

Title:

**Design and Simulation of Multi-Tiered 4D Printed Objects**

Authors:

Joel Pepper, jcp353@drexel.edu, Drexel University

David E. Breen, david@cs.drexel.edu, Drexel University

Keywords:

FDM, 4D Printing, Simulation, Optimization, Inverse Design

DOI: 10.14733/cadconfP.2022.140-144

Introduction:

“4D printing” refers to the 3D printing of objects which deform over time in response to some stimulus. In its most common setup, stress is built into thermoplastics by varying print speed and layer thickness, then the stress is partially released via heating and the plastic differentially shrinks. This process shows promise in decreasing the packaging size and material usage for manufactured products. Parts with 2D topology are able to be printed flat, sent to their destination, then stimulated (via a heated water bath for example) and warp into their final 3D shape. Complex surfaces and structures can also be produced without multiple printed parts or specially created molds, potentially saving on material and assembling costs. Work on the design of fused deposition modeling (FDM) 4D prints has to date mostly involved the printing of thin, flat initial configurations, which are then warped to produce 3D shapes [3, 4].

In this work, we present an inverse design and forward simulation process for 4D prints consisting of multiple-tiered layers of warping material. The model begins as a flat 3D grid of cubes, each defined as 8 vertices and 12 edges, whose edges shrink in accordance with the behavior of 4D printed thermoplastics. The modeling system is able to recreate 3D curved surfaces defined by Bézier patches and triangle meshes by variably shrinking edges across sets of “nodes” in order to produce the same local curvatures found in the input, target surface. Each set of vertically aligned cubes and the vertices and edges that comprise them constitute a node. This work represents an extension of 4D print modeling from single-layer initial conditions to multi-layered three dimensional initial conditions, while still being theoretically actualizable; thus increasing the diversity of objects that can be created via 4D printing and allowing for greater double curvature than is typically available with single-layer 4D prints.

Surface Modeling with 4D Printed Grids:

The core modeling structure of our framework is a rectangular grid of cubes consisting of three or more layers of vertices, as seen in Fig. 1. The function of each individual node within the grid is to reproduce the curvature present within a node’s respective region of the target surface. To accomplish this, we variably shrink each layer of each node, as seen in Fig. 2. If layers are considered to lie in a plane with axes  $u$  and  $v$ , the inverse design process detailed in the *Inverse Design of Surfaces* section determines the lengths  $L_u^n$  and  $L_v^n$  for each layer  $n$  that will produce the target directionally signed curvature values  $\kappa_u$  and  $\kappa_v$  for every node. Given a  $\kappa$ , the radius of curvature  $R$  for the region is defined by

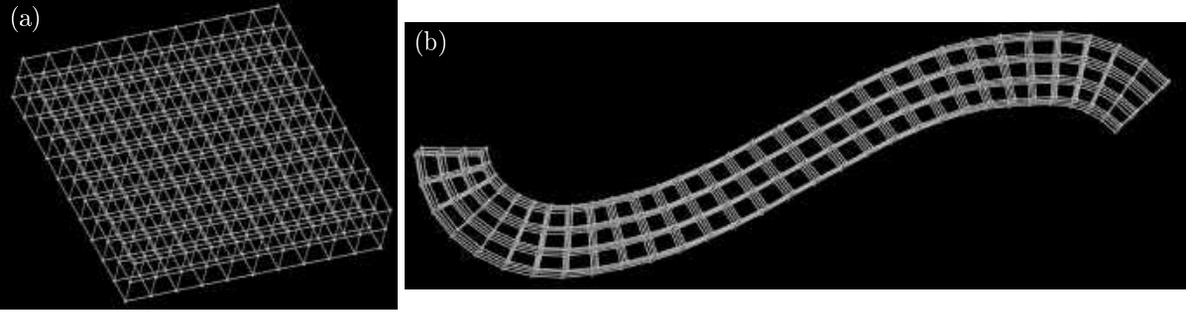


Fig. 1: (a) A  $10 \times 10 \times 3$  grid made up of 100 nodes. (b) Reversing curvature on a strip of three by thirty four-layer nodes.

$$R = \frac{1}{|\kappa|} . \quad (2.1)$$

In order to determine how much to shrink each layer within a node, the angle  $\theta$  is needed.  $\theta$  refers to the angle created at the intersection of the vertical edges of a node if they were to extend infinitely. Using  $R$  and  $L^0$ ,  $\theta$  can be calculated via

$$\theta = 2 \arcsin \left( \frac{L^0/2}{R} \right) . \quad (2.2)$$

Depending on the direction of the curvature,  $L^0$  is placed either at the top or bottom layer of the node. Its neighboring layer is referred to as  $L^1$ , then next to that  $L^2$ , and so on. The following formula is used to determine what length a layer  $L^n$  should shrink to:

$$L^n = L^0 - 2n\tau \sin \left( \frac{\theta}{2} \right) , \quad (2.3)$$

where  $\tau$  is the vertical distance between layers. The curvature results for this process can be seen in Fig. 1(b). On a strip thirty nodes long, the value for  $\theta$  is set to 1.0 radian on one end,  $-1.0$  on the other end, and interpolated for the nodes in between to create bending in both directions.

#### Inverse Design of Surfaces:

Input target surfaces can take the form of a bicubic Bézier patch or a triangle mesh. In the case of a Bézier patch, the input consists of sixteen control points  $P_{ij}$ , where  $0 \leq i, j \leq 3$ , which determine the shape of the surface. For values of  $u$  and  $v$  where  $0 \leq u, v \leq 1$ , points  $p$  on a patch are determined via

$$p(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 b_i(u) b_j(v) P_{ij} , \quad (3.1)$$

where  $b_i$  and  $b_j$  are cubic Bernstein polynomials. With Eq. (3.1), arc lengths  $L^0$  can be measured by sampling  $p(u, v)$  values within the region of each node and measuring the distance between neighbors. In order to measure curvature  $\kappa$ , we generate four Bézier control points for the Bézier curve found at one fixed value of either  $u$  or  $v$ . These curves are defined by

$$Q(u) = \sum_{i=0}^3 P_{i+1} \binom{3}{i} u^i (1-u)^{3-i} \quad (3.2)$$

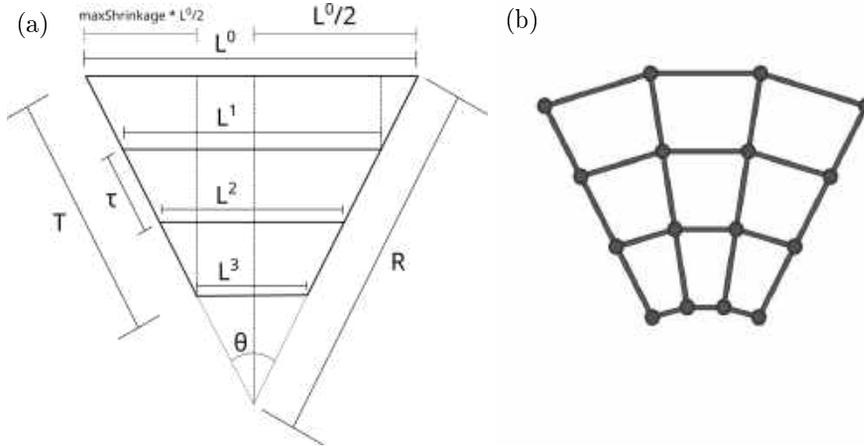


Fig. 2: (a) Elements and dimensions of a single node.  $R = 1/|\kappa|$ .  $\kappa$  is the local surface curvature captured by the node.  $L^n$  are the rest lengths for horizontal edges.  $T$  is the vertical height of a node, and  $\tau$  is the spacing between layers. Also shown are the similar triangles used in the derivation of Eq. (2.3). (b) Three simulated four-layer nodes after edge shrinkage.

and its derivatives are  $Q'(u)$  and  $Q''(u)$ .  $\kappa$  is calculated via

$$\kappa = \text{sgn}(Q''(u)_w) \frac{\|Q'(u) \times Q''(u)\|}{\|Q'(u)\|^3}, \quad (3.3)$$

where  $w$  is the dependent component  $x$ ,  $y$  or  $z$  for the input surface. Eq. (3.3) is then sampled and averaged across the region of each node.

In the case of triangle mesh inputs, the goal is to approximate continuous isolines along the surface of the model, similar to what is provided by the Bézier equations. To start, equally spaced parallel rays are shot at the model in order to determine a series of isolines in both  $u$  and  $v$  comprised of ray-triangle intersection points. Catmull–Rom splines are then fit across every isoline. These being made up of Bézier curves, we then evaluate the  $Q$  equations for the curves between each intersection point to determine  $L^0$ 's and  $\kappa$ 's for every node.

#### Forward Simulation of Models:

In order to simulate the deformation of models, we use the *ShapeOp* geometry optimization library [1]. To mimic the physical properties a printed plastic structure would theoretically exhibit, edges of nodes are modeled as basic springs, referred to as *EdgeStrainConstraint*'s in *ShapeOp*. With these alone however, the structure will undergo little to no out-of-plane deformation—and thus no curvature—in the event it roughly maintains its original shape. It is also possible for the grid to collapse in on itself during edge contraction. To avoid these unwanted outcomes, we add angular spring (*AngleConstraint*'s) at every vertex of the grid. There are up to twelve of these constraints per vertex, one going to each neighboring vertex. Lateral edge spring rest lengths for a given layer  $n$  are set to  $L^n$  from Eq. (2.3), vertical edge spring rest lengths are set to their initial lengths, and all angular springs are set to  $90^\circ$  as per the initial conditions of all grids (see Fig. 1). Following initialization, *ShapeOp* is run for a fixed number of iterations (15,000 for the results in Fig. 3) and then returns the final vertex positions that it determines as optimal. The optimization process is conducted in two stages: the first with only a single layer shrinking in-plane, and the second with all layers and curvature-based shrinkage included. These

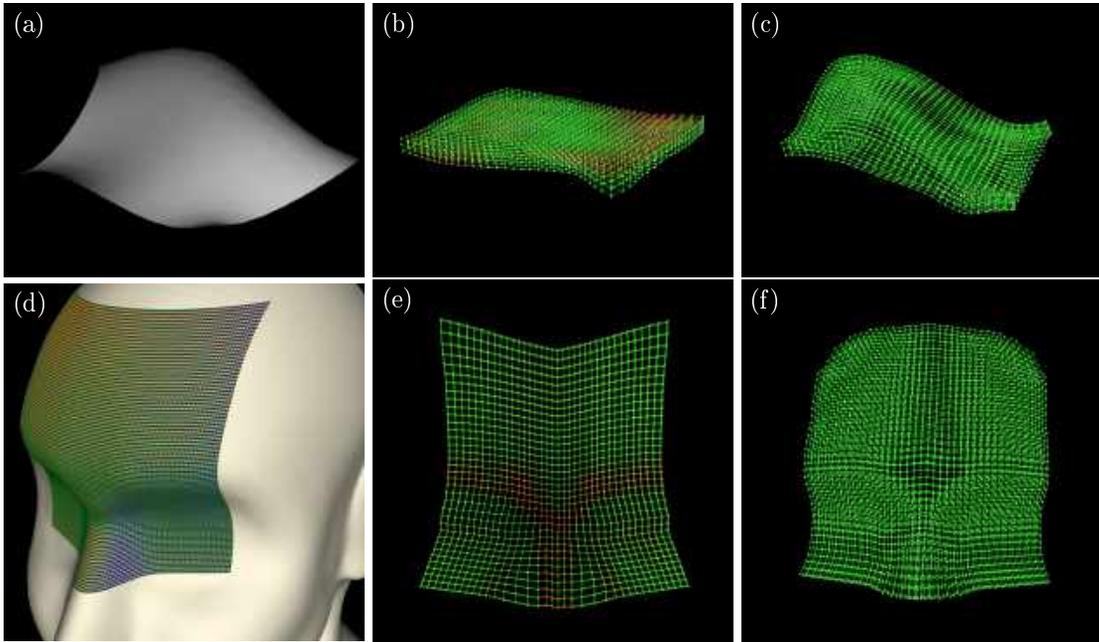


Fig. 3: (a-c) Results for a Bézier patch input. (d-f) Results for a triangle mesh input.

two stages are in line with the understanding that 3D curved surfaces can be defined via a combination of in-plane strain and out-of-plane curvatures [2].

#### Results:

A Bézier patch result is presented in Fig. 3(a-c). This patch curls up at one of its corners, down at the other three, has one heavily elongated edge, and a bulbous center. The patch measures 866.3 by 734.8 units along its largest isolines, giving it an aspect ratio of 1.18. The most any edge needs to shrink from the default spacing distance to its  $L^0$  value during the in-plane phase is 31.5%. To create the largest curvature found in the patch ( $\kappa = 0.0065$ ), an  $L^b$ , where  $b = \text{layers} - 1$ , needs to shrink an additional 26.0%. The grid is configured with an inter-layer spacing ( $\tau$ ) of 20 units, three layers, a max allowable material shrinkage (i.e. from  $L^0$  to  $L^b$ ) of 40%, and a long edge node count of 30. The short edge node count is set to 26 based on the aspect ratio, and the initial length of edges is set to 45.7 (corresponding to the largest  $L^0$  in the grid).

Following full grid optimization (which can be seen in (c)), on average  $L^0$  edges are 0.52% away from the measured distance on their associated locations on the target patch. This indicates close replication of isoline lengths as they were measured on the input patch. As an assessment of curvature replication,  $L^b$  edges are an average of 1.19% from their rest lengths. Since  $L^b$  and  $L^0$  directly correlated to  $\kappa$  via Eqs. (2.2) and (2.3), these low errors indicate close curvature replication as well. Angular springs deviate an average of  $7.06^\circ$  from their  $90^\circ$  rest angles, indicating the differential layer shrinking process created a large amount of out-of-plane deformation while the optimization software worked to balance the two groups of springs.

A triangle mesh result for a simplified human head [5] is shown in Fig. 3(d-f). The intersected region exhibits reversing curvature as well as high curvature around the nose and brow ridge. The region measures 15.98 units by 15.65 units along its largest isolines, giving it an aspect ratio of 1.02. The most

any edge needs to shrink from the default spacing distance to its  $L^0$  value is 20.4%. For this example, we chose a maximum shrinkage value of 50.0%. Given that the maximum curvature over the surface is  $\kappa = 1.35$ , our analysis indicates that the maximum value of  $\tau$  should be 0.23, which is the value employed for these results. Additional settings are a three layer grid, a long edge node count of 35, a short edge node count of 35 as well, and an initial edge length of 0.73.

Following full grid optimization (which can be seen in (f)), on average  $L^0$  edges are 0.11% from their rest lengths, and  $L^b$  edges are 0.34% off. Angular springs are off by an average of 7.55% from their rest angle of  $90^\circ$ . After color-coding deformation magnitude, it is notable that in (d) the bridge of the nose and brow ridge are clearly visible in red, as would be expected of these high curvature regions. Lighter red curves can be seen in the eye sockets where curvature reverses back in the other direction. In (f), the accurate recreation of the face's structure is visually evident.

### Conclusions:

We have described a novel technique for the inverse design and simulation of 4D printed objects consisting of multiple-tiered layers of deforming material. This technique takes the form of a grid made up of nodal subsections which differentially shrink and curve to match the shape of an input target surface. This modeling setup allows for a greater breadth of design choices than single-layer approaches, while also offering a better method for creating bi-directional curvature in 4D prints than previous approaches. Compared to other simulation work in thick 4D printing [6], our models have the potential to be produced by current FDM technology by printing each layer separately then assembling the grid before heating.

### Acknowledgments:

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2041772. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. We would like to thank the EPFL Computer Graphics and Geometry Laboratory for the *ShapeOp* library, as well as Jianzhe Gu and Lining Yao of CMU for technical assistance with utilizing the library.

### References:

- [1] Deuss, M.; Deleuran, A.H.; Bouaziz, S.; Deng, B.; Piker, D.; Pauly, M.: ShapeOp—A Robust and Extensible Geometric Modelling Paradigm. In M.R. Thomsen; M. Tamke; C. Gengnagel; B. Faircloth; F. Scheurer, eds., *Modelling Behaviour: Design Modelling Symposium 2015*, 505–515. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24208-8. [https://doi.org/10.1007/978-3-319-24208-8\\_42](https://doi.org/10.1007/978-3-319-24208-8_42).
- [2] Efrati, E.; Sharon, E.; Kupferman, R.: The metric description of elasticity in residually stressed soft materials. *Soft Matter*, 9, 8187–8197, 2013. <https://doi.org/10.1039/C3SM50660F>.
- [3] Gu, J.; Narayanan, V.; Wang, G.; Luo, D.; Jain, H.; Lu, K.; Qin, F.; Wang, S.; McCann, J.; Yao, L.: Inverse Design Tool for Asymmetrical Self-Rising Surfaces with Color Texture. *Symposium on Computational Fabrication*, 2020. <https://doi.org/10.1145/3424630.3425420>.
- [4] Momeni, F.; Hassani, N., S.M.M.; Liu, X.; Ni, J.: A review of 4D printing. *Materials & Design*, 122, 42 – 79, 2017. ISSN 0264-1275. <https://doi.org/10.1016/j.matdes.2017.02.068>.
- [5] printable\_models: GenericHead v2 3D Model. <https://free3d.com/3d-model/generichhead-v2--650180.html>. Accessed: 2022-03-28.
- [6] Sossou, G.; Demoly, F.; Belkebir, H.; Qi, H.J.; Gomes, S.; Montavon, G.: Design for 4D printing: Modeling and computation of smart materials distributions. *Materials & Design*, 181, 108074, 2019. ISSN 0264-1275. <https://doi.org/10.1016/j.matdes.2019.108074>.