Cut-Cell Microstructures for Two-scale Structural Optimization

DAVI COLLI TOZONI, nTop / New York University, USA
ZIZHOU HUANG, New York University, USA
DANIELE PANOZZO, New York University, USA
DENIS ZORIN, New York University, USA

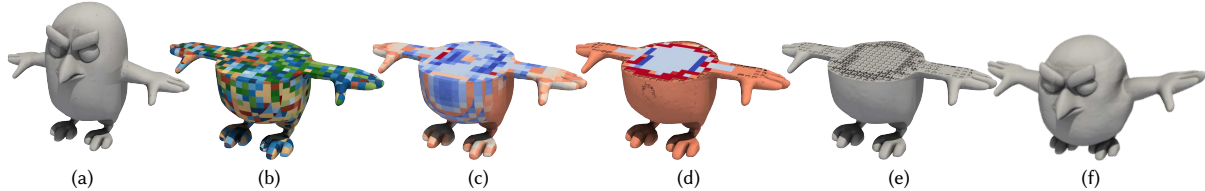(a)   (b)   (c)   (d)   (e)   (f)

Fig. 1. (a) Our method starts with an arbitrary initial surface mesh. (b) We generate a regular 3d grid enclosing the object, further subdivided into tetrahedra with the surface of the tetrahedral mesh conforming to the object's boundary. (c) We perform optimization for a particular deformation behavior (wings going up by a prescribed distance with pressure on the top) and obtain a material distribution (the color shows Young's modulus). (d) The cut-cell microstructure is inserted into boundary cells to match the Young's modulus computed by the optimization, and then a second optimization is performed only on interior cells to compensate for the change in the deformation due to deviation of effective elastic properties of cut-cell microstructure from the optimized Young modulus (e) The full-cell microstructures with effective Young modulus and Poisson ratio matching the optimized values are inserted into the interior cells (f) The deformation result on the final model.

Two-scale topology optimization, combined with the design of microstructure families with a broad range of effective material parameters, is increasingly widely used in many fabrication applications to achieve a target deformation behavior for a variety of objects. The main idea of this approach is to optimize the distribution of material properties in the object partitioned into relatively coarse cells, and then replace each cell with microstructure geometry that mimics these material properties. In this paper, we focus on adapting this approach to complex shapes in situations when preserving the shape's surface is important.

Our approach extends any regular (i.e. defined on a regular lattice grid) microstructure family to complex shapes, by enriching it with individually optimized cut-cell tiles adapted to the geometry of the cut-cell. We propose an automated and robust pipeline based on this approach, and we show that the performance of the regular microstructure family is only minimally affected by our extension while allowing its use on 2D and 3D shapes of high complexity.

1 INTRODUCTION

Additive manufacturing enables easy fabrication of complex geometric structures designed with specific effective material properties in mind, with a broad range of applications from lightweight but strong structures in aerospace to shoe soles, prosthetic devices, and flexible robot parts [10, 32]. While the more complex geometry can be easily handled by modern 3D printing software and hardware, optimized design of such objects is a fast-growing research area. Most traditional engineering analysis and optimization software is not suitable for this task, due to limitations on the complexity and scale of geometry that can be handled.

Authors' addresses: Davi Colli Tozoni, nTop / New York University, USA, davi.tozoni@nyu.edu; Zizhou Huang, New York University, USA, zizhou@nyu.edu; Daniele Panozzo, New York University, USA, panozzo@nyu.edu; Denis Zorin, New York University, USA, dzorin@cs.nyu.edu.

A popular approach is to directly optimize the shape of the entire structure by using topology optimization and/or shape optimization. However, the computation cost of these direct methods is high, especially for large objects, often requiring HPC clusters [1]. An alternative is to use a two-scale approach: (1) a material optimization is performed on a voxelized grid to compute material properties varying between voxels and resulting in desired target deformation behavior, and (2) each voxel is replaced by a geometrical structure (microstructure tile) with effective properties matching the computed solid material properties in the voxel. The advantage of this strategy, compared to whole object high-resolution topology optimization, is that step (1) can be very efficient and step (2) can be reduced to a lookup table: a family of microstructure tiles, mapping material properties to geometry, which can be precomputed using a cluster, and then used for multiple designs.

Unavoidably, the voxelization of a shape leads to artifacts in the boundary, becoming problematic in applications where the shape's boundary matters, which is common in many settings, from moving engine parts to prosthetic devices. Trivial approaches, such as trimming the microstructure to follow the boundary or filling the voxels on the shape's boundary, lead to a major change in effective microstructure properties in this cell, and may result in significant deviations from the intended behavior under surface loads (see Figure 2 and 3 for a comparison and Section 4 for a detailed analysis). To address this problem, these approaches have been extended, in 2D only, from regular grids to boundary-aligned structured meshes [46]. Still, the strong requirement for mesh uniformity and quality makes them less robust than the grid-aligned methods, which are still the standard de facto in industry [32]. In 3D, such an extension is much more difficult: while planar quad mesh generation is possible (although quads may be far from parallelograms required by [46]), generating boundary-aligned hex meshes is still an open problem,
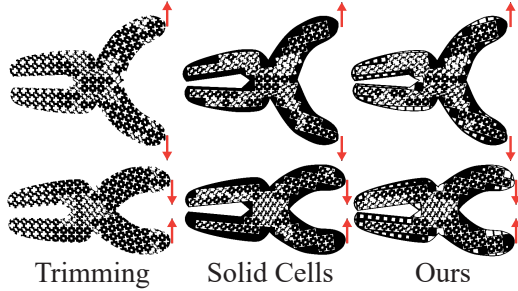
Fig. 2. Two examples comparing simpler baselines, trimming (left) and solid cells (middle), with our approach (right). The scissor is optimized to close when the handles are pulled apart (top row) or pulled together (bottom row). Our approach succeeds in both cases, while trimming performs very poorly, and solid cells are somewhere in-between. We provide a detailed comparison against the solid cells baseline in Section 4.
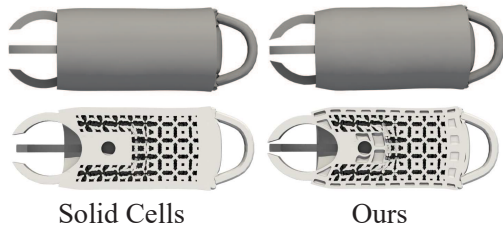

Fig. 3. Comparison between the solid baseline (left) and our method (right) in 3D. The top row shows both versions seen from the outside (as someone using the gripper would see it), while the bottom row shows section cuts of both shapes. The structure is optimized so that its jaws close together under a compression force on the handle (gripper's right side). The baseline example fails to reach the prescribed deformation.

with a recent state-of-the-art method yielding a 50% success rate on a recent benchmark [23].

We propose a novel family of cut-cell microstructures to extend existing 2D or 3D regular tiling microstructure families to shapes with a complex boundary. Our construction ensures that the shape boundary exactly matches the input (with the exception of tunnels needed to remove the internal material after printing), and, in combination with an interior structure, leads to accurate matching of the prescribed deformation behavior. Our algorithm is the first method enabling two-scale shape optimization on very complex shapes in 3D, in a fully automated and robust way, with the resulting complex geometric structures including the input boundary precisely.

We demonstrate its effectiveness by combining our cut-cell family with a variant of the microstructure family proposed in [46] to realize complex 3D shapes with controllable material properties, and evaluate their properties by fabricating them using 3D printing and testing them in a set of controlled load experiments.

To summarize, the contributions of this paper are:

- An extended microstructure construction that includes a structure for boundary cells which ensures their connection to the interior cells and supports arbitrary surface geometry.

- An algorithm for two-scale structural optimization, optimizing the material properties for interior and boundary cells, and a mapping from material to shape parameters of the microstructure, in two and three dimensions, with a refinement optimization pass.
- A robust geometric pipeline for constructing microstructure in-fill for arbitrary input shapes.
- Quantitative and qualitative physical evaluation of the optimized structures.

2 RELATED WORK

Microstructure design and optimization. There is extensive literature on the design of elastic and other types of metamaterials/microstructures. We refer to the books, e.g., [11, 29, 45] for the foundational work and the recent survey [21] for more up-to-date references.

The approaches to the design of periodic microstructures with prescribed homogenized properties rely on topology optimization, going back to [5, 6], and shape optimization, as well as combinations of the two. Examples in computer graphics literature include [40] (topology optimization) and [34] for shape optimization, with many more in studies in material science and engineering. These works, as well as, e.g., [52], consider constructing families spanning a broad range of elastic properties, with the potential maximally achievable ranges characterized in [30].

Global topology optimization. In [1, 24, 49], topology optimization was scaled up to high-resolution uniform and adaptive 3D grids. [48] performs high-resolution topology optimization with additional constraints to create an evenly distributed porous small-scale structure minimizing compliance for specific loading scenarios. Even with these improvements, direct topology optimization at the microstructure level remains computationally expensive: e.g., a modest cell resolution of 32^3 combined with an equally modest 32^3 resolution at the coarse level requires over 1 billion cells to work with. To avoid these high computational costs, we favor a two-scale optimization approach.

Two-scale optimization using microstructures. Two-scale optimization uses microstructures to optimize the deformation behavior of objects efficiently by separating the problem into two scales through the partitioning of an object into cells. Fine-scale structures for individual cells are often precomputed to yield a particular range of effective material properties within the limit of infinitely fine cells. The coarse-scale optimization is performed by treating the whole cells as made of homogeneous material. Some works, e.g., [25], do not precompute microstructures and optimize at both macro- and microscale simultaneously. This approach, while most flexible, is computationally very expensive and does not support well typical workflows. Recent surveys [50] and [36] provide an overview of the approaches and recent work in this domain.

All two-scale optimization approaches rely on tiling the input shapes with microstructures. The most common approach, and the current standard de facto in industry, is voxelization. Papers using voxelization focus on rectangular structures, as examples [34, 53]. Other works [12, 37, 51] approximate arbitrary shapes with

collections of cubes. This approach is unsuitable for applications that require preserving the shape of the surface (e.g., a wheel would not roll) or its appearance. This problem is exacerbated for relatively coarse grids of cells, which are often unavoidable due to limitations in the minimal beam thickness in many 3D printing technologies: an exception is [38], where an expensive and slow printing process is used to reduce the size of voxels. Yet, despite the high resolution, the resulting surfaces are still not smooth.

The conforming tiling construction is introduced in [46] for 2D and [47] for 3D, where a voxelization is replaced by a boundary adapted, isotropic, low distortion quad or hexahedral tiling, and the domain of the microstructure families is extended accordingly. In addition to the increased complexity in the family design, the main drawback of these methods is that structured meshes with these properties are challenging to generate for arbitrary geometries. While some solutions exist for 2D [46], the solution to conforming hexahedral mesh generation in 3D is a challenging open problem [4], where the failure rate of state-of-the-art methods is shown to be around 50%. Finally, the approach [13–15] for 2D and surface structures can be viewed as a partial conforming tiling construction: it uses a simple rectangular microstructure with two parameters, and a frame field to define the directions and scale for cells from a simple family. The frame field orientations and other fields are chosen by optimization. Similarly to the previous category, this approach is also difficult to generalize reliably to 3D volumes due to a more complex structure of 3D fields [35].

Our approach shares the benefits of conforming tiling constructions, as it allows reproduction of boundaries for complex geometries, but is doing so by extending the voxelization methods using a cut-cell approach. Our algorithm combines the robustness and simplicity of the voxelization methods with the advantages of conforming tiling constructions without requiring a difficult-to-build quad or hexahedral mesh.

Special Shape-Adapted Microstructures. [22] uses a special type of 2D triangular auxetic structure to effect conformal surface deformations. This method requires domain meshing with triangles close to regular. Similarly, a recent paper [26] uses 2D spiral microstructures. As an alternative to periodic microstructures, [27, 28] construct randomized printable structures with control over Young's moduli both for isotropic and anisotropic target properties. However, it cannot independently control the Poisson's ratio, and the behavior of randomized structures is less controllable overall.

Variable Base Material. Our focus is fabrication methods supporting single base material due to their wide availability, lower cost, and support for stiff materials (metal). Having access to controllable base material further widens the range covered by microstructures: [7] designs and fabricates objects satisfying an input deformation using actual material properties variation, with fabrication done using a multi-material printer. [43] applies discrete material optimization to achieve desired deformations of complex characters, also using multi-material printing for fabrication.

3 METHOD

3.1 Material Optimization Primer

We provide a short introduction to material optimization and to two-scale optimization using microstructures to keep the paper self-contained. We refer to [34, 40] for more details.

Let Ω be the domain of a solid. Loads (Neumann boundary conditions) and/or Dirichlet boundary conditions are applied on some parts of Ω, typically on the boundary. The deformed state is obtained by solving the linearized elasticity equation for the displacement u,

$$-\nabla \cdot [C : \varepsilon(\mathbf{u})] = 0 \quad \text{in } \Omega, \tag{1}$$

where C_{ijkl} represents the 4th order elasticity tensor and $\varepsilon(\mathbf{u}) := \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the linearized Cauchy strain tensor of the averaged deformation [1]. The boundary conditions have the form $\mathbf{u}|_{\partial\Omega_D} = \mathbf{u}_0$, where $\mathbf{u}_0$ are prescribed boundary displacements on the part of the boundary $\partial\Omega_D$ and $C : \varepsilon(\mathbf{u})|_{\partial\Omega_N} = \mathbf{f}_0$, where $\mathbf{f}_0$ are external forces applied to the part of the surface Ω_N.

Our goal is to minimize a function $\mathcal{E}$ of the displacement u. For example, to obtain a desired deformation of a part of the object's boundary under a certain load condition (Figure 1), $\mathcal{E}$ is the L_2 norm of the difference between u and the target deformation. We do this by changing the spatially varying material properties (Young's modulus and Poisson ratio) encoded in the spatially varying tensor C. The problem is constrained by the linearized elasticity PDE in (1):

$$\min_{C} \mathcal{E}(\mathbf{u}, C), \text{ subject to (1).} \tag{2}$$

This problem is closely related to widely used topology optimization formulations [6]. Additional constraints are usually added to this problem to force, in every point in space, either homogeneous material properties or zero density, thus obtaining an object with homogeneous material properties and possibly holes. These constraints make this large optimization problem even harder to solve.

The two-scale approach. A two-scale optimization approach [34, 46] uses a microstructure family $\mathcal{P}$ to make solving this problem orders of magnitude cheaper. $\mathcal{P}$ is a map from an elasticity tensor C (or equivalently material properties such as Young's modulus and Poisson's ratio) to a periodic geometric pattern with the same homogenized stiffness tensor (see Figure 4). The periodic patterns are typically characterized by a set of shape parameters. Equipped with $\mathcal{P}$, the two-scale optimization solves the problem in two steps. First, a material optimization problem, in which the per-voxel value of the stiffness tensor is optimized, is solved on a voxel grid at a coarse scale:

$$\min_{C \in D} \mathcal{E}(\mathbf{u}, C), \text{ subject to (1),} \tag{3}$$

where D is the domain of the map $\mathcal{P}$, which is a subset of all stiffness tensors that can be realized by using a family of cell microstructures. After solving this small problem (e.g., this problem has around 500 dofs for the 2D examples in Figure 15 and can be solved within seconds), the map $\mathcal{P}$ is used to replace the cell with the corresponding microstructure geometry; the resulting object with complex geometric structure can be fabricated using a single material.

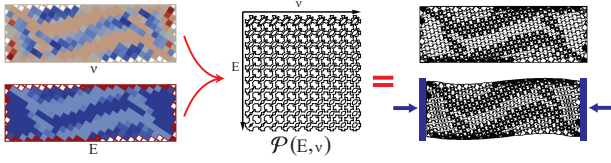[1]The notation $A : B$ is used to denote the contraction $\sum_{k,l} A_{ijkl} B_{kl}$.

Fig. 4. Two-scale optimization pipeline. (left) we run material optimization on a bar with square cells; (middle) a map $\mathcal{P}$ is used to map material properties to geometry; and (right) we obtain the final shape, which deforms as expected.
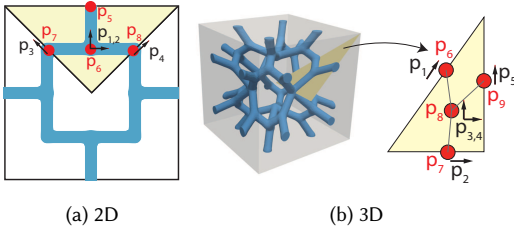

(a) 2D   (b) 3D

Fig. 5. Cell parameters for 2D and 3D, interior cells.

The two-scale approach described above has two important challenges: (1) designing the microstructure family $\mathcal{P}$ and (2) reducing the error introduced in the object's shape due to a coarse voxel grid, as fabrication constraints typically do not allow one to make the voxels too small. Our focus in this paper is on the latter problem, which has received little attention in the literature. For (1), we borrow and slightly modify the family proposed in [46], as detailed in Section 3.5.

3.2 Overview

We describe our approach to computing an infill structure producing desired deformation behavior for an object for 3D, but the same construction works for 2D: we highlight the required modifications in Section 3.8. In Section 4, we demonstrate how these steps contribute to the quality of the results. The approach is composed of five steps:

(1) **Cell Partition.** We produce a cell partition of the domain Ω by intersecting it with a regular grid; the partition consists of cubic (square, in 2D) cells in the interior and cut cells, obtained by intersecting a cubic cell with the interior of the object. Each cell is partitioned into tetrahedra, with the same material property variable C_i assigned to all tetrahedra corresponding to a cell.

(2) **Initial Material Optimization.** We solve the problem defined in Equation 2, with constant properties per cell, computing material parameters C_i by minimizing the objective $\mathcal{E}$ (see Equation 4), with material parameter bounds for the interior cells determined by the domain of $\mathcal{P}$, and for boundary cells inferred from cell geometry. For interior cells, (E, ν) are used as material parameters. For boundary cells, E is the only optimized material parameter and we keep ν fixed at the base material value.

(3) **Generation of Cut-Cell Geometry.** For the boundary cells, we use a one-parametric family of microstructures, with a
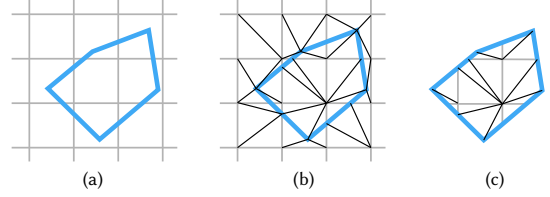

Fig. 6. Illustration of cell partition. (a) Initial surface mesh with a background regular grid. (b) Triangular/Tetrahedral mesh of the bounding box. (c) Final mesh after removing triangles/tetrahedra outside of the surface mesh.

single parameter for each cell, the interior void size. We determine this parameter for each boundary cell so that the effective material properties approximately match the target material and compute the cell geometry for this void size (Section 3.5).

(4) **Interior Material Refinement.** We build a hybrid geometry using solid interior cells with variable material C_i assigned to each interior cell but the actual microstructure geometry from Step 2 for the boundary cells. For this new volumetric mesh, with the complex structure on the boundary only, we repeat the material optimization for interior cells, but, this time, with material properties fixed to the base material for already inserted boundary cell geometry and variable material properties for the internal cells (Section 3.6).

(5) **Surface Extraction.** The geometry of the boundary cells is merged with the microstructure geometry produced from optimized material properties from the previous step via the map $\mathcal{P}$. Optionally, we add tunnels to the exterior to allow fabrication using 3D printing methods requiring residual material removal (Section 3.7).

3.3 Step 1. Cell Partition

The input for this step is a surface mesh and a background regular grid, with each grid cell corresponding to a microstructure tile in the final structure. The output is a tetrahedral mesh of the interior of the surface mesh, with the regular grid cells split into tetrahedra. This mesh is used for the first material optimization.

This problem could be solved by computing a 3D arrangement [17] or by cut-cell meshing tools [2, 44], followed by postprocessing to tesselate the interior of each cell with tetrahedra while ensuring a conforming mesh at the cell boundaries.

As shown in Figure 6, we opted for a more direct and robust solution, creating a tetrahedral mesh with fTetWild [18], while providing both the surface mesh and grid faces as input. The fTetWild tool outputs a tetrahedral mesh of the bounding box of the input mesh. We then filter the tetrahedra outside the object surface using the winding number [20], and assign the resulting tetrahedra to the cell of the grid containing its barycenter. Eventually, we obtain a tetrahedral mesh of the surface, while each tetrahedron is inside a grid cell. We use a geometrical tolerance of 10^{-3} for fTetWild. This approach is insensitive to surface mesh imperfections and simultaneously allows us to maintain the high mesh quality needed for simulation.
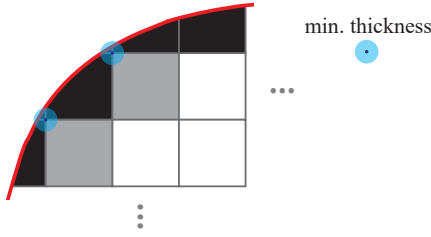
Fig. 7. Interior Cells that are too close to the boundary may cause thin features in the shape. To avoid issues during printing and non-smoothness in displacement behavior, we consider those as boundary cells. The drawing shows boundary cells (in black) touching the (red) boundary and two quad cells (in dark gray) that are too close to the boundary, according to a min. thickness parameter.
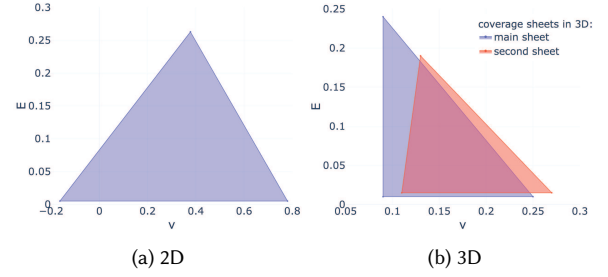

(a) 2D   (b) 3D

Fig. 8. Coverage of material parameter properties in the families we use. Our coverage regions are approximated to triangles, so they can be easily added as constraints in our optimization.
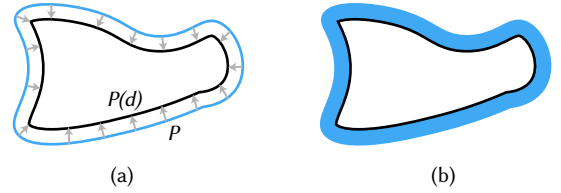

(a)   (b)

Fig. 9. Illustration of creating cavity to reach a certain volume ratio based on the surface offset. (a) The surface P is offset by d to get $P(d)$. (b) Volume inside $P(d)$ is dropped to match target volume ratio r.

To satisfy the minimum thickness requirement in 3D printing, the maximum size of the internal voids of boundary cells is restricted. However, some boundary cells might be too thin even without internal voids. If a thin boundary cell is adjacent to an interior cell, after the interior cell is replaced by the microstructure, this boundary cell may violate the minimum thickness. To avoid this, we change the type of some interior cells into boundary cells, although they are not cut by the boundary surface. Specifically, we compute the distance between the surface cutting a boundary cell and its internal boundary faces, edges, and vertices, shared with other internal cells; if the distance is too small, we transform all internal cells sharing this element into a boundary cell. See Figure 7 for an example.

3.4 Step 2. Initial Material Optimization

We solve problem (3) with the following functional:

$$\mathcal{E}(\mathbf{u}, C) = \int_{\partial \Omega^t} (\mathbf{u}(C) - \mathbf{u}^t)^2 dA + w \int_{\Omega} \|\nabla C\|^2 dA, \qquad (4)$$

where $\partial \Omega^t$ is typically a part of the boundary of our domain for which we prescribe target displacements $\mathbf{u}^t$, w is a weight to adjust the smoothness of the material distribution. The constraints on the problem consist of a set of linear bounds for the domain of material properties. For interior cells, we constraint the material properties ((E, ν) to be inside the domain of $\mathcal{P}$, shown in Figure 8. For boundary cells, we add constraints on Young's modulus, preventing the creation of features smaller than a predefined thickness.

As seen in Equation 4, our objective has two terms. The first corresponds to the deformation error, which measures the difference between the simulated displacement $\mathbf{u}(C)$ and the target $\mathbf{u}^t$. The second term is a regularizationterm, aiming to achieve a smoother spatial material variation, which helps to reduce large differences in microstructure geometry when transitioning between neighboring cells, which may lead to local deformation artifacts.

The problem is discretized using the finite element method [39], and the adjoint method is used to compute the gradient of the target functional with respect to elasticity parameters [19]. The optimization itself is performed using L-BFGS with box constraints.

3.5 Step 3. Generation of Cut-Cell Microstructure

While we can use $\mathcal{P}$ to find microgeometry for the internal cells, the construction of the map from material properties to geometry is based on periodic-material homogenization and requires cells to be cubes: in this step, we provide a construction of a microstructure family for boundary cut cells with arbitrary geometry.

Our approach is to create a 1D bijective smooth map $\mathcal{P}_b$ between volume fraction r of the filled part of a cell and Young's modulus E, using the fact that there is usually a strong connection between stiffness and solid material volume fraction (see Subsection 3.5) in simple cell geometries with a single void, while the shape of the void has less impact. We can obtain such single-void one-parametric families for arbitrary cell shapes by defining our microstructure geometry to be solid between the polygonal boundary P of the cell and an offset $P(d)$ of P (Figure 9), where $d \geqslant d_{\min}$ is a parameter interpolating between the minimal wall thickness ($d_{\min}$) and a fully filled cell. We construct $P(d)$ for an arbitrary cell shape using a signed distance function (SDF). The cell boundary mesh P is converted to an SDF (we use OpenVDB [31]), and then we call the **offset** function in OpenVDB, and finally convert the offset SDF back to a mesh. Figure 10 shows three examples of boundary cell geometry in real models. Notice that, in 3D, we may need to add additional openings to avoid enclosed voids, as discussed in Section 3.7.

Next, we determine the correspondence between volume fraction r and the offset d for a specific cut cell geometry. Denote the volume enclosed by surface P as V, and the volume enclosed by surface $P(d)$ as $V(d)$. Given a fixed r, our goal is to find the offset value d so that

$$V(d) = (1 - r)V.$$

(a) Boundary cell with different volume ratio

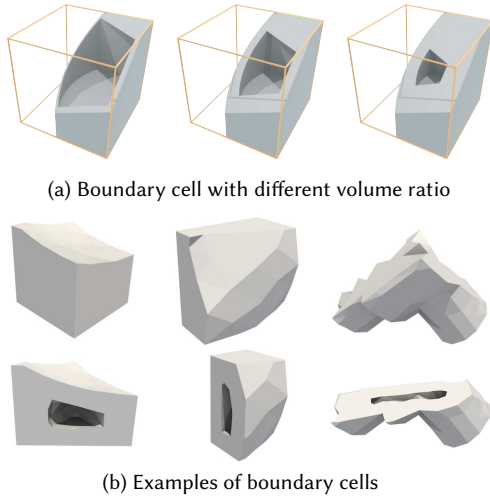(b) Examples of boundary cells

Fig. 10. Boundary cell geometry in 3D. (a) The cross-section of a cut-cell enclosed by the grid partition in yellow. (b) The bottom row shows the cross-section of the top row.

(a) 2D

(b) 3D

Fig. 11. Volume ratio to average Young's modulus map. The left figure represents the map for 2D microstructures, where each color represents a different shape of the opening, while the figure on the right represents the map in 3D.

This is a 1D root-finding problem for a monotonically increasing function, so we use a binary search to find the root d with a relative tolerance of 0.01.

From material properties to volume fraction. The relationship between the effective Young's modulus and volume ratio, in general, depends on the shape of the cell and, in the case of non-tileable cells, may even be difficult to define precisely, as it is unclear how to perform correct homogenization in this case (homogenization requires periodic tiling). However, a rough approximation of this dependence can be obtained from a one-parametric family of complete cubic cells, which can be tiled periodically. In this case, we can compute the effective Young's modulus by averaging Young's modulus on orthogonal directions (x, y, z) (obtained from the homogenized elasticity tensor), for a variety of interior void shapes, to verify that the dependence on the shape of the void is not strong (Figure 11). As a result, we have a function $E(r)$ which we invert to map the optimized Young modulus values C_i (obtained at Step 2 for boundary cells) to the volume ratios r. For a specific choice of cell boundary, we determine the offset d from r as described above, building a complete map from C_i to a boundary cell microstructure.

The need for precise knowledge of the effective material properties of a cell is made less critical for this stage by the next step, which compensates for the errors introduced in the boundary cells.

3.6 Step 4. Interior Material Refinement

To increase the accuracy of the solution, we run a second round of material optimization fixing the geometry and material properties of the boundary cells, and optimizing only the regular, interior cells.

Given a target volume ratio for each boundary cell determined by their target Young modulus computed in Step 1, we determine the offset for the cell microstructure in Step 3. We perform the offsets for all boundary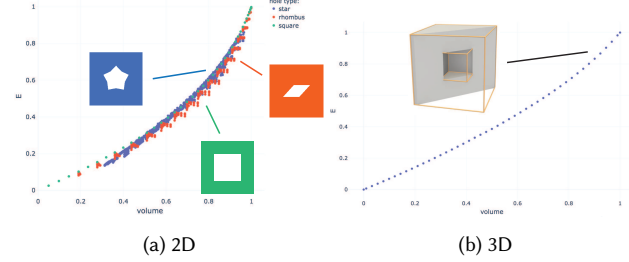 cells in parallel, append the offset surface mesh to the original mesh, and then the approach of Step 2 is used again to generate a new volumetric mesh.

On this mesh (which has explicitly meshed cut-cell microstructures on the boundary and regular grid cells in the interior), we solve a material optimization problem with fixed base material in boundary cells (as their microstructure is already integrated into mesh geometry). While the overall complexity of the optimization increases, as the simulation needs to be done on a finer mesh in boundary cells, it is still much lower than the complexity of direct microstructure shape or topology optimization, as the number of optimization variables is lower. In principle, the process can be iterated, by switching boundary and interior cell optimization, but using explicit geometry for interior cells is expensive, and little additional benefit can be expected.

Note that converting the Young's modulus to volume fraction in boundary cells is not precise resulting in the change of deformations relative to the optimized ones. As a consequence, the objective value obtained by the first optimization, can be significantly increased by this transformation. While the second optimization brings it back closer to the originally obtained values, even after the second optimization the objective may be higher than after the first one (Table 1) – but it is always lower than the objetive after the cut-cell microstructure insertion.

3.7 Step 5. Surface Extraction

To extract the surface from implicit volumetric functions (see Section 3.9) of internal and boundary cells, we first compute the union of all interior cell SDFs and intersect it with the SDF obtained in Step 5 which included boundary cell geometry only. The surface mesh is extracted from the resulting SDF using the marching cubes method. We use a resolution of $50 \times 50 \times 50$ per cell for the SDF (note that this effective resolution is used in a sparse volumetric representation in OpenVDB).

By construction, our boundary cells have a solid layer covering the model surface, which will prevent uncured resin or unmelted powder to be removed after fabricating the model. To make the model fabricable, we add tunnels to each boundary cell (Figure 12). We first go over boundary cells with internal voids and consider the parts B_E of their surfaces that belong to the outer surface. If the sphere centered at the barycenter of the internal void and with

(a) 2D view of tunnels
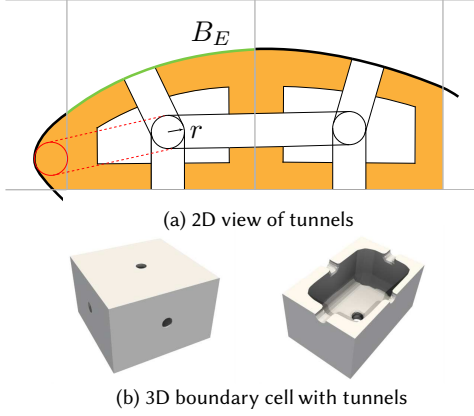

(b) 3D boundary cell with tunnels

Fig. 12. Tunnels connecting boundary cells. (a) The gray lines represent the grid partition, B_E for one boundary cell is shown in green, the tunnels are cylinders with radius r, solid parts are shown in yellow. The red circle is too large to fit into the boundary cell, so we fill the cell completely. (b) A boundary cell with internal void and tunnels (left) and its cross-section (right).

radius equal to the tunnel radius intersects B_E, constructing a tunnel to the exterior is impossible, and we completely fill this cell. We connect the centers of every two adjacent boundary cells with voids by a tunnel, and these segments form a graph consisting of disjoint connected components. For each such connected component, we create a cylindrical tunnel from one void center to the object's surface. To remove the materials inside interior cells, at least two tunnels are created from internal cells to adjacent boundary cells with voids. To avoid the decrease in volume caused by creating tunnels, we pick the smallest possible radius as long as the 3D printing materials can be removed.

3.8 2D Algorithm

Our 2D pipeline closely follows the 3D pipeline, using Triangle [41] instead of fTetWild, and using our own implementation of marching squares as OpenVDB does not support 2D domains. An advantage of Triangle is that it allows us to represent the boundaries exactly, without introducing a geometric approximation error. Tunnels are not added to the 2D boundary cells as they are not necessary for fabrication.

3.9 Construction of $\mathcal{P}$

We use the method introduced in [46] for creating microstructure families for the interior cells both for 2D and 3D. However, we do not use their rhombic shape parameter, as in our case we will use only square/cubic cells. We briefly review this construction here.

First, *a homogenization map* $H(\mathbf{p})$ is obtained for a low-parametric family of microstructures, with parameters p_i shown in Figure 5, consisting of graph vertex positions and radii at these vertices determining how thick the microstructure is at this location. The method of [33] is used to construct the implicit curve/surface bounding the microstructure geometry from the graph and radii.
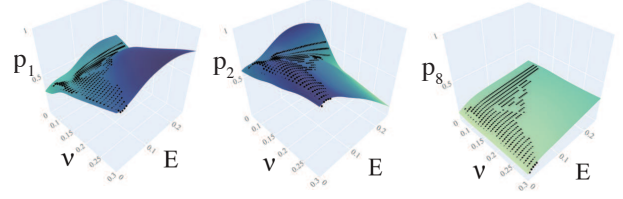

Fig. 13. Examples of shape parameter dependence on materials in our 3D family. These maps correspond to our main coverage sheet.

We run a sweep of the parameter values in a given range $[p_i^{min}, p_i^{max}]$ for each parameter p_i. Consider B_p the n-dimensional box of admissible parameter values. Each combination of values is used to generate a microstructure cell geometry using the implicit surface definition of [33]. The value of $H(\mathbf{p})$ is obtained as the effective value of the elasticity tensor parameters (with additional symmetry conditions on our structures, there are only 3 independent parameters in the tensor, Young's modulus E, Poisson's ratio v and the shear modulus G, so the function $H(\mathbf{p})$ maps $\mathbf{p}$ into $\mathbb{R}^3$.

The map $H(\mathbf{p})$ is not bijective as it typically maps a higher-dimensional space into a lower-dimensional space. In [46], a sequence of dimension reductions is used to identify a subspace D in the space of shape parameters equal to the dimension of the elasticity material parameter space (3 in our case). The map $H(\mathbf{p})$ restricted to this subspace may still be not bijective, but its image $H(B_p)$ can be partitioned into parts on which it is invertible; specifically, we construct simplices in the subspace of shape parameters and split $B_p \cap D$ into connected subdomains D_j by simplex orientation. On each of D_j, we can invert $H(\mathbf{p})$. We further restrict it to the two-dimensional space of isotropic material parameters parameterized by (E, v) to obtain the *material-to-geometry* map $\mathcal{P}_i(E, v) : H(D_i) \rightarrow B^p$. Figure 8 presents the coverage regions for our 2D and 3D families. Notice that for 3D we have two different material coverage sheets, with the blue one covering most of the space and being chosen as our main material range.

Figure 13 shows examples on the final map from material properties (v and E) to geometric parameters p_i.

4 EVALUATION

Our algorithm is implemented in C++ and uses Eigen [16] for the linear algebra routines, PolyFEM [39] for finite element simulation in 3D and MeshFEM [34] in 2D, fTetwild [18] and triangle [42] for meshing, and Pardiso [3, 8, 9] for solving linear systems. We run our experiments on a cluster node with an Intel Cascade Lake Platinum 8268 processor limited to 32 threads.

4.1 2D examples

We experimented with six 2D models shown in Figure 15. Visually, our results closely match the specified target, both in direct simulation (done on a dense mesh of the final geometry) and in our physical validations. For physical validation, we fabricated the models using FDM printing with a TPU 95A filament on a Prusa i3 MK3S and an Ultimaker 3 (see Figures 16 and 17). Numerical results,
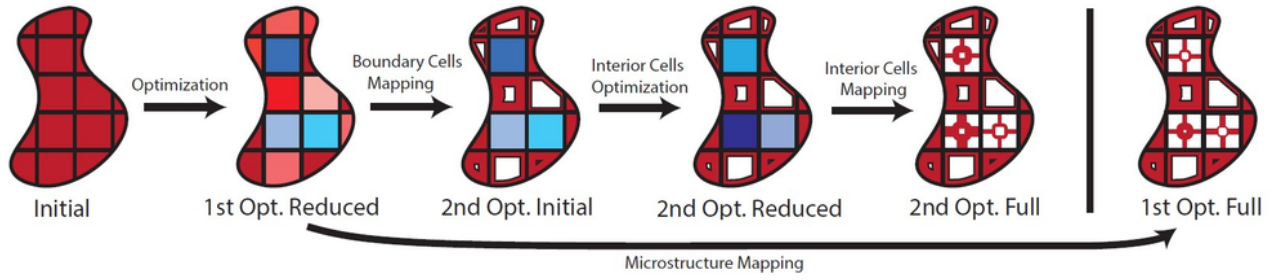
Fig. 14. Summary of our method, showing the main steps of our algorithm. Quantitative results of each step are shown in Table 1. Dark red represents our base material. Different tones of red represent variable material on the boundary, while tones of blue are used for material properties of interior cells.
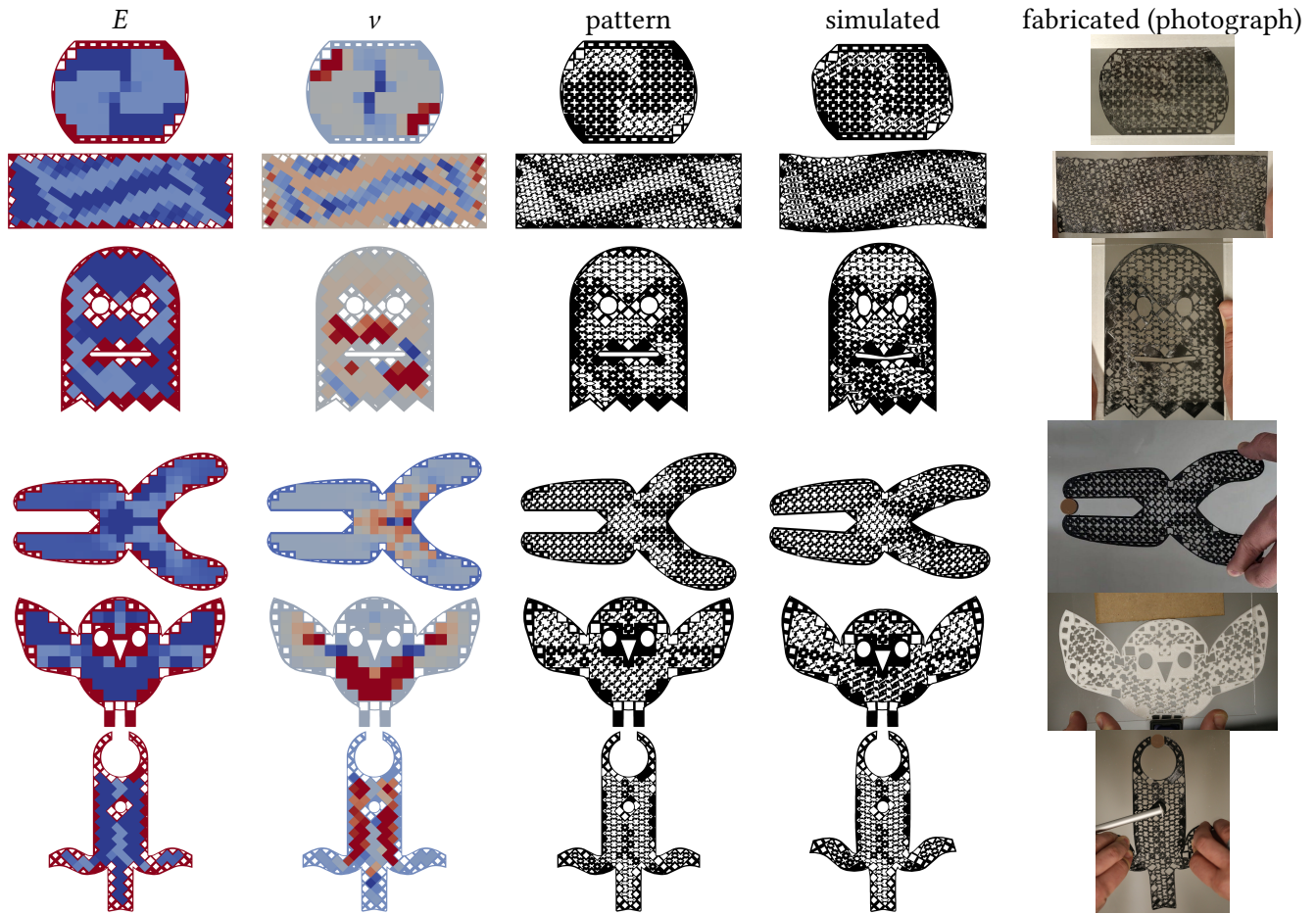

Fig. 15. A gallery of 2D examples. From left to right: optimized material distribution (Young's modulus and Poisson's ratio), final geometry at rest, deformed geometry (simulated) and photograph of deformed pattern (fabricated). From top to bottom: **Disk.** The objective is to obtain a shear-like deformation, bulging on the top left and bottom right, whenever the disk is compressed in the vertical direction. **Sine bar.** The bar is optimized to match a sine wave shape. **Happy Ghost.** The pacman ghost shape is optimized to smile when compressed on the sides. We add target displacements such that the middle region (of both the top and bottom) of the mouth displace down. **Pliers.** The pliers is optimized to close when the handles are pulled together. **Bird.** The shape is optimized to flap its wings whenever its head is pushed down. In this case, we add a Dirichlet boundary condition on the bird's feet to keep it in place. **Sword Gripper.** The gripper has a rounded hole in the middle that is used to hold the shape in place and should not deform during the operation. The mechanism closes whenever the handle moves up.

Fig. 16. Fabricated pliers (black TPU). Rest shape on the left; moving the handles apart in the middle; compressing the handles to hold the object on the right.
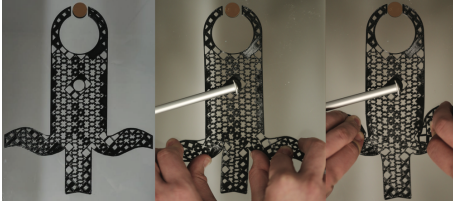

Fig. 17. Fabricated sword gripper (black TPU). Rest shape on the left; moving handles back in the middle; moving handles to the front, holding the object, on the right.

breaking down the effect of the different optimization stages, are available in Table 1 and discussed in more detail below.

The target displacement for the six examples is described in the caption of Figure 15. Additionally, we show two deformation sequences for the plier (Figure 16) and sword gripper (Figure 17).

4.2   3D Examples

Our algorithm reliably handles complex geometrical models. We show a gallery of our models and some fabricated results below. Numerical results for these models are shown in Table 18, which show a similar trend to the 2D results discussed above. For physical validation, we fabricated the gripper and bar example with TPU 88A, using the 3D printing service Xometry.

Sine Bar. The first two rows in Figure 18 show an example of a bar optimized to match a sine function when compressed. In the first row, the tiling orientation is aligned with the bar, while in the second row, we rotate the tiling to obtain similar deformation. This example shows that our method is not very sensitive to the orientation. Figure 20 shows the fabricated result of the first row, which matches closely with our simulated result.

Gripper. The third row in Figure 18 shows an example of a gripper, with four hands on the left and a handle on the right. We assign very stiff material to the handle (to avoid any deformation) and apply a compression force to it. The structure is optimized so that the gripper closes when the force is applied to the handle. We show a fabricated gripper in Figure 20, with force applied.

Shoe sole. The fourth row in Figure 18 shows an example of a box with a curved top surface. Its structure is optimized so that the deformed top surface is as flat as possible for a uniform applied force.

Bird. The last row in Figure 18 shows a bird model whose structure is optimized so that its wings go upwards as its head is pushed down.

Batch Examples. To further illustrate the robustness and generality of our algorithm, we show an additional gallery of 10 examples in Figure 19, computed in batch mode using a fixed objective and fixed boundary conditions. The model surface in the top 5% of the bounding box is fixed, and the surface in the bottom 5% is compressed by 10% using Dirichlet boundary conditions. The optimization maximizes the horizontal displacement under deformation integrated over the middle 10% surface. We observe that the algorithm generates intricate geometries (for reference, the size of the tetrahedral meshes used for the final simulation varies between 0.5 and 15 million tetrahedra), adapting to the complex boundaries while filling the interior with the microstructure geometry.

4.3   Alternative methods

We compare our method against two simple baseline approaches to handle boundary cells and the method of [46] (applicable only to 2D). In addition, we perform an ablation study to evaluate the effect of the second optimization.

Trimmed boundary microstructures. A simple approach to extend two-scale microstructures to shapes with a complex boundary is trimming the parts outside the boundary shape [40]. The downsides are obvious: arbitrarily cutting a microstructure may rxesult in dramatic uncontrollable changes in the effective material properties, as well as entirely disconnected structures, in addition to not having a surface layer, one of the major goals of our method. We show a comparison between this method and ours in Figure 2, where it is possible to see that the method does not generally behave well – the results strongly depend on how the grid lines intersect the object.

Solid boundary cells. An alternative, which produces a surface layer, is filling all boundary cells with the base material, i.e. forcing the volume fraction in boundary cells to be 1 in the optimization. However, this adds uncontrollable stiffness to the object, especially in thinner parts. The numerical evaluation in Table 1 shows that our approach consistently outperforms this method in both reduced (i.e., using solid cells with variable material properties, as in the output of Step 2) and full (using full microstructure geometry) simulations. Additional comparisons can also be seen in Figures 2 and 3.

Rhombic microstructures, 2D only. [46] describes a method for partitioning a mesh to cells close to rhombic and conforming to the boundary, and constructs a suitable microstructure family. This eliminates the need for a separate cut-cell microstructure family, but is hard to generalize to 3D as it requires high-quality conforming quad meshing. We compare four 2D optimization problems (disk, bar, ghost, and pliers, Figure 15) with this approach, making the best effort to replicate the problem setups, which is not entirely possible due to differences in formulation. Specifically, our objective computes an integral of the target error over the object's boundary, which is the original surface for our case, while for [46] it corresponds to the microstructures' interface areas for every quad in the boundary since there is no solid boundary. We observe in Figures 21 and 22 that the results are similar between the two methods. We
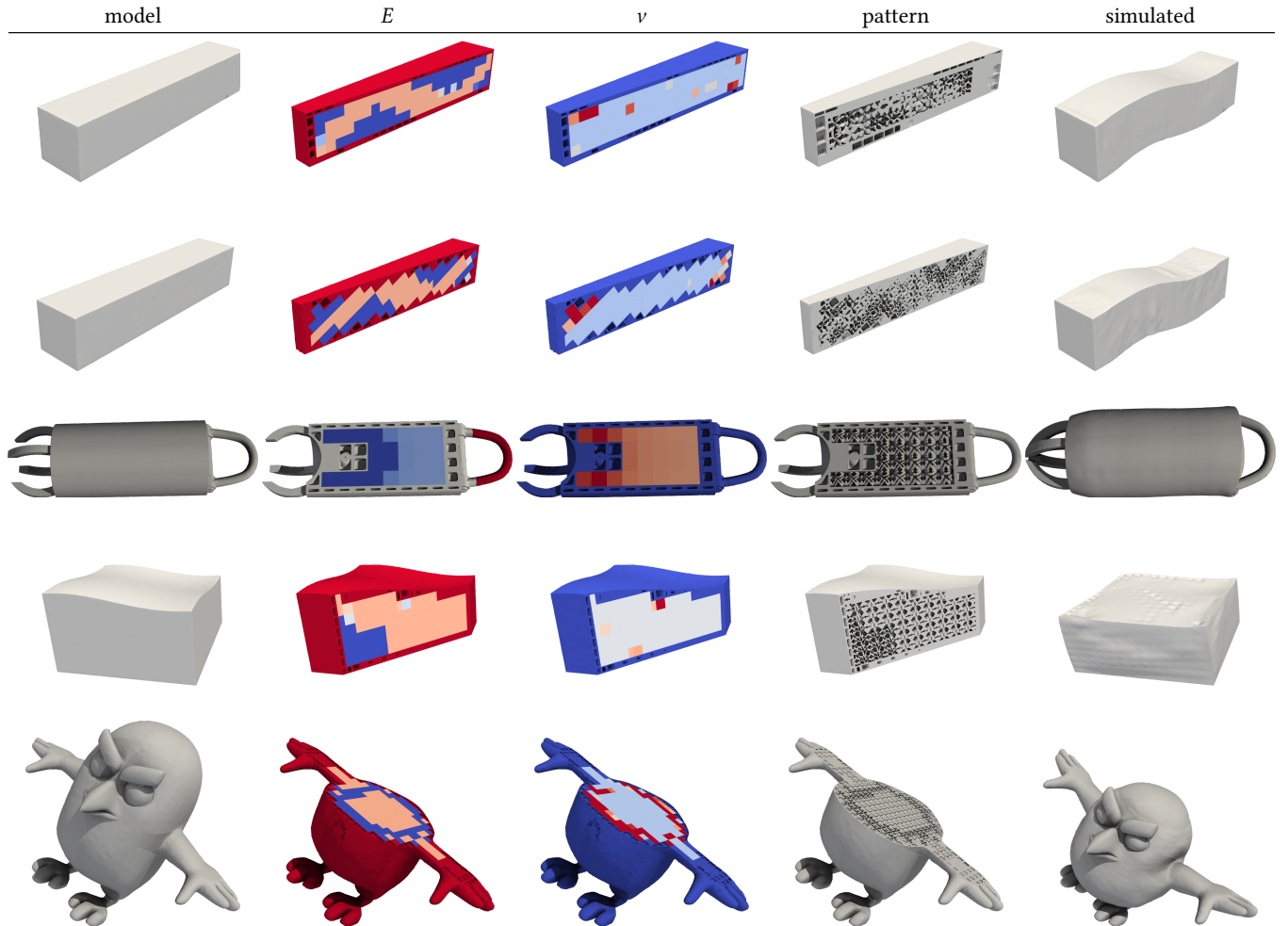
Fig. 18. A gallery of 3D examples. From left to right: original model, optimized material distribution (Young's modulus and Poisson's ratio), final geometry at rest and deformed geometry (simulated).

note that [46] only applies to 2D, and its extension to 3D is highly non-trivial due to the requirement of a high-quality, isotropic, hexahedral mesher (see discussion in Section 2). In contrast, our method generalizes well to 3D.

Numerical results and an ablation study for second optimization. Numerical comparisons for the examples in the paper are shown in Table 1. The first optimization stage is very effective for all models (*1st Optimization Reduced* row); however, inserting cut-cell microstructures significantly increases the objective values in some cases (*1st Optimization Full* row); The second optimization stage (*2nd Optimization Reduced/Full* rows) significantly reduces this effect in many cases (e..g, 2D Bar, 3D Bird, Gripper), but may slightly increase errors in some cases, (e.g. 2D Disk and Ghost). This small objective increase is dominated by the objective increase due to insertion of the microstructure. Because first and second optimizations produce different target material parameter values for interior cells, corresponding microstructures in two cases may have different

errors in approximation of these material properties and random variation in these errors may result in the observed increase in the objective.

Both the full pipeline with the second optimization and the reduced pipeline produce a large improvement compared to the baseline in all cases.

5 CONCLUSIONS

We have presented a *fully automated* pipeline starting from an arbitrary 2D or 3D shape and prescribed boundary conditions, and target deformation, to a complete manufacturable geometric shape with microstructure infill, retaining the input surface, and approximating the target behavior.

Limitations and future work. Our approach has several limitations. First, for thin objects and/or relatively coarse cells, most cells would be cut cells, and the quality of the results may be lower. Some of our examples suffer from buckling of the surface layer, as no terms
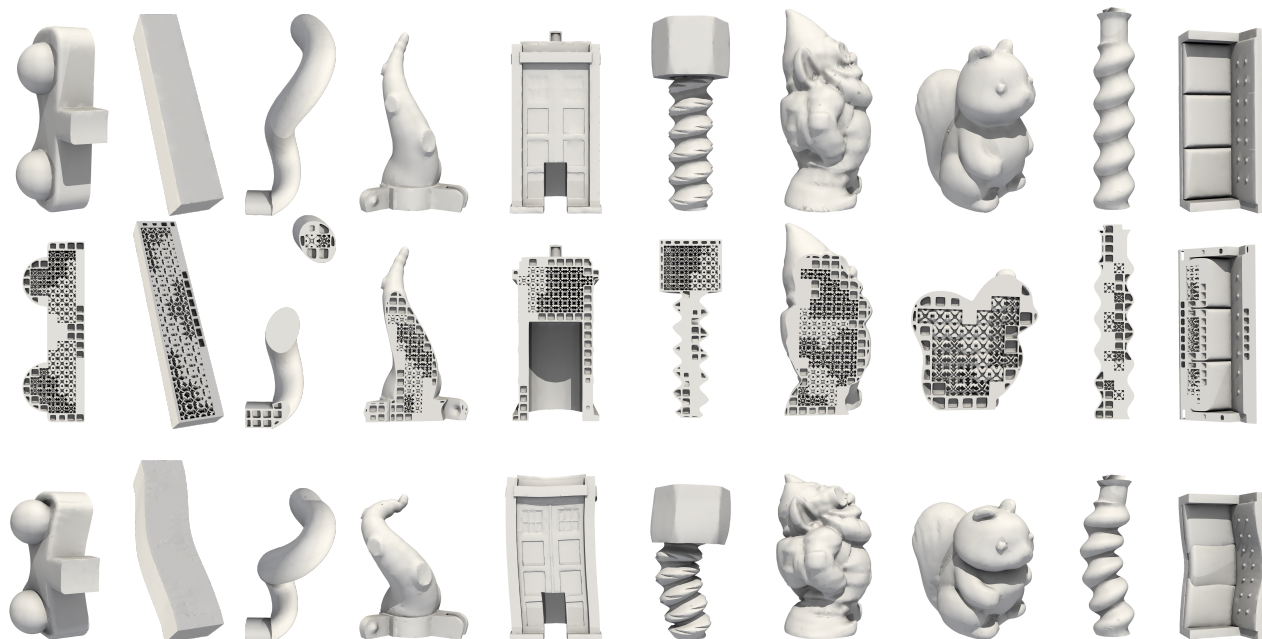
Fig. 19. 3D examples computed in batch mode. Top: rest shape. Middle: optimized pattern. Bottom: deformed shape.
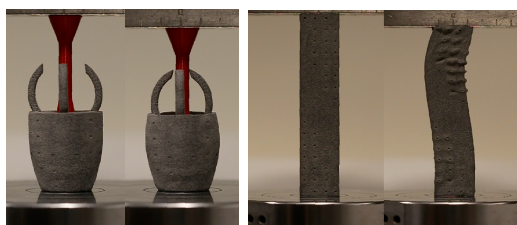

Fig. 20. Left image: Fabricated gripper. Rest shape on the left; force applied to the handle on the right. Right image: Fabricated bar. Rest shape on the left; compressed bar deforms in a controllable way, on the right.
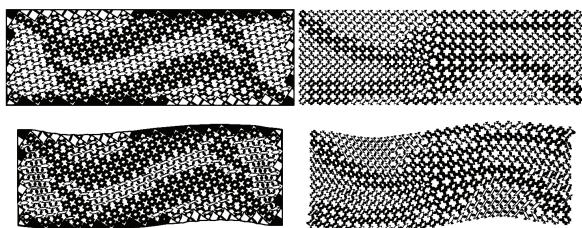

Fig. 21. Bar example. Dirichlet conditions on both sides compress the object while targeting a sin wave shape at the top and bottom. On the left, the result obtained with our cut-cell method; on the right, the original result obtained by [46].

	Disk	Bar	Ghost	Pliers	Bird	Sword
Baseline	0.01213	0.01270	0.00792	0.00189	0.01207	0.02132
1st Opt. Reduced	0.00710	0.00214	0.00469	0.00009	0.00847	0.01202
2nd Opt. Reduced	0.00826	0.00260	0.00342	0.00006	0.00719	0.01240
2nd Opt. Full	0.01060	0.00565	0.00590	0.00089	0.01001	0.01834
1st Opt. Full	0.01055	0.01498	0.00579	0.00094	0.01030	0.01852
cells	128	356	186	287	168	209
runtime	538	687	1013	309	406	487

	Bird	Gripper	Sine bar	Sine bar rotated	Sole
Baseline	2.058	0.799	1.211	1.327	1.331
1st Opt. Reduced	1.540	0.232	0.550	0.452	0.010
2nd Opt. Reduced	1.080	0.224	0.635	0.499	0.030
2nd Opt. Full	1.620	0.030	0.586	0.437	0.259
1st Opt. Full	1.729	0.085	0.550	0.432	0.289
cells	4442	400	936	1060	1475
runtime	20033	1738	2461	3062	3249

Table 1. Numerical comparisons of our approach vs. a baseline using solid boundary cells filled with base material in 2D and 3D. We report the displacement objective term in Equation 4, which represents the average point-wise error from the prescribed target for different stages of the pipeline, see Figure 14 for illustrations. The last two rows show the number of cells and runtime (sec).

penalizing buckling are included in our optimization. Second, we use the linear elasticity model in our examples. Extending our work to nonlinear elasticity, including contact, would reduce the simulation-to-reality gap and could extend the applicability of our method to more complicated scenarios. However, this would require a more complex mapping between material properties and microstructure geometry.

In addition, in this work, we explored one family of structures for boundary cells. There are many possible ways to develop it further; for example, rather than keeping it entirely fixed during the second optimization stage, one can combine material optimization for interior cells with low parametric shape optimization for the
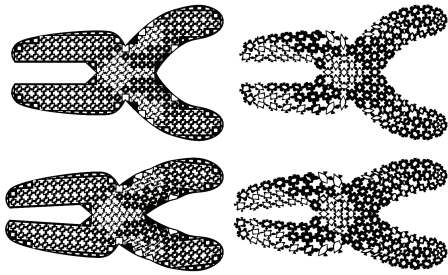
Fig. 22. Pliers example (black TPU). We use Dirichlet conditions on the right to compress the handles with the target of closing the jaws. On the left, is the result obtained with our cutcell method; on the right, is the result of [46].

boundary cells. The results can also be improved by using a broader spectrum of microstructure topologies.

REFERENCES

[1] Niels Aage, Erik Andreassen, Boyan S. Lazarov, and Ole Sigmund. 2017. Giga-Voxel Computational Morphogenesis for Structural Design. *Nature* 550, 7674 (10 2017), 84–86. https://doi.org/10.1038/nature23911

[2] M. J. Aftosmis, M. J. Berger, and J. E. Melton. 1998. Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry. *AIAA Journal* 36, 6 (1998), 952–960. https://doi.org/10.2514/2.464 arXiv:https://doi.org/10.2514/2.464

[3] Christie Alappat, Achim Basermann, Alan R. Bishop, Holger Fehske, Georg Hager, Olaf Schenk, Jonas Thies, and Gerhard Wellein. 2020. A Recursive Algebraic Coloring Technique for Hardware-Efficient Symmetric Sparse Matrix-Vector Multiplication. *ACM Trans. Parallel Comput.* 7, 3, Article 19 (June 2020), 37 pages.

[4] Pierre-Alexandre Beaufort, Maxence Reberol, Denis Kalmykov, Heng Liu, Franck Ledoux, and David Bommes. 2022. Hex Me If You Can. *Computer Graphics Forum* (2022). https://doi.org/10.1111/cgf.14608

[5] Martin Philip Bendsøe. 1989. Optimal shape design as a material distribution problem. *Structural optimization* 1, 4 (1989), 193–202.

[6] Martin Philip Bendsøe and Ole Sigmund. 2003. *Topology optimization: theory, methods and applications.* Springer.

[7] 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. *ACM Trans. Graph.* 29, 4, Article 63 (July 2010), 10 pages. https://doi.org/10.1145/1778765.1778800

[8] Matthias Bollhöfer, Aryan Eftekhari, Simon Scheidegger, and Olaf Schenk. 2019. Large-scale Sparse Inverse Covariance Matrix Estimation. *SIAM Journal on Scientific Computing* 41, 1 (2019), A380–A401.

[9] Matthias Bollhöfer, Olaf Schenk, Radim Janalik, Steve Hamm, and Kiran Gullapalli. 2020. State-of-the-Art Sparse Direct Solvers. (2020), 3–33.

[10] Carbon3D. 2023. Carbon Design Engine. https://www.carbon3d.com/products/carbon-design-engine Carbon Design Engine.

[11] Doina Cioranescu and Patrizia Donato. 1999. *An introduction to homogenization.* Oxford University Press.

[12] Tristan Djourachkovitch, Nawfal Blal, Nahiene Hamila, and Anthony Gravouil. 2021. Multiscale topology optimization of 3D structures: A micro-architectured materials database assisted strategy. *Computers & Structures* 255 (2021), 106574.

[13] Francisca Gil-Ureta, Nico Pietroni, and Denis Zorin. 2019. Structurally Optimized Shells. *arXiv preprint arXiv:1904.12240* (2019).

[14] Jeroen P. Groen and Ole Sigmund. 2017. Homogenization-Based Topology Optimization for High-Resolution Manufacturable Microstructures. *Internat. J. Numer. Methods Engrg.* 113, 8 (08 2017), 1148–1163. https://doi.org/10.1002/nme.5575

[15] Jeroen P. Groen, Jun Wu, and Ole Sigmund. 2019. Homogenization-Based Stiffness Optimization and Projection of 2D Coated Structures With Orthotropic Infill. *Computer Methods in Applied Mechanics and Engineering* 349 (06 2019), 722–742. https://doi.org/10.1016/j.cma.2019.02.031

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

[17] Peter Hachenberger and Lutz Kettner. 2022. 3D Boolean Operations on Nef Polyhedra. In *CGAL User and Reference Manual* (5.5.1 ed.). CGAL Editorial Board. https://doc.cgal.org/5.5.1/Manual/packages.html#PkgNef3

[18] Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast tetrahedral meshing in the wild. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 117–1.

[19] Zizhou Huang, Davi Colli Tozoni, Arvi Gjoka, Zachary Ferguson, Teseo Schneider, Daniele Panozzo, and Denis Zorin. 2022. Differentiable solver for time-dependent deformation problems with contact. arXiv:2205.13643 [cs.GR]

[20] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. 2013. Robust Inside-Outside Segmentation Using Generalized Winding Numbers. *ACM Trans. Graph.* 32, 4, Article 33 (jul 2013), 12 pages. https://doi.org/10.1145/2461912.2461916

[21] Muamer Kadic, Graeme W Milton, Martin van Hecke, and Martin Wegener. 2019. 3D metamaterials. *Nature Reviews Physics* 1, 3 (2019), 198–210.

[22] Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid deployment of curved surfaces via programmable auxetics. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 106.

[23] Heng Liu and David Bommes. 2023. Locally Meshable Frame Fields. *ACM Trans. Graph* 42, 4 (2023).

[24] Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. 2018. Narrow-Band Topology Optimization on a Sparsely Populated Grid. *ACM Trans. Graph.* 37, 6, Article 251 (12 2018), 14 pages. https://doi.org/10.1145/3272127.3275012

[25] Yang Liu, Zuyu Li, Peng Wei, and Wenwen Wang. 2018. Parameterized level-set based topology optimization method considering symmetry and pattern repetition constraints. *Computer Methods in Applied Mechanics and Engineering* 340 (2018), 1079–1101.

[26] 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 (12 2018), 14 pages. https://doi.org/10.1145/3272127.3275076

[27] Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. 2016. Procedural Voronoi foams for additive manufacturing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 44.

[28] Jonàs Martínez, Haichuan Song, Jérémie Dumas, and Sylvain Lefebvre. 2017. Orthotropic K-Nearest Foams for Additive Manufacturing. *ACM Transactions on Graphics* 36, 4 (07 2017), 1–12. https://doi.org/10.1145/3072959.3073638

[29] Graeme W Milton. 2002. *The theory of composites.* Cambridge University Press.

[30] Graeme W. Milton, Mark Briane, and Davit Harutyunyan. 2017. On the possible effective elasticity tensors of 2-dimensional and 3-dimensional printed materials. *Mathematics and Mechanics of Complex Systems* 5, 1 (2017), 381–391.

[31] Ken Museth. 2013. VDB: High-Resolution Sparse Volumes with Dynamic Topology. *ACM Trans. Graph.* 32, 3, Article 27 (jul 2013), 22 pages. https://doi.org/10.1145/2487228.2487235

[32] nTopology. 2023. nTop Platform. https://ntopology.com/ Engineering software built for digital manufacturing.

[33] Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-Case Stress Relief for Microstructures. *ACM Transactions on Graphics* 36, 4 (07 2017), 1–16. https://doi.org/10.1145/3072959.3073649

[34] 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), 135.

[35] Nico Pietroni, Marcel Campen, Alla Sheffer, Gianmarco Cherchi, David Bommes, Xifeng Gao, Riccardo Scateni, Franck Ledoux, Jean Remacle, and Marco Livesu. 2022. Hex-Mesh Generation and Processing: A Survey. *ACM Trans. Graph.* 42, 2, Article 16 (oct 2022), 44 pages. https://doi.org/10.1145/3554920

[36] János Plocher and Ajit Panesar. 2019. Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures. *Materials & Design* 183 (2019), 108164.

[37] Sina Rastegarzadeh, Jun Wang, and Jida Huang. 2022. Two-Scale Topology Optimization with Isotropic and Orthotropic Microstructures. *Designs* 6, 5 (2022), 73.

[38] ED Sanders, A Pereira, and GH Paulino. 2021. Optimal and continuous multilattice embedding. *Science Advances* 7, 16 (2021), eabf4838.

[39] Teseo Schneider, Jérémie Dumas, Xifeng Gao, Denis Zorin, and Daniele Panozzo. 2019. PolyFEM. https://polyfem.github.io/.

[40] 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), 136.

[41] Jonathan Richard Shewchuk. 1996. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Workshop on Applied Computational Geometry.* Springer, 203–222.

[42] Jonathan Richard Shewchuk. 2005. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry Towards Geometric Engineering: FCRC'96 Workshop, WACG'96 Philadelphia, PA, May 27–28, 1996 Selected Papers.* Springer, 203–222.

[43] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational design of actuated deformable characters. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 82.

[44] Michael Tao, Christopher Batty, Eugene Fiume, and David Levin. 2019. Mandoline: Robust Cut-Cell Generation for Arbitrary Triangle Meshes. *ACM Transactions on Graphics* (2019).

[45] Salvatore Torquato. 2002. *Random heterogeneous materials: microstructure and macroscopic properties*. Vol. 16. Springer.

[46] 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.

[47] Jun Wang, Jesse Callanan, Oladapo Ogunbodede, and Rahul Rai. 2021. Hierarchical combinatorial design and optimization of non-periodic metamaterial structures. *Additive Manufacturing* 37 (2021), 101710.

[48] Jun Wu, Niels Aage, Rudiger Westermann, and Ole Sigmund. 2018. Infill Optimization for Additive Manufacturing—Approaching Bone-Like Porous Structures. *IEEE Transactions on Visualization and Computer Graphics* 24, 2 (02 2018), 1127–1140. https://doi.org/10.1109/tvcg.2017.2655523

[49] Jun Wu, Christian Dick, and Rudiger Westermann. 2016. A System for High-Resolution Topology Optimization. *IEEE Transactions on Visualization and Computer Graphics* 22, 3 (03 2016), 1195–1208. https://doi.org/10.1109/tvcg.2015.2502588

[50] Jun Wu, Ole Sigmund, and Jeroen P Groen. 2021. Topology optimization of multi-scale structures: a review. *Structural and Multidisciplinary Optimization* 63, 3 (2021), 1455–1480.

[51] Liang Xia and Piotr Breitkopf. 2015. Multiscale structural topology optimization with an approximate constitutive model for local material microstructure. *Computer Methods in Applied Mechanics and Engineering* 286 (2015), 147–167.

[52] Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization With Microstructures. *ACM Trans. Graph.* 36, 4, Article 120b (07 2017). https://doi.org/10.1145/3072959.3095815

[53] 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.