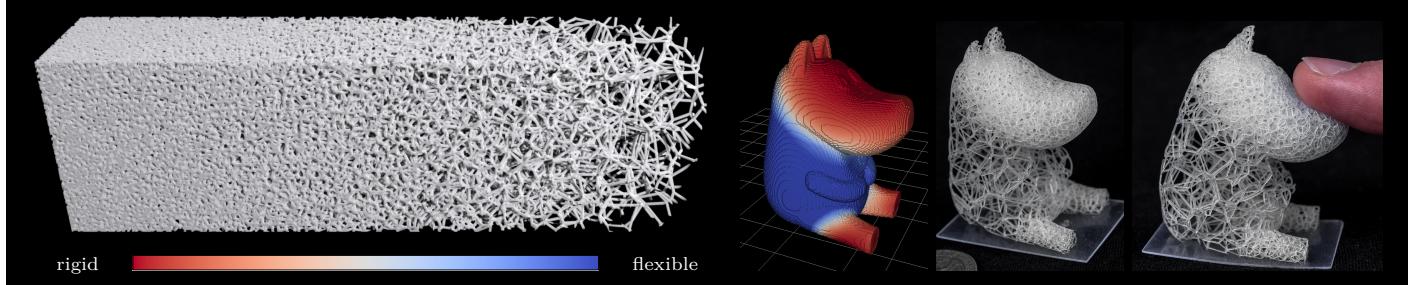


# Procedural Voronoi Foams for Additive Manufacturing

Jonàs Martínez  
INRIA

Jérémie Dumas  
Université de Lorraine, INRIA

Sylvain Lefebvre  
INRIA, Université de Lorraine



**Figure 1:** Our method procedurally generates structures with graded material elasticities, which can be directly fabricated. Here the user paints elasticity on a 3D model to create a flexible figurine. Model: Moomin ([thing:1173447](#)) by Jeroentjj.

## Abstract

Microstructures at the scale of tens of microns change the physical properties of objects, making them lighter or more flexible. While traditionally difficult to produce, additive manufacturing now lets us physically realize such microstructures at low cost.

In this paper we propose to study procedural, aperiodic microstructures inspired by Voronoi open-cell foams. The absence of regularity affords for a simple approach to grade the foam geometry — and thus its mechanical properties — within a target object and its surface. Rather than requiring a global optimization process, the microstructures are directly generated to exhibit a specified elastic behavior. The implicit evaluation is akin to procedural textures in computer graphics, and locally adapts to follow the elasticity field. This allows very detailed structures to be generated in large objects without having to explicitly produce a full representation — mesh or voxels — of the complete object: the structures are added on the fly, just before each object slice is manufactured.

We study the elastic behavior of the microstructures and provide a complete description of the procedure generating them. We explain how to determine the geometric parameters of the microstructures from a target elasticity, and evaluate the result on printed samples. Finally, we apply our approach to the fabrication of objects with spatially varying elasticity, including the implicit modeling of a frame following the object surface and seamlessly connecting to the microstructures.

**Keywords:** 3D printing, additive manufacturing, procedural modeling, material design

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

© Jonàs Martínez, Jérémie Dumas, Sylvain Lefebvre [2016]. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive version was published in SIGGRAPH '16 Technical Paper.

SIGGRAPH '16 Technical Paper, July 24–28, 2016, Anaheim, CA  
ISBN: 978-1-4503-4279-7/16/07  
DOI: <http://dx.doi.org/10.1145/2897824.2925922>

## 1 Introduction

Additive manufacturing enables the fabrication of objects having unprecedented complexity. This capability is often understood in terms of the overall shape of objects. However, it is also possible to fabricate parts filled with *microstructures* — having intricate internal details in the order of tens of microns. The macro-scale mechanical properties of the object are then directly influenced by the geometry of the microstructures. In particular, careful design of the structures affords for parts that are lighter while remaining rigid enough for their intended use. This reduces material usage, shipping and transportation costs. In addition, inner structures can progressively vary within the object and adapt to varying rigidity requirements between regions subject to different stresses.

There are several challenges to achieve these goals. First, the fine scale geometry of the structures has to produce the desired large scale elastic behavior. This often implies formulating challenging global optimizations, either to directly synthesize the fine scale geometry or to fill the shape with precomputed microstructures (see Section 2). Second, the structures are very small compared to the size of the objects (e.g. tens of microns versus tens of centimeters). Therefore, the meshes describing the models become quickly prohibitively large, posing important computational challenges for simulation, visualization, and fabrication. Third, the structures have to enforce *fabricability constraints*. The main industrial processes have different requirements depending on whether they locally deposit material (e.g. fused filament fabrication, resin droplets) or whether they locally solidify a bed of material itself acting as a support (e.g. selective laser sintering). In the first case the structures should not present any disconnected parts during fabrication, while in the second case they should not enclose non-solidified material in pockets.

To answer these challenges we draw inspiration from procedural noises in computer graphics, where an infinite amount of content is produced at low, constant memory cost while precisely controlling the statistical properties of the produced noises [Lagae et al. 2009]. This hints at the possibility of generating *procedural, stochastic* microstructures that directly exhibit the desired elastic behavior, without further

optimization. Our approach explores this idea and defines a procedure to synthesize open-cell foams that enforce fabrication requirements while having precisely controlled elastic properties. The foam parameters can vary spatially to follow gradients of elasticity.

As the microstructures are procedural we only generate their details when needed for fabrication. Typically, evaluation happens at the slice level, just before sintering or curing a layer of material. The microstructures are stochastic and aperiodic in nature. Stochasticity results in an exceptionally good isotropic behavior, and lets us grade the properties without introducing discontinuities along pre-defined boundaries. Aperiodicity removes the need for a global optimization when conforming the structures to a surface.

### Our contributions are:

- The definition of procedural Voronoi foams that can be evaluated very efficiently, have precisely controlled isotropic elastic behavior, and can be spatially graded to produce gradients of elasticity.
- A methodology to derive an inverse mapping, from a target elasticity to the parameters driving the microstructure generation.
- A complete implementation that maps well to stackless, massively parallel architectures.
- Applications to the 3D fabrication of objects filled with procedural Voronoi foams, with proper handling of the outer object frame.

The result is an algorithm that generates microstructures with a prescribed elastic behavior on the fly, during fabrication. No optimization process is required to adapt to a different object or to match a graded elasticity field. There is no limit to the size of the printable objects as the microstructures never fully reside in memory. Therefore, our approach will naturally scale with future technology as the resolution and size of printable object increases.

**Scope of the Paper.** Isotropic elastic materials are described by two parameters: Young's modulus and Poisson's ratio. Intuitively the Young's modulus captures how rigid or soft an object is, while the Poisson's ratio captures how one dimension stretches with another. In this work we focus on varying the Young's modulus while preserving the Poisson's ratio of the base material.

For spatially varying elasticity we assume the target scalar field is given as input and do not consider its computation.

## 2 Related Work

**Modeling for Fabrication.** In the recent years there has been a strong interest in techniques helping the user to design physical objects with controlled behaviors, for instance to avoid fragile regions and reinforce them [Umetani and Schmidt 2013; Zhou et al. 2013; Stava et al. 2012], to allow the user to balance shapes [Prévost et al. 2013; Bächer et al. 2014], or to produce strong yet lightweight inner structures [Wang et al. 2013; Lu et al. 2014]. These latter approaches are related to our work in that they seek to produce structures of maximum rigidity under a prescribed volume to strengthen 3D printed objects. However, these approaches rely on global optimization of as rigid as possible macro-scale

structures, while we focus on the procedural generation of *micro*-structures with controlled elastic properties.

**Graded Elasticity.** In the field of mechanical engineering Sigmund et al. [1995; 1999] introduced the optimization of micro-scale structures to achieve specific macro-scale behaviors, such as controlled elasticity. A key idea is to consider the behavior of a composite material, made of infinitely many repetitions of a base material tile. This limit behavior is captured by the theory of *homogenization* for linear elasticity [Allaire 2012] which relates the unit material tile to the elastic properties of its (infinite) periodic tiling. We provide more details in the supplemental material. Therefore, most techniques pose an inverse homogenization problem, optimizing for a tile producing a target elastic behavior [Sigmund 1994; Zhou and Li 2008; Radman et al. 2013; Xia and Breitkopf 2015]. Andreassen et al. [2014] include fabrication constraints in the structure optimization.

In the field of computer graphics the design of materials for fabrication has become an important direction of research, as we strive to enable artists and designers to physically realize virtual models of deformable objects [Skouras et al. 2013; Xu et al. 2015; Pérez et al. 2015]. Bickel et al. [2010] proposed a data-driven approach to design materials with prescribed deformation. Base materials are stacked by an optimizer to obtain the target properties. Schumacher et al. [2015] extend this idea in two ways. First, elementary material tiles are optimized by homogenization to cover a large spectrum of elastic behaviors. Second, a process globally optimizes for a choice of tiles in a grid covering the object, to achieve the desired, spatially varying elastic behavior. The process considers fabrication constraints and connectivity between adjacent tiles. Panetta et al. [2015] take a different approach, by optimizing for an optimal family of elementary tiles among a large — but restricted — set of possibilities. The tiles take into account fabrication and connectivity constraints. Once the best family is determined, it is possible to arrange the tiles in a grid such as to obtain a spatially varying elasticity.

These works achieve impressive results at relatively low computational costs, thanks to the regularity of the grid and the periodicity assumptions: homogenization is performed on a single base tile instead of having to rely on a global optimization, and the periodicity affords for compact descriptions [Pasko et al. 2011]. However, the tile-based approach has a number of drawbacks. First, grading the material by changing the tiles within the grid requires careful handling of the interface between neighboring tiles. Either the tiles are pre-optimized to have matching boundaries [Panetta et al. 2015], or a global selection process is required to strike a compromise between continuity across neighbors, grid discretization and elasticity objectives [Schumacher et al. 2015]. The global optimization does not scale well with the size of the structures and objects — especially as it is desirable to produce the smallest possible tiles to converge towards the limit behavior computed by homogenization. Our approach solves this thanks to a direct and simple relationship between elastic properties and geometric parameters. Second, it is often desirable to conform the elasticity field to the object surface, for instance having a more rigid crust within a distance of the surface. The global periodic nature of the grid makes this difficult. One possible way is to optimize for a 3D parameterization of the grid within the object. While this is a topic of intense research (related to hexahedral meshing), this remains difficult [Staten 2007], and the effect on the final elastic properties is hard to precisely establish. Instead,

we seek to produce *stochastic*, *aperiodic* structures which are by nature simpler to conform to the gradients of a field as they do not require specific spatial alignments. Finally, the isotropy of periodic tilings is equal to that of the optimized base tile, while the isotropy of stochastic foams further improves with larger extents of foams.

**Efficient Microstructure Generation.** Pasko et al. [2011] considered procedural definitions of periodic microstructures. The parameters of the microstructures can vary spatially to produce graded materials [Fryazinov et al. 2013], for instance to reinforce an object following a cross-sectional stress analysis [Li et al. 2015]. Brennan-Craddock [2011] computes the intersection of the object and periodic microstructures on a per-slice basis. A frame structure is built on the object surface by subtracting the microstructure cells from a thick surface shell. Our frame structure follows a similar intuition for the case of Voronoi foams. OpenFAB [Vidimče et al. 2013] provides a specialized language to describe procedural microstructures. The geometric details are efficiently evaluated at slicing time, streaming voxels to the printer. Similar to these approaches, we evaluate the microstructures procedurally during slicing. However, the microstructures we generate are aperiodic graded foams whose Young’s modulus is directly and precisely controlled.

Recent works consider the problem of generating non-periodic microstructures with varying density. The adaptive voids approach [Medeiros e Sá et al. 2015] relies on a subdivision scheme to produce denser structures near the surface of an object. Brackett et al. [2014] perform a sequential dithering of a density field to keep a subset of points, which are then used to define an open-cell foam. While closely related to our work, these approaches do not explicitly control the Young’s modulus of the produced structures, nor afford for an efficient parallel evaluation.

Recent software for additive manufacturing propose microstructure generation packages. In particular, Within-Labs [Autodesk 2016] proposes trabecular structures resembling Voronoi foams. While the parameters can be varied, to the best of our knowledge there is no direct control of the Young’s modulus and the structures are not defined by a procedure akin to procedural solid textures.

**Open-Cell Foams.** Our procedural microstructures belong to a specific class of microstructures known as *open-cell foams*. These structures occur naturally and can be obtained from physical processes in a variety of materials including metals. Therefore, in the field of mechanical engineering there has been a strong interest in modeling and analyzing the properties of these structures.

Interestingly, naturally occurring open-cell foams are often idealized as edges of Voronoi cells [Gibson and Ashby 1997]. As we propose to generate foams from procedurally generated Voronoi diagrams, these works are highly relevant for the study of the mechanical behavior of our structures. In particular, from these studies we can expect the following properties: 1) the elastic behavior of open-cell irregular foams is highly isotropic [Luxner et al. 2007], and 2) their Young’s modulus relate *almost linearly* to their geometric parameters (thickness, density) [Gibson and Ashby 1997; Roberts and Garboczi 2002]. This is an exceptionally good property for our purpose of producing graded elastic materials, as the simple relationship affords for a direct derivation of structure parameters given an elasticity target. We verify

that our structures meet these observations in Section 4.3.

Prior studies also indicate that the Poisson’s ratio of open-cell foams does not significantly vary [Gibson and Ashby 1997], and our structures indeed share this limitation.

### 3 Procedural Voronoi Foam Generation

We now introduce our approach for the procedural generation of open-cell foams. We describe the procedural synthesis of the structures in Section 3.1 and discuss implementation in Section 3.2. We explain how to derive the parameters of the structures from a desired target elastic behavior in Section 4 and present results and applications in Section 5.

#### 3.1 Procedural Generation

We seek to define aperiodic *procedural* structures akin to procedural textures in computer graphics. The structure is defined as a function  $\mathcal{F} : \mathbb{R}^3 \rightarrow \{0, 1\}$  which is evaluated during display and slicing at every point in space, at the desired resolution.

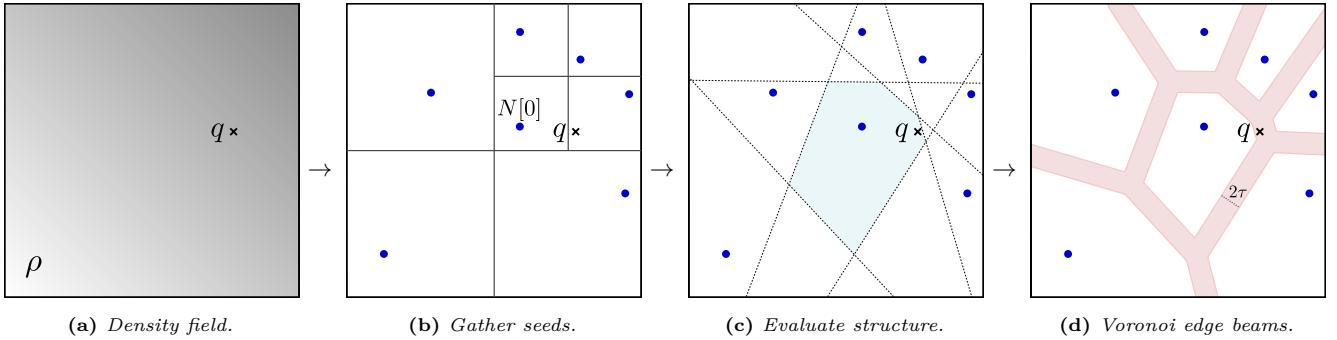
To have the computational advantages of procedural textures,  $\mathcal{F}$  has to follow a number of requirements [Lagae et al. 2010], that we summarize as follows: 1)  $\mathcal{F}$  has to evaluate in constant time and constant memory regardless of the point of evaluation. 2) The size of  $\mathcal{F}$  — its program and built-in data — has to be independent from the size of the generated content. This is the case of our technique which produces arbitrary large aperiodic content from a constant, small memory footprint (a few hundreds of bytes).

In addition, the structure has to enforce geometric requirements to be printable. First, there should be a minimal number of pockets (holes) enclosing printing material. This advocates for an open-cell structure made of beams along the edges of a cellular structure. Second, there should be no disconnected parts appearing during fabrication. It is easy to see that convex cells enforce this property everywhere but at the boundary (which we discuss in Section 5.3). The convex cells of Voronoi diagrams are therefore well suited.

Our procedural Voronoi foams are defined by two parameters: the density  $\rho$  of Voronoi seeds per unit volume ( $\text{seeds/mm}^3$ ), and the radius  $\tau$  of the beams along the edges ( $\text{mm}$ ). Density may vary spatially and is given as a function, i.e. at a given point  $x$  the desired density is  $\rho(x)$ . We assume the variations to be smooth compared to the size of the structure cells.

**Open-Cell Voronoi Foams.** We seek to design a procedural function  $\mathcal{F}_{\rho, \tau}$  that produces beams of thickness  $2\tau$  along the edges of a Voronoi cell structure having a density  $\rho$ . In addition, we would like to allow for  $\rho$  to be spatially varying. Our procedural generation is inspired by the seminal work on cellular solid textures by Worley [1996], revised to produce an open-cell structure. We further extend the procedural scheme to afford for spatially varying densities.

Worley defines procedural cellular textures by using pseudo-random sequences to generate seeds in a virtual grid, following a Poisson distribution (not to be confused with a Poisson disc distribution). Given an evaluation point  $q$  the seed closest to  $q$  is determined. The seed id is used to derive a color value at  $q$ , for instance coloring each Voronoi cell differently. The set of grid cells that can have an influence is limited to cells neighboring  $q$ . This stems from the fact that

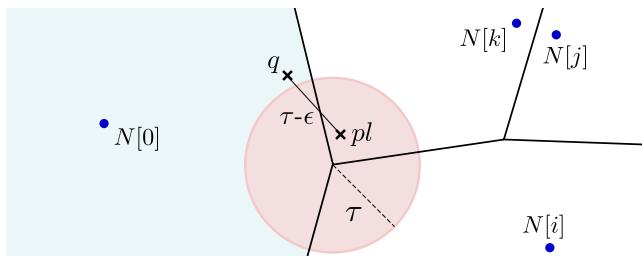


**Figure 2:** 2D overview of Algorithm 1. (a) Input query point  $q$  and density field  $\rho$ . (b) The seeds that could contribute to the Voronoi cell of  $N[0]$  (which is the seed closest to  $q$ ) are gathered (Algorithm 2). (c) The bisectors of the seed pairs influencing  $q$  are computed. (d) Finally, the algorithm checks whether  $q$  lies inside a beam of radius  $\tau$  along the Voronoi edge.

each cell contains at least one seed — a deviation from a pure Poisson distribution to enable efficiency [Worley 1996] — and, therefore, the closest seed is necessarily within the 2-ring of neighboring cells. This leads to the constant time evaluation property, as the number of considered seed points remains below a constant everywhere in space.

Our algorithm achieves similar properties while generating the edges of a Voronoi open-cell foam. Given an evaluation point  $q$  our goal is to write a function returning 0 (empty) if  $q$  is not inside a beam of the structure and 1 (solid) otherwise. The pseudo-code of our structure generation is in Algorithm 1, and a graphical overview is in Figure 2. It starts by generating all the seeds that could contribute to the definition of the Voronoi cell by calling algorithm GATHERSEEDS (line 1) — we detail this algorithm later. We next enumerate all the line equations of the edges (line 4) and for each compute  $pl$ , the point closest to  $q$  on the line (line 5). If the distance between  $q$  and the edge is greater than the beam radius  $\tau$  (line 6),  $q$  is not influenced by this edge. Otherwise,  $q$  might be inside a beam. To be certain that it is the case, we have to verify that  $pl$  is indeed *on an edge* of the Voronoi cell containing  $q$ . A counter example is illustrated in Figure 3. We verify this by considering whether  $pl$  belongs to the Voronoi cell of a neighboring seed (lines 8–11). If that is the case, then we ignore this line equation (line 10). Otherwise,  $q$  is in a solid region and we return 1 (solid) line 13.

We next discuss how the seeds are gathered and generated.



**Figure 3:** In this counterexample, all the seeds lie on the same 3D plane (cross section shown). The closest point  $pl$  to the bisector line of  $N[0]$ ,  $N[i]$ , and  $N[j]$  is at a distance  $\tau - \epsilon$ , where  $\epsilon > 0$ . However,  $pl$  does not belong to any edge of the Voronoi cell of  $N[0]$  (shaded in blue).

---

#### Algorithm 1: EVALSTRUCTURE $\mathcal{F}_{\rho,\tau}(q)$

---

**Input:** Density field  $\rho$ , beam radius  $\tau$ , query point  $q$ .  
**Output:** Voxel state  $\in \{0, 1\}$

```

1  $N \leftarrow \text{GatherSeeds}(q); // \text{seeds influencing } q$ 
2 for  $i \leftarrow 1$  to  $|N|$  do
3   for  $j \leftarrow i + 1$  to  $|N|$  do
4      $bl \leftarrow \text{BisectorLineEquation}(N[0], N[i], N[j]);$ 
5      $pl \leftarrow \text{ClosestPointOnLine}(q, bl);$ 
6     if  $\|q - pl\| \leq \tau$  then
7        $accept \leftarrow \text{true};$ 
8       for  $k \leftarrow 1$  to  $|N|$  do
9         if  $\|N[0] - pl\| > \|N[k] - pl\|$  then
10           $accept \leftarrow \text{false};$ 
11        break;
12     if  $accept$  then
13       return 1
14 return 0

```

---

**Gathering Seeds.** Algorithm 1 requires all seeds that can influence the result at  $q$ . If some required seeds were to be missed, the produced structure could fail to print or break. However, we only need to be conservative: as long as we have a superset of the required seeds, the algorithm will produce the correct result.

We generate seeds in a grid, and we guarantee that all grid cells receive at least one seed. This bounds the number of grid cells that we have to consider. As explained in Figure 4, the Voronoi cell of a seed cannot be influenced beyond a 2-ring of neighbors.

Algorithm 2 gives the pseudo-code for gathering the seeds around  $q$ . The evaluation point  $q$  might belong to the Voronoi cell of any of the seeds within (at most) a 2-ring radius. We therefore first search for the seed closest to  $q$ , and then gather seeds with a 2-ring around the closest seed. SUBDIVIDECELL produces at least one seed per grid cell — possibly more with spatially varying density. We detail this algorithm next.

**Seed Generation and Density Control.** We now describe SUBDIVIDECELL. It generates seeds in each cell, at least one and possibly more by subdividing to locally adapt to density. Conceptually our technique is based on a primal subdivision

---

**Algorithm 2:** GATHERSEEDS( $\rho, q$ )

---

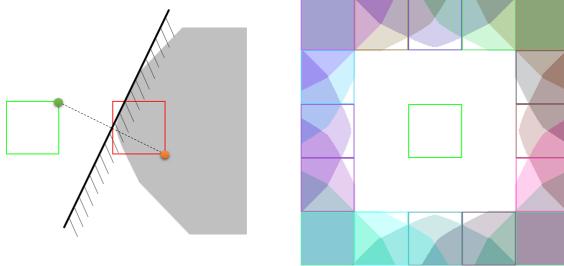
**Input:** Density field  $\rho$ , query point  $q$ .  
**Output:** Set of seeds possibly influencing  $q$

```

1  $N \leftarrow \emptyset$  ;
2  $Visited \leftarrow \emptyset$  ;
3  $cq \leftarrow GridCellEnclosing(q)$  ;
4  $closest \leftarrow \infty$  ;
5 for  $cell \in TwoRingNeighborhood(cq)$  do
6    $Visited \leftarrow Visited \cup \{cell\}$ ;
7    $seeds \leftarrow SUBDIVIDECELL(\rho, cell.center, cell.length)$ ;
8    $N \leftarrow N \cup seeds$ ;
9   for  $s \in seeds$  do
10    if  $\|s - q\| < \|closest - q\|$  then
11       $closest \leftarrow s$ ;
```

12  $cs \leftarrow GridCellEnclosing(closest)$  ;
13 **for**  $cell \in TwoRingNeighborhood(cs) \setminus Visited$  **do**
14  $N \leftarrow N \cup SUBDIVIDECELL(\rho, cell.center, cell.length)$ ;
15 **return**  $N$ 


---



**Figure 4:** *Left:* We consider the Voronoi cell of a seed located anywhere in the green square, and how it is influenced by another seed in the red square. The red grid square is within the second ring of neighbors of the green grid square. Each pair of seeds in the green/red squares defines a possible bisector, which might be a face of the Voronoi cell of the green seed. The opposite half-space cannot belong to the Voronoi cell. This is illustrated for a single pair of seeds in the figure (green/red dots). We define the **shadow** of the red square as the intersection of all the half-spaces from all possible pairs of green/red seeds. The shadow is shown in gray ; it represents the region of space that **cannot** possibly be part of the Voronoi cell of the green seed: regardless of the position of the seeds, the shadow is always cutout by the bisectors. *Right:* The shadows of all grid squares in the 2-ring completely cover the space beyond. Therefore, no seed outside of the 2-ring can have an influence on the Voronoi cell of the green seed.

of the coarsest density grid, where each parent cell is split regularly in eight children. It is important to recall, however, that the grids are never stored: all computations happen implicitly and on the fly. We refer to the first level of grid cells as the *coarse* grid cells (between 2mm and 5mm in our implementation). Another important design goal of our subdivision process is to avoid bias: The statistics of the point distributions remain constant — up to a scaling factor — at all density levels.

The pseudo-code is given in Algorithm 3 for the general case of a spatially varying density field. Given a grid cell of size  $l$  and center  $c$ , we compute the number of seeds it has to contain as  $l^3 \times \rho(c)$  (line 2, cell volume times density). We

evaluate the density at the cell center, but more elaborate schemes could be used (e.g. multi-point evaluation, or pre-integration of  $\rho$  in a summed area table). The field  $\rho$  is clamped to a minimum value to ensure that the coarsest cells always receive at least one seed. When the target number of seeds in a cell is above  $2^3$  — there is more than exactly one seed per subdivision child of the current cell — we recurse and subdivide the cell (line 13). Otherwise, we randomly select  $n = \lfloor l^3 \times \rho \rfloor$  distinct children and draw exactly one seed in each (lines 4-7). We take into account the remaining fraction  $f = t - n$  by drawing an additional sample in a next child cell, with probability  $f$  (lines 8-10). Note that all random number generators are pseudo-random sequences seeded by the grid cell coordinate. Therefore, for a same initial grid cell, the exact same set of seeds is produced.

The pseudo-code in Algorithm 3 cannot be implemented directly on massively parallel architectures (GPUs) due to the recursive calls. We present in the supplemental material a stackless iterative version that maps well to massively parallel processors.

---

**Algorithm 3:** SUBDIVIDECELL

---

**Input:** Density field  $\rho$ , cell center  $c$ , cell size  $l$ .  
**Output:** A set of seed, with a density driven by  $\rho$

```

1  $N \leftarrow \emptyset$  ;
2  $t \leftarrow l^3 \times \rho(c)$  ; // target number of seeds in current cell
3 if  $t \leq 2^3$  then
4    $I = RandomPermutation(subcells)$ ;
5    $n_{min} = \lfloor t \rfloor$  ; // minimum number of samples to draw
6   for  $i \leftarrow 0$  to  $n_{min} - 1$  do
7      $N \leftarrow N \cup \{RandomSampleInSubcell(I[i])\}$ ;
8    $p \leftarrow random(0, 1)$ ;
9   if  $p \leq (t - n_{min})$  then
10     $N \leftarrow N \cup \{RandomSampleInSubcell(I[n_{min}])\}$ ;
11 else
12   for  $subcell \in currentCell$  do
13      $N \leftarrow N \cup SUBDIVIDECELL(\rho, subcell.center, l/2)$ ;
```

14 **return**  $N$ 


---

**Beam Radii.** The radii of the beams are directly controlled by  $\tau$  during evaluation (Algorithm 1, line 6). While this value may also vary spatially ( $\tau(q)$ ) we only vary density in our approach (see Section 4.2).

We next discuss how to select the parameters to reach a target elastic behavior.

### 3.2 Implementation

We implement our technique in an image-based slicer, which either directly sends images to the printer (SLA) or extracts contours (SLS). We slice at a resolution between 10µm and 50µm per pixel.

The procedural foam is implemented as an OpenCL kernel processing all pixels of a slice in parallel on a GPU. In addition, we implement a supersampling procedure to obtain gray-scale exposure levels on SLA processes. This is done by performing supersampling around the evaluation point  $q$  in a small loop around lines 6-13 of Algorithm 1, counting the number of times  $q + \epsilon$  lies within the beam.

Exact beam contours could be extracted by *locally* constructing the explicit geometry of the Voronoi cells around  $q$  in the manner of Algorithm 1. We found this approach less convenient than our implicit description that fits very well existing massively parallel architectures, and avoids having to union beam geometries explicitly.

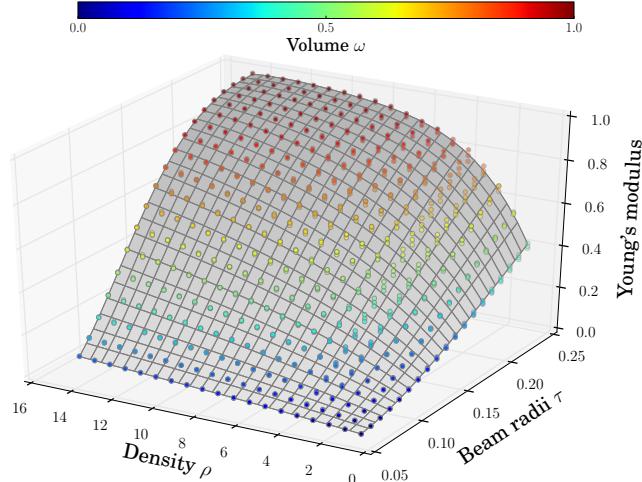
## 4 Foams With Controlled Elasticity

We now consider the problem of selecting the microstructure parameters  $(\rho, \tau)$  to achieve a target elastic behavior. This is done by statistical analysis of the homogenized elasticity tensor of  $\mathcal{F}_{\rho, \tau}$ , for different parameters.

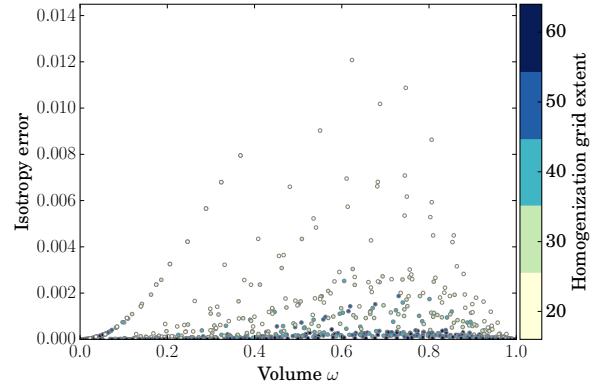
### 4.1 Analysis Using Homogenization

For a choice of  $(\rho, \tau)$  we produce a periodic version of the structure in a base volume, and apply homogenization to compute the elastic tensor of the corresponding periodic media. By doing this for many choices of  $(\rho, \tau)$  we reconstruct the underlying relationship between  $\rho, \tau$  and the elastic properties (Young's modulus, Poisson's ratio). The graph resulting from this analysis is shown in Figure 5. It is obtained for a material with a unit Young's modulus ( $= 1$ ) and a Poisson's ratio  $v = 0.3$  which matches most plastics. Only changing the Poisson's ratio requires re-running the simulation as results scale linearly with Young's modulus.

We perform homogenization on a grid of hexahedral elements (see supplemental material for details). The size of the elements is chosen to properly capture the beams; our experiments showed that using half the diameter gives stable results. The stiffness of each element is computed by calling our implicit procedural function with super-sampling, which returns the volume of structure intersected by the element.



**Figure 5:** Young's modulus as a function of radius and density. Each dot is the result of homogenization for a given  $(\rho, \tau)$ . For each  $(\rho, \tau)$  we generate multiple instances of foams using different random seeds. As can be seen the dots tightly cluster, indicating that randomness has little impact on the elastic properties. The maximum deviation after the fitting is 3.3% of the Young's modulus. The data points are used to fit a polynomial function (see Section 4). For clarity, the graph shows a subset of our data that extends to higher densities for lower beam radii.



**Figure 6:** Relative volume — volume of the structure over simulated volume — and error  $\xi(C^H)$  of the approximation of the ideal isotropic tensor (i.e. divergence from isotropy). As the spatial extent of the simulated volume grows the tensor tends towards the ideal isotropic tensor.

**Homogenization of Aperiodic Foams.** With this approach we are making an important assumption. Our structures are *not* periodic: they form an aperiodic foam of constant density in space. We therefore assume that the overall behavior of the aperiodic foam is similar to the periodic behavior of a *sufficiently large base volume*. To determine the size of the base volume we rely on the expected isotropic behavior of irregular open-cell foams [Luxner et al. 2007]. We perform homogenization for volumes of increasing spatial extent and consider the deviation of the computed tensor from a perfectly isotropic tensor.

The homogenized tensor is a full symmetric tensor composed of 21 independent variables. We approximate the full tensor with the tensor of an isotropic material and consider the residual error. The tensor of an isotropic material is expressed from only two independent variables — Young's modulus and Poisson's ratio — and we denote it as  $C^I(E, v)$ . Its expression is given in the supplemental material.

Similarly to [Schumacher et al. 2015] we approximate the homogenized elasticity tensor  $C^H$  by minimizing:

$$\xi(C^H) = \min_{E, v} \|C^I(E, v) - C^H\|_F^2 \quad (1)$$

$$-1 \leq v \leq 0.5$$

$$0 \leq E \leq E_M$$

, where  $E_M$  is the upper bound Young's modulus of the solid base material. The expression of  $C^I$  is nonlinear with respect to  $E, v$  and we therefore solve this minimization with a nonlinear optimizer [Johnson 2007].

The approximation error measures whether homogenization was able to properly capture the isotropic behavior of the structure. Figure 6 reports the error in isotropy for various grid sizes. In practice, we find that a volume having at least  $60 \times 60 \times 60$  hexahedral elements provides a stable result across a wide range of beam radii.

**Collecting Data Points.** In a one-time precomputation we generate a large number of pairs  $(\rho, \tau)$ . We uniformly sample  $\rho$  and  $\tau$  to cover the range that can be fabricated (see Section 4.2). For each selection of  $(\rho, \tau)$ , we generate a number of different structure realizations by varying the global random seed. We compute the Young's modulus and Poisson's ratio

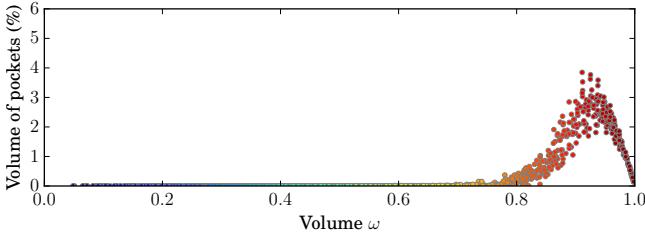
obtained by applying homogenization on each unit volume. This gives us a large number of data points characterizing the elastic behavior of the structures, as shown in Figure 5.

We then fit a polynomial function  $\mathcal{P}$  on the experimental results to correlate the density, beam radius, and Young's modulus. We optimize for the coefficients of a degree 4 polynomial by least square fitting, minimizing  $\sum_i (\mathcal{P}(\rho_i, \tau_i) - E_i)^2$ . The resulting polynomial fit is shown in Figure 5.

## 4.2 Deriving Parameters for a Target Elasticity

We now describe how to select  $\rho$  and  $\tau$  to achieve a target Young's modulus. As can be seen in Figure 5, for a given target elasticity multiple choices of  $\rho$  and  $\tau$  are possible: the full isocurve where  $\mathcal{P}(\rho, \tau) = E_{\text{target}}$ . Our preferred strategy to select the values of  $\rho$  and  $\tau$  is to obtain structures that are as dense as possible, to remain close to the limit behavior computed by homogenization. The limiting factor is the minimal printable beam radius, which we denote as  $\tau_{\min}$ . To maximize density, we fix  $\tau = \tau_{\min}$ . As densities increases with a fixed radius, we reach a point where the block of matter is full. There is no need to further increase density, which leads to a density upper bound  $\rho_{\max} = \frac{1}{(2\tau_{\min})^3}$ . The resulting curve is extracted from the polynomial shown in Figure 5 by intersecting it with the plane  $\tau = \tau_{\min}$  — in practice we use  $\tau_{\min} \in [0.2, 0.3]$  mm depending on the printer.

For high densities, just before reaching a full block of matter, small pockets start to appear. This is due to the edges of the Voronoi cells merging together. As measured in Figure 7 the volume captured by pockets never exceeds 4%, and is negligible for relative volumes below 0.7.



**Figure 7:** Relative volume  $\omega$  and volume of pockets (%). Pockets have a negligible impact under a relative volume of 0.7 and go up to 4% of the full volume around 0.9 relative volume. Thanks to the quasi-linear relationship between relative volume and Young's modulus, we can expect at most a similar error on the elastic behavior.

Other strategies could be developed. For instance, for aesthetic purposes a user could use thicker beams and a lower density to reveal the structures at the same elasticity.

## 4.3 Elastic Behavior: Properties and Analysis

The collection of data points we produced for the parameter fitting lets us fully characterize our procedural foams. We now compare our results to the open-cell foams literature and verify our microstructures behave as expected.

### 4.3.1 Elasticity

The linear elastic response of *low density* ( $0.04 < \omega < 0.5$ ) open-cell foams is described by the following model [Gibson

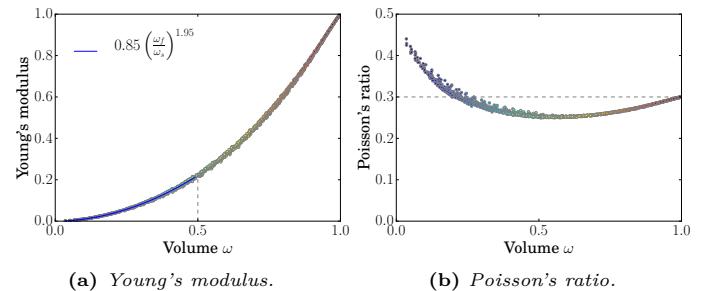
and Ashby 1997]:

$$\frac{E_f}{E_s} = C \left( \frac{\omega_f}{\omega_s} \right)^n \quad (2)$$

, where the subscript  $f$  denotes the foam and  $s$  denotes a solid block of base material.  $\frac{\omega_f}{\omega_s}$  defines the relative volume  $\omega$ . The constants  $C$  and  $n$  change depending on the foam family [Roberts and Garboczi 2002].

Figure 8a shows the relationship between the relative volume and the Young's modulus  $E$  of our procedural foams. We fit the model for low density foams on the range  $0.04 < \omega < 0.5$ , using  $C = 0.85$  and  $n = 1.95$ . As can be observed, our data fits this model very well.

Figure 8b shows the Poisson's ratio of our structures, and as can be seen it remains stable. While different materials will give different values of Poisson's ratio, it always remains stable around the Poisson's ratio of the base material as observed by Gibson and Ashby [1997] on open-cell foams.



**Figure 8:** Relative Young's modulus and Poisson's ratio. The base material has a Poisson's ratio of 0.3. The Poisson's ratio tends to 0.5 for low densities, and to the base material Poisson's ratio for high densities.

### 4.3.2 Randomization

Our procedural foams are based on a (pseudo) random process. Therefore, it is important to consider whether different realization using the same base parameters  $\rho, \tau$  produce a consistent Young's modulus. Figure 5 reveals this as for each pair  $(\rho, \tau)$  we produced three different realizations. Close inspection reveals that the different realizations do not perfectly match. The largest deviation to the fitted polynomial is 3.3% of the Young's modulus and therefore is negligible. This is in agreement with the literature where it was observed that randomization has little influence on the linear elastic properties of open-cell foams [Van Der Burg et al. 1997].

### 4.3.3 Regularity

Studies of naturally occurring foams often discuss the *regularity* of the foam, which in our case measures the minimum distance between Voronoi seeds in a bounded domain using the same number of seeds. A Poisson disc distribution provides the most regular cases, while a random point process is the less regular. Our procedural foams lie in-between.

Our seed generation draws at least one sample per grid cell. This is known to be a crude — yet very efficient — approximation of a Poisson disc distribution [Cook 1986]. The approximation is less evenly spaced than a high-quality Poisson disc distribution. However, less regular foams exhibit

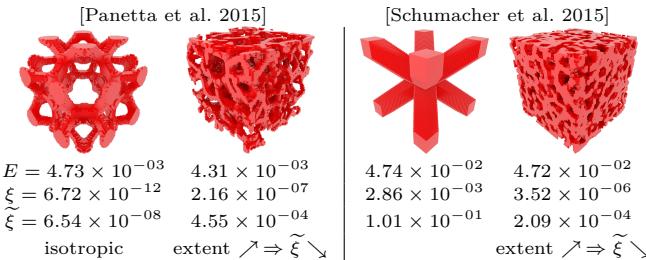
a better isotropy [Luxner et al. 2007], which is also desirable. Therefore, we believe that our jittered grid approximation is a good compromise between computational efficiency and elastic properties, while allowing for graded densities.

## 5 Applications and Results

We now experimentally challenge the behavior of our foams. In Section 5.1 we compare their isotropy to recent tile-based techniques, in Section 5.2 we measure actual printed samples, and in Section 5.3 we describe a complete application to fill 3D models with graded elasticity.

### 5.1 Isotropic Behavior Versus Periodic Tiles

Figure 9 compares the isotropy of the tensors computed by homogenization using tile-based methods. The method of Panetta et al. [2015] explicitly constructs isotropic tiles. When discretized in our homogenization process they provide close to perfect isotropy; the measured error is due to the limited numerical precision. At a same scale and resolution our structures exhibit lower isotropy; however, performing homogenization with increasingly larger volumes reveals that the residual quickly decreases (see Figure 6). This shows that filling larger volumes with procedural foams improves isotropy — a property that stems from the aperiodic stochastic nature of Voronoi foams. Figure 9 also shows a tile with a less isotropic behavior (manually designed to match [Schumacher et al. 2015]).

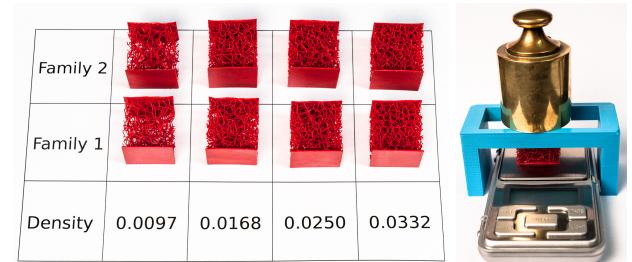


**Figure 9:** Relative distance  $\tilde{\xi}$  to an ideal isotropic tensor, for a fixed voxel size and a resolution of  $80 \times 80 \times 80$ . Both pairs have a similar Young’s modulus. The small isotropy error of the tile from [Panetta et al. 2015] (left) is due to limited numerical precision and decreases with finer discretization. As larger volumes are simulated, the isotropy of our structures quickly improves (see also Figure 6).

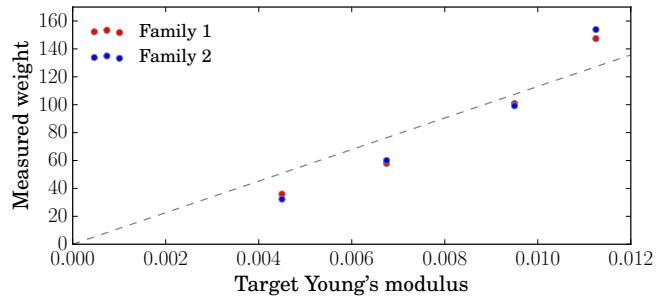
### 5.2 Experimental Results on Printed Samples

We verify the predicted elastic behavior of our foams on printed samples. We print two families of four samples with varying Young’s modulus. Each family uses a different random seed. All samples are printed on a B9Creator V1.2 using red-cherry material. The two families are visible in Figure 10.

We put each sample on a high precision scale and impose to each a same small displacement. We then measure the weight applied to the sample onto the scale (canceling out the structure weight). Due to the limited maximum measurable weight of the scale we consider samples with relatively close Young’s modulus. The results are reported in Figure 10. We obtain a good correspondence between the measured data and the prediction; the average deviation of the Young’s modulus from linear correlation (dashed line in Figure 5)



Family	Target Young’s modulus	Density by polynomial	Object weight (g)	Measured Weight (g)
1	0.0045	0.0097	1.47	35.9
	0.00675	0.0168	1.66	57.98
	0.009	0.025	1.82	100.82
	0.01125	0.0332	1.91	147.36
2	0.0045	0.0097	1.42	32.32
	0.00675	0.0168	1.67	60.06
	0.009	0.025	1.78	99.2
	0.01125	0.0332	2.07	153.86



**Figure 10:** Compression tests on printed samples. The measured weight is expected to be linearly correlated with the target Young’s modulus.

0.001434, that is less than 0.2% absolute error, and 25% relative error.

For reproducibility please note that the SLA print process, the subsequent cleanup and the material curing all have significant variabilities. We noticed on a few prints (2 out of 16) an incorrect final weight — we discarded these samples. We also noticed that the elastic behavior significantly changes in the first few hours after the print completed. Similarly to [Panetta et al. 2015] we wait for 24 hours before taking measurements.

### 5.3 Procedural Foam with Elasticity Gradients

The process we have described in Section 4 is capable of generating graded structures, with a spatially varying Young’s modulus. We exploit this property to generate objects with controlled elastic properties. The user inputs the spatially varying Young’s modulus as a scalar field in space, which we denote  $E(x)$ , with  $x$  a point in space. Our technique does not put any requirement on how the field is encoded (e.g. implicit function or interpolated from a grid), but it is expected that the field varies smoothly compared to the size of the Voronoi cells. Violating this expectation will *not* result in an incorrect structure, but the produced elasticity gradient will not be a good match to the input field.

The target Young's modulus field is converted upon lookup into a target density following the approach proposed in Section 4.2 — this requires a simple tabulation computed from Figure 5. This directly drives  $\rho$  in Algorithm 1. Unless otherwise specified, the beam radius remains fixed at  $\tau_{min}$ .

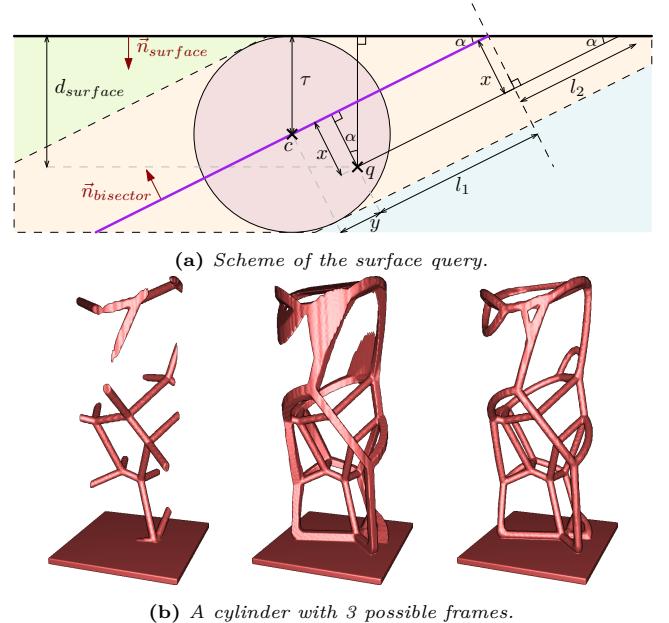
There are two main usage scenarios for our technique. A first scenario is to fill the inside of objects that have to remain rigid — simply adapting the inner density to varying stresses. In such a case the outer hull of the object remains solid and there is no additional challenge. The second scenario is to produce objects that can deform — flexible prosthesis and robot parts, toys. In such a case, printing the outer hull of the object would be detrimental to its flexibility. We therefore propose an object frame generation well suited to our approach. This frame is visible on all our 3D printed objects. It is fully procedural and only assumes that we have access to a (narrow band) distance field from the surface.

**Object Frame.** The key idea of our frame generation is to intersect the *faces* of the Voronoi diagram with a thick shell just below the surface, while taking care of cases where faces are almost parallel to the surface. Figure 11a explains our frame generation process.  $q$  is the query point and we want to know if it belongs to the solid structure or not. Here we consider two seeds  $s_1, s_2$  among the closest seeds from  $q$  as obtained from Algorithm 1. If the point  $q$  is too close from the surface, we need to keep the intersection of the Voronoi faces with the frame (in addition to the regular Voronoi edges). Simply using the distance from  $q$  to the bisector of  $\{s_1, s_2\}$  would yield the region colored in light orange, whereas we would like to select the point  $q$  iff it belongs to the beam of center  $c$  and thickness  $t$  (in light purple). However, we don't know the exact position of  $c$ , but only the distance to the border  $d$ , the distance to the bisector  $x$ , and an estimate of the angle formed by the surface and the bisector at point  $q$ , that we call  $\alpha$ . Assuming the surface is locally planar the rest follows from basic trigonometry: knowing  $l_1 + l_2 = \frac{d}{\sin(\alpha)}$ , and  $l_2 = \frac{x}{\tan(\alpha)}$ , we can compute  $y$  and test whether  $x^2 + y^2 \leq t^2$ .

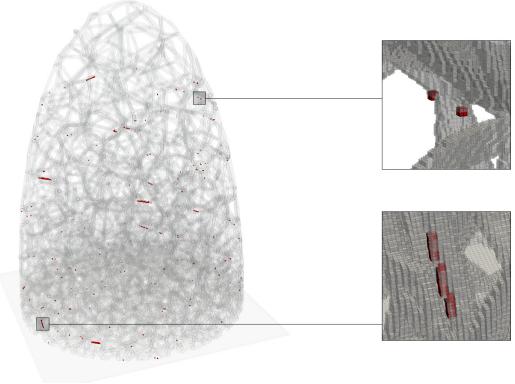
**Printed Results.** We now apply our approach to produce a variety of 3D printed results. We use two different printers: a B9Creator with red-cherry resin and an Autodesk Ember with standard clear resin. All these prints are prepared using our image based in-house slicer using the implicit procedural foam generation.

Our SLA printers require support structures. The objects we printed do not themselves require support — a property preserved by our microstructures but for a few cases along the object frame. These cases are rare, for illustration we selected one in Figure 11b (top of rightmost case). Table 1 shows the amount of volume filtered out due to those constraints. We filter them out during out-of-core slice generation, keeping track of connected components from one slice to the next. Adapting support techniques to our microstructures is left as future work — a possibility would be to connect a standard support to the closest microstructure beam. Figure 12 shows the location of isolated voxels filtered out before printing on SLA printers. SLS printers do not require support and can directly print our objects.

Figure 13 shows a very simple case of graded material applied to a 3D model. Figure 14 is a detailed 3D model filled with a spatially varying structure, denser along the surface and



**Figure 11:** *Top:* 2D planar cut orthogonal to the surface (top black line). The goal is the produce the beam which cross section is drawn as a circle. The purple line is a Voronoi face intersected by the view plane. *Bottom:* Extreme case of a frame on a small cylinder. From left to right, no frame, frame obtained by intersecting the Voronoi faces with the surface naively, frame obtained with our approach.



**Figure 12:** Illustration of the isolated voxels filtered out for resin printers. Removed percentage for each example is reported in Table 1. Greyscaled values in voxels (section 3.2) have been thresholded at isovalue 0.5 to show a binary design.

coarser inside. Note how the frame preserves the surface details. Figure 17 is a similar case on a sliced sphere, revealing the deformation behavior. Figure 18 is a model with varying elasticity. The model deforms as expected when pushing its head sideways. The object frame has a limited impact on the deformation behavior, while producing a much more visually pleasing object. It is however difficult to quantify precisely the mechanical influence of the frame. Figure 15 illustrated an android finger with built-in flexible articulations. Figure 1 shows a more complex case of a painted object, where we added a support manually. Figure 16 shows how the elasticity field can be controlled, e.g. by the SIGGRAPH

logo. Figure 19 illustrates how the elasticity field can be used to produce additional effects, such as anisotropic behaviors. Table 1 gives detailed statistics on all these prints, including average time per-slice. The throughput of our implementation averages to 7.8M pixel/sec on the slice images on an Intel® Core™ i7-4770K @ 3.50GHz, 16 GB RAM with a Titan Black NVidia GPU.

**Large Objects.** Our approach scales trivially with object size. While we are currently limited in the size of objects we can print, we illustrate this by producing a microstructure in a model that is about one meter in size. The sliced equivalent has 5 *tera*-voxels — but of course only a single slice ( $15402 \times 19814$ ) would have to fit in the printer memory. Figure 20 provides closeups of the depth map obtained by raycasting the implicit structure generated by our algorithm.

## 6 Discussion and Limitations

Our technique has a number of limitations. The foams only exhibit their target properties when a sufficiently large volume is printed — a limitation shared by all approaches relying on homogenization. This is however aligned with our goal of producing dense microstructures in large objects.

Compared to regular structures the stochastic nature of the foams produces more localized stresses. We have observed that a few beams fail under large deformations, perhaps earlier than on regular structures. This is specially the case along object frames with the Ember standard resin, which is more brittle. This would require further studies, noting that crushing behaviors of naturally occurring open-cell foams have received some interest [Gaitanaros et al. 2012].

Contrary to tile-based approaches we currently cannot provide spatially varying Poisson’s ratios. Studying stochastic structures that can vary both Young’s modulus and Poisson’s ratio is an interesting direction of future work. Generating structures with anisotropic behaviors is another natural venue for further studies.

## 7 Conclusion

The main advantages of our approach stem from the implicit formulation of stochastic microstructures. This has significant computational advantages by allowing for evaluation at slicing time and by avoiding to resort on global optimizations for each new object. The aperiodic and stochastic nature of the foams provide a simple and efficient way to grade the structures and to conform to target elasticity fields in space, without introducing artificial boundaries.

While we took a strict interpretation on the procedural nature of our structure generation, it is clear that other computational schemes could be envisioned. The important fundamental properties are 1) that the evaluation of the structure remains *local* and independent from the overall size of the domain and 2) that the elastic properties relate to the structure parameters through a simple relationship avoiding complex parameter fitting during evaluation.

Our approach deviates significantly from both the periodic tiling of microstructures and the optimization of macrostructures, by making a link between microstructures and procedural solid textures with controlled statistics in computer graphics. We believe there are many other such structures to

be discovered, and hope our work will spark further interest in procedurally generated, stochastic microstructures.

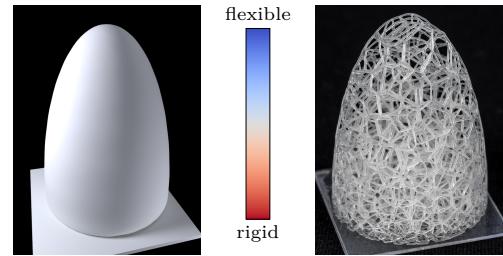


Figure 13: A simple graded ellipsoid.

## Acknowledgments

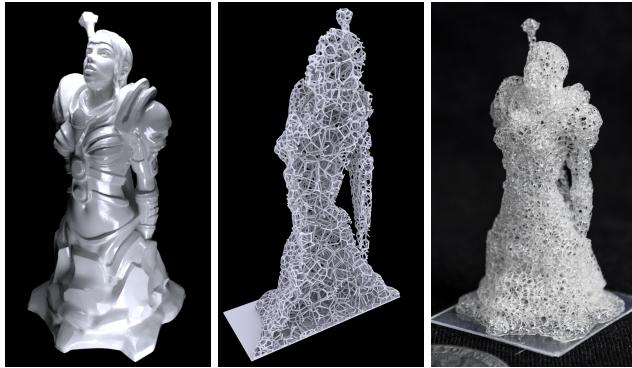
We would like to thank anonymous reviewers for their valuable comments, and the members of our research team for help with printing. This work was supported by ERC grant ShapeForge (StG-2012-307877).

## References

- ALLAIRE, G. 2012. *Shape optimization by the homogenization method*, vol. 146. Springer Science & Business Media.
- ANDREASSEN, E., LAZAROV, B. S., AND SIGMUND, O. 2014. Design of manufacturable 3D extremal elastic microstructure. *Mechanics of Materials* 69, 1, 1 – 10.
- AUTODESK, 2016. [www.autodesk.com/products/within](http://www.autodesk.com/products/within).
- BÄCHER, M., WHITING, E., BICKEL, B., AND SORKINE-HORNUNG, O. 2014. Spin-it: Optimizing moment of inertia for spinnable objects. *ACM Trans. Graph.* 33, 4.
- BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.* 29, 4, 63:1–63:10.
- BRACKETT, D., ASHCROFT, I., WILDMAN, R., AND HAGUE, R. 2014. An error diffusion based method to generate functionally graded cellular structures. *Computers & Structures* 138, 102–111.
- BRENNAN-CRADDOCK, J. 2011. *The investigation of a method to generate conformal lattice structures for additive manufacturing*. PhD thesis, Loughborough University.
- COOK, R. L. 1986. Stochastic Sampling in Computer Graphics. *ACM Trans. Graph.* 5, 1, 51–72.
- FRYAZINOV, O., VILBRANDT, T., AND PASKO, A. A. 2013. Multi-scale space-variant FRep cellular structures. *Computer-Aided Design* 45, 1, 26–34.
- GAITANAROS, S., KYRIAKIDES, S., AND KRAYNIK, A. M. 2012. On the crushing response of random open-cell foams. *Int. J. Solids Struct.* 49, 19–20, 2733 – 2743.
- GIBSON, L. J., AND ASHBY, M. F. 1997. *Cellular solids: structure and properties*. Cambridge university press.
- JOHNSON, S. G., 2007. The NLOpt nonlinear-optimization package.

Example		Extent (mm)	# Voxels	Volume	% Filtered	Time per slice (ms)
Moomin	fig. 1	26.7 × 40.8 × 51.9	534 × 815 × 1038	6.44%	0.005%	68.34
Ellipsoid	fig. 13	30.9 × 30.9 × 41.1	617 × 617 × 822	6.30%	0.001%	37.28
Knight	fig. 14	26.1 × 30.0 × 50.55	521 × 600 × 1011	12.50%	0.023%	20.25
Finger	fig. 15	25.0 × 23.25 × 70.5	500 × 465 × 1410	23.35%	0.006%	28.03
SIGGRAPH logo	fig. 16	20.0 × 40.0 × 80.0	400 × 800 × 1600	5.73%	0.003%	69.18
Half-dome	fig. 17	25.0 × 50.0 × 25.0	500 × 1000 × 500	19.49%	0.025%	71.22
Octopus	fig. 18	41.7 × 41.1 × 28.8	833 × 822 × 576	17.27%	0.009%	150.22
Anisotropic cube	fig. 19	40.0 × 40.0 × 40.0	800 × 800 × 800	26.86%	0.005%	113.52
Forest dragon	fig. 20	770.1 × 990.7 × 961.7	15402 × 19814 × 19234	N/A	N/A	1666.91

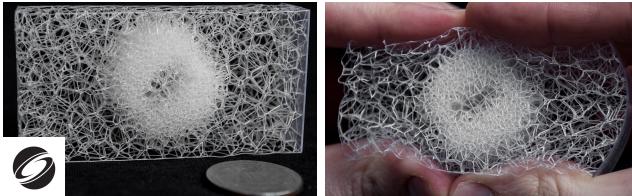
**Table 1:** Statistics on the examples shown in the paper. Beam thickness varies between 0.2mm and 0.4mm.



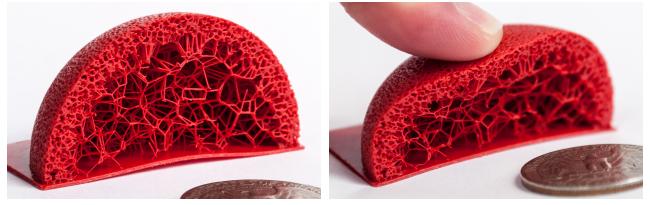
**Figure 14:** Knight model with a dense crust, and low-density interior. Model: Knight (thing:33804) by andreas.



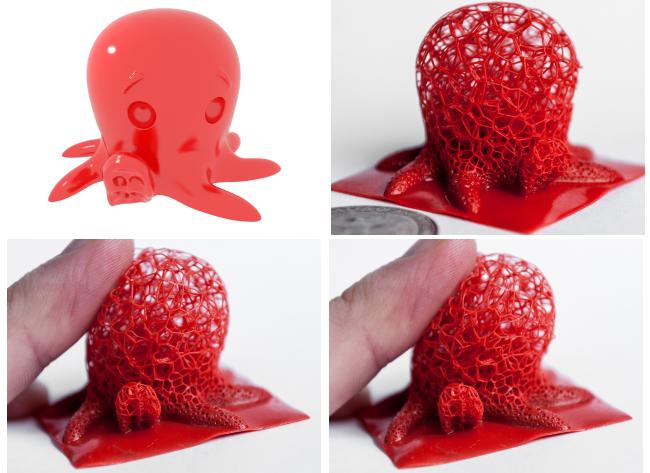
**Figure 15:** Finger with articulations. Right: deformed object. Note that the inferior articulation is made wider and thus bends further. Model: Olivier hand (AIM@SHAPE Project).



**Figure 16:** SIGGRAPH logo driving the density field.



**Figure 17:** A dome with variable density. No frame is generated on the cut revealing the interior.

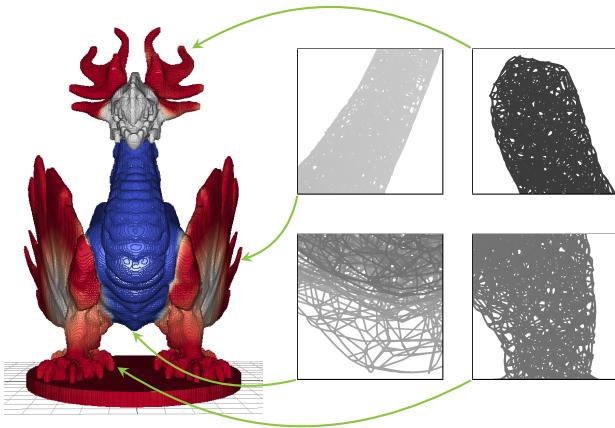


**Figure 18:** Cute octopus. From left to right: 3D model (original), printed model, printed with deformation (finger pressure). Model: Cute Octopus (thing:27053) by MakerBot.



**Figure 19:** Varying the density in one direction only can also produce parts with anisotropic behavior. The same weight of 8 kg has been applied to the cube placed in different orientations, leading to a visible deformation in the direction of the anisotropy.

- LAGAE, A., LEFEBVRE, S., DRETTAKIS, G., AND DUTRÉ, P. 2009. Procedural noise using sparse Gabor convolution. *ACM Trans. Graph.* 28, 3, 54:1–54:10.
- LAGAE, A., LEFEBVRE, S., COOK, R., DEROSSE, T., DRETTAKIS, G., EBERT, D., LEWIS, J., PERLIN, K., AND ZWICKER, M. 2010. A survey of procedural noise functions. *Computer Graphics Forum* 29, 8.
- LI, D., DAI, N., JIANG, X., AND CHEN, X. 2015. Interior structural optimization based on the density-variable shape modeling of 3D printed objects. *The International Journal of Advanced Manufacturing Technology* 83, 9.
- LU, L., SHARF, A., ZHAO, H., WEI, Y., FAN, Q., CHEN, X., SAVOYE, Y., TU, C., COHEN-OR, D., AND CHEN, B. 2014. Build-to-last: Strength to weight 3D printed objects. *ACM Trans. Graph.* 33, 4, 97:1–97:10.
- LUXNER, M. H., STAMPFL, J., AND PETTERMANN, H. E. 2007. Numerical simulations of 3D open cell structures – influence of structural irregularities on elasto-plasticity and deformation localization. *International Journal of Solids and Structures* 44, 9, 2990 – 3003.
- MEDEIROS E SÁ, A., MELLO, V. M., ECHAVARRIA, K. R., AND COVILL, D. 2015. Adaptive voids. *The Visual Computer* 31, 6-8, 799–808.
- PANETTA, J., ZHOU, Q., MALOMO, L., PIETRONI, N., CIGNONI, P., AND ZORIN, D. 2015. Elastic textures for additive fabrication. *ACM Trans. Graph.* 34, 4.
- PASKO, A., FRYAZINOV, O., VILBRANDT, T., FAYOLLE, P.-A., AND ADZHIEV, V. 2011. Procedural function-based modelling of volumetric microstructures. *Graphical Models* 73, 5, 165–181.
- PÉREZ, J., THOMASZEWSKI, B., COROS, S., BICKEL, B., CANABAL, J. A., SUMNER, R., AND OTADUY, M. A. 2015. Design and fabrication of flexible rod meshes. *ACM Trans. Graph.* 34, 4, 138:1–138:12.
- PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make it stand: Balancing shapes for 3D fabrication. *ACM Trans. Graph.* 32, 4, 81:1–81:10.
- RADMAN, A., HUANG, X., AND XIE, Y. 2013. Topology optimization of functionally graded cellular materials. *Journal of Materials Science* 48, 4, 1503–1510.
- ROBERTS, A., AND GARBOCZI, E. 2002. Elastic properties of model random three-dimensional open-cell solids. *Journal of the Mechanics and Physics of Solids* 50, 1, 33 – 55.
- SCHUMACHER, C., BICKEL, B., RYS, J., MARSCHNER, S., DARAIO, C., AND GROSS, M. 2015. Microstructures to control elasticity in 3D printing. *ACM Trans. Graph.* 34, 4, 136:1–136:13.
- SIGMUND, O., AND TORQUATO, S. 1999. Design of smart composite materials using topology optimization. *Smart Materials and Structures* 8, 3, 365.
- SIGMUND, O. 1994. Materials with prescribed constitutive parameters: An inverse homogenization problem. *Int. J. Solids Struct.* 31, 17, 2313 – 2329.
- SIGMUND, O. 1995. Tailoring materials with prescribed elastic properties. *Mechanics of Materials* 20, 4, 351–368.
- SKOURAS, M., THOMASZEWSKI, B., COROS, S., BICKEL, B., AND GROSS, M. 2013. Computational design of actuated deformable characters. *ACM Trans. Graph.* 32, 4.
- STATEN, M. L. 2007. Why is hex meshing so hard? Presentation at Sandia National Laboratories ([URL](#)).
- STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: Improving structural strength of 3D printable objects. *ACM Trans. Graph.* 31, 4, 48:1–48:11.
- UMETANI, N., AND SCHMIDT, R. 2013. Cross-sectional structural analysis for 3D printing optimization. In *SIGGRAPH Asia 2013 Technical Briefs*, 5:1–5:4.
- VAN DER BURG, M. W. D., SHULMEISTER, V., VAN DER GEISSEN, E., AND MARISSEN, R. 1997. On the linear elastic properties of regular and random open-cell foam models. *Journal of Cellular Plastics* 33, 1, 31–54.
- VIDIMČE, K., WANG, S.-P., RAGAN-KELLEY, J., AND MATUSIK, W. 2013. OpenFab: A programmable pipeline for multi-material fabrication. *ACM Trans. Graph.* 32, 4, 136:1–136:12.
- WANG, W., WANG, T. Y., YANG, Z., LIU, L., TONG, X., TONG, W., DENG, J., CHEN, F., AND LIU, X. 2013. Cost-effective printing of 3D objects with skin-frame structures. *ACM Trans. Graph.* 32, 6, 177:1–177:10.
- WORLEY, S. P. 1996. A cellular texturing basis function. In *Proceedings of SIGGRAPH*, 291–294.
- XIA, L., AND BREITKOPF, P. 2015. Design of materials using topology optimization and energy-based homogenization approach in Matlab. *Struct. Multidiscip. Opt.* 52, 6.
- XU, H., LI, Y., CHEN, Y., AND BARBIĆ, J. 2015. Interactive material design using model reduction. *ACM Trans. Graph.* 34, 2, 18:1–18:14.
- ZHOU, S., AND LI, Q. 2008. Design of graded two-phase microstructures for tailored elasticity gradients. *Journal of Materials Science* 43, 15, 5157–5167.
- ZHOU, Q., PANETTA, J., AND ZORIN, D. 2013. Worst-case structural analysis. *ACM Trans. Graph.* 32, 4, 137:1–137:12.



**Figure 20:** Closeups of the depth map of a large dragon (about 1m<sup>3</sup>) with painted elasticity, sliced at a resolution of 0.05mm. The computation domain is a volume of about 10<sup>12</sup> voxels. Model: Forest dragon (thing:87458) by dutchmogul.