

Two-Scale Topology Optimization with Microstructures: Supplemental Material

BO ZHU, MÉLINA SKOURAS, DESAI CHEN, WOJCIECH MATUSIK, MIT CSAIL

1 DISCRETE SAMPLING OF MICROSTRUCTURES

Given a set of microstructures, a new population of microstructures is generated using the Stochastically-Ordered Sequential Monte Carlo (SOSMC) method introduced by Ritchie et al. (2015).

In our implementation, the samples (or particles) are our microstructures, i.e. binary assignments of the base materials, and the desired distribution is the one that maximizes the number of particles located near or outside the boundary of the gamut of material properties. We evaluate the contribution of each sample towards the desired goal thanks to the scoring function

$$s(\mathbf{p}_i) = \frac{\Phi(\mathbf{p}_i)}{D(\mathbf{p}_i)} \times \frac{1}{D(\mathbf{p}_i)}, \quad (1)$$

where $\Phi(\mathbf{p}_i)$ is the signed distance of the material properties of particle i to the gamut boundary and $D(\mathbf{p}_i)$ is the local sampling density at the location \mathbf{p}_i . The sample density is defined as

$$D(\mathbf{p}_i) = \sum_{\mathbf{k}} \phi_{\mathbf{k}}(\mathbf{p}_i), \quad (2)$$

where $\phi_{\mathbf{k}}(\mathbf{p}) = \left(1 - \frac{\|\mathbf{p} - \mathbf{p}_{\mathbf{k}}\|_2^2}{h^2}\right)^4$ are locally-supported kernel functions that vanish beyond their support radius h , set to a tenth of the size of the lattice used for the continuous representation of the material gamut.

As described by Algorithm 1, given an initial microstructure, we generate a new microstructure by randomly swapping materials in the material assignments. However, as explained in Ritchie’s paper, executing the procedure sequentially – in our case visiting the voxels in a fixed order – is often suboptimal since the best order is often unknown a priori. To introduce randomness in the program execution, and because of the simplicity of our procedure primitives, i.e. swapping voxel materials, we do not rely on Stochastic Future as in Ritchie’s implementation, but directly modify the original program into the one described by Algorithm 2.

Starting with the microstructures corresponding to the entire gamut, we initialize the population of microstructures to evolve by sampling N microstructures using systematic resampling (Douc 2005) based on their scores as computed by Equation 1. We then run, for each microstructure, the program described by Algorithm 2 in

Algorithm 1 Initial procedure for generating a new microstructure

```

procedure GENMICROSTRUCTURE(input: microstructure  $M_i$ , output: microstructure  $M_o$ )
   $M_o \leftarrow M_i$ 
  for all voxels do
    swap material of the current voxel  $v$  with probability 0.5
  end for
end procedure

```

Algorithm 2 Procedure for generating a new microstructure

```

procedure GENMICROSTRUCTURE(input: microstructure  $M_i$ , output: microstructure  $M_o$ )
   $M_o \leftarrow M_i$ 
  while some voxels of  $M_o$  have not been visited do
    while microstructure  $M_o$  is unchanged do
      pick a random voxel  $v$  of  $M_o$  that has not been visited
      assign a randomly chosen material to  $v$ 
      if  $M_o$  is manifold and  $M_o \neq M_i$  then
        accept the change
      end if
    end while
    // Synchronization point
  end while
end procedure

```

Algorithm 3 SOSMC for discrete sampling of microstructures

```

procedure SOSMC(input: set of  $n$  microstructures  $m$ , output: set of microstructures  $p_o$ )
   $p_o \leftarrow m$ 
  for  $i=1..n$  do
    // Evaluate scores of all the particles  $m_i \in m$ 
     $w(i) \leftarrow s(m_i)$ 
  end for
  // Sample  $N$  particles
   $p \leftarrow \text{universal\_sampling}(m, N, w)$ 
  for  $i=1..N$  do
    start program genMicrostructure( $p_i, q_i$ ) for the input microstructure  $p_i \in p$ 
  end for
  while some particles  $q_i \in q$  have unvisited voxels do
    for all unterminated programs do
      run the program until the synchronization point is reached
     $p_o \leftarrow p_o \cup q$ 
    for  $i=1..N$  do
      // Evaluate scores for modified microstructures
       $w(i) \leftarrow s(q_i)$ 
    end for
    // Sample particles
     $q \leftarrow \text{universal\_sampling}(q, N, w)$ 
  end for
end while
end procedure

```

order to evolve the population. The program is not executed entirely

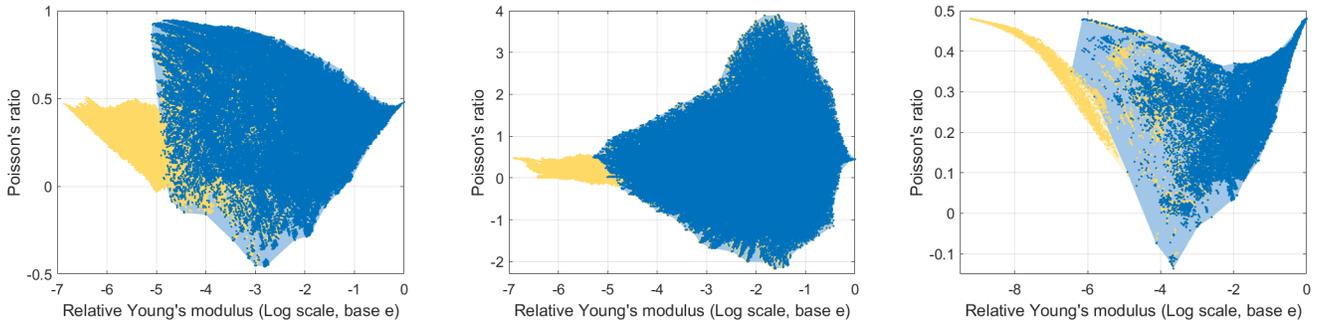


Fig. 1. Gamuts computed with our discrete-continuous sampling scheme for 2D cubic structures (*left*), 2D orthotropic structures (*middle*) and 3D cubic structures (*right*). The plots show the results for the projection of the gamuts on the plane defined by the macroscale Young's modulus along the x axis (normalized by the Young's modulus of the stiffest base material) and the Poisson's ratio corresponding to a contraction along the y-direction when the material is stretched along the x-direction. All these plots correspond to microstructures that use 0.48 as Poisson's ratio for the base material. The blue dots correspond to the filtered one-material microstructures while the yellow dots correspond to the original two-material microstructures.

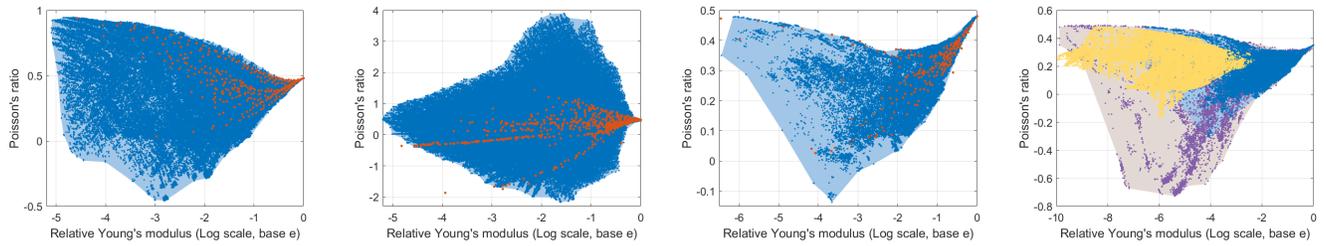


Fig. 2. Gamuts computed with our discrete-continuous sampling scheme for 2D cubic structures (*left*), 2D orthotropic structures (*second from left*) and 3D cubic structures (*second from right*) using 0.48 as Poisson's ratio, and 3D cubic structures with 0.35 as Poisson's ratio (*right*). The plots show the results for the projection of the gamuts on the plane defined by the macroscale Young's modulus along the x axis (normalized by the Young's modulus of the stiffest base material) and the Poisson's ratio corresponding to a contraction along the y-direction when the material is stretched along the x-direction. The blue dots correspond to the generated samples for $16 \times 16 \times 16$ microstructures, the purple dots correspond to generated samples for $64 \times 64 \times 64$ microstructures, the orange dots correspond to the microstructures from Schumacher et al. (2015) and the yellow dots correspond to the microstructures from Panetta et al. (2015).

but paused after the microstructure is modified, i.e. after the inner loop of the procedure has been executed, which corresponds to a so-called *barrier synchronization point*. When all the programs corresponding to all the microstructures of the population have reached this synchronization point, the scores of the partially modified microstructures are evaluated again and the population is resampled using systematic resampling. We again sample N microstructures, but since the scores have changed, the most interesting microstructures will appear several times, while the less promising ones will leave the evolving population. Note that the microstructures, and the associated programs, are duplicated together with their program execution history, i.e. the information regarding which voxels have been already visited, so that these voxels are not modified a second time. This ensures that interesting changes in the material assignments are preserved. After the synchronization point is reached and the microstructures have been resampled, the execution of the programs is resumed and the algorithm continues until all the voxels of all the microstructures have been visited. The entire SOSMC algorithm is summarized by Algorithm 3. In our implementation, we used $N = 3000$ microstructures for the 2D and 3D cubic databases, and $N = 10000$ for the 2D orthotropic database.

2 MATERIAL GAMUTS

We initially targeted multi-material printers and therefore computed databases of two- and three-dimensional microstructures made of two materials. We used isotropic base materials whose Young's modulus differed by a factor of 1000 and having 0.48 as Poisson's ratio. For comparison purposes with previous research (2), we adapted these databases to one-material microstructures by replacing the softer material by void, filtering out all the microstructures with disconnected components, filling the enclosed voids in the 3D case, and recomputing their homogenized properties. The two sets of gamuts are depicted in Figure 1. We observed that the gamuts corresponding to two-material microstructures and one-material microstructures have a very similar shape, except in the area corresponding to very soft microstructures. This is to be expected since microstructures made of soft material connecting small blocks of stiff materials are not realizable in the one-material setting. From an implementation point of view, it is worthwhile to note that, if the final intent of the user is to design one-material structures, these additional fabrication constraints can be directly accounted for during the sampling stage by preventing any change that would affect the validity of the microstructures. This avoids the need of filtering

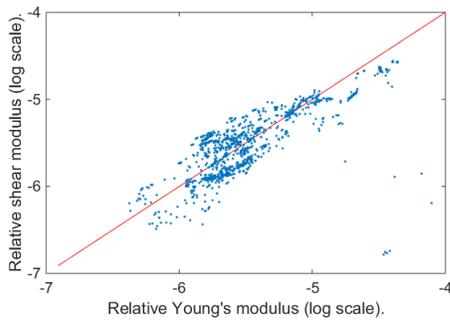


Fig. 3. The 64^3 microstructures span a wide range of relative shear modulus even they have negative Poisson's ratios. As an example, we plotted the distribution of structures with Poisson's ratio $\nu = -0.5 \pm 0.03$. Points on the diagonal line $\mu = \mu_{iso}$ are isotropic within the linear elasticity regime.

the database a posteriori and would improve the sampling density in regions that might be undersampled if one filters the database in a subsequent step. Note that we sampled the microstructures using a non-logarithmic scale for the Young's modulus and that the estimated density in the soft regions is higher than what it appears to be in Figures 1 and 2.

Our initial databases were computed for microstructures corresponding to $16 \times 16 \times 16$ arrangements of voxels. This limits the sizes of the thinnest features of the microstructures and therefore the softness of the softer material that can be achieved. When increasing the lattice size to 64, the gamut of the microstructure properties expands in this area and reaches what can be obtained when using other parametrization methods (see Figure 2, right). Due to high computation costs (each microstructure takes 29s to simulate in average), we limited our initial analysis of highly discretized geometries to the study of a database comprising about 10k microstructures. Notably, our search method allows us to find a wide range of single-material structures with negative Poisson's ratio ($\nu = -0.7$). While lower Poisson's ratio is theoretically achievable, such structures contain extremely thin joints unsuitable for manufacturing. These structures demonstrate a variety of relationships between Young's modulus and shear modulus. To be more precise, we define μ_{iso} as the shear modulus computed from Young's modulus E and Poisson's ν ratio using the relationship $\mu_{iso} = E/(1 - 2 \times \nu)$. We then compare μ_{iso} with the actual shear modulus μ of each structure. For structures with $\nu = -0.5 \pm 0.03$, the ratio μ/μ_{iso} achieves a range from 0.09 to 1.39 (Figure 3).

REFERENCES

- DOUC, R. 2005. Comparison of resampling schemes for particle filtering. In *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*. 64–69.
- PANETTA, J., ZHOU, Q., MALOMO, L., PIETRONI, N., CIGNONI, P., AND ZORIN, D. 2015. Elastic textures for additive fabrication. *ACM Trans. Graph.* 34, 4 (July), 135:1–135:12.
- RITCHIE, D., MILDENHALL, B., GOODMAN, N. D., AND HANRAHAN, P. 2015. Controlling procedural modeling programs with stochastically-ordered sequential monte carlo. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4.
- SCHUMACHER, C., BICKEL, B., RYS, J., MARSCHNER, S., DARAIO, C., AND GROSS, M. 2015. Microstructures to control elasticity in 3D printing. *ACM Trans. Graph.* 34, 4.