 13

Table 2. Summary of parameters and performance statistics for all our experiments.



Fig. 14. Functional Elastic Shirt. Using a stretchable textile as substrate, we optimize for stripe-shaped reinforcements such as to minimize the strain energy of the garment. *Left*: initial design with parallel stripes in front and back view. Our optimized design (*right*) yields a significant decrease in energy.



Fig. 15. Influence of initial guess on optimization results. We optimize a reinforced fabric to exhibit a tetragonal stiffness profile using a unidirectional (*left*) and a radial (*right*) vector field as initial guess. Initial patterns and stiffness profiles are shown as bottom and top inset images, respectively.

we resolved ambiguities due to eigenvalue multiplicity by establishing a unique parameterization of the corresponding eigenplane, resulting in well-defined eigenvector derivatives. Second, to allow for accurate modeling of moving material interfaces, we proposed a combination of solid shells and extended finite elements. Finally, we introduced design space regularizers to avoid numerical singularities and improve stripe neatness. We combined these components with equilibrium state derivatives into an end-to-end differentiable pipeline that enables inverse design with high-level performance objectives. Our results indicate that stripe patterns are indeed a promising design space for bi-material surfaces, and that gradientbased optimization is an effective tool for exploring this space. There are, nevertheless, several limitations and corresponding opportunities for future improvements.

8.1 Limitations

We have assumed that our bi-material designs have material interfaces extending through the entire thickness. This assumption is a somewhat coarse approximation of our reinforced fabric examples, for which we added 3D-printed stripes *on top* of a textile substrate. While our results showed fairly good accuracy compared to real-world experiments, Jourdan et al. [2022a] have recently demonstrated that a thin shell model can be adapted to reflect this nonsymmetric patterning. We expect that similar modifications could likewise be effective for our solid shell model. Another alternative would be to use a bi-layer solid shell, with a homogeneous bottom layer and a structured top layer consisting of stiff and void material.

Our method is currently limited to a single linear cut per element. This is not much of a restriction in practice, since our focus is on lowfrequency stripe patterns and we can simply choose mesh resolution accordingly. An exception are branches, which involve two cuts in a single element that form an inward dent. Our method currently ignores this non-convex feature and approximates it with its convex closure. Nevertheless, extending our method to handle multiple cuts per element using the hierarchical scheme by Zehnder et al. [2017] should pose no major problems.

We believe that the combination of solid shells and XFEM offers unique advantages in terms of accuracy for computation time. In this work, we only explored constant thickness shells with two materials. However, solid shells could be useful for many other graphics applications where through-the-thickness deformation is an important effect, including simulation of skin, faces, and filled garments.

Finally, we have only started to explore the possibilities of stripe patterns for functional garment design. In the future, we would like to investigate the design of haptic garments that use stripe patterns to provide feedback on, e.g., posture [Vechev et al. 2022]. Extending passive stripes toward active control and sensing is likewise an exciting avenue for future work.



Fig. 16. Structural Optimization with Differentiable Stripe Patterns. From *left* to *right*: a 3D-printed thin-walled vase (1) buckles under a vertical load of 45N (2). To prevent this failure mode, we optimize for normal offsets in the form of stripe patterns such as to maximize the stiffness of the vase with respect to vertical loading (3). Our optimized design deforms much less under vertical loading (4) and shows no signs of buckling even for 90N (5).

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their valuable comments. This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 866480), and the Swiss National Science Foundation through SNF project grant 200021_200644.

REFERENCES

- Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin Van Hecke. 2017. Flexible mechanical metamaterials. Nature Reviews Materials 2, 11 (2017), 1–11.
- Gaurav Bharaj, David I. W. Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. 2015. Computational Design of Metallophone Contact Sounds. ACM Trans. Graph. 34, 6, Article 223 (Oct. 2015), 13 pages. https://doi.org/ 10.1145/2816795.2818108
- 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
- Narasimha Boddeti, Yunlong Tang, Kurt Maute, David W. Rosen, and Martin L. Dunn. 2020. Optimal design and manufacture of variable stiffness laminated continuous fiber reinforced composites. *Scientific Reports* 10 (12 2020). Issue 1. https://doi.org/ 10.1038/s41598-020-73333-4
- R. Bridson, S. Marino, and R. Fedkiw. 2005. Simulation of Clothing with Folds and Wrinkles. In ACM SIGGRAPH 2005 Courses (Los Angeles, California) (SIGGRAPH '05). Association for Computing Machinery, New York, NY, USA, 3–es. https: //doi.org/10.1145/1198555.1198573
- Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman. 2017. Dynamics-Aware Numerical Coarsening for Fabrication Design. ACM Trans. Graph. 36, 4, Article 84 (jul 2017), 15 pages. https://doi.org/10.1145/3072959.3073669
- Honglin Chen, Hsueh-TI Derek Liu, Alec Jacobson, and David I. W. Levin. 2020. Chordal Decomposition for Spectral Coarsening. ACM Trans. Graph. 39, 6, Article 265 (nov 2020), 16 pages. https://doi.org/10.1145/3414685.3417789
- Hsiao-Yu Chen, Arnav Sastry, Wim M. van Rees, and Etienne Vouga. 2018. Physical Simulation of Environmentally Induced Thin Shell Deformation. ACM Trans. Graph. 37, 4, Article 146 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201395
- Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly. 2021. Bistable auxetic surface structures. ACM Transactions on Graphics 40 (7 2021). Issue 4. https: //doi.org/10.1145/3450626.3459940
- Yu Ju (Edwin) Chen, David I. W. Levin, Danny Kaufmann, Uri Ascher, and Dinesh K. Pai. 2019. EigenFit for Consistent Elastodynamic Simulation across Mesh Resolution. In Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Los Angeles, California) (SCA '19). Association for Computing Machinery, New York, NY, USA, Article 5, 13 pages. https://doi.org/10.1145/3309486. 3340248
- Keenan Crane, Mathieu Desbrun, and Peter Schröder. 2010. Trivial Connections on Discrete Surfaces. Computer Graphics Forum (SGP) 29, 5 (2010), 1525–1533.
- Akash Garg, Eitan Grinspun, Max Wardetzky, and Denis Zorin. 2007. Cubic Shells. In Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (San Diego, California) (SCA '07). Eurographics Association, Goslar, DEU, 91–98.

- Yotam Gingold, Adrian Secord, Jefferson Y Han, Eitan Grinspun, and Denis Zorin. 2004. A discrete model for inelastic deformation of thin shells. In ACM SIG-GRAPH/Eurographics symposium on computer animation. Citeseer.
- Aldair E. Gongora, Kelsey L. Snapp, Richard Pang, Thomas M. Tiano, Kristofer G. Reyes, Emily Whiting, Timothy J. Lawton, Elise F. Morgan, and Keith A. Brown. 2022. Designing lattices for impact protection using transfer learning. *Matter* 5, 9 (2022), 2829–2846. https://doi.org/10.1016/j.matt.2022.06.051
- 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. Eurographics Association, 62–67.
- 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
- Christian Hafner, Christian Schumacher, Espen Knoop, Thomas Auzinger, Bernd Bickel, and Moritz Bächer. 2019. X-CAD: Optimizing CAD Models with Extended Finite Elements. ACM Trans. Graph. 38, 6, Article 157 (Nov. 2019), 15 pages. https: //doi.org/10.1145/3355089.3356576
- R. Hauptmann and K. Schweizerhof. 1998. A systematic development of 'solid-shell' element formulations for linear and non-linear analyses employing only displacement degrees of freedom. *Internat. J. Numer. Methods Engrg.* 42, 1 (1998), 49–69. https: //doi.org/10.1002/(SICI)1097-0207(19980515)42:1<49::AID-NME349>3.0.CO;2-2
- Gerhard A. Holzapfel and Thomas C. Gasser. 2001. A viscoelastic model for fiberreinforced composites at finite strains: Continuum basis, computational aspects and applications. Computer Methods in Applied Mechanics and Engineering 190, 34 (2001), 4379–4403. https://doi.org/10.1016/S0045-7825(00)00323-6
- Caigui Jiang, Florian Rist, Hui Wang, Johannes Wallner, and Helmut Pottmann. 2022. Shape-morphing mechanical metamaterials. *Computer-Aided Design* 143 (2022), 103146. https://doi.org/10.1016/j.cad.2021.103146
- David Jourdan, Skouras Mélina, Romero Victor, Etienne Vouga, and Adrien Bousseau. 2022a. Simulation of printed-on-fabric assemblies. Association for Computing Machinery (ACM), 1–11. https://doi.org/10.1145/3559400.3562001
- David Jourdan, Mélina Skouras, Etienne Vouga, and Adrien Bousseau. 2022b. Computational Design of Self-Actuated Surfaces by Printing Plastic Ribbons on Stretched Fabric. Computer Graphics Forum 41 (5 2022), 493–506. Issue 2. https://doi.org/10. 1111/cgf.14489
- David Jourdan, Mélina Skouras, Etienne Vouga, Adrien Bousseau, D Jourdan, M Skouras, E Vouga, and A Bousseau. 2021. Printing-on-Fabric Meta-Material for Self-Shaping Architectural Models. AAG 2020-Advances in Architectural Geometry. , 19 pages. https://hal.inria.fr/hal-02925036
- Peter Kaufmann, Sebastian Martin, Mario Botsch, Eitan Grinspun, and Markus Gross. 2009. Enrichment Textures for Detailed Cutting of Shells. ACM Trans. Graph. 28, 3, Article 50 (jul 2009), 10 pages. https://doi.org/10.1145/1531326.1531356
- Lily Kharevych, Patrick Mullen, Houman Owhadi, and Mathieu Desbrun. 2009. Numerical coarsening of inhomogeneous elastic materials. ACM Transactions on graphics (TOG) 28, 3 (2009), 1–8.
- Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2015. Stripe patterns on surfaces. ACM Transactions on Graphics 34. Issue 4. https://doi.org/10.1145/2767000
- Yeongbin Ko and Phill Seung Lee. 2017. A 6-node triangular solid-shell element for linear and nonlinear analysis. Internat. J. Numer. Methods Engrg. 111 (9 2017), 1203–1230. Issue 13. https://doi.org/10.1002/nme.5498
- Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid Deployment of Curved Surfaces via Programmable Auxetics. ACM Trans. Graph. 37, 4, Article 106 (July 2018), 13 pages. https://doi.org/10.1145/3197517.3201373
- Dan Koschier, Jan Bender, and Nils Thuerey. 2017. Robust extended finite elements for complex cuting of deformables. ACM Transactions on Graphics 36. Issue 4.

1:14 • Montes et al.

https://doi.org/10.1145/3072959.3073666

- Kurt Leimer and Przemyslaw Musialski. 2020. Reduced-Order Simulation of Flexible Meta-Materials. Proceedings - SCF 2020: ACM Symposium on Computational Fabrication. https://doi.org/10.1145/3424630.3425411
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM Trans. Graph. 39, 4 (2020), 49.
- Hsueh-Ti Derek Liu, Alec Jacobson, and Maks Ovsjanikov. 2019. Spectral Coarsening of Geometric Operators. ACM Trans. Graph. 38, 4, Article 105 (jul 2019), 13 pages. https://doi.org/10.1145/3306346.3322953
- Zhenyuan Liu, Jingyu Hu, Hao Xu, Peng Song, Ran Zhang, Bernd Bickel, and Chi-Wing Fu. 2022. Worst-Case Rigidity Analysis and Optimization for Assemblies with Mechanical Joints. *Computer Graphics Forum* 41, 2 (2022). https://doi.org/10.1111/ cgf.14490
- 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 (Dec. 2018), 14 pages. https://doi.org/10.1145/3272127.3275076
- Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. 2016. Procedural voronoi foams for additive manufacturing. ACM Transactions on Graphics (TOG) 35, 4 (2016), 1–12.
- Jonàs Martínez, Mélina Skouras, Christian Schumacher, Samuel Hornus, Sylvain Lefebvre, and Bernhard Thomaszewski. 2019. Star-shaped metrics for mechanical metamaterial design. ACM Transactions on Graphics (TOG) 38, 4 (2019), 1–13.
- Jonàs Martínez, Haichuan Song, Jérémie Dumas, and Sylvain Lefebvre. 2017. Orthotropic k-nearest foams for additive manufacturing. ACM Transactions on Graphics (TOG) 36, 4 (2017), 1–12.
- Nicolas Moës, Mathieu Cloirec, Patrice Cartraud, and J-F Remacle. 2003. A computational approach to handle complex microstructure geometries. *Computer methods in applied mechanics and engineering* 192, 28-30 (2003), 3163–3177.
- Juan Montes, Bernhard Thomaszewski, Sudhir Mudur, and Tiberiu Popa. 2020. Computational Design of Skintight Clothing. ACM Trans. Graph. 39, 4, Article 105 (jul 2020), 12 pages. https://doi.org/10.1145/3386569.3392477
- Ella Moore, Ioannis Karamouzas, and Michael Porter. 2018. Precision control of tensile properties in fabric for computational fabrication. Proceedings - SCF 2018: ACM Symposium on Computational Fabrication. https://doi.org/10.1145/3213512.3213514
- Przemyslaw Musialski, Christian Hafner, Florian Rist, Michael Birsak, Michael Wimmer, and Leif Kobbelt. 2016. Non-linear Shape Optimization Using Local Subspace Projections. ACM Trans. Graph. 35, 4, Article 87 (July 2016), 13 pages. https: //doi.org/10.1145/2897824.2925886
- William Neveu, Ivan Puhachov, Bernhard Thomaszewski, and Mikhail Bessmeltsev. 2022. Stability-Aware Simplification of Curve Networks. In ACM SIGGRAPH 2022 Conference Proceedings (Vancouver, BC, Canada) (SIGGRAPH '22). Association for Computing Machinery, New York, NY, USA, Article 20, 9 pages. https://doi.org/10. 1145/3528233.3530711
- Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. 2021. Computational inverse design of surface-based inflatables. ACM Transactions on Graphics 40 (7 2021). Issue 4. https://doi.org/10.1145/3450626. 3459789
- J. Panetta, M. Konakoviundefined-Lukoviundefined, F. Isvoranu, E. Bouleau, and M. Pauly. 2019. X-Shells: A New Class of Deployable Beam Structures. ACM Trans. Graph. 38, 4, Article 83 (July 2019), 15 pages. https://doi.org/10.1145/3306346. 3323040
- Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case stress relief for microstructures. ACM Transactions on Graphics (TOG) 36, 4 (2017), 1–16.
- Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic textures for additive fabrication. ACM Transactions on Graphics (TOG) 34, 4 (2015), 1–12.
- 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. https://doi.org/10.1145/3072959.3073695
- Stefan Pillwein and Przemysław Musialski. 2021. Generalized deployable elastic geodesic grids. ACM Transactions on Graphics 40 (12 2021). Issue 6. https: //doi.org/10.1145/3478513.3480516
- Abhinit Sati, Ioannis Karamouzas, and Victor Zordan. 2021. DIGISEW: Anisotropic Stitching for Variable Stretch in Textiles. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3485114.3485121
- 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 (Tog) 34, 4 (2015), 1–13.
- 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
- Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing Inflatable Structures. 33, 4 (2014).

- Krister Svanberg. 1995. A globally convergent version of MMA without linesearch. In Proceedings of the first world congress of structural and multidisciplinary optimization, Vol. 28. Goslar, Germany, 9–16.
- Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational Design of Linkage-Based Characters. ACM Trans. Graph. 33, 4, Article 64 (July 2014), 9 pages. https://doi.org/10.1145/ 2601097.2601143
- Davi Colli Tozoni, Jérémie Dumas, Zhongshi Jiang, Julian Panetta, Daniele Panozzo, and Denis Zorin. 2020. A low-parametric rhombic microstructure family for irregular lattices. ACM Transactions on Graphics (TOG) 39, 4 (2020), 101–1.
- Thibault Tricard, Vincent Tavernier, Cédric Zanni, Jonàs Martínez, Pierre-Alexandre Hugron, Fabrice Neyret, and Sylvain Lefebvre. 2020. Freely orientable microstructures for designing deformable 3D prints. ACM Trans. Graph. 39, 6 (2020), 211–1.
- Nobuyuki Umetani, Jun Mitani, and Takeo Igarashi. 2010. Designing Custom-made Metallophone with Concurrent Eigenanalysis. In Proceedings of the International Conference on New Interfaces for Musical Expression. Zenodo, 26–30. https://doi.org/ 10.5281/zenodo.1177917
- Velko Vechev, Juan Zarate, Bernhard Thomaszewski, and Otmar Hilliges. 2022. Computational Design of Kinesthetic Garments. Computer Graphics Forum (2022). https://doi.org/10.1111/cgf.14492
- Yi Yang, Katherine Vella, and Douglas P. Holmes. 2021. Grasping with kirigami shells. Science Robotics 6, 54 (2021), eabd6426. https://doi.org/10.1126/scirobotics.abd6426 arXiv:https://www.science.org/doi/pdf/10.1126/scirobotics.abd6426
- 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. https://doi.org/10.1145/2897824.2925888
- Jonas Zehnder, Espen Knoop, Moritz Bächer, and Bernhard Thomaszewski. 2017. Metasilicone: design and fabrication of composite silicone with desired mechanical properties. ACM Transactions on Graphics (TOG) 36, 6 (2017), 1–13.
- Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-scale topology optimization with microstructures. ACM Transactions on Graphics (TOG) 36, 4 (2017), 1.