Generative Deformation: Procedural Perforation for Elastic Structures

Shane Transue† and Min-Hyung Choi‡

***Abstract*— Procedural generation of elastic structures provides the fundamental basis for controlling and designing 3D printed deformable object behaviors. The automation through generative algorithms provides flexibility in how design and functionality can be seamlessly integrated into a cohesive process that generates 3D prints with variable elasticity. Generative deformation introduces an automated method for perforating existing volumetric structures, promoting simulated deformations, and integrating stress analysis into a cohesive pipeline model that can be used with existing consumer-level 3D printers with elastic material capabilities. In this work, we present a consolidated implementation of the design, simulate, refine, and 3D print procedure based on the automated generation of heterogeneous lattice structures. We utilize Finite Element Analysis (FEA) metrics to generate perforated deformation models that adhere to deformation behaviors created within our design environment. We present the core algorithms, automated pipeline, and 3D print deformations of various objects. Quantitative results illustrate how the heterogeneous geometric structure can influence elastic material behaviors towards design objectives. Our method provides an automated open-source tool for quickly prototyping elastic 3D prints.**

I. Introduction

3D printing technology is developing at a blistering pace and has expanded to wide reaching implications for both academic and industrial research in product design, engineering, and manufacturing [16]. Yet, the control of elastic deformations of 3D printed objects remains challenging due to the complex interrelationship between the properties of available elastic materials, the density of internal lattice structures, and the connectivity of procedurally generated print geometry. To this end, several recent contributions have significantly improved the iterative design process to incorporate Finite Element Analysis (FEA) [6] through various software packages to facilitate print optimizations for stress reduction [3], print material minimization [9], internal lattice generation [4], vibration reduction [15], print structures [2], implications of layering orientation, and numerous other metrics. While the design, simulate, refine, print pipeline has been well established for modifying characteristics of 3D prints, there still remains a limited set of tools that provide an end-to-end solution for providing precise control of deformation behaviors in prints based on design constraints.

Several variations of simulation-based optimization pipelines have been introduced [17] within the domain of 3D printing to address a numerous challenges with printing controllable, structurally robust objects and parts. Most pipelines attempt to introduce an overarching bridge between the initial design of a part and the resulting print subject to several additional constraints that can be defined to optimize the print with respect to a given set of objectives. Significant contributions have been introduced through the precise control of elastic structures using micro-structures [11] and the behavioral optimization of multi-material micro-structures [19]. However, these advanced techniques assume that 3D printers with an extremely high print resolution or multiple mixable materials are available. The problem is that the cost and limited set of tools that are specifically designed for these expensive 3D printers are challenging to use with consumer-level printers that have limited elastic material capabilities.

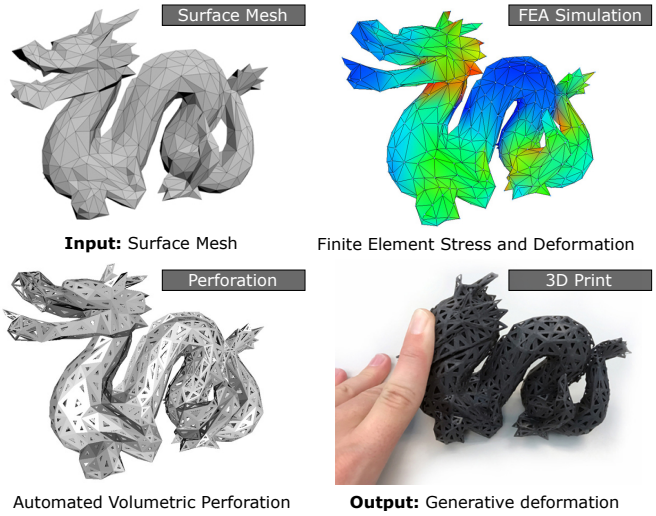

Fig. 1. Generative deformation through automated perforation. Heterogeneous internal structure is defined through a simulation model to promote deformation and minimize stress. The input is a simple surface mesh and the output is a 3D printable surface mesh.

In this work, we introduce a consolidated pipeline that integrates both automated perforation, deformation behavior, and stress analysis to provide an automated process that allows for the development of elastic 3D prints on consumer-level printers. Based on this objective, we focus on the following set of contributions:

- Combine generative modeling with FEA dynamics to introduce deformation-driven procedural geometry
- Integrate a 1-to-1 pipeline between volumetric simulation meshes and elastic 3D print geometry
- Provide a procedural algorithm for generative deformation control using any input surface mesh
- Automate the process of generating heterogeneous internal structures within a single application for 3D printing

These contributions incorporate a design application that integrates existing volumetric mesh generation algorithms, FEA simulation, and perforated internal structure genera-

† This work is partially supported by DoEd GAANN Fellowship: P200A150283. Shane Transue, and Min-H. Choi are with the University of Colorado Denver Department of Computer Science and the Comcast Media and Technology Center‡. Emails: {shane.transue, min.choi}@ucdenver.edu

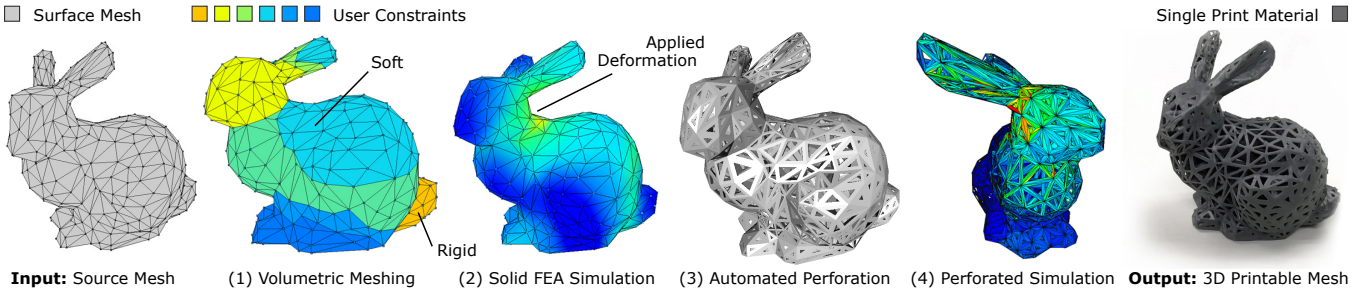

Fig. 2. Automated volumetric perforation pipeline. The input is a simple surface mesh and the main process consists of a four stages: (1) volumetric meshing and element-wise user constraints, (2) FEA simulation of the solid mesh, (3) the automated perforation, and (4) the simulation of the perforated mesh. The result produced by the perforation algorithm provides a valid 3D printable surface mesh that can be printed using consumer-level 3D printers.

tion to quickly obtain 3D prints specifically targeted for consumer-level 3D printers. Building on this pipeline, shown in Figure 2, we facilitate a method for *generative deformation* which allows for a heterogeneous lattice structure to be generated with respect to deformation objectives. We define this process as the procedural generation of internal geometry that is derived from two primary metrics obtained through FEA simulation: (1) measures of tetrahedral element deformation using dihedral angles and (2) internal stress distributions derived from nonlinear FEA simulations [8], [5] using VegaFEM [14]. This objective does not only impact passive deformations of 3D prints, but can also be used to modify activated behaviors of 4D prints based on how heterogeneous structures impact actuated movements [18].

II. Automated Volumetric Perforation

Printing elastic structures that follow a prescribed deformation behavior is difficult for numerous reasons including: (1) material properties are limited to those that are compatible with the 3D printer, (2) the internal geometric structure plays a significant role in how the object deforms, (3) the printing process itself (layers, method, etc.) largely influences behavior, (4) the curing process is not only unique between different materials but even slight variances in applying the same curing method to the same material can result in different behaviors, and (5) simple and common operations such as scaling can drastically alter how a print behaves. These five contributing factors contribute to substantial challenges in developing generalized algorithms that can provide specific deformation behaviors.

To address the wide variances in how these factors contribute to the deformation behavior of a 3D print, procedural geometry and simulation have become critical to the design process. Based on these requirements, it is beneficial to have a methodology that can quickly adjust the geometric distributions that modify behavior based on the conditions of the simulated objective behavior. To achieve this with a real-time design element, we have implemented a geometrically reduced form of *inset-based* procedural geometry to formulate an automated form elastic control: automated *perforation*. This provides a variety of different design tools that can be employed to modify the geometric material distribution of the print based on various functions including: lattice thickness, gradient functions, per-element constraints, and precise painting-based user adjustments.

A. Geometric Perforation

The process of perforating a volumetric structure is derived from an inset operation based on the shape of the interface between adjacent elements. This operation defines a single scalar value i where $i \in [0, 1]$ defines the *inset* of the element. For a tetrahedral mesh, the interface element is a triangle and the inset operation is defined as follows: given an interface triangle t and inset i, compute the lines parallel to each edge of the triangle (a, b, c) that create three intersection points of the inset as shown in Figure 3. If two neighboring tetrahedra have different inset values $(i,\ j)$, then each 2D edge element is divided into two regions, one for each inset value. This results in voids within both elements A and B that provide the basis for altering effective elasticity of each element.

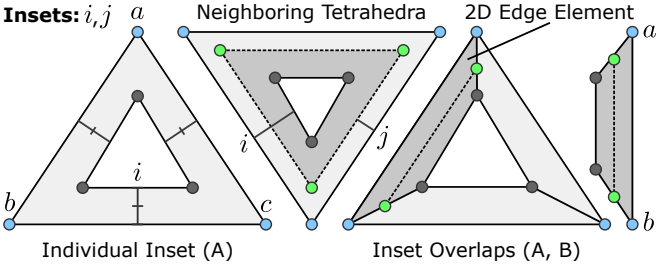

Fig. 3. Inset of the shared interface triangle between tetrahedral elements (A) and (B). The inset values $i, j \in [0, 1]$ define a percentage inset that perforates the surface of the triangle between the elements.

This operation has been introduced in 3D printing for biomass structures [7]; however, the presented formulation is *invalid* with respect to the requirements of simulation geometry. This is because this naïve form of the algorithm introduces coplanar and unconnected faces and no internal elements. To simulate an elastic object using FEA, the volumetric mesh must provide a cohesive and valid set of elements that discretely represent the continuum of the material. Therefore, we must generate non-overlapping internal elements with coherent indices that form a continuum.

To define the 3D form of the perforation algorithm we assume: a surface mesh is provided and an existing meshing algorithm such as *Delaunay*-based Tetgen [13] is used to generate a volumetric tetrahedral mesh. Based on this mesh, we generate a graph G of all internal triangle faces F shared between neighboring tetrahedral elements and perform the 2D inset operation in parallel for all faces. From this, we consolidate all nodes and generate element indices to form a collection of *micro-tetrahedra* that compose the

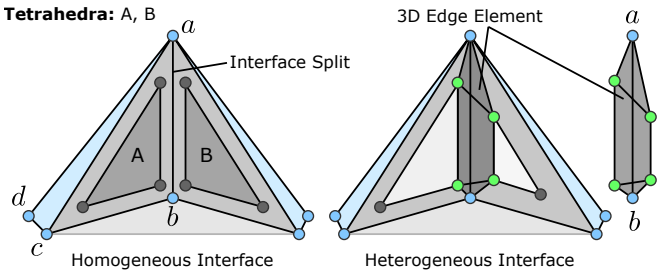

Fig. 4. Inset diagram for neighboring tetrahedral elements for 3D perforation. In the homogeneous case, all nodes align and can be welded. For the heterogeneous case, the interface results in discontinuous node positions from each inset. These must be resolved using a predetermined enumerated set of edge configurations to form a cohesive FEA mesh.

edge elements of the perforated mesh. The complication comes from three challenges: (1) element indices can be arbitrarily ordered, (2) the configuration of the pairwise interface between different insets requires an enumeration of possible element configurations that depend on how the inset values i and j relate, and (3) based on the topology of the volumetric mesh, each edge may have no adjacency, one adjacent or two adjacent tetrahedra. Thus, each edge may require the generation of 3, 6, or 9 internal micro-tetrahedra depending on the local interface topology. Additionally, with the generation of all edge elements (from Figure 4), there remain voids within the corners of each original tetrahedra that must also be filled with micro-tetrahedra elements.

For arbitrarily ordered element indices we define a simple correlation map for each interface between adjacent elements. For each inset configuration case, $(i = j)$, $(i \neq j)$, $(i < j)$, and $(i > j)$, we employ an enumerated generation of all possible edge configurations. This results in an element generating lookup table of edge elements as shown in Figure 5 (left) shared between any two tetrahedra as: 3-Tet edge, 6-Tet edge, or 9-Tet edges. Similarly, the inset values also impact the generation of the corner elements, resulting in another element lookup table shown in Figure 5 (right). In the instance where node positions become arbitrary close (ϵ), they are welded to reduce geometric complexity and minimize the number of low quality tetrahedra. The generation of the micro-tetrahedra for all elements results in a perforated, continuous mesh that can be simulated for stress evaluation.

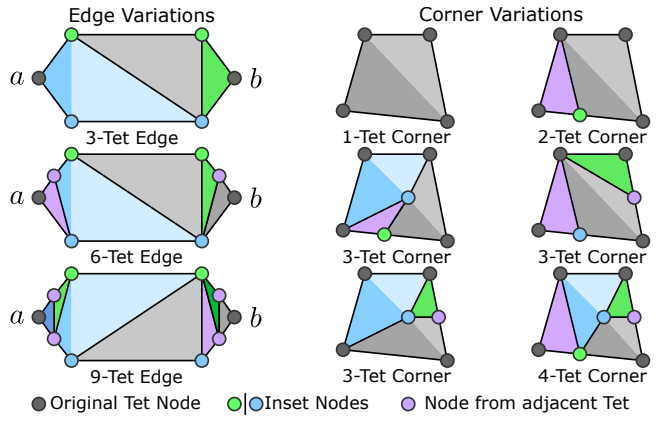

Fig. 5. Edge and corner micro-tetrahedra element lookup tables. These configurations are used to define all possible micro-tetrahedra combinations generated between adjacent tetrahedral elements defined by insets i and j.

B. Mesh Perforation Algorithm

The algorithm assumes as input a tetrahedral mesh M, and a real-valued inset value i or set of inset values I where $|I|$ is equal to the number of elements in the mesh and $i \in [0, 1]$ $\forall i \in I$. First, all element insets are computed in parallel based on each element's associated inset value $I[i]$. Then for each interface between adjacent elements, micro-tetrahedra are generated according to the edge relationships (3, 6 or 9-Tet variants) between the element inset geometry using the lookup configurations in Figure 5. Finally, the micro-tetrahedra elements for all non-interface edges on the surface of the object and internal corners are computed. The output is a collection of all generated micro-tetrahedra elements and faces that represent the perforated tetrahedral mesh P. This mesh contains the set of nodes, elements, and faces that can be used for both simulation and as a printable surface mesh.

Algorithm 1 PERFORATION(M, I)

Input: Tetrahedral Mesh: M($nodes$ N, $elements$ T)
Real-valued Element Insets: I ($|I| = |T|$)
Output: Tetrahedral Mesh: P

$P.nodes \leftarrow N$
Parallel for:
for *each* $t \in T$ **do**
  InsetTetrahedraElement(t, $I[i]$)
end for
`// Compute graph of internal faces`
G = GenerateInternalFaceGraph(M, F)
`// Compute interfaces/types (3,6,9)`
E = GenerateElementInterfaces(G)
`// Generate all micro-tetrahedra elements`
$P.elements \leftarrow$ Internal_MicroTets(E, G)
$P.elements \leftarrow$ Surface_MicroTets(E, G)
$P.elements \leftarrow$ Corner_MicroTets(E, G)
return P

This algorithm inherently covers both homogeneous and heterogeneous element perforations depending on the inset values within I (constant or multiple unique values). The full version of the perforation algorithms are implemented in C++ and available at our lab website: http://graphics.ucdenver.edu/generativedeformation.html

C. Heterogeneous Perforation

To provide meaningful control within the deformation of an object printed with one material, the mass distribution, internal structure, and member thickness must be varied at specific regions to modify the elastic behavior within the print. Towards this objective, we introduce the ability to specify *per-element* inset values as the basis for creating a heterogeneous mesh structure. Due to the numerous potential methods for defining the entire array of inset values, and the resolution of the provided mesh, the composition of the object may vary drastically. To illustrate the potential differences in heterogeneous mesh generations, Figure 7 shows how different gradients within the Stanford bunny model result in different geometric compositions that will result in different localized deformation behaviors.

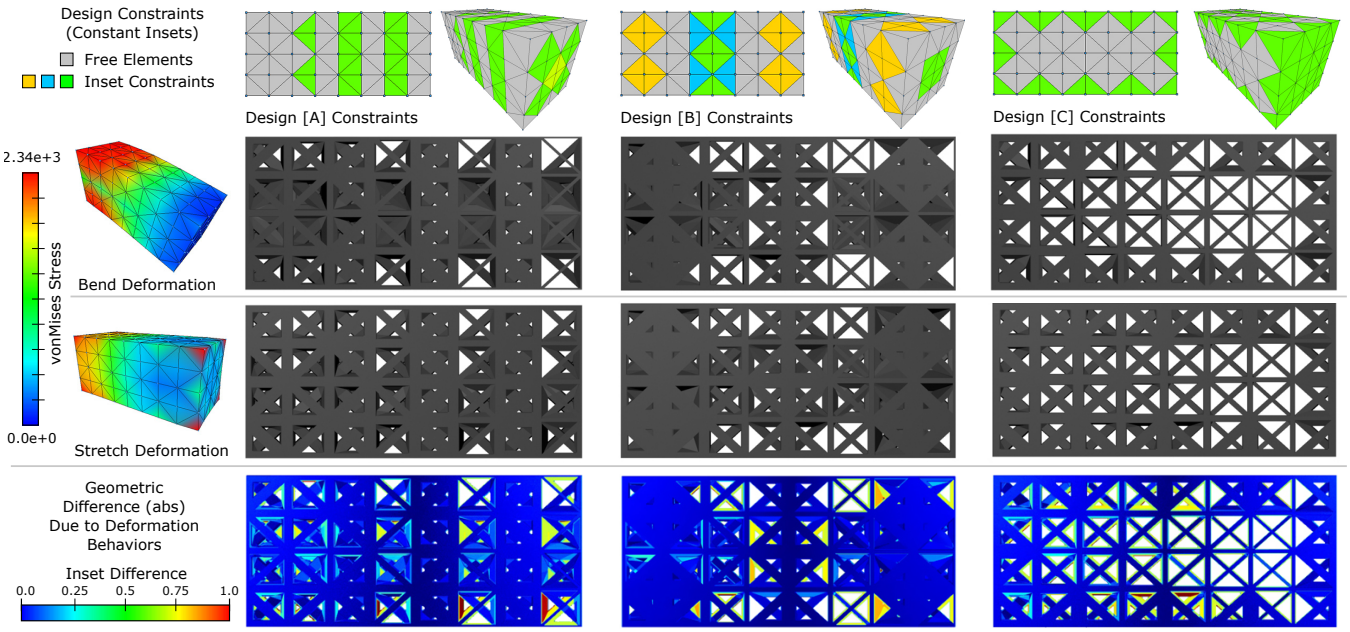

Fig. 6. Generative deformation: integration of design constraints, deformation behavior, and stress analysis to generate heterogeneous geometric structures consisting of a single elastic material model. The design shape (top row) has specific design constraints that are optimized to generate different deformation and stress patterns (bend and stretch) that alter inset values. The differences between these optimizations (bottom row) are shown for each design.

Providing an automated method for generating heterogeneous internal structures is only as powerful as the methods that can generate meaningful inset values for all elements. Therefore, our generative deformation process relies on deformation and stress analysis as a basis for modifying the dynamic behaviors of the 3D printed model by automatically generating inset values that promote specific behaviors. In addition to the simulated model driving the deformation behavior, we also consider design objectives and user defined element constraints. To address these design features we introduce a stage within the automated pipeline for *element constraint design* to alter how element behaviors are generated through inset variation driven by the FEA simulation.

D. Element Constraint Design

Simulated prototyping and optimization provide an ideal basis for refining and adjusting control over deformable objects; however, from a design perspective, objectives are typically achieved through specifying boundary conditions and sparse set of constraints. Any algorithmic scheme that can automate the process of controlling deformations should also allow for the definition of boundary and behavioral characteristics critical to the functional design of the print. To achieve this within the perforation pipeline, we introduce an intermediate design stage where explicit constraints can be defined as *design-specific* inset values. Utilizing selection or painting-based techniques, the inset values of elements can be directly controlled. For higher-level more indirect control of these constraints we introduce a *gradient-falloff* algorithm for editing insets within general regions. This algorithm uses internal face adjacencies to perform a breadth-first search through which the mathematically defined falloff function will apply smooth inset transitions based on adjacency. This provides a method for introducing controlled smooth gradients within the perforation that do not depend on the FEA simulation and allow for explicit control of the perforation. To allow for a mixture between sets of user defined inset constraints or boundary conditions and simulation-driven inset values, individual elements can be assigned to have free (simulation controlled) or constant value (user defined) insets as shown in Figure 6.

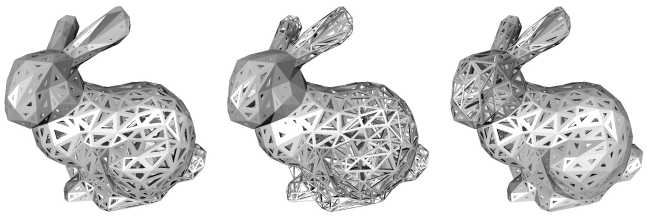

Fig. 7. Heterogeneous geometric structure due to per-element specified inset values. This method enables smooth gradients between elements and allows the elasticity of specific regions of the mesh to be controlled through an FEA simulation result or user-defined constraints.

III. Generative Deformation

The process of *generative deformation* incorporates the procedural generation of geometric structures that promote design-specific deformation behaviors. Through the simple observation that existing methods explore [11], varying the thickness or design of internal micro-structures varies the elastic characteristics of the object and doing this to carefully selected regions can significantly change deformation behaviors. To build on this observation and provide an automated pipeline for generating perforated structures, we introduce a method that combines design, element deformation behavior, and stress analysis into a single consolidated deformation model. This model is then constructed using volumetric perforation, resulting in a 3D printable surface mesh that defines the generated structural elasticity.

In the generative process, we form an iterative design cycle that incorporates the design, refinement, and recording of simulation behaviors to provide control over how mesh structures are generated using the perforation operation. To provide this control, we present a weighted three component formalization of our generative algorithm based on: (1) user design constraints, (2) elastic deformation objectives, and (3) stress-based reinforcement. These control mechanisms allow for application specific constraints, deformation objectives of the print, and strength optimization through simulation-driven stress analysis. Each of these components are integrated into this weighted model based on a recording of the mesh M under an applied load in the simulation:

$$R(M) = c(e_i) + \alpha \cdot d(e_j) + \beta \cdot s(e_j), \quad \forall e \in M \qquad (1)$$

where R is a recorded FEA simulation of the mesh including the set of user defined inset-constrained elements $c(e_i)$ and normalized deformation and stress values of the element sets $d(e_j)$, $s(e_j)$ represent the deformation and stress of all free elements e_j in mesh M. The coefficients α, $\beta \in [0, 1]$ represent a bias towards promoting larger deformations (α) or reducing stress ($\beta = 1 - \alpha$). For the deformation function $d(e)$, we employ a simple metric based on the dihedral angles [10] of the tetrahedral element e. This provides a rotation invariant measure of deformation within an element defined as the magnitude of six independent angles $\theta_{ij} \in [0, \pi]$ between adjacent faces in each element. For the stress states, we store the Cauchy stress tensor (evaluated post polar decomposition) that can then be used to compute the *von Mises* scalar representation of the stress in each element. Both of these values are evaluated with respect to Equation 1 to generate the final inset values within the array I that defines all of the per-element insets for the mesh based on recording R. This process is outlined in Algorithm 2.

Algorithm 2 GENDEFORMATION(M, $R(M)$, α, β)

Input: Tetrahedral Mesh: M
FEA Recording R of M with n element states
Deformation and stress coefficients α, β
Output: Real-valued Inset Array: I (scalar insets $\in [0, 1]$)

```
for each constrained element e_i ∈ M do
    I[i] = c(e_i)
end for
for each free element e_j ∈ M do
    for each frame f in R do
        d[j] ← R.dihedralAngleMag(e_j)
        s[j] ← R.vonMisesStress(e_j)
    end for
    I[j] = α·mean(d) + β·mean(s)
end for
return I
```

To facilitate the generative deformation pipeline within a manageable design environment, we provide a design application that provides: importing surface meshes, volumetric meshing, FEA simulation, perforation tools, and the ability to export perforated meshes for printing. A screenshot of our perforation design studio is shown in Figure 8.

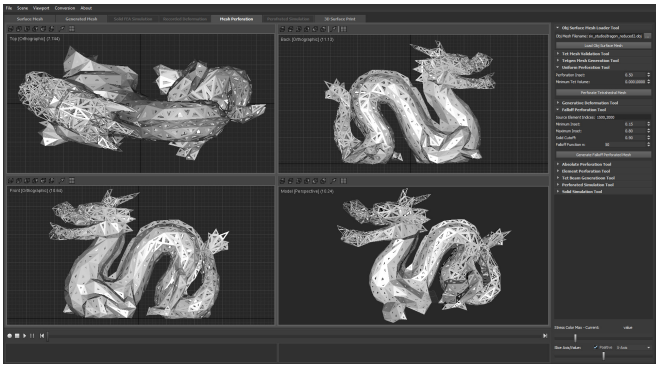

Fig. 8. Perforation design studio. This application includes the ability to import, perforate, and export 3D printable surface meshes.

IV. EXPERIMENTAL RESULTS

Stemming from the introduction of the generative deformation pipeline and the perforation algorithm, we experimentally verify various deformation behaviors on 3D printed perforated meshes. The experimental prints were obtained through simple perforation and use of a Formlabs Form 1+ stereolithography (SLA) 3D printer [1] with its single *flexible* resin and generated support structures. This process is shown in Figure 9. We also experimented with Fused Deposition Modeling (FDM) printers using Thermoplastic polyurethane (TPU) filaments and obtained several successful prints.

Fig. 9. Volumetric perforation printing process (SLA) illustrating the resin print (left), alcohol wash (center), and $405[nm]$-UV cure (right).

Qualitatively, the perforation algorithm provides promising results for printing heterogeneous internal structures, even when support structures are required. In our experimentation, we were able to obtain successful prints without inserting internal support structures in the perforated voids. However, an implication of the tetrahedral insets is that stress regions may be induced at the corners of insets which are prone to tearing, but this is material dependent. Our 3D printed bunny and dragon models are shown in Figure 10. For our quantitative experimentation, five perforations were generated and printed using a 8x4x4 tetrahedral bar as the initial input mesh.

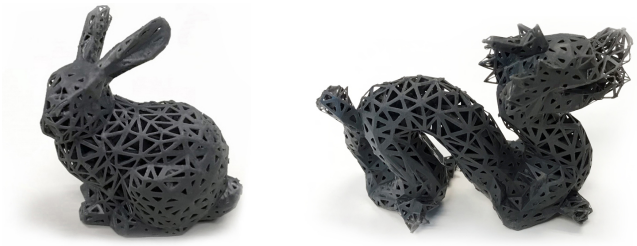

Fig. 10. Final result of the perforated Stanford Bunny and Dragon models printed using an SLA-based printer with a single flexible resin. Each region has variability in the elastic behavior (hard, soft) due to the perforation.

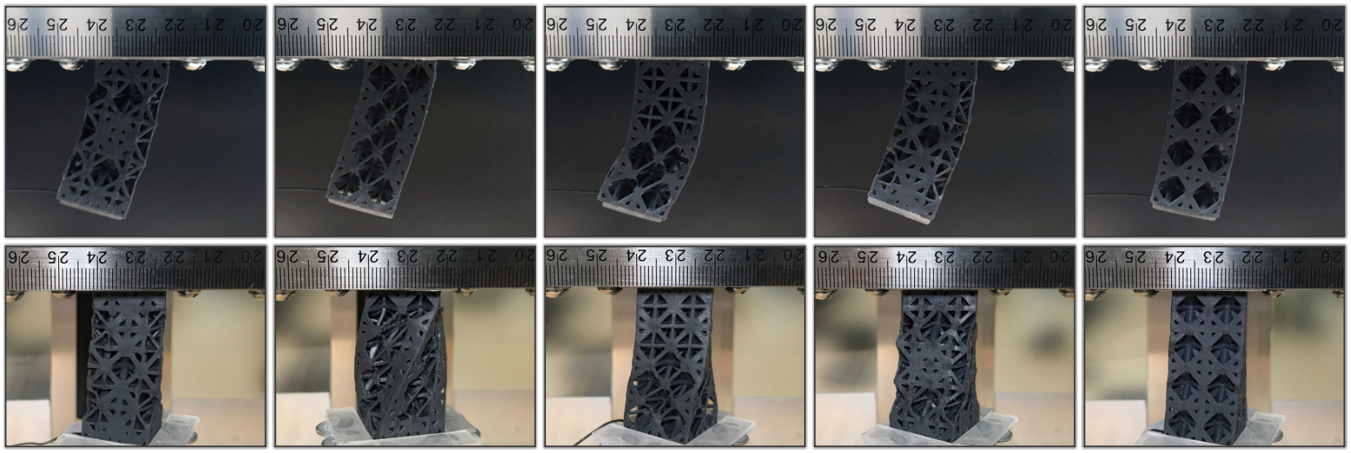

Fig. 11. Result of deforming five beams generated using the generative deformation algorithm. The top row illustrates each beam subject to a deformation imposed by a horizontal load, inducing various deflection behaviors. The twist deformation (bottom row) is imposed by an applied torque to demonstrate nonlinear rotational deformations. Each exhibits unique deflections and localized deformations due to the changes in the heterogeneous perforations.

The premise of this experiment is to generate perforated meshes that alter deflection and rotational flexibility of each print subject to design constraints and measure the resulting deformation behaviors under incremental loads. The models are shown in Figure 12 as designs [0]-[4].

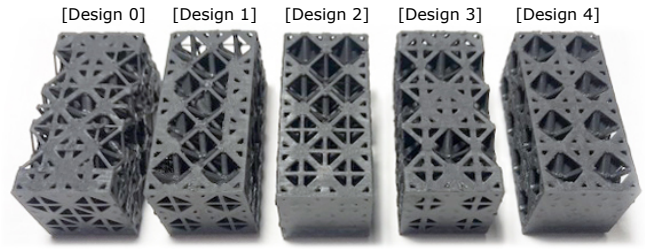

Fig. 12. 3D printed perforation meshes for testing the generative deformation pipeline. Each is a 2.0x2.0x4.5[cm] single material perforated print.

For both experiments, the deflection displacements and rotation angles from the reference configuration have been measured [12] to evaluate the deformation of each model. These loads were applied using incremental weights (at 100g intervals under gravity) to obtain the forces indicated in Figure 13 and torque applications in Figure 14. These measurements have been recorded to identify the impact of the geometric variance introduced by each generated design. In both experiments, the heterogeneous design of each model introduces changes in both behavioral deformations and deflections at equilibrium that can be adjusted to achieve specific design-oriented objectives using a single material.

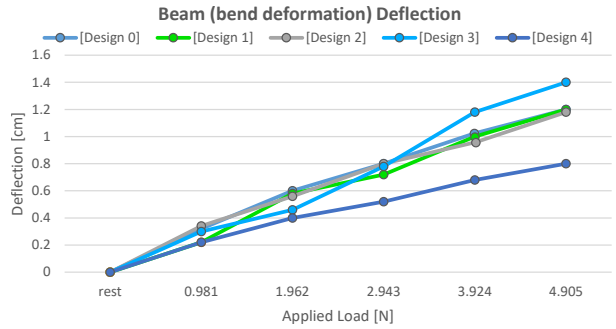

Fig. 13. Plot of the external loads used to induce beam deflection for each of the five perforation designs as shown in Figure 11 (top row).

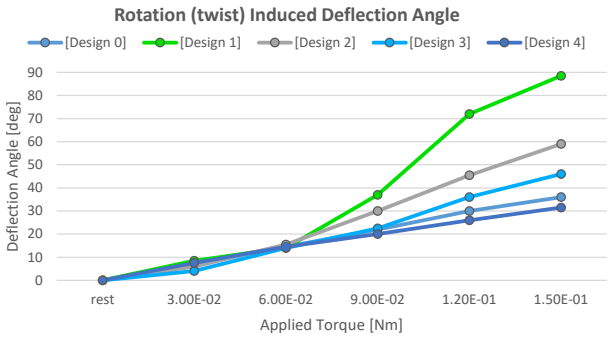

Fig. 14. Resulting rotational angles achieved by each of the five designs.

The deformations of the perforated prints shown in Figure 11 indicate that (1) the elastic behavior of the uniform material can be controlled by modifying internal geometric insets and (2) through simulation-driven inset optimization we can generate heterogeneous structures that promote specific localized deformation characteristics. This indicates that from different geometric structures we can enable common objectives such as material volume minimization, stress reduction, and user-defined constraints while maintaining prescribed deformation behaviors. This provides an important tool for rapid prototyping through model simulation for deformable prints when a limited number of elastic materials (such as curable resins and filaments) are available.

V. Conclusion

In this work, we introduced an automated method for generating heterogeneous material structures through the use of volumetric perforation. The presented method provides a generative deformation pipeline that enables user constraints and stress analysis using recorded FEA simulations to control the elastic behaviors of elastic 3D printed objects. The contributions of this work are consolidated within an open-source tool for use with various types of consumer-level 3D printers with limited selections of flexible materials.

Acknowledgment

This work was partially supported by the Department of Education (DoE) GAANN Fellowship P200A150283.

REFERENCES

[1] Formlabs - designs and manufacturers the form 2 stereolithography (sla) 3d printer. https://formlabs.com/.

[2] Grasshopper 3d™ - a graphical algorithm editor for rhino's 3-d modeling tools. https://www.grasshopper3d.com/.

[3] ntopology - advanced cad for advanced manufacturing. functional, generative design and optimization software for high-performance parts. https://www.ntopology.com/.

[4] Rhinoceros (rhino 3d) - design, model, present, analyze, realize (3d cad for printing). https://www.rhino3d.com/.

[5] J. Barbic. *Real-time Reduced Large-deformation Models and Distributed Contact for Computer Graphics and Haptics*. PhD thesis, 2007.

[6] K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, 1996.

[7] F. P. Melchels et al. Additive manufacturing of tissues and organs. *Progress in Polymer Science*, 37, 2012.

[8] M. Müller and M. Gross. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, pages 239–246, 2004.

[9] A. Nadal, H. Cifre, J. Pavon, and O. Liebana. Material use optimization in 3d printing through a physical simulation algorithm. *Automation in Construction*, 78:24 – 33, 2017.

[10] G.-P. Paillé et al. Dihedral angle-based maps of tetrahedral meshes. *ACM Trans. Graph.*, 34(4):54:1–54:10, July 2015.

[11] C. Schumacher et al. Microstructures to control elasticity in 3d printing. *ACM Trans. Graph.*, 34(4):136:1–136:13, July 2015.

[12] M. Shahzad, A. Kamran, M. Z. Siddiqui, and M. Farhan. Mechanical Characterization and FE Modelling of a Hyperelastic Material. *Materials Research*, 18:918 – 924, 10 2015.

[13] H. Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41(2):11:1–11:36, Feb 2015.

[14] F. S. Sin, D. Schroeder, and J. Barbic. Vega: Nonlinear fem deformable object simulator. *Computer Graphics Forum*, pages 1–13, 2013.

[15] W. P. Syam, W. Jianwei, B. Zhao, I. Maskery, W. Elmadih, and R. Leach". Design and analysis of strut-based lattice structures for vibration isolation. *Precision Engineering*, 52:494 – 506, 2018.

[16] H. N. G. Wadley. Multifunctional periodic cellular metals. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 364 1838:31–68, 2006.

[17] O. Weeger, N. B. andd S.-K Yeung, S. Kaijima, and M. L. Dunn. Digital design and manufacture of soft lattice structures. *Preprint, Singapore University of Technology and Design*, 2017.

[18] O. Weeger, Y. S. B. Kang, S.-K. Yeung, and M. L. Dunn. Optimal design and manufacture of active rod structures with spatially variable materials. *3D Printing and Additive Manufacturing*, 3(4):204–215, 2016.

[19] B. Zhu, M. Skouras, D. Chen, and W. Matusik. Two-scale topology optimization with microstructures. *ACM Trans. Graph.*, 36(4), July 2017.