Freely orientable microstructures for designing deformable 3D prints

Thibault Tricard, Vincent Tavernier, Cédric Zanni, Jonàs Martínez, Pierre-Alexandre Hugron, Fabrice Neyret, Sylvain Lefebvre

Freely orientable microstructures for designing deformable 3D prints

THIBAULT TRICARD, Université de Lorraine, CNRS, Inria, LORIA
VINCENT TAVERNIER, Université Grenoble Alpes, CNRS, Laboratoire Jean Kuntzmann, Inria
CÉDRIC ZANNI, Université de Lorraine, CNRS, Inria, LORIA
JONÀS MARTÍNEZ, Université de Lorraine, CNRS, Inria, LORIA
PIERRE-ALEXANDRE HUGRON, Université de Lorraine, CNRS, Inria, LORIA
FABRICE NEYRET, Université Grenoble Alpes, CNRS, Laboratoire Jean Kuntzmann, Inria
SYLVAIN LEFEBVRE, Université de Lorraine, CNRS, Inria, LORIA

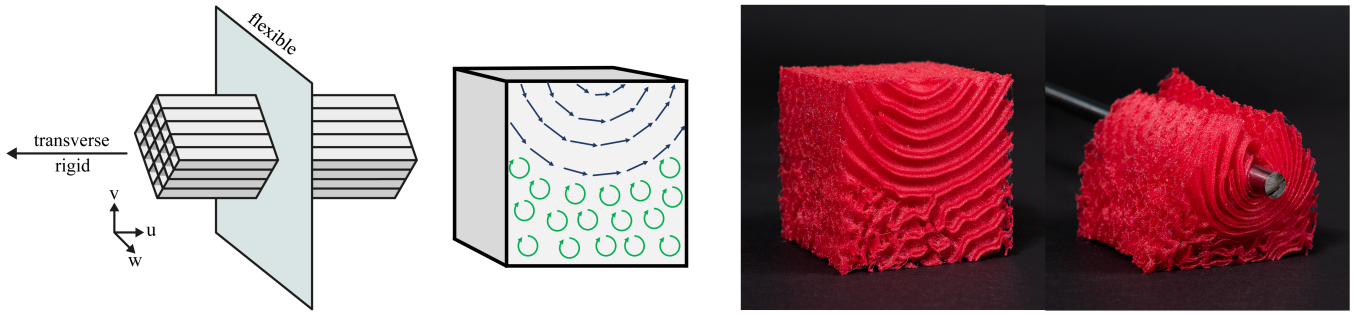


Fig. 1. Our technique synthesizes microstructures within 3D printed objects. We focus on controlling a specific type of elastic response that we call *rigid-transverse*: the microstructures are very rigid along a transverse axis, while being comparatively very flexible in the orthogonal plane. **Left:** Our microstructures without any gradation. They are formed by square diamond profiles defined in the v, w plane and extruded along the axis u. The result is very rigid along u and comparatively very flexible in the v, w plane. **Right:** 3D printed object embedding our synthesized microstructures. The rigidity axis is freely oriented to control how the cube volume reshapes under large deformations. Here, the top face collapses while gripping the bar. Note the progressively less strongly oriented structures at the bottom, which cancels the directional effect in this area. Our technique also allows to grade the microstructure density.

Nature offers a marvel of astonishing and rich deformation behaviors. Yet, most of the objects we fabricate are comparatively rather inexpressive, either rigid or having simple homogeneous behaviors when interacted with.

In this work, we focus on controlling how a 3D printed volume reacts under large deformations. We propose a novel microstructure that is extremely rigid along a transverse direction, while being comparatively very flexible in the orthogonal plane. By allowing free gradation of orientation within the object, the microstructure can be designed such that, under deformation, some distances in the volume are preserved while others freely change. This allows to control the way the volume reshapes when deformed, and results in a wide range of design possibilities. Other gradations are possible, such as locally and progressively canceling the directional effect.

To synthesize the structures we propose an algorithm that builds upon procedural texturing. It produces a cellular geometry that can be fabricated reliably despite 3D printing walls at a minimal thickness, for maximal flexibility. The synthesis algorithm is efficient, and scales to large volumes.

1 INTRODUCTION

With the advent of additive manufacturing (AM), we can now envision the design and fabrication of objects exhibiting a wide palette of deformation behaviors. The elastic response of a part can be significantly modified by carefully crafting a small scale architecture within its volume. Yet, despite major advances in recent years, we are still relatively limited when it comes to triggering large controlled deformations within an object, across different directions.

We believe that a promising approach to unlock this possibility is to produce freely orientable metamaterials having very high *contrast* in their directional elastic response: being near non-stretchable along some directions while being comparatively very flexible along others. Under large deformations, the contrast triggers internal curvature — internal distances are preserved along rigid directions but freely change along others. This results in controllable non-homogeneous deformations within volumes. This is illustrated in Figure 1 where the fiber-like microstructures — produced by our method — make the top of the deformed cube wrap around the bar being pressed on it.

Our approach allows to freely orient and grade such an elastic directional response throughout the inner volume of an object. We call our material *transverse-rigid* as it exhibits a strong rigidity along a transverse direction compared to its response in the orthogonal plane, mimicking the inclusion of rigid fibers in an otherwise very flexible medium (Figure 1). We also consider grading the contrast between the most rigid and most flexible directions. This allows to locally dampen or cancel the directional contrast, progressively transitioning towards a structure that has the same elastic response

Authors' addresses: Thibault Tricard, Université de Lorraine, CNRS, Inria, LORIA; Vincent Tavernier, Université Grenoble Alpes, CNRS, Laboratoire Jean Kuntzmann, Inria; Cédric Zanni, Université de Lorraine, CNRS, Inria, LORIA; Jonàs Martínez, Université de Lorraine, CNRS, Inria, LORIA; Pierre-Alexandre Hugron, Université de Lorraine, CNRS, Inria, LORIA; Fabrice Neyret, Université Grenoble Alpes, CNRS, Laboratoire Jean Kuntzmann, Inria; Sylvain Lefebvre, Université de Lorraine, CNRS, Inria, LORIA.

in all directions (ideally, an *isotropic* material). We refer to this control as the *rigidity contrast gradation*.

Defining periodic structures having a transverse-rigid behavior in a reference frame is relatively straightforward. However, without the ability to freely orient the transverse direction and grade the rigidity contrast, the design possibilities are severely restricted. Making such controls possible, however, requires synthesizing a geometry with specific spatially varying characteristics in terms of local shapes and distribution of orientations (Section 3 and Figures 2 and 3).

Besides triggering the target elastic response, a key challenge is to ensure that the synthesized geometry can be fabricated with AM. For the large deformations we target, internal voids and thin walls are paramount, as the combination of both will produce the least dense and most flexible objects. On most technologies, this leads to the major difficulty of avoiding any sort of support structure within the volume. 3D printing thin walls — at the minimal possible thickness — also implies enforcing a maximum overhang constraint: no wall should have an angle that exceeds a maximum with respect to the vertical direction of fabrication. Otherwise, the solidified curves forming the walls in each slice will not vertically bond to one another [Livesu et al. 2017].

The microstructure we synthesize is carefully crafted to achieve all the required properties. It prints reliably on even the most affordable AM process (filament printers), it exhibits the desired mechanical properties, it can be freely spatially graded in terms of orientation and rigidity contrast, and it is very fast to synthesize (supported by embarrassingly parallel algorithms).

To achieve this, we define the microstructures from a stochastic process introduced in the field of procedural texturing [Lagae et al. 2010a]. The stochastic component affords for a direct control over the distribution of geometric features, which translates into a control over the mechanical response. However, this comes with a specific challenge: such processes generate structures that can have a large number of localized defects (distortion, discontinuity). While this seems to make them less than ideal for our purpose, we propose a specialized filtering method that removes most of these issues from the final results.

We describe the geometry synthesis process in Section 3 and Section 4. To verify that the geometry we synthesize indeed produces a transverse-rigid oriented response, we fabricate a variety of samples covering the parameter space and submit each to mechanical testing. We report results in Section 5. We also fabricate a number of designs with interesting deformations, shown in Section 6 as well as in the accompanying video.

Contributions. In summary, our contributions are:

(1) The definition, from stochastic processes, of a microstructure with high transverse-rigidity, which orientation, density and rigidity contrast can be freely graded.
(2) A filtering approach to regularize the synthesized structures and strongly reduce the occurrence of defects.
(3) An evaluation of the obtained results in terms of manufacturability and mechanical properties.

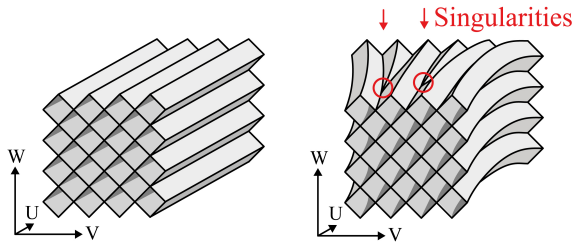


Fig. 2. **Left**: Microstructure without gradations. The extruded diamond shape has a high rigidity in the direction of extrusion u and is comparatively very flexible in the two orthogonal directions v and w. **Right**: Illustration of orientation gradation along an arbitrary direction field. Conforming to changes in direction while maintaining a near constant structure scale inevitably yields scattered singularities (red circles).

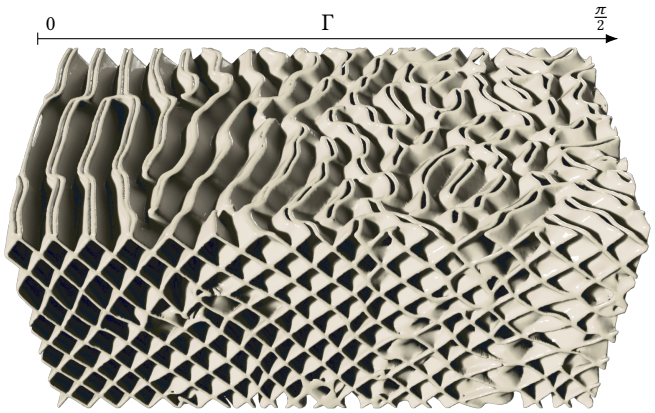


Fig. 3. *Rigidity contrast gradation*: the structure transitions from a high directional rigidity contrast (left) to a material with similar response in all directions (right).

2 MICROSTRUCTURES FOR ADDITIVE MANUFACTURING

There is a large body of work considering how microstructures, or small-scale architectures, modify the macro-scale behavior of an object. We describe here those most related to our approach, and in particular those studied in the context of AM.

Throughout this discussion, and in the remainder of the paper, it is important to recall that the mechanical behavior of microstructures is to be understood as the *average behavior* of a sufficiently large volume filled with the microstructure [Gibson and Ashby 1999].

Microstructures are synthesized inside an object following a variety of approaches. A traditional way is to repeat carefully optimized tiles within a periodic grid [Panetta et al. 2017, 2015; Schumacher et al. 2015; Zhu et al. 2017]. The periodicity affords for efficient simulation of material properties [**?**] and slicing [Brennan-Craddock 2011]. Gradation is achieved by choosing different tiles in different locations, ensuring their boundaries match, either by construction [Panetta et al. 2015], optimization [Schumacher et al. 2015] or parametric interpolation [Fryazinov et al. 2013; Li et al. 2015].

Microstructures can also be defined through subdividing geometries [Kuipers et al. 2019; Medeiros e Sá et al. 2015; Wu et al. 2016b], in which case density is directly controlled by the subdivision level.

Another approach is to generate the microstructures from stochastic point processes [Martínez et al. 2016, 2018, 2017]. This affords for scalable, efficient generation during slicing [Pasko et al. 2011; Vidimče et al. 2013], but also offers mechanical advantages [Pham et al. 2019; Tarantino et al. 2019]. These approaches allow for very progressive gradation and increased design freedom, as no grid or subdivision rule is constraining the structures.

When using microstructures, spatial gradations may be directly designed [Ion et al. 2016; nTopology 2019], or may be optimized for various objectives such as rigidity [Li et al. 2015; Zhu et al. 2017], target deformations [Panetta et al. 2015], or balance [Wu et al. 2016b].

Microstructures dedicated to additive manufacturing typically take fabrication constraints into account. These include minimal geometry size, maximal overhang [Martínez et al. 2018; Wu et al. 2016b], the existence of unsupported islands in slices [Panetta et al. 2015] as well as overlaps in deposition paths [Kuipers et al. 2019].

Freely orientable microstructures. Several works orient structures in a volume or along a surface, for instance conformal truss structures [Wang et al. 2005], periodic structures following a texture mapping analogy [Chen 2007], or procedural synthesis with free orientation [Martínez et al. 2017].

It is also possible to directly optimize a microstructure within a part, performing high-resolution topology optimization [Aage et al. 2017; Wu et al. 2016a] that allows complex oriented structures to emerge. A recent development consists in synthesizing microstructures following the output of a topology optimization process [Geoffroy-Donders et al. 2020; Groen et al. 2019; Pantz and Trabelsi 2008; Wu et al. 2019]. These techniques produce cellular (hexahedral) and lattice (truss) structures following an optimized direction field. This has direct links to the field of hex-meshing [Gao et al. 2017; Jakob et al. 2015a; Ray et al. 2006].

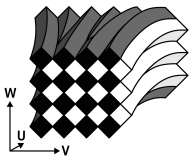
To the best of our knowledge, none of the approaches we mentioned allow for directional grading of a structure that can trigger deformations as large as ours within the object. Martínez et al. [2017] orient orthotropic microstructures within a volume. However, the open-cell beam structures are delicate to fabricate (requiring SLA or SLS) and fragile under large deformations. Specifically, for the same density closed-cell structures have in general higher fracture toughness than open-cell structures [Maiti et al. 1984]. The closed-cell structure proposed by Kuipers et al. [2019] is very flexible and collapsible along the vertical axis; however, no gradations other than density are possible.

3 SYNTHESIZING TRANSVERSE-RIGID MICROSTRUCTURES

In this section, we describe how we define and synthesize our microstructures. Note that despite guidelines (e.g. [Ashby 2006]) and in-depth studies of specific cases (e.g. [François et al. 2017; Schumacher et al. 2018]), it remains very challenging in general to precisely predict the link between a small-scale geometry and its macro-scale mechanical behavior. This is made even more difficult if taking into account large deformations and fabrication with layered AM processes.

In this section we seek to define a geometry with characteristic features that we *anticipate* to result in the target mechanical behavior. We back up our prediction with measurements in Section 5.2, and demonstration samples in Section 6.

We synthesize a cellular pattern, which walls are meant to be printed using the minimal thickness of the target AM process. Using thin walls allows to produce the highest number of cells per unit of length, which helps reach the average structural behavior in even small volumes.



We do not directly model the walls of the structure, but instead design a function that synthesizes a black and white cellular pattern which boundaries will represent the walls (see Figure inset). So our procedure returns the color of the cell enclosing an evaluation point at a coordinate $\mathbf{x}$. We later extract the wall geometry in each AM fabrication layer (Section 3.3).

Our structure forms elongated cells aligned with an input direction field D (normalized direction vectors). The cross-section size of the cells in the plane orthogonal to D is controlled by a frequency field F (cells per unit of length). A third scalar field, Γ, defines an angular spread around D (in radians, 0 to $\pi/2$). Γ controls how much the cells locally deviate from D, locally taking random angles within the defined angular spread.

In terms of physical properties, we design the structure such that the transverse rigidity is aligned with D. The density ρ is controlled by F, while Γ impacts the rigidity contrast, from 0 (highest contrast) to $\pi/2$ (no contrast, close to isotropic elasticity). Please refer to Section 5.2 for measurements validating these relationships on actual samples.

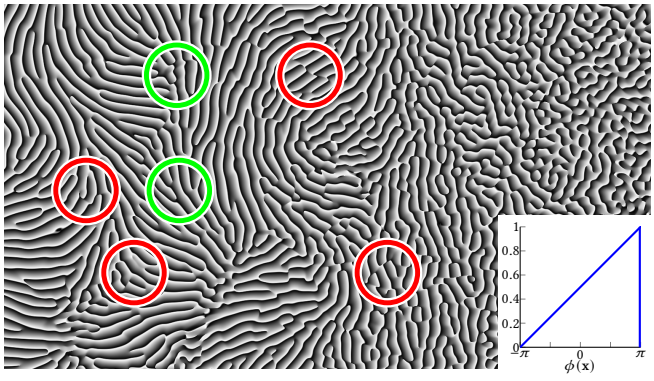


Fig. 4. Phasor noise with a sawtooth wave profile (inset). The main orientation of the kernels and the amount of angular spread is controlled by user-defined fields. The paradoxical requirements of following orientation and preserving a constant wave length yield two singularity types: curves — breaking the pattern (in red) — and points — creating branching waves (in green). The profile is distorted in those areas, introducing unwanted high-frequency content and discontinuities.

3.1 Design challenges

Consider a very flexible periodic 2D pattern in the xy-plane, typically a bending dominated structure [Ashby 2006]. By extruding it vertically along the z-axis, we produce a 3D pattern that remains flexible in the xy-plane but is very rigid along the z-axis (as vertical walls strongly resist extension and compression). Thus, we can expect that by sweeping such a 2D pattern along the orientation field D, the alignment of the extruded walls with D would locally trigger a similarly aligned local transversal rigidity.

To control the rigidity contrast, we expect to have to modulate the microstructure geometry from a pattern that has mainly walls aligned with D, to a pattern that has a uniform distribution of wall orientations. This is achieved through the angular spread Γ.

We are now facing an especially difficult task, as we seek to have strongly oriented regular patterns in some locations (locally an extruded 2D pattern), and progressively grade towards evenly distributed spatial structures. Prior work exists for generating very regular stripe patterns, e.g. [Jakob et al. 2015a; Knöppel et al. 2015]. However, we are not aware of any that allows for a gradation towards a pattern with no directional preference.

Meanwhile, the walls have to obey everywhere the overhang constraints of AM processes, as they will be printed at minimal thickness. We address these challenges in the following Sections.

3.2 Stochastic structures

Stochastic procedural texturing methods offer rich controls over the patterns that can be synthesized. These are usually focused on producing seemingly random patterns that obey a specific spectral content in terms of frequency, direction and bandwidth [Lagae et al. 2010b]. We note that a perfectly regular stripe pattern can be described in terms of spectral content. For instance, a perfect sine wave is a Dirac in the spectral domain. As procedural methods with increasingly precise spectral content are developed [Goldberg et al. 2008; Lagae et al. 2009b], these offer an interesting alternative to parameterization methods for generating stripe patterns (e.g. [Jakob et al. 2015a; Knöppel et al. 2015]).

We thus choose to define our microstructures from a phasor noise [Tricard et al. 2019]; a stochastic pattern generator that can produce sharp; contrasted oscillating patterns (see Figure 4). The oscillations frequency and orientation can be spatially graded at will, and can spatially evolve from a locally regular periodic oscillation towards a stochastic pattern having no clear preferred orientation.

The produced oscillations are significantly less regular than those produced by parametrization methods. However, this trade-off is compensated by the ability to spatially grade the patterns at the computational efficiency of procedural methods.

We therefore use this methodology to define a geometric microstructure having properties similar to those of phasor noise, and later focus on improving the quality of the synthesized oscillations (Section 4).

3.2.1 Phasor noise. A phasor noise is an instantaneous phase field ϕ which is defined from a complex version $\mathcal{G}$ of Gabor noise:

$$\phi(\mathbf{x}) = \mathrm{Arg}\left(\mathcal{G}(\mathbf{x})\right) \tag{1}$$

with:

$$\mathcal{G}(\mathbf{x}) = \sum_j e^{-b\|\mathbf{x}-\mathbf{x}_j\|^2} e^{2\mathrm{i}\pi f_j \mathbf{d}_j\cdot(\mathbf{x}-\mathbf{x}_j)+\mathrm{i}\varphi_j} \tag{2}$$

This is a sparse convolution process [Lewis 1989]. A finite set of kernels located at random locations $\mathbf{x}_j$ in a domain combine into the final signal. Each kernel is a product of two complex functions. The first is a Gaussian of bandwidth b centered on the kernel, the second is an oscillator of frequency f_j, phase φ_j and direction $\mathbf{d}_j$. In our case, $\mathbf{x}$, $\mathbf{x}_j$ and $\mathbf{d}_j$ are 3D vectors, thus $\mathcal{G}$ and ϕ define solid noises (see Figure 5, left).

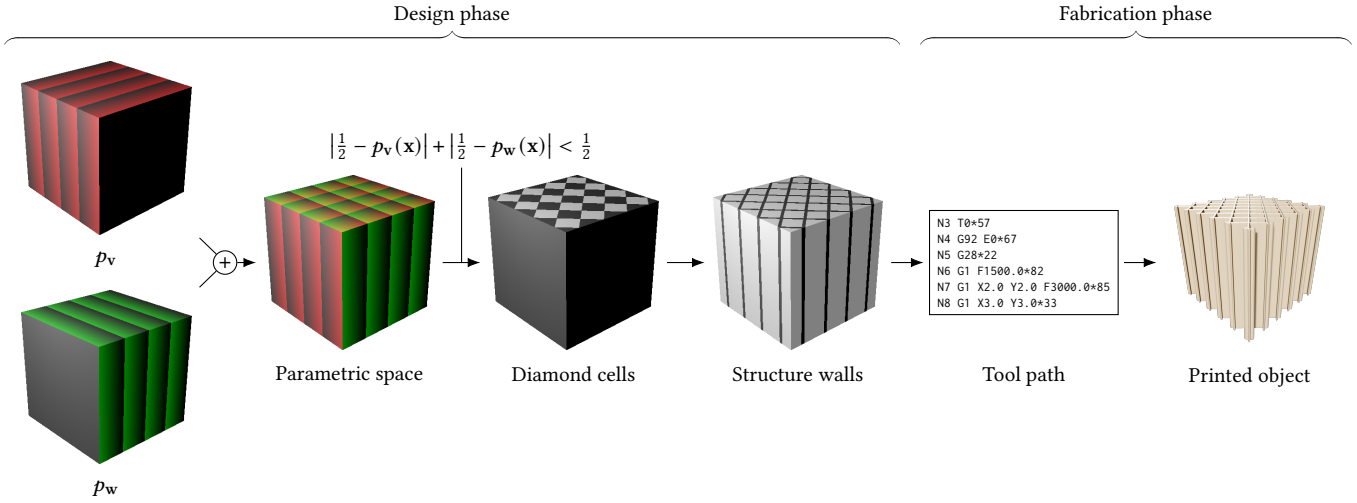


Fig. 5. Overview of the microstructure geometry synthesis, without any spatial gradation for clarity. We combine solid phasor noises p_v and p_w to define local u, v parametrizations in $[0,1]^2$. These local parametrizations are then used to define diamond cells, from which the structure walls are extracted. AM tool paths are generated from this structure in order to 3D print the resulting object.

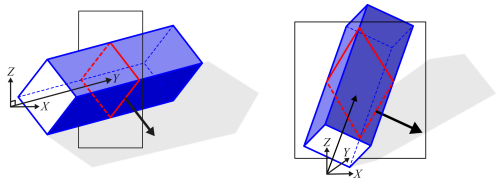


Fig. 6. A square diamond cell can cover the full unit-sphere $\mathcal{S}^2$ while verifying a 45 degrees overhang angle constraint. **Left**: Square diamond shape extruded along its normal, Y. All walls have angles of exactly 45 degrees with respect to the horizontal plane. **Right**: Rotation of the diamond cell around its horizontal diagonal (X). This further increases the vertical cross-section wall angles (red lines), which still verify the overhang constraint.

By feeding the phase Arg $(\mathcal{G})$ into a periodic function, one obtains oscillations having the profile of the periodic function, together with the ability to freely grade the orientation and frequency of the oscillations (see Figure 4).

3.2.2 Local parametrizations. We define our structures using local parameterizations synthesized from phasor noise. We compute two oscillating waves, each with a sawtooth profile. The sawtooth oscillations vary linearly from 0 to 1 in each period, and thus define local 1D parameterizations. We orient the two waves such that their directions are locally orthogonal to the transverse direction D: the first follows $\mathbf{v}(\mathbf{x}) = \mathbf{z} \times D(\mathbf{x})$, the second follows $\mathbf{w}(\mathbf{x}) = (\mathbf{z} \times D(\mathbf{x})) \times D(\mathbf{x})$. We denote the waves respectively $p_\mathbf{v}$ and $p_\mathbf{w}$. At a given point $\mathbf{x}$, the phasor sawtooth evaluates as:

$$p_\mathbf{v}(\mathbf{x}) = \frac{\text{Arg}\left(\mathcal{G}_\mathbf{v}(\mathbf{x})\right)}{2\pi} \tag{3}$$

where $\mathcal{G}_\mathbf{v}$ has the form given in Equation 2, with the kernel directions $\mathbf{d}_j$ computed as $\mathbf{v}(\mathbf{x}_j)$ (and similarly for $p_\mathbf{w}$, using $\mathbf{w}(\mathbf{x}_j)$), see Figure 5 (left). The phasor sawtooth waves depicted in the remainder of the paper are planar slices of the 3D noise.

The crossing of the two waves defines small local $[0,1]^2$ parametrization patches that sweep along $\mathbf{u} = D(\mathbf{x})$. We use these parametric patches to define the swept shapes forming the elongated structures. This is illustrated in Figure 5. The choice of extrusion profile will impact the final mechanical response as well as determine whether the result is fabricable under overhang constraints. We discuss it in Section 3.2.3.

To generate cells with an increasing angular spread of wall directions we follow the user field Γ, perturbing the kernel orientations $\mathbf{d}_j$ only in $p_\mathbf{v}$ to remain within overhang constraints. As in phasor noise, this is done by adding a random angle based on the angular spread in Γ. When $\Gamma = \pi/2$, this results in oscillations in all directions (Figure 4 right), that still properly define parametric patches.

While very effective, defining the local parameterizations with phasor noise comes with its own challenges: the phasor fields have local defects that damage the structure. We discuss these in Section 3.2.4.

Implementation. We implement our phasor noises as follows. The positions of the kernels $(\mathbf{x}_j)$ are obtained from a random sampling providing a dense spatial coverage of the domain. In practice we use a grid approach as in [Tavernier et al. 2019], with a constant number of kernels per cell (8). The position of a kernel is initialized with a uniform random value within the virtual grid cell. For each kernel, f_j and $\mathbf{d}_j$ are directly sampled onto the user input fields F and D. The kernel orientations of $p_\mathbf{v}$ are perturbed to handle the angular spread Γ. This perturbation should keep both wave patterns $p_\mathbf{v}$ and $p_\mathbf{w}$ locally orthogonal in order to preserve the squared diamond shape. Therefore, the perturbation is computed by applying a rotation of axis $\mathbf{w}(\mathbf{x})$ to the kernel direction $\mathbf{d}_j$ of $p_\mathbf{v}$. The rotation angle is uniformly sampled in $[-\Gamma(\mathbf{x}), \Gamma(\mathbf{x})]$. φ_j is initially set to 0.

The choice of bandwidth impacts the size of the kernels in the spatial domain, with a larger bandwidth resulting in smaller kernels. We choose the bandwidth large enough so that the kernels are small enough to properly reproduce rapid spatial changes of the user fields. However, we keep the kernels large enough to not allow introducing changes smaller than the structure feature size, which would break the pattern. Using a large bandwidth also induces an increase of the total number of kernels to fill a same target volume, thus increasing the computational cost of the method. The frequency being a direct indicator of the final structure feature size, we use the minimal frequency to determine the bandwidth. In practice we use $b = \frac{c}{F_{min}}$ with $c = 4.5$ which gave the best results based on our experiments.

3.2.3 Diamonds and overhangs. To allow for free orientation, the transverse axis of the generated microstructures should be able to cover any direction in the full unit-sphere $\mathcal{S}^2$. However, the structure also has to always verify the overhang constraints for correct fabrication.

Under this objective, a square diamond extrusion profile has desirable properties. Let us consider a diamond shape having one of its diagonal aligned with the vertical printing direction, z. The extrusion of the diamond shape along its normal defines a cell with a diamond cross-section (see Figure 6, left). In this case, the wall angles with the horizontal plane are exactly 45 degrees, which typically enforces the maximum overhang angle constraint [Livesu et al. 2017]. Any such diamond cell remains printable for arbitrary rotations around the z-axis.

Now, if we rotate the diamond cell around the horizontal diagonal of the base diamond shape, the wall angles with respect to the horizontal plane actually increase (see Figure 6, right). Thus, any such rotated cell remains well within overhang constraints. (Of course, the bottom edge along the extrusion direction is unsupported in this discussion; however in the final structure it is supported from below by another diamond cell.)

By composition of the two previous types of rotations, all possible directions for the transverse axis can be achieved. In practice, we directly sample the transverse direction $\mathbf{u}$ from the input direction field D. We obtain the two diamond diagonal directions as $\mathbf{v} = \mathbf{z} \times \mathbf{u}$ and $\mathbf{w} = (\mathbf{z} \times \mathbf{u}) \times \mathbf{u}$.

We obtain the black diamond cells by thresholding the local parametrizations, using the following formula:

$$\left|\frac{1}{2} - p_\mathbf{v}(\mathbf{x})\right| + \left|\frac{1}{2} - p_\mathbf{w}(\mathbf{x})\right| < \frac{1}{2}$$

White cells are obtained by reversing the inequality.

In terms of mechanical response, the diamonds provide a very flexible structure overall, but are not ideal as their straight walls

produce rigid spikes in the **v**, **w** plane. We further analyze this trade-off in Section 5.2.

Additional rotational freedom. Even though our implementation does not exploit it, there is, in fact, additional freedom under overhang constraints. In addition to the two rotations mentioned before, we can also consider a rotation around the diamond shape normal (roll).

For a square diamond, it is possible to derive a closed-form formula for the constraints on the rotation θ around the extrusion axis. Noting α the rotation around the horizontal diagonal, and ϵ the cosine of the limit overhang, we have:

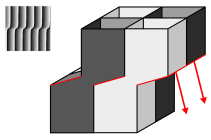
$$|\theta| < \frac{1}{2}\arcsin\left(2\frac{\epsilon^2}{\cos^2\alpha} - 1\right) \text{ if } \cos^2\alpha > \epsilon^2, |\theta| < \frac{\pi}{4} \text{ otherwise} \tag{4}$$

The derivation of Equation 4 is given in the supplemental material. Also note that, under stricter overhang constraints, a vertically stretch diamond could be considered.

3.2.4 Singularities. Using phasor noises as a basis to generate our microstructures has several intrinsic advantages. The synthesis process is fast, as it is supported by an embarrassingly parallel algorithm implemented on the GPU. The approach easily scales to generating small-scale structures within comparatively large volumes. The user has great freedom of design, and can quickly visualize the final structure produced from the control fields.

However, this comes with a specific challenge: the fields produced by phasor noise exhibit *singularities*. These are locations where the behavior diverges from an ideal regular oscillation. We know from the hex-meshing literature [Bommes et al. 2013] that given a non trivial orientation field a number of singularities have to exist. However, these singularities should ideally be points (in a slice) or curves (in 3D) where oscillations meet or split (see Figure 2, right). For the sake of clarity, since we show in figures 2D slices of 3D noises we will refer to the ideal case as point singularities.

Point singularities at the apex of branching oscillations (forks) do not endanger the overall properties of the structure: around the singular point, the produced shape still exhibits a diamond extrusion profile and prints reliably, with only a small hole at their base.



Unfortunately, we also observe *curve* singularities in the phasor noise (*surface* singularities in the 3D case); across such curves the local parameterization values abruptly change and oppose on both sides (see red circles in Figure 4). These are more problematic than point singularities as they induce a change from a black to a white diamond (and vice-versa) resulting in walls orthogonal to the local diamond extrusion direction — see red walls in the inset. Such walls violate the overhang constraint if the orientation is vertical, and introduce undesirable rigidity in other cases.

We thus focus in Section 4 on the problem of filtering *curve* singularities from phasor noises.

3.3 Toolpath extraction

For the sake of completeness, before we proceed to filtering let us briefly describe how we extract the final structure for 3D printing.

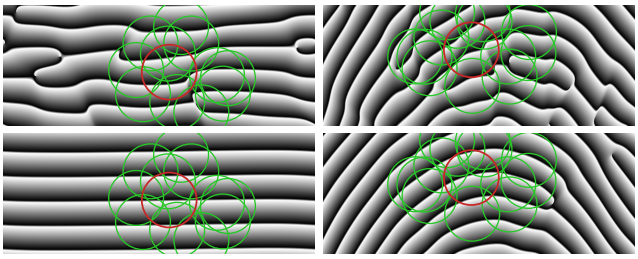


Fig. 7. **Top**: Sawtooth oscillations directly produced by a phasor noise, with constant orientation (left) and smooth changes (right). A kernel (red) and its neighbors (green) are shown. The misalignment in kernel phases produce complex interactions. **Bottom**: Our algorithm aligns kernel phases to produce a more regular result.

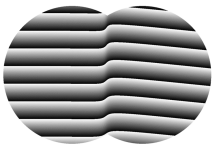
We seek to produce a single, as thin as possible wall between diamonds. Given a slice, we first draw an image of the sliced structure by evaluating whether each pixel belongs to a black or white diamond cell, as defined in Figure 5. This results in a binary image, from which we extract a solidification trajectory along each frontier between black and white.

This is a robust process, which only drawback is the image discretization error. However, we can use a high sampling density — smaller than half of the minimal thickness — since memory consumption is low: we work in a single slice and evaluate the pixel 'color' using a GPU pixel shader. The shader evaluates the phasor noises with sawtooth profile directly from the kernels. The kernels themselves use little memory, and the memory allocated for a slice is freed immediately after trajectory extraction.

4 FILTERING SINGULARITIES

The main drawback of relying on phasor noises is that the oscillations exhibit many singularities, even when D is smooth overall and $\Gamma = 0$. In this case one would expect a very regular result with only a limited number of singularities, with smooth and progressive changes of frequency and orientation.

We now describe our oscillation regularization approach to strongly reduce the occurrence of singularities in the synthesized fields, while preserving the variability associated to changes of orientation, frequency and angular spread.



Recall that our focus is on *curve* singularities, which have a detrimental effect on the synthesized geometry (see Section 3.2.4 and Figure 4). Such singularities most notably arise when two neighboring kernels with similar direction $\mathbf{d}_j$ are out of phase — see Figure inset.

Our regularization works at two fundamentally different levels. We first observe that the stochastic process of the phasor noise cannot possibly produce phase coherency between kernels across long distances. Thus, we apply an iterated process that operates on the phasor noise kernels, and seeks to align the phases of neighboring kernels — removing inconsistencies — in order to produce longer coherent wavefronts (see Figure 7 left). It is described in Section 4.1.

Note that a perfect phase alignment rarely exists in presence of variations in the orientation field D. Because of this, even after phase alignment, some local singularities remain where neighboring kernels of different orientation or frequency combine. To further improve the result, we define a specialized filter that locally removes deviations from an idealized oscillation estimated in every sampling point. We describe the filter in Section 4.2.

4.1 Iterated phase alignment

To regularize the oscillations of the phasor noise, we first adjust the phase shifts of neighboring kernels, as illustrated in Figure 7. Interestingly, this problem appears in a related form in recent interactive quad meshing techniques [Evgeny and Harders 2019; Lichtenberg et al. 2018]. These operate on triangles and have one oriented oscillator (akin to a phasor kernel) per vertex. They attempt to globally align the phases of the oscillators, so as to define continuous patterns across triangles by applying an iterative diffusion process. To achieve high performance, the phase at every vertex is iteratively updated, in parallel for all vertices, from the average of phases suggested by the neighbors. Note that the orientation and frequency of the oscillators remains fixed.

We adopt a similar approach in the context of phasor noise.

4.1.1 Algorithm. At each iteration, the algorithm adjusts the phase of each kernel such that it complies better with the phases of its neighbors. As we manipulate complex numbers, for a kernel k_j this amounts to evaluating the oscillator of each neighboring kernel at $\mathbf{x}_j$. Aligning phases between kernels having large differences in orientation is not sound. Thus, the average is weighted by the dot product of orientation vectors between k_j and its neighbor, clamped to zero. This weighting scheme accommodates in particular for the varying angular spread Γ. We then take the phase (argument) of the average oscillator.

The algorithm for one iteration is given in Algorithm 1. It trivially maps to an efficient GPU implementation, with one thread per kernel. The neighborhood $\mathcal{N}(k_j)$ of kernels interacting with k_j is efficiently retrieved from the kernel grid, as is done when evaluating a Gabor (or phasor) noise [Lagae et al. 2009b]. In addition, this neighborhood is the same for all k_j in a same grid cell, enabling an efficient pre-fetching in shared memory of the set of neighbor kernels.

Algorithm 1 Improving the phase of kernel k_j

$a \leftarrow 0$
for all $k_i \in \mathcal{N}(k_j)$ **do**
  $a \leftarrow a + \mathbf{d}_j \cdot \mathbf{d}_i e^{2i\pi f_i \mathbf{d}_i \cdot (\mathbf{x}_j - \mathbf{x}_i) + i\varphi_i}$
end for
$\varphi_j \leftarrow \mathrm{Arg}(a)$

4.1.2 Iterating. We repeat Algorithm 1 for a fixed number of iterations (20 in our implementation), which we experimentally adjusted by observing convergence with respect to a *singularity energy* that we derive in Section 4.2.2 (we postpone detailed explanations as we need to introduce additional concepts first). The average time for one iteration on an NVIDIA GTX 1080 is 7*ms* for 551368 kernels for a 100 mm × 100 mm × 100 mm cube.

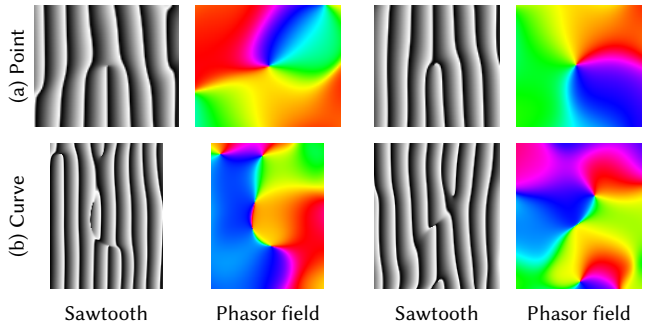


Fig. 8. Some singularities in a phasor sawtooth example. Point singularities (a) represent forks and merges in the pattern. They correspond to rotations around a singular point in the phasor field. Curve singularities (b) introduce major discontinuities in the result as well as profile distortions in various directions. They are characterized by long curves with a high phase gradient. Phasor field images show Arg $(\varphi(\mathbf{x}))$ encoded as the hue.

4.2 Evaluation Filter

While phase alignment removes overall inconsistencies, the complex local interactions between disagreeing kernels still creates local singularities. We propose a filter that regularizes the signal, filtering away discontinuities which locally distort the noise profile.

We start by examining the singularities and their properties in order to identify and classify them, and quantify their presence in the generated noise.

4.2.1 Identifying singularities. Let us first consider a simple case where the frequency and the orientations are constant in space, noted respectively f and $\mathbf{d}$. In this case we can extract the *phasor field* of the phasor noise. It is obtained at each point $\mathbf{x}$ by dividing the complex Gabor noise by a global oscillator ω_o:

$$\omega_o(\mathbf{x}) = e^{2i\pi f \mathbf{d}\cdot\mathbf{x}} \tag{5}$$

The phasor field is thus obtained as:

$$\varphi(\mathbf{x}) = \frac{\mathcal{G}(\mathbf{x})}{\omega_o(\mathbf{x})} \tag{6}$$

It represents the phase shift between a specific phasor noise pattern and a hypothetical perfectly regular oscillator (i.e. a simple sine wave). Examples of phasor fields are shown in Figure 8.

Remark 1. *The smoother the phasor field is, the less the pattern deviates from an ideal oscillator. Thus the phasor field variations characterize deviations from the ideal signal (the singularities).*

Note that there is a notion of scale in this remark: deviations become problematic when they happen over a short distance compared to the oscillations wavelength. This wavelength is determined by the local frequency:

Remark 2. *The frequency of the phasor noise defines the relative variation scale at which deviations can be considered as singularities.*

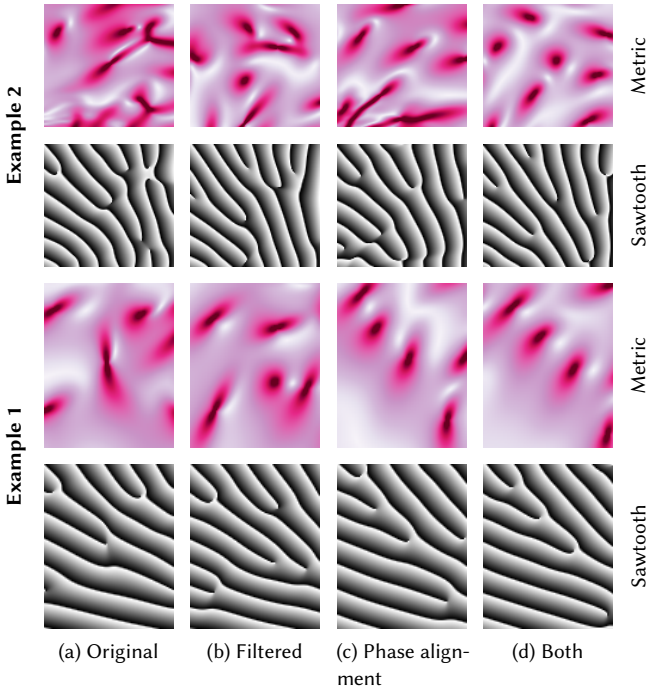


Fig. 9. Singularity measure $\mathcal{S}(\mathbf{x})$. Darker spots in pink metric images indicate areas with a high $\mathcal{S}(\mathbf{x})$ value. Aligning kernel phases through regularization outright removes curve singularities and some point singularities (c). The filter reduces the extent of remaining singularities to turn them as much as possible into point singularities with a smooth phasor field gradient (b). The best results are obtained by combining both approaches (d).

Indeed, variations that are smooth with respect to the local frequency are not problematic, as the phasor noise smoothly and progressively shifts around the ideal oscillator. However, where the phasor field varies too abruptly, the noise profile is locally compressed, which results in displeasing discontinuities. This is coherent with the visual examination of the fields in Figure 8.

Remark 1 and Remark 2 offer two opportunities, both considering the gradient of the phasor field. The first is to derive a measure of the quantity of singularities by considering the gradient of the phasor field (Section 4.2.2). The second is to perform filtering by directly smoothing the phasor field (Section 4.2.3). However, this only holds for a constant orientation and frequency over the whole surface. A primary objective of our work is to allow arbitrary spatial gradations, therefore a more general point of view is needed.

We note that under the assumption that the control fields are smooth, they can be *locally* considered constant. We start by rewriting the global oscillator for an arbitrary origin point $\mathbf{c}$:

$$\omega_{\mathbf{c}}(\mathbf{x}) = e^{2i\pi f \mathbf{d}\cdot(\mathbf{x}-\mathbf{c})} \tag{7}$$

The field $\frac{\mathcal{G}(\mathbf{x})}{\omega_{\mathbf{c}}(\mathbf{x})}$ is the same as $\varphi(\mathbf{x})$ with an additional constant phase shift $e^{2i\pi f \mathbf{d}\cdot(-\mathbf{c})}$. Thus, it has the same gradient, and gives the same result under Remark 1.

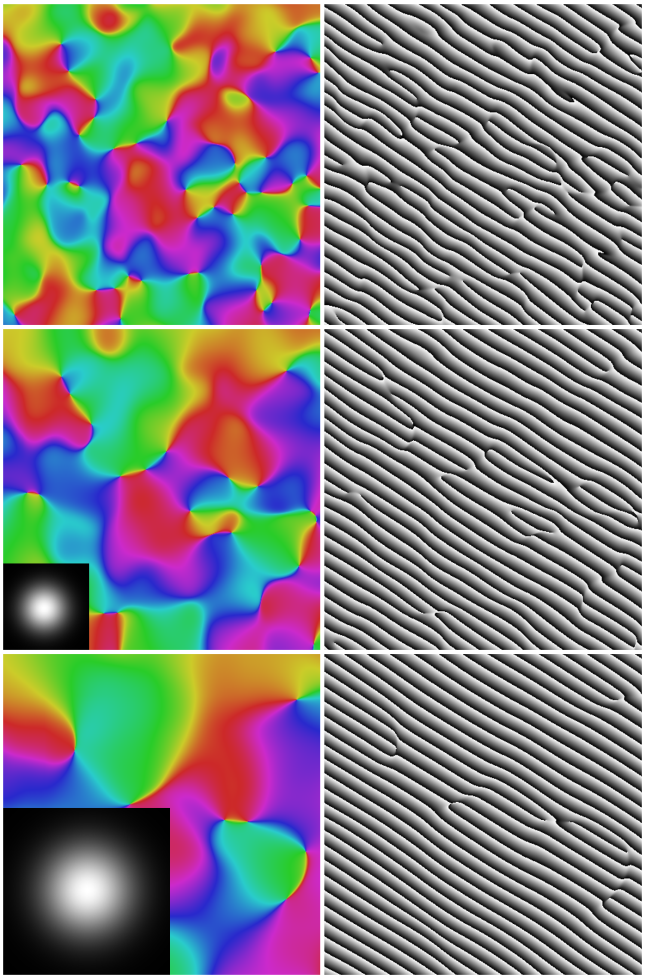


Fig. 10. Phasor noise filtering with increasing filter kernel sizes. **Left**: phasor field argument, encoded as hue. **Right**: phasor sawtooth. **Inset**: filtering kernel. **Top to bottom**: increasing kernel size. Note how the phasor field becomes smoother and remaining singularities become point singularities.

Then, substituting f and $\mathbf{d}$ for $F(\mathbf{c})$ and $D(\mathbf{c})$ — the approximated parameters around $\mathbf{c}$ — in $\omega_{\mathbf{c}}$, we obtain a local definition of the phasor field:

$$\varphi_{\mathbf{c}}(\boldsymbol{\delta}) = \frac{\mathcal{G}(\mathbf{c}+\boldsymbol{\delta})}{\omega_{\mathbf{c}}(\mathbf{c}+\boldsymbol{\delta})} \tag{8}$$

$\varphi_{\mathbf{c}}(\boldsymbol{\delta})$ provides a good approximation of the phase shift relative to an ideal oscillator at any offset $\boldsymbol{\delta}$ *around* $\mathbf{c}$, as long as $\boldsymbol{\delta}$ remains small compared to the spatial variations of the control fields.

This allows us to evaluate functions on the locally-defined phasor field in a neighborhood centered around $\mathbf{c}$. We use this property to define a measure for singularities and then express the evaluation filter.

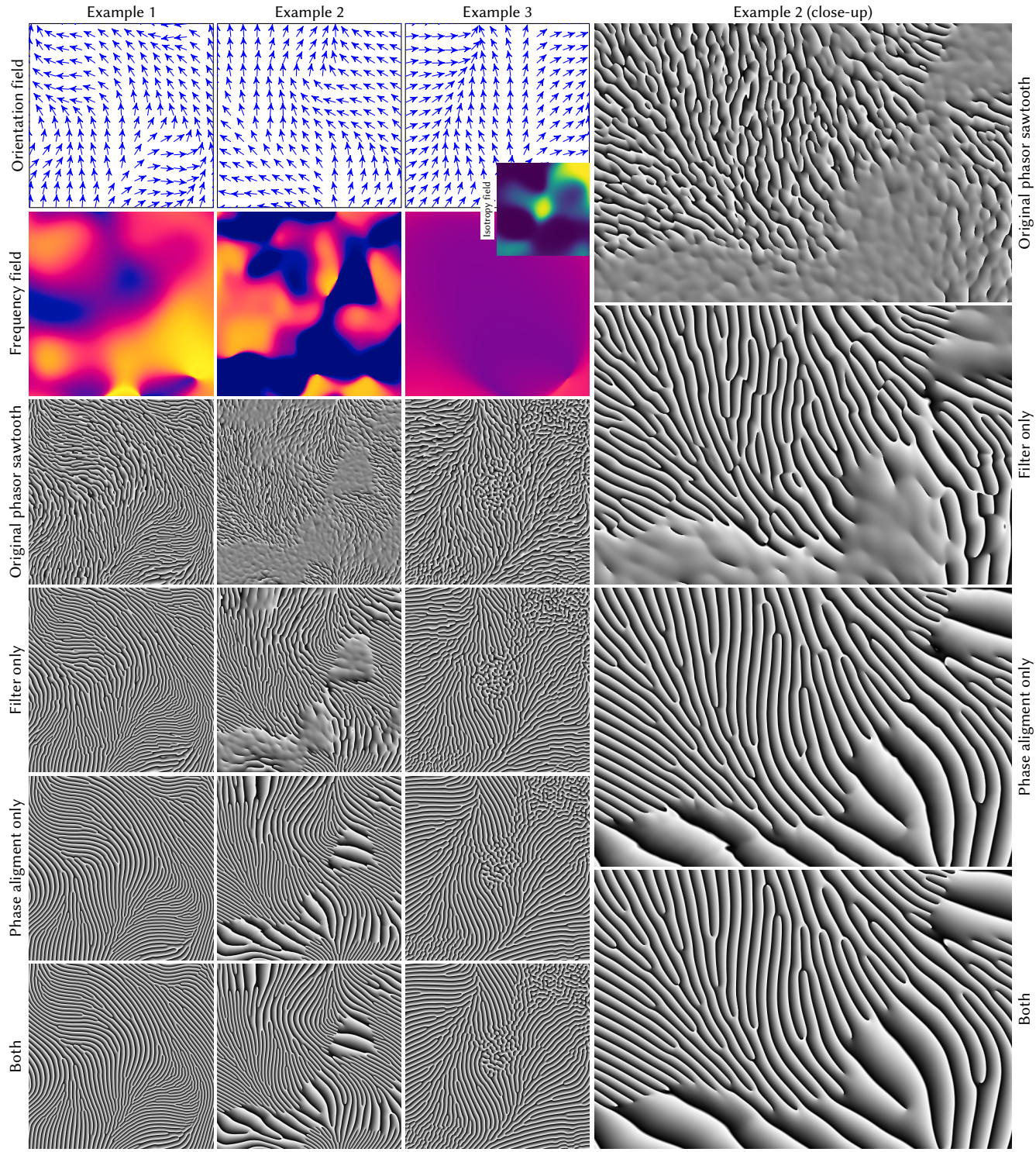


Fig. 11. Comparison of the contribution of the different parts of the method to the reduction of singularities. **Left**: from top to bottom: orientation field, frequency field, original phasor sawtooth wave, phasor sawtooth wave after iterated kernel phase alignment, phasor sawtooth wave after filtering, phasor sawtooth wave after iterated phase alignment and filtering. From left to right: various input fields and random seeds. **Right**: close-up of the effect of the iterated phase alignment, filtering and both methods on example 3.

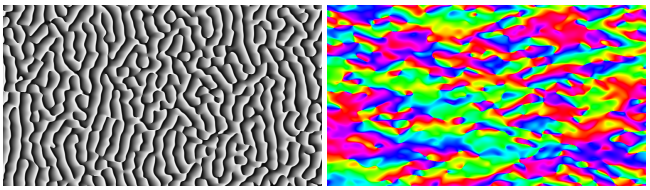


Fig. 12. Phasor field (right) obtained by division of the input noise (left) by the corresponding oscillator. Note how there are strong oscillations along a preferred direction in the field, which is unexpected from a phasor field such as in Figure 10.

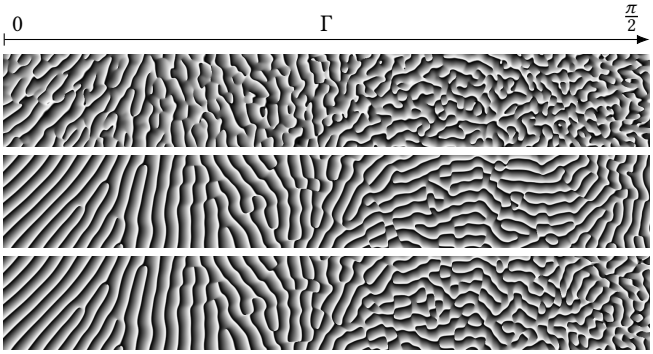


Fig. 13. Non-attenuated filter (middle) vs. attenuated filter (bottom). Without attenuation, the filter privileges the direction given by the input orientation field and discards other orientations. This erases the angular spread (right part), which is a superposition of kernels of spread-out directions. The attenuated filter does not exhibit this behavior, and thus better preserves the properties of the original noise (top).

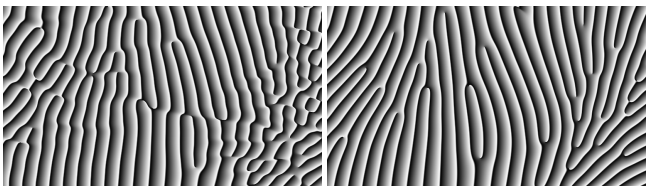


Fig. 14. Filtered phasor sawtooth (right) vs. phasor sawtooth with an equivalent noise bandwidth (left). Our filter formulation results in a pattern that matches the input fields more closely and is less prone to leading to undesired angular spread.

Fig. 15. Effect of removing a singularity vs. changing its shape. Given our metric $\mathcal{S}$, removing a singularity through phase alignment has a large impact on the cumulative count of pixels above the threshold (left, green). Filtering a singularity which cannot be removed changes its shape and reduces its area slightly, although with a much smaller impact on the cumulative count (right, green). This however greatly reduces the profile distortion that happens in the result (Figure 11).

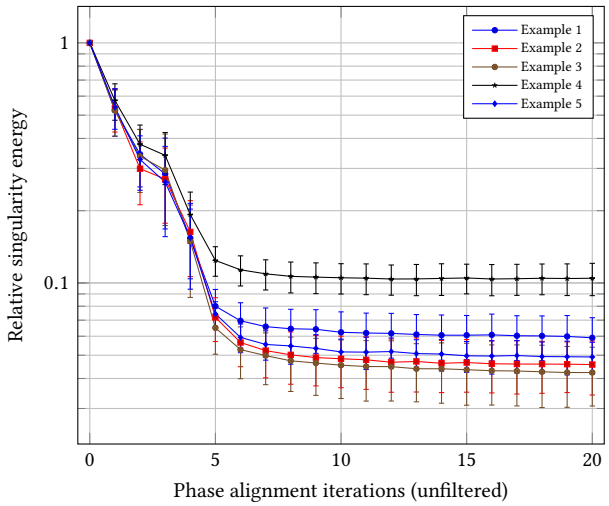


Fig. 16. Relative change in singularity energy as the iterative phase alignment converges. As the number of iterations increases, the singularity energy can be reduced by up to 95%. Each line uses a different set of parameters for generating the control fields, with each point being averaged over 50 different random seeds. This allows smoothing out variations due to specific instances in order to evaluate the average behavior of the phase alignment. The error bars show the standard deviation around the mean value among those seeds. The effect of the phase alignment is stable over varying parameters (examples have similar average behavior), and varying seeds (the standard deviation is small). Note the log scale.

4.2.2 Singularity energy. We define the *local singularity measure* $\mathcal{S}(\mathbf{x})$ at location $\mathbf{x}$ as:

$$\mathcal{S}(\mathbf{x}) = \frac{||\nabla \operatorname{Arg}(\varphi_{\mathbf{x}}(\boldsymbol{\delta}))||}{2\pi F(\mathbf{x})} \tag{9}$$

Note that the gradient operator is applied to the local phasor field around $\mathbf{x}$, differentiating with respect to the offset variable $\boldsymbol{\delta}$. The singularity measure is dimensionless and represents the relative phase change speed, as it is normalized by the local frequency $F(\mathbf{x})$, following Remark 2. In our implementation — which is used only for analysis — the gradient operator is computed using finite differences on the generated images of $\operatorname{Arg}(\varphi_{\mathbf{x}}(\boldsymbol{\delta}))$ (taking into account the 2π wrapping).

$\mathcal{S}(\mathbf{x})$ provides a strong spatial indicator of the two different families of singularities, points and curves, as shown in Figure 9.

In order to objectively measure the quantity of singularities (point or curves) in a result pattern, we introduce for an image domain Ω the *singularity energy*:

$$F_{\mathcal{S}}(q) = \frac{|\{\mathcal{S}(\mathbf{x}) > q, \mathbf{x} \in \Omega\}|}{|\Omega|} \tag{10}$$

The *singularity energy* effectively counts how much surface area of a given image is covered by singularities. The threshold q is to be determined visually depending on how much profile distortion should be considered a singularity. In all our examples we chose a demanding threshold of $q = 2$. In other words, all points whose phase shift is twice the frequency of the noise are considered part of a singularity.

Equipped with this measure, we can now analyze the effect of our regularization and filter.

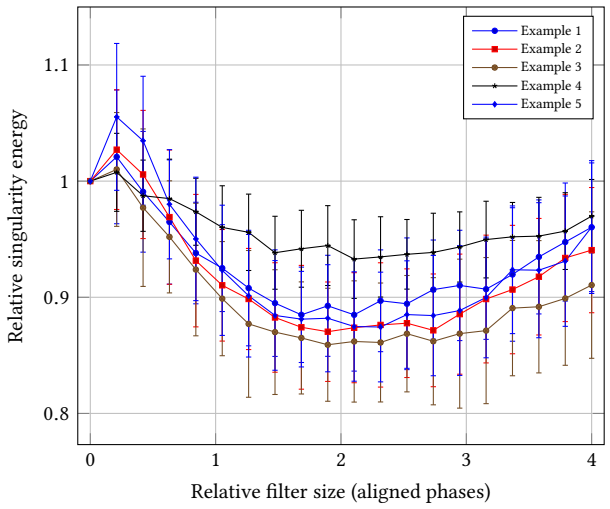


Fig. 17. Relative change of singularity energy as the filter size increases. The filter radius is defined relative to the noise kernel radius. All points are computed with 20 phase alignment iterations. As in Figure 16, results are averaged over 50 samples to reduce noise. The energy reduction reaches a minimum value at a relative filter size of 2 (the filter bandwidth is $1/\sqrt{2}$ the noise kernel bandwidth) on all examples. When the filter size increases beyond the tipping point, we tend toward a unique phase shift per direction of oscillation. Such a global choice of phase is not valid for a varying direction field, reducing the filter performance. The relative reduction in energy is weaker than what the regularization achieves (at best 15% on these examples) as illustrated by Figure 15, however this still improves the visual result significantly.

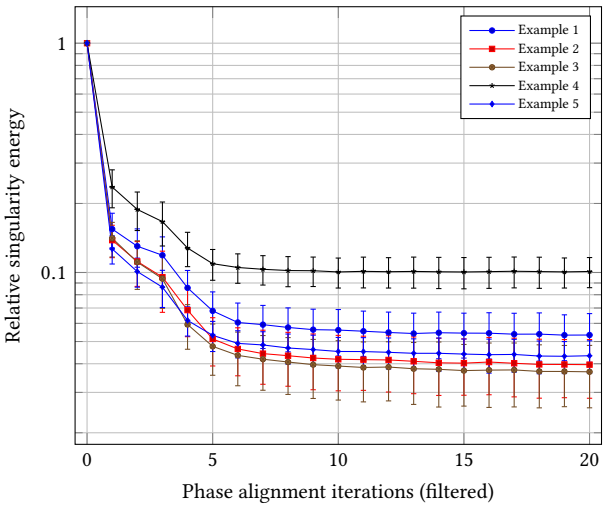


Fig. 18. Relative change of singularity energy as the phase alignment converges, while the evaluation filter bandwidth is set to its optimal value. The results are similar to Figure 16, but the singularity energy reduction converges to 96% of the initial value.

4.2.3 Filter. From Equation 6 we see that the initial complex Gabor noise can be reconstructed from the phasor field, multiplying it back by the oscillator. This opens the possibility to directly manipulate the phasor field, and in particular, apply a smoothing filter that will remove abrupt variations. Following Remark 1, doing so will reduce the extent of singularities in the phasor noise.

Given g a Gaussian kernel of bandwidth a:

$$g(\mathbf{x}) = e^{-a\|\mathbf{x}\|^2} \tag{11}$$

The filtered complex Gabor noise is obtained as:

$$\mathcal{F}(\mathbf{x}) = (g \circledast \varphi_{\mathbf{x}})(\mathbf{x}) \cdot \omega_{\mathbf{x}}(\mathbf{x}) \tag{12}$$

Note that the convolution is applied to the variable δ, around point $\mathbf{x}$, hence filtering around the local phasor field.

Since $\omega_{\mathbf{x}}(\mathbf{x}) = 1$ (see Equation (7)), the filter simply becomes:

$$\mathcal{F}(\mathbf{x}) = (g \circledast \varphi_{\mathbf{x}})(\mathbf{x}) \tag{13}$$

An example of filtering is given in Figure 10.

Making the filter practical. At this stage we have defined a filter that improves the result. Unfortunately it is also computationally expensive due to the convolution in Equation 13. Used as is, obtaining the convolution result would require at each evaluation point $\mathbf{x}$ a dense local sampling of both $\varphi_{\mathbf{x}}$ and the Gaussian filter. As we operate in 3D, this would add a very significant computational and/or memory overhead.

Fortunately, the convolution-based noise formulation has desirable properties with respect to Gaussian products that we can use to our advantage. As was done for texture filtering in the original Gabor noise [Lagae et al. 2009b] we can analytically integrate the filtering operation directly into the kernels.

At a given evaluation point, applying the filter to the kernels we obtain new filtered kernels $\mathcal{K}_j$:

$$\mathcal{K}_j(\mathbf{x}) = e^{-\frac{ab}{a+b}\|\mathbf{x}-\mathbf{x}_j\|^2 - \frac{\pi^2\|\Delta_{j\mathbf{x}}\|^2}{a+b}} e^{2\mathrm{i}\left(\pi f_j \mathbf{d}_j \cdot (\mathbf{x}-\mathbf{x}_j) + \frac{a}{a+b}\Delta_{j\mathbf{x}}\cdot(\mathbf{x}-\mathbf{x}_j)\right) + \mathrm{i}\varphi_i} \tag{14}$$

where

$$\Delta_{j\mathbf{x}} = f_j \mathbf{d}_j - f_{\mathbf{x}} \mathbf{d}_{\mathbf{x}} \tag{15}$$

$f_{\mathbf{x}}$ and $\mathbf{d}_{\mathbf{x}}$ are the frequency and orientation sampled at the evaluation point $\mathbf{x}$. Recall that a and b are respectively the bandwidth of the filter and the phasor noise. The full derivation of this equation is given in the supplemental material.

This leads to a formulation with only a small overhead. Indeed, using the kernels defined in Equation (14) in place of the original ones *directly gives the filtered result*. Thus, evaluating the filtered pattern remains a simple sum of kernels, as with the original phasor noise. The kernel expression is only slightly more complex, while we entirely eliminate the sampling that was required to implement the convolution in Equation 13. The only remaining overhead comes from the larger footprints of the filtered kernels. This requires fetching a larger neighborhood compared to the non-filtered version. However, this only adds a limited, constant cost to the evaluation.

The $\Delta_{j\mathbf{x}}$ term can be interpreted as a weight that weakens the contribution from kernels whose frequency and orientation are too different from the ones given by the input field. Interestingly, when this term goes towards zero the filter becomes equivalent to decreasing the bandwidth of the noise (larger kernels). The importance of

this term is revealed in Figure 14, where our filter is compared to simply reducing the noise bandwidth. In areas of quick orientation changes, decreasing the bandwidth leads to mixing orientations (the pattern locally contains multiple orientations, see bottom right part of the first image in Figure 14). The filter, on the contrary, preserves the input orientation everywhere.

4.2.4 Filtering and angular spread. To control the rigidity contrast, we allow the user to manipulate the angular spread Γ of the phasor noise. This locally perturbs the kernel orientations, resulting in a pattern locally mixing different orientations. It is important to keep these variations in the final result as they allow grading the rigidity contrast.

By construction, the filter we propose assumes only a single direction is present at any given point in the noise. This is not true of a phasor noise with angular spread, as illustrated in Figure 12. Thus if we would just extract the phasor field from such a noise and convolve it with a Gaussian filter, the extra oscillations introduced by the angular spread would be removed, see Figure 13, top.

To avoid this, we introduce an attenuation factor $A(\mathbf{x}) \in [0, 1]$ that permits a smooth transition between the original filter definition and a Gaussian filter with no preferred orientation. This is done by substituting $\Delta_{j\mathbf{x}}$ for $\Delta'_{j\mathbf{x}}$ in Equation (14):

$$\Delta'_{j\mathbf{x}} = A(\mathbf{x})\Delta_{j\mathbf{x}} \tag{16}$$

The factor A is expected to vary smoothly so as to not introduce discontinuities in the result. We derive it from the local angular spread of the phasor noise, which is a combination of Γ and the gradient of the orientation control field D.

4.3 Results of singularity reduction

In this section we have a closer look at the reduction of singularities achieved on various examples by each approach separately (iterated phase alignment and filtering) and then combined (oscillation regularization). We use the singularity energy (Section 4.2.2) to analyze the results.

4.3.1 Regularization convergence and filter size. There is a lower bound to the singularity energy for each example we consider, since there is no combination of phase shifts that could possibly align all kernels given a varying orientation field, in the general case. Figure 16 analyzes the reduction in singularity energy with more iterations of the phase alignment algorithm. We clearly see the plateau appear after a few iterations. Visual inspection of examples in Figure 11 reveals that a large number of singularities has been removed, and those that remain have shrunk toward points.

Even though the result is greatly improved by the phase alignment, there is still a noticeable amount of remaining distortion. By smoothing the phasor field using our filter, most of the remaining defects are removed and the area covered by singularities further shrinks.

Note that the energy reduction of the filter is smaller than what the regularization achieves, due to the choice of the threshold in the definition of F_S. This is explained in Figure 15.

Figure 17 analyzes the filter effect for varying bandwidths a. This reveals a U-shaped curve. Figure 16 and Figure 17 allow us to chose the number of regularization iterations and filtering bandwidth. We use 20 iterations in all our examples. Since the filter smoothes discontinuities in the phasor field, its bandwidth should be determined by the spatial dimensions of singularities. However the filter should not be too large relative to variations in the control fields, or it would break the assumptions made in order to recover the local phasor field $\varphi_{\mathbf{c}}$. Thus, we pick the filtering bandwidth to be $1/\sqrt{2}$ the unfiltered noise kernel bandwidth (i.e. the filter kernel radius is twice the radius of the noise kernel).

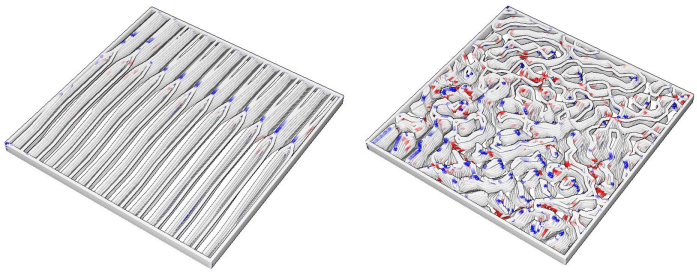


Fig. 19. Two slabs of tool paths visualized in our printer emulation tool. A red color indicates an overhang violation, a blue color an overlap. Sliced with 0.2 mmlayers for a 0.4 mmnozzle, cube filled at 25% density. **Left**: $\Gamma = 0$. **Right**: $\Gamma = \frac{\pi}{2}$.

4.3.2 Contribution of each approach. Figure 11 shows a variety of examples using the sawtooth profile, and for each, the effect of both approaches separately, and then combined. In all cases, we observe a significant reduction in the number of singularities by the regularization, as well as a much more regular oscillation pattern after the evaluation filter is applied. Applying only the filter results in a high number of curve discontinuities which cannot be solved without aligning phases, but applying only the iterated phase alignment results in remaining local distortions. Using both approaches gives the best results (see also Figure 18).

5 ANALYSIS

We perform our analysis for the case of fused-filament 3D printing. This is, by far, the most widespread and inexpensive additive manufacturing technique.

For analysis and measurements we consider cubes of 35^3 mm with different densities ρ and angular spreads Γ. The fields are always constant within a sample cube. All the cubes are 3D printed using the same flexible *NinjaFlex 85A* filament with the same machine. We print on a *Creality CR10* modified to replace the initial Bowden extruder by a direct extrusion mechanism (The E3D Titan Aero); this increases reliability by ensuring the gear pushing the (flexible) filament is as close as possible to the nozzle.

We measure fabrication statistics on the Gcode produced by our slicer in Section 5.1 and provide mechanical measurements in Section 5.2. We perform all compressive testing using the same uniaxial *Instron 3345* testing machine.

5.1 Fabricability

The stochastic nature of our synthesis process creates a geometry that is not perfectly regular. This calls for additional analysis of the

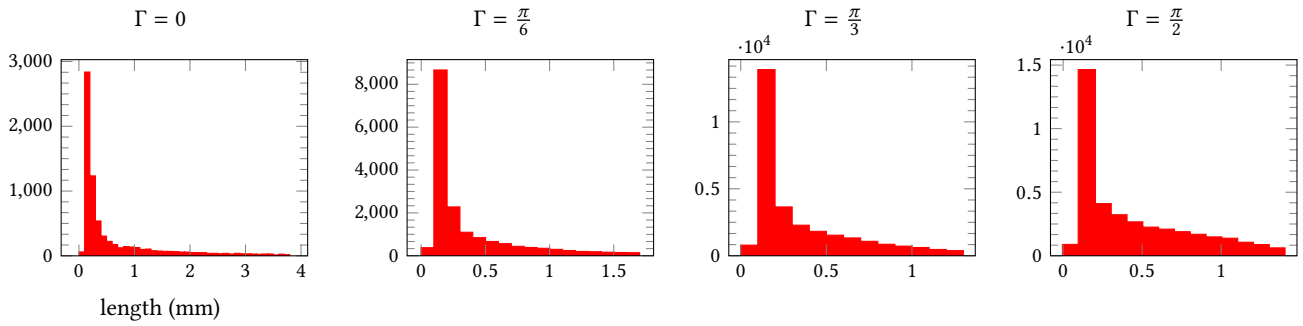


Fig. 20. Overhang statistics for 35 mmcubes sliced at 0.2 mmthickness printing with a 0.4 mmnozzle, filled at 25% density for varying value of Γ. The histograms count the number of segments of some length exceeding the overhang constraint (less than 40% of the extrusion disc supported from below). The graphs are clamped after the first 95% of segment counts for readability. These results were computed on a cube at 25% density.

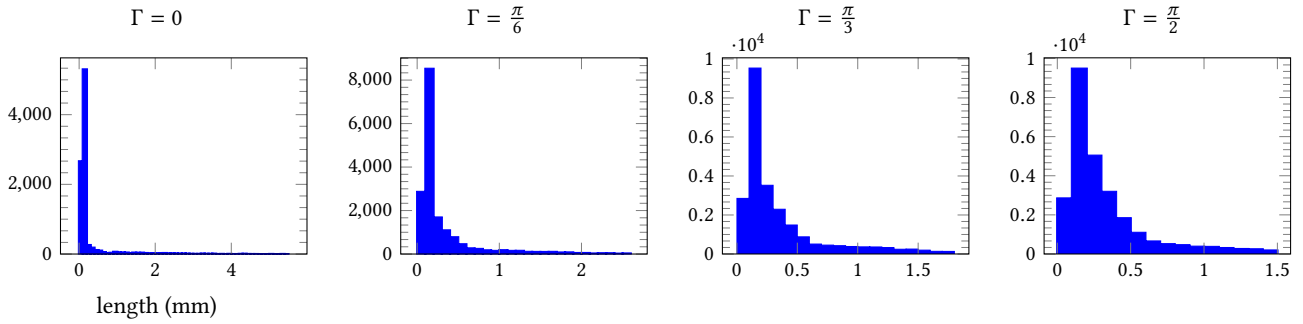


Fig. 21. Same as Figure 20 for overlap statistics. A segment is counted when the overlap along it exceeds 10% of the nozzle diameter. The graphs are clamped after the first 95% of segment counts for readability.

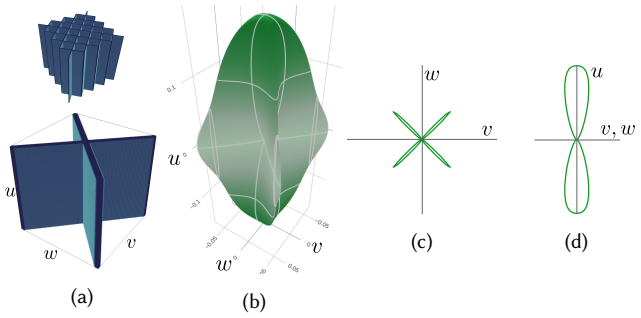


Fig. 22. Numerical simulation (small deformations, i.e. linear elasticity) of the periodic diamond structure. **(a)** Periodic domain used for the homogenization [?] (grid with resolution 100^3, solid phase with Poisson's ratio 0.3 and Young's modulus 1). The solid phase (in blue) has a density of 17%. The homogenized linear elasticity tensor [?] is computed with the software CrAFT [Boittin et al. 2014]. **(b)** Visualization of the directional Young's modulus given by ELATE [Gaillac et al. 2016]. The minimal Young's modulus is achieved in the axes v and w, while the maximal one is achieved in the axis u. **(c)** Young's modulus in the vw plane, the central point corresponds to the origin. There is a sharp increase in the Young's modulus for directions closely aligned with the diamond walls, as illustrated by the cross. **(d)** Young's modulus in the uv and uw plane.

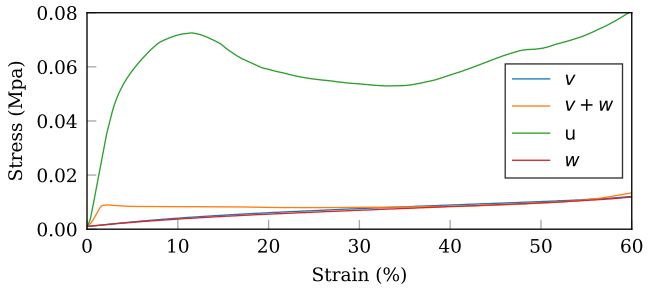


Fig. 23. Compression test with two specimens, having equal density $\rho = 10\%$ and angular spread $\Gamma = 0$. In this case, the printing direction is along the axis u. One of the specimens has been rotated 45° degrees along the u axis. Please refer to text for details.

printability of the synthesized microstructures. Therefore, we report here statistics obtained by emulating the fused filament fabrication process based on the generated trajectories (Gcode). Our emulation is purely geometric and performed by rasterizing the paths in high resolution images (0.04 mm per pixel). We track overlap between paths and violation of overhang constraints.

The statistics are reported in Figures 20 (overhangs) and 21 (overlap). Two slabs of tool paths are visualized in Figure 19, revealing that overhang violations and overlaps occur essentially along small

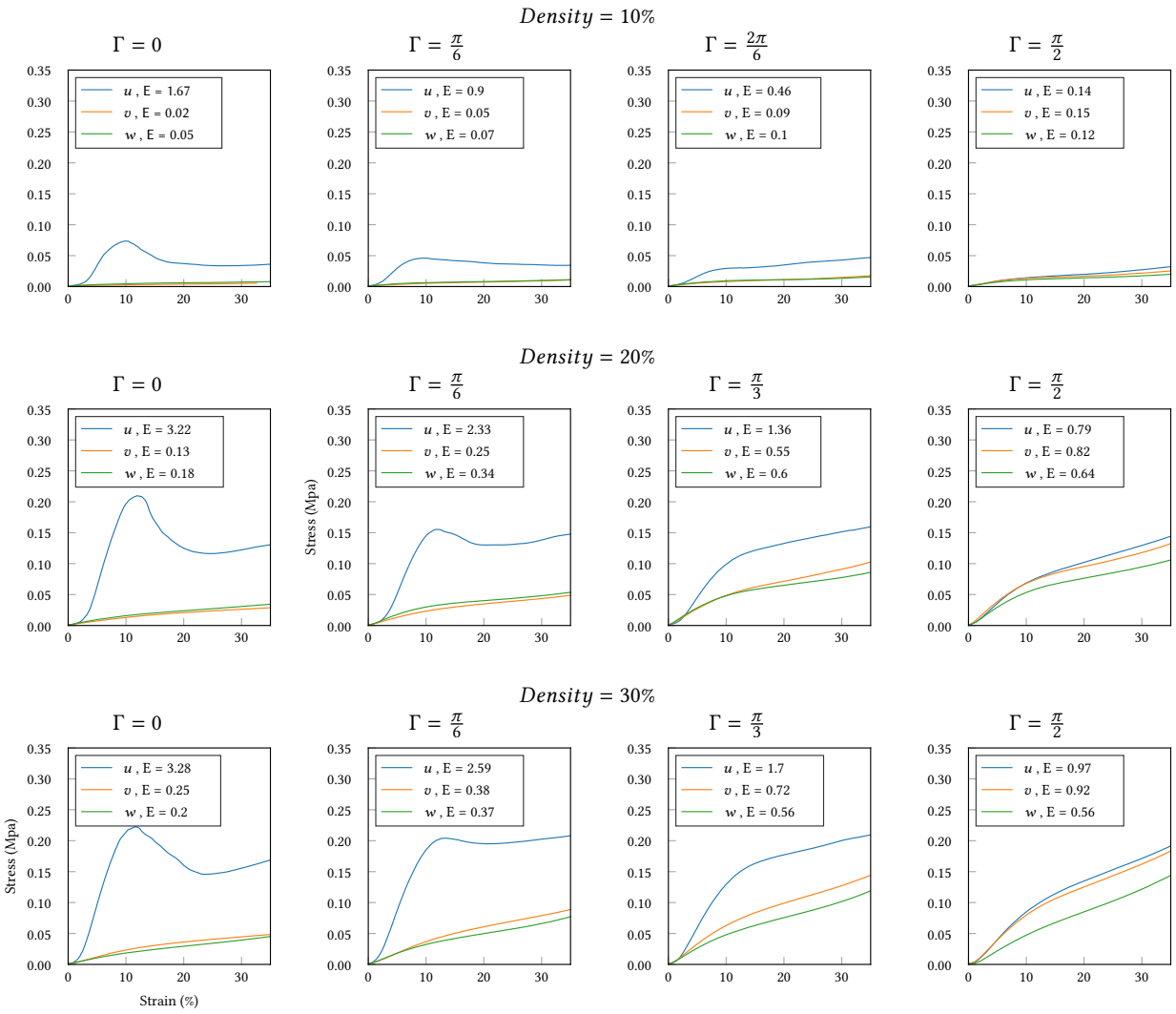


Fig. 24. Compression test with 12 different specimens (3D printed cubes) with varying density ρ (columns) and amount of angular spread Γ (rows), from 0 to isotropy ($\pi/2$). For each specimen we measure the stress-strain response [?] in the three orthogonal directions u, v, w denoted with different colors. The Young's modulus E is in Mpa. The printing direction is along the axis w. Please refer to text for discussion.

segments (comparable to the nozzle diameter). Overall, these defects do not jeopardize fabricability, especially as many overhang violations are actually bridging segments supported on both ends.

We did not observe any significant printing issues on the microstructures. However, the existence of overlaps indicates that they are not well suited for high densities.

5.2 Mechanical behavior

In this section we study the macroscopic mechanical behavior produced by our microstructures. They are meant to be used under large deformations, beyond the linear regime. Numerically simulating under such large deformations is extremely challenging. Notably due to thin walls, complex stochastic geometry, and the difficulty to model the effect of layered fabrication. For large deformations we thus resort to physical measurements on actual fabricated samples. This has the advantage to take into account the actual final product as opposed to an idealization of it. This is especially important as we print walls at the smallest possible thickness. The downside is that we cannot explore the material space in a numerical fashion.

Before moving to actual samples, we first provide a numerical analysis of the linear elastic behavior of the diamond structure in Figure 22. This confirms the transverse-rigid nature of the structure (u is significantly stiffer). However, it also shows that directions exactly aligned with the diamond walls are stiffer than any other direction in the vw plane. This is not ideal but is a trade-off we had to make to accommodate for AM constraints. Interestingly, this behavior is less problematic under large deformations: Figure 23 considers a specimen with maximal rigidity contrast ($\Gamma = 0$), testing

for compression along u, v, w as well as $u+w$ (strain in the direction of the diamond structure walls). For this latter measure we use a second cubic specimen with the diamond walls aligned with the specimen sides. We observe collapsing (explaining the plateau) in the measures along $u+w$, followed by a progressive approach to the v and w curves (almost identical for 30% strain). Thus, under large deformations the collapsing produces a response that is roughly equivalent to v and w, especially when compared with the much more rigid u axis.

We report additional stress-strain curves measured on printed samples in Figure 24, for varying frequencies and angular spreads. Note that regardless of the parameter Γ all structures of a given frequency have the same density (up to 2% variations). We selected the frequency to match densities of 10%, 20% and 30%. As expected, a lower density results in an overall less rigid structure. We observe three other major experimental trends in Figure 24, across the three different densities. First, as we increase the angular spread Γ the elastic response in u becomes fairly close to v and w. The w axis ends up being less rigid than u and v, which is mainly explained by the fact that the angular spread is solely occurring in the u, v plane. While we could imagine compensating for this by changing the frequency of $p_{\mathbf{w}}$ overhang constraints make this difficult. Nevertheless, the rigidity contrast is significantly reduced compared to the case of $\Gamma = 0$ verifying our original assumption. Second trend, for $\Gamma = 0$ and $\Gamma = \frac{\pi}{6}$ there is buckling in the u axis. However, this buckling is much stronger in the case of $\Gamma = 0$, indicating that a small degree of angular spread allows to reduce it. We do not consider buckling along u as a problem for our use cases: the amount of stress required to trigger it is very significantly above the stress required to fully compress the v or w axes. Third trend, for $\Gamma = 0$, there exists a small but noticeable difference between the response of v and w, while ideally both should have the same response. This is primarily explained by the anisotropy introduced by the layered printing along the w axis, used for this particular test.

Figure 25 shows the behavior of the basic case ($\Gamma = 0$, D constant) when w is fully compressed. As can be seen, the lateral expansion is very dissimilar in u and v: it is large along v and inexistent along u (the same occurs for w, u compressing along v, by symmetry). This behavior relates to Poisson's ratios, which, in linear elasticity is effectively zero for pairs u, v and u, w while being close to one for pairs v, w and w, v. This is particularly interesting for our application as compression along w (resp. v) triggers a large expansion along only v (resp. w). Through the orientation field D we control the v, w plane where these expansions occur (recalling the u is aligned with D).

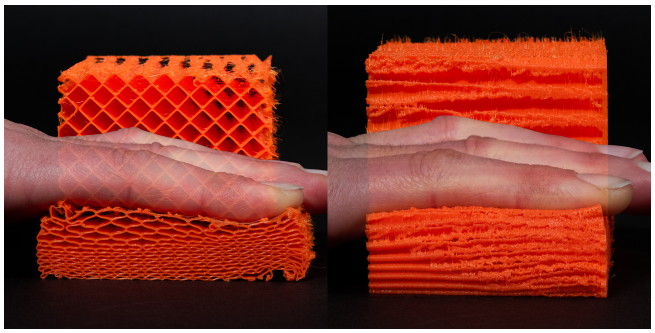


Fig. 25. Basic structure at rest and under full compression along w. **Left**: v, w face. **Right**: u, w face. Note how the structure does not expand along u, even when w is fully compressed.

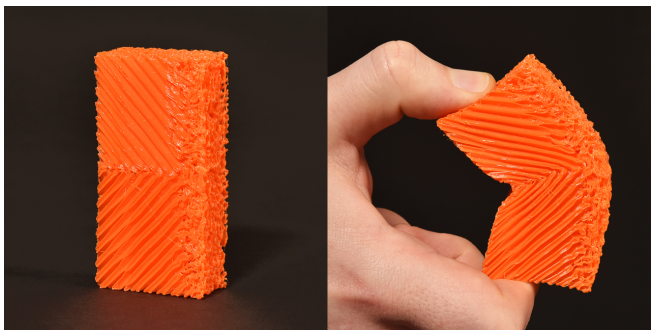


Fig. 26. Principle of a knee-like flexure. A flexible front plate provides the overall strength. The V-shaped transverse-rigid structures behind resist lateral bending, while freely collapsing at the rear to let the front flex in the unique allowed direction (at comparable stress levels).

Fig. 27. A structured plate. The orientations produce a strong collapsing effect when pressing on a specific location.

6 PRINTED EXAMPLES

In this section we design and fabricate several examples with our method, demonstrating the variety of deformation behaviors that can be achieved by manipulating the control fields (orientation, rigidity contrast, density). As can be seen throughout our various examples, non trivial shapes embedding our microstructures can be reliably printed.

The fields are stored and manipulated as low resolution 3D textures (64^3, 8-bit voxels). A scripting language helps defining an initial value within the fields, and the designer is then free to directly paint them through a basic brush interface.

Our first example is a simple cube printed with a constant direction, density and no angular spread ($\Gamma = 0$ everywhere). It is shown in Figure 25. This example gives an insight on what the structure looks like in simple cases, and also demonstrate the large, directional deformations that can be obtained.

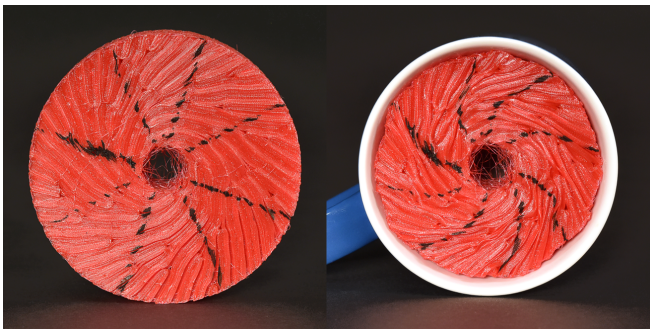


Fig. 28. A wheel design with a spiraling direction field. **Left**: The structure at rest, with black lines drawn on it to better outline the deformation. **Right**: The wheel under radial compression. Note the spiraling effect occurring due to the deformation being concentrated orthogonally to the spiraling transverse rigidity.

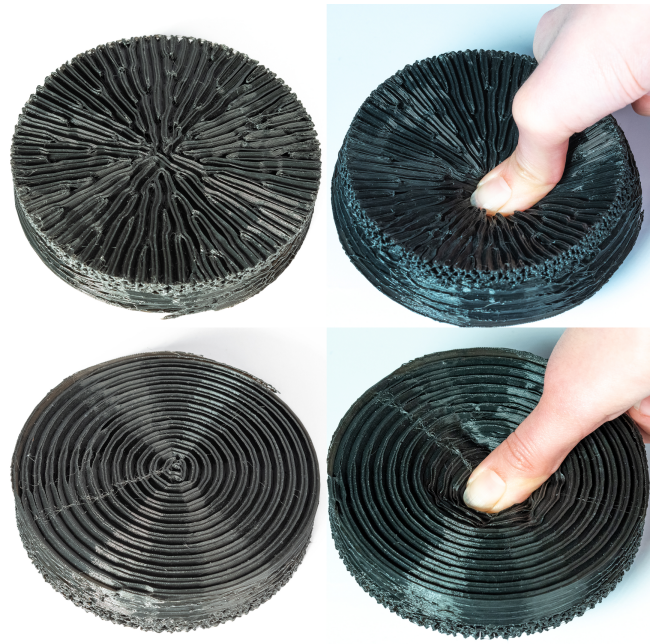


Fig. 29. Disc printed with a circular direction field, when stress is allied on the center, it bends to form a cone.

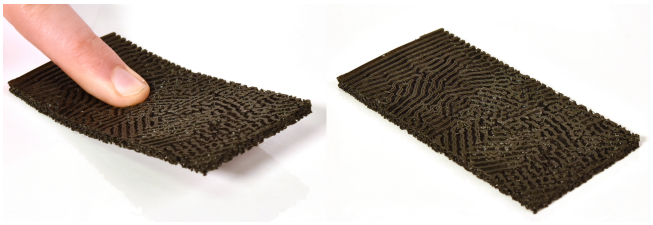


Fig. 30. Plate showing a gradation of the angular spread. If pressed on specific point, it bends out of plane

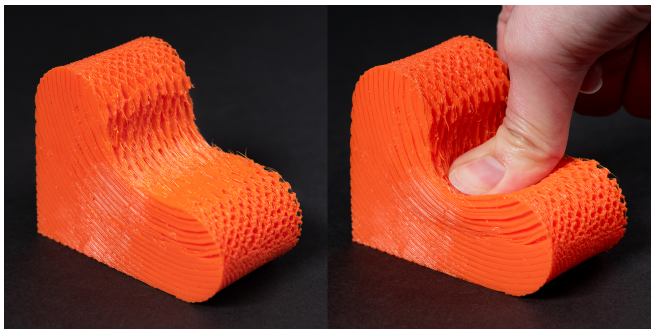


Fig. 31. Seat printed with our method, lower Frequency are used on the top of the structure to make it softer

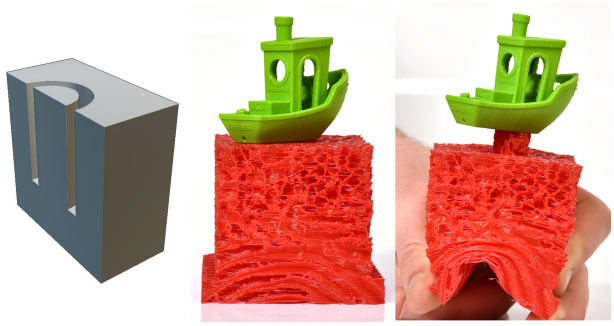


Fig. 32. Example of soft mechanism that can be create with our method, here a lateral pressure of the side of the object create a deformation that make a cylinder move up and lift a small object

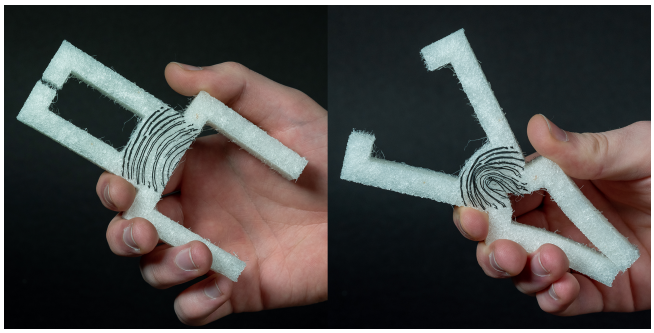


Fig. 33. Pliers printed with our method. The direction field induces the movement, the frequency field rigidifies the hand grips and the jaws to only allow the structure to bend in the joint part.

Our second example, shown in Figure 1, is another cube filled with a more complex structure. The direction field rotates such that the transverse rigidity remains parallel to an axis, located in the middle of the *top* face. Some angular spread is added to the opposite (bottom) face, making it more rigid in all directions. Pressing on the top face collapses the cube while wrapping the top around the pressing object, an unexpected gripping behavior for the unsuspecting observer. Pressing on the bottom face produces nothing special, the

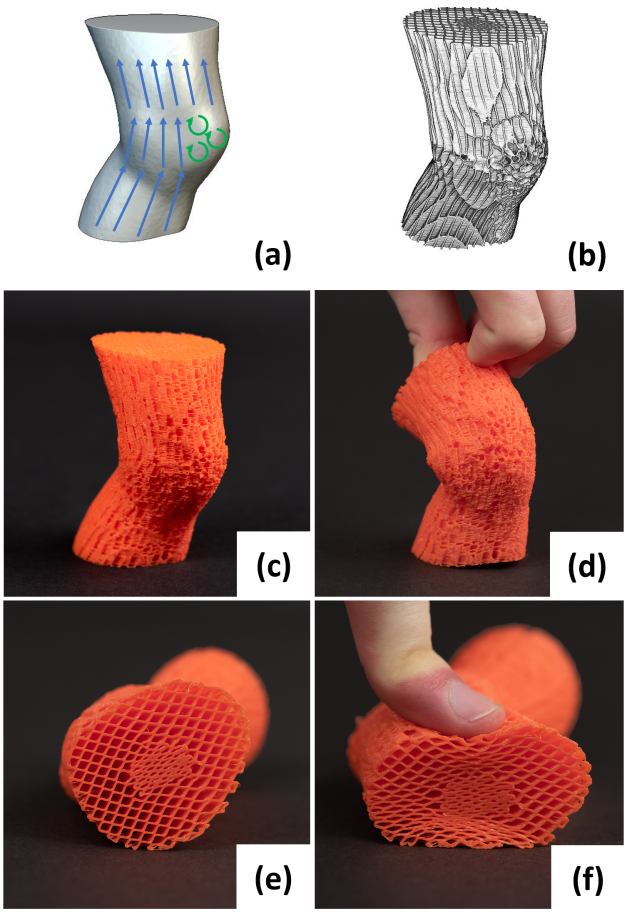


Fig. 34. Knee 3D printed using our diamond structures. **(a)** Show the input model with in blue the direction field D, and in green the angular spread field, (b) how the generated toolpaths. **(c) and (e)** show the printed model at rest **(d)**: Knee bended. **(f)**: A strength is applied on the section to show the impact of a denser structure, the less dense structures are folded, while the denser parts remain the same

cube reacts as if made of an homogeneous material. Such gripping behaviors are promising for soft robotics.

Figure 29 shows a thick circular plate, where one face has rigidity oriented radially and the other in a tangential manner. This reveals how the direction field alone strongly influences the emerging mechanical behavior. In Figure 29 (bottom) we see that pressing the center has no global effect, since the structures flex and absorb the distortion. In Figure 29 (top) the radial arrangement of rigidity makes the entire plate bend under pressure. This is easily explained by the orientation of the structures that forbid distances to change within the volume along specific directions. Note that this very different behavior occurs on both sides of the same object.

Figure 28 is another circular plate where a tangential spiral displacement occurs under radial compression, thanks to the specific arrangement produced by the direction field. This example could also act as a shock-absorbing wheel for a robot or rover.

Figure 32 is a toy mechanism inspired from Figure 1 right. The fields are overall the same, but an internal geometry allows a column to raise from the opposite face when pressing on the sides. This can serve as an actuator or a lateral pressure sensor (measuring the displacement of the column).

Figure 31 is an example design of a simple chair. We oriented the structure to follow the seat surface and be less dense near it.

Figure 33 is a pliers synthesized with our method. The jaws and the hand grips are denser whereas the joint part is softer and re-use the same kind of radial field presented in Figure 1 to amplify the deformation.

Finally, we conclude with an illustration of a knee-like 'joint-less' articulation, shown in Figure 34 and which principle is more clearly isolated in Figure 26. A flexible front plate is created using a low rigidity contrast. Behind, two main directions are used in way that 1. allows the volume to collapse where we want the fold to occur and 2. resists lateral bending thanks to converging rigidity direction. We also added a region of low contrast to mimic the bone in the upper and lower leg. This inner core is easily felt when pressing on the sides, and gives vertical rigidity to the overall part.

To the best of our knowledge, no other available approach can produce results with such deformation behaviors under large displacements. Free orientation, density, and rigidity contrast gradation combine to provide a very effective control over the volume reshaping. Our examples are only scratching the surface of potential applications in soft robotics, soft mechanisms, prosthetics and orthoses design, as well as shock or vibrations absorbing pads.

7 LIMITATION

Even though they print reliably, the stochastic nature of the generated structures makes it difficult to provide a strict guarantee regarding fabrication constraints. The existence of overlaps prevents reaching high densities: the overlaps would accumulate and overfill, resulting in printing issues.

In addition, if an overhang constraint stricter than 45 degrees is required, square diamonds may not enforce self-supportability anymore. In this case, vertically stretched diamonds could be used at the cost of a rigidity anisotropy in the direction orthogonal to the transverse axis.

The mechanical response is not perfect, with rigidity spikes along specific directions in the v, w plane ($\mathbf{v} + \mathbf{w}$ and $\mathbf{v} - \mathbf{w}$). On less demanding fabrication technologies, using a circular profile would avoid this. Also, the profile could be changed or interpolated based on the field D: as the structures align with the vertical axis much more freedom is available to design the 2D profile of extrusion.

Finally, our structures grade from rigid-transverse to a roughly isotropic response. They do not intrinsically cover a case akin to a laminate where two directions would be rigid and one very flexible in comparison. Such a case can be obtained by sandwiching layers of our structures, but this requires a larger volume to reach an average behavior.

8 CONCLUSION

By significantly modifying the way internal distances are allowed to change, our method enables unprecedented control over the

deforming volume of a 3D printed object. We made careful trade-offs to obtain a balance between reliability of fabrication — including on the most affordable filament printers — and quality of mechanical properties. The mechanical tests confirm that the main mechanical properties are achieved, especially under large deformations. The ability to print walls at the thinnest thickness with a single material allows to produce very flexible objects using off-the-shelf materials.

Our technique is defined from a stochastic approach. This affords for free gradation of the rigid axis orientation, as well as spatial gradation in density and rigidity contrast. Thanks to it favorable computational properties, our technique allows rapid, efficient extraction of toolpaths and is highly scalable. As future work, we will explore how to provide direct visualization of the structures during modeling, e.g. by ray-casting.

It can be difficult to understand and predict the effects that the input fields will have on the large deformations. This opens interesting venues for future work. A first venue of research would be the development of a method to simulate interactively the effect of the generated structures, even if approximately. This would provide early feedback, and help in designing increasingly complex large deformation responses. Finally, it would be interesting to couple our method with inverse mechanical design problems, that would automatically solve for the control field given a target deformation.

REFERENCES

Niels Aage, Erik Andreassen, Boyan S Lazarov, and Ole Sigmund. 2017. Giga-voxel computational morphogenesis for structural design. *Nature* 550, 7674 (2017), 84.

Erik Andreassen and Casper Schousboe Andreasen. 2014. How to Determine Composite Material Properties Using Numerical Homogenization. *Computational Materials Science* 83 (2014), 488–495. https://doi.org/10.1016/j.commatsci.2013.09.006

M.F Ashby. 2006. The properties of foams and lattices. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 364, 1838 (2006), 15–30.

Guylaine Boittin, Daniela Garajeu, Alice Labé, Hervé Moulinec, Fabrice Silva, and Pierre Suquet. 2014. CraFT. http://craft.lma.cnrs-mrs.fr/

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013. Quad-Mesh Generation and Processing: A Survey. *Comput. Graph. Forum* 32, 6 (Sept. 2013), 51–76. https://doi.org/10.1111/cgf.12014

James Brennan-Craddock. 2011. *The Investigation of a Method to Generate Conformal Lattice Structures for Additive Manufacturing.* Ph.D. Dissertation. Loughborough Univ. https://dspace.lboro.ac.uk/2134/9146

Yong Chen. 2007. 3d texture mapping for rapid manufacturing. *Computer-Aided Design and Applications* 4, 6 (2007), 761–771.

Zuenko Evgeny and Matthias Harders. 2019. Wrinkles, Folds, Creases, Buckles: Small-Scale Surface Deformations as Periodic Functions on 3D Meshes. *IEEE Transactions on Visualization and Computer Graphics* PP (05 2019), 1–1. https://doi.org/10.1109/TVCG.2019.2914676

Marc L.M. François, Letian Chen, and Michel Coret. 2017. Elasticity and symmetry of triangular lattice materials. *International Journal of Solids and Structures* 129, Supplement C (2017), 18–27.

Oleg Fryazinov, Turlif Vilbrandt, and Alexander Pasko. 2013. Multi-Scale Space-Variant FRep Cellular Structures. *Computer-Aided Design* 45, 1 (2013), 26–34. https://doi.org/10.1016/j.cad.2011.09.007

Romain Gaillac, Pluton Pullumbi, and François-Xavier Coudert. 2016. ELATE: an open-source online application for analysis and visualization of elastic tensors. *Journal of Physics: Condensed Matter* 28, 27 (2016), 275201.

Xifeng Gao, Wenzel Jakob, Marco Tarini, and Daniele Panozzo. 2017. Robust Hex-Dominant Mesh Generation Using Field-Guided Polyhedral Agglomeration. *ACM Transactions on Graphics* 36, 4, Article Article 114 (July 2017), 13 pages.

Perle Geoffroy-Donders, Grégoire Allaire, and Olivier Pantz. 2020. 3-d topology optimization of modulated and oriented periodic microstructures by the homogenization method. *J. Comput. Phys.* 401 (2020), 108994. https://doi.org/10.1016/j.jcp.2019.108994

Lorna J Gibson and Michael F Ashby. 1999. *Cellular solids: structure and properties.* Cambridge university press.

Alexander Goldberg, Matthias Zwicker, and Frédo Durand. 2008. Anisotropic Noise. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–8. https://doi.org/10.1145/1360612.1360653

Jeroen Groen, Florian Stutz, Niels Aage, J. Andreas Bærentzen, and Ole Sigmund. 2019. De-homogenization of optimal multi-scale 3D topologies. arXiv:cs.CE/1910.13002

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

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015a. Instant Field-Aligned Meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA)* 34, 6 (Nov. 2015). https://doi.org/10.1145/2816795.2818078

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015b. Instant Field-Aligned Meshes. *ACM Trans. Graph.* 34, 6, Article Article 189 (Oct. 2015), 15 pages. https://doi.org/10.1145/2816795.2818078

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2015. Stripe Patterns on Surfaces. *ACM Trans. Graph.* 34, 4 (2015).

Tim Kuipers, Jun Wu, and Charlie C.L. Wang. 2019. CrossFill: Foam Structures with Graded Density for Continuous Material Extrusion. *Computer-Aided Design* 114 (2019), 37 – 50. https://doi.org/10.1016/j.cad.2019.05.003

Ares Lagae, Sylvain Lefebvre, Rob Cook, Tony Derose, George Drettakis, David S. Ebert, J.P. Lewis, Ken Perlin, and Matthias Zwicker. 2010a. A Survey of Procedural Noise Functions. *Computer Graphics Forum* 29, 8 (2010), 2579–2600. https://doi.org/10.1111/j.1467-8659.2010.01827.x

Ares Lagae, Sylvain Lefebvre, Rob Cook, Tony DeRose, George Drettakis, D. S. Ebert, J. P. Lewis, Ken Perlin, and Matthias Zwicker. 2010b. A Survey of Procedural Noise Functions. *Computer Graphics Forum* 29, 8 (2010).

Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. 2009a. Procedural Noise Using Sparse Gabor Convolution. *ACM Trans. Graph.* 28, 3 (2009), 54:1–54:10.

Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. 2009b. Procedural Noise using Sparse Gabor Convolution. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2009)* 28, 3 (July 2009), 54–64. https://doi.org/10.1145/1531326.1531360

J. P. Lewis. 1989. Algorithms for Solid Noise Synthesis. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '89).* ACM, New York, NY, USA, 263–270. https://doi.org/10.1145/74333.74360

Dawei Li, Ning Dai, Xiaotong Jiang, and Xiaosheng Chen. 2015. Interior Structural Optimization Based on the Density-Variable Shape Modeling of 3D Printed Objects. *The International Journal of Advanced Manufacturing Technology* 83, 9 (2015), 1627–1635. https://doi.org/10.1007/s00170-015-7704-z

Nils Lichtenberg, Noeska Smit, Christian Hansen, and Kai Lawonn. 2018. Real-time field aligned stripe patterns. *Computers & Graphics* 74 (Aug. 2018), 137–149. https://doi.org/10.1016/j.cag.2018.04.008

Marco Livesu, Stefano Ellero, Jonàs Martínez, Sylvain Lefebvre, and Marco Attene. 2017. From 3D Models to 3D Prints: An Overview of the Processing Pipeline. *Computer Graphics Forum* 36 (2017), 537–564.

S.K. Maiti, M.F. Ashby, and L.J. Gibson. 1984. Fracture toughness of brittle cellular solids. *Scripta Metallurgica* 18, 3 (1984), 213 – 217. https://doi.org/10.1016/0036-9748(84)90510-6

Jonàs Martínez, Jérémie Dumas, and Sylvain Lefebvre. 2016. Procedural Voronoi Foams for Additive Manufacturing. *ACM Trans. Graph.* 35, 4 (2016), 44:1–44:12. https://doi.org/10.1145/2897824.2925922

Jonàs Martínez, Samuel Hornus, Haichuan Song, and Sylvain Lefebvre. 2018. Polyhedral Voronoi diagrams for additive manufacturing. *ACM Transactions on Graphics* 37, 4 (Aug. 2018), 15. https://doi.org/10.1145/3197517.3201343

Jonàs Martínez, Haichuan Song, Jérémie Dumas, and Sylvain Lefebvre. 2017. Orthotropic k-nearest foams for additive manufacturing. *ACM Transactions on Graphics* 36, 4 (July 2017), 121:1–121:12. https://doi.org/10.1145/3072959.3073638

Asla Medeiros e Sá, Vinícius Moreira Mello, Karina Rodriguez Echavarria, and Derek Covill. 2015. Adaptive voids. *The Visual Computer* 31, 6 (2015), 799–808. https://doi.org/10.1007/s00371-015-1109-8

Fabrice Neyret and Eric Heitz. 2016. *Understanding and controlling contrast oscillations in stochastic texture algorithms using Spectrum of Variance.* Research Report. LJK / Grenoble University - INRIA. 8 pages. https://hal.inria.fr/hal-01349134

nTopology. 2019. nTop design platform. https://ntopology.com.

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case Stress Relief for Microstructures. *ACM Transactions on Graphics* 36, 4 (2017), 122:1–122:16. https://doi.org/10.1145/3072959.3073649

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Trans. Graph.* 34, 4 (2015), 135:1–135:12. https://doi.org/10.1145/2766937

Olivier Pantz and Karim Trabelsi. 2008. A Post-Treatment of the Homogenization Method for Shape Optimization. *SIAM Journal on Control and Optimization* 47, 3 (2008), 1380–1398. https://doi.org/10.1137/070688900

Alexander Pasko, Oleg Fryazinov, Turlif Vilbrandt, Pierre-Alain Fayolle, and Valery Adzhiev. 2011. Procedural function-based modelling of volumetric microstructures. *Graphical Models* 73, 5 (2011), 165–181.

Minh-Son Pham, Chen Liu, Iain Todd, and Jedsada Lertthanasarn. 2019. Damage-tolerant architected materials inspired by crystal microstructure. *Nature* 565, 7739 (2019), 305.

Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. 2006. Periodic Global Parameterization. *ACM Transactions on Graphics* (2006). https://hal.inria.fr/inria-00104853

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to Control Elasticity in 3D Printing. *ACM Trans. Graph.* 34, 4 (2015), 136:1–136:13. https://doi.org/10.1145/2766926

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

M.G. Tarantino, O. Zerhouni, and K. Danas. 2019. Random 3D-printed isotropic composites with high volume fraction of pore-like polydisperse inclusions and near-optimal elastic stiffness. *Acta Materialia* 175 (2019), 331 – 340. https://doi.org/10.1016/j.actamat.2019.06.020

Vincent Tavernier, Fabrice Neyret, Romain Vergne, and Joëlle Thollot. 2019. Making Gabor Noise Fast and Normalized. In *Eurographics 2019 - Short Papers*, Paolo Cignoni and Eder Miguel (Eds.). The Eurographics Association. https://doi.org/10.2312/egs.20191009

Thibault Tricard, Semyon Efremov, Cédric Zanni, Fabrice Neyret, Jonàs Martínez, and Sylvain Lefebvre. 2019. Procedural Phasor Noise. *ACM Transactions on Graphics* (2019), 1–13. https://doi.org/10.1145/3306346.3322990

Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik. 2013. OpenFab: A Programmable Pipeline for Multi-material Fabrication. *ACM Trans. Graph.* 32, 4 (2013), 136:1–136:12.

Hongqing Wang, Yong Chen, and David W Rosen. 2005. A hybrid geometric modeling method for large scale conformal cellular structures. In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 421–427.

Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. 2016a. Infill Optimization for Additive Manufacturing–Approaching Bone-Like Porous Structures. *arXiv* (2016). http://arxiv.org/abs/1608.04366

Jun Wu, Charlie C.L. Wang, Xiaoting Zhang, and Rüdiger Westermann. 2016b. Self-supporting rhombic infill structures for additive manufacturing. *Computer-Aided Design* 80 (2016), 32–42. https://doi.org/10.1016/j.cad.2016.07.006

Jun Wu, Weiming Wang, and Xifeng Gao. 2019. Design and Optimization of Conforming Lattice Structures. *IEEE Transactions on Visualization and Computer Graphics* (2019). https://doi.org/10.1109/TVCG.2019.2938946 to appear.

Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Trans. Graph.* 36, 5, Article 164 (July 2017), 16 pages. https://doi.org/10.1145/3095815