

# Slice Coherence in a Query-Based Architecture for 3D Heterogeneous Printing

Ulas Yaman<sup>a,b,\*</sup>, Nabeel Butt<sup>a</sup>, Elisha Sacks<sup>a</sup>, Christoph Hoffmann<sup>a</sup>

<sup>a</sup>Purdue University, USA

<sup>b</sup>Middle East Technical University, Turkey

---

## Abstract

We report on 3D printing of artifacts with a structured, inhomogeneous interior. The interior is decomposed into cells defined by a 3D Voronoi diagram and their sites. When printing such objects, most slices the printer deposits are topologically the same and change only locally in the interior. The slicing algorithm capitalizes on this coherence and minimizes print head moves that do not deposit material. This approach has been implemented on a client/server architecture that computes the slices on the geometry side. The slices are printed by fused deposition, and are communicated upon demand.

**Keywords:** 3D printing, additive manufacturing, 3D Voronoi diagram, inhomogeneous volumes, client-server architecture, geometric queries

---

## 1. Introduction

Object properties and performance characteristics depend, to a large extent, on the structure of the object's interior. For instance, the strength of polycrystalline material cannot be determined without information about the shape and orientation of the interior cells, and their material composition. The outer boundary of the object normally determines the geometry of the object and reflects intended functionality, but the functionality must be supported by appropriate interior structure. In particular, the structure of the interior determines properties such as density, elasticity, stiffness, fatigue, current flow, sound absorption, thermal conductivity, and other salient properties of the material.

In order to analyze and study these properties in detail, many researchers model the interior structures of the objects as an aggregation of cells. A cell structure based on Voronoi tessellation is one of the most popular approaches in these studies, since it can generate realistic homogeneous and heterogeneous structures in both 2D and 3D. In fact, Voronoi tessellations can be considered to subsume many types of cell decomposition where the Voronoi sites are arranged in a special way. For instance, when the sites lie on the vertices of a regular grid we obtain a regular subdivision into blocks. In the following we assume a general position site arrangement unless otherwise noted.

We consider representations in which the volume is partitioned into cells where each cell has an interior with specific geometric and/or material specifications. For example, a particular cell might be stipulated to have

- a particular strength in a given direction,

- or a particular regular geometric structure of a certain characteristic,
- or be a specific, homogeneous material, and so on.

We investigate a manufacturing approach in which cells are separated by a membrane structure, and the interior is fabricated using filament deposition equipment or other fabrication techniques that proceed in the usual layer-by-layer methodology. We restrict to manufacturing the membrane separating the cells of the object partition. Specifically, we assume a cell geometry defined by a 3D Voronoi partition of the interior. The Voronoi partition is defined by a set of interior points, the *sites* of the partition. The sites may be specified explicitly as part of the design. Each such point has attributes that are a description of the cell interior. The attributes are sufficient to unambiguously specify cell interior. Examples include stipulated material, homogeneously filling the cell; variable density material where the density at a point  $x$  in the cell might be a function of the distance of the point from the cells site; empty cells; and so on. Cells are therefore convex polyhedra, except for cells that share part of the exterior boundary of the object. The representation is *implicit* in the sense that the cell specification is given by the sites along with the outer geometry of the object.

We will describe how to manufacture the cell membrane throughout the interior. Our algorithm is *incremental*, exploiting the fact that many successive slices are topologically the same, and that slices that differ topologically often differ only locally. Those differences are characterized by only a few cases. More than that, the algorithm seeks to deposit each membrane layer such that, at no time, the head requires repositioning without depositing material, unless the external boundary requires it.

The remainder of the paper is structured as follows. After a review of the literature is presented in the next section, the algorithm for our proposed manufacturing paradigm is given

---

\*Corresponding author

Email addresses: uyaman@purdue.edu, uyaman@metu.edu.tr (Ulas Yaman), butt@purdue.edu (Nabeel Butt), eps@purdue.edu (Elisha Sacks), cmh@purdue.edu (Christoph Hoffmann)

in the third section. The paradigm is evaluated with various experiments and the results are summarized in Section 4. The paper concludes with some remarks in Section 5.

## 2. Background

Voronoi based modeling is used in different fields of science and engineering for many applications. Material scientists employ Voronoi tessellations to model and simulate polycrystalline materials under different conditions to analyze their physical properties. In these simulations, each Voronoi cell corresponds to a crystalline cell in the aggregate and the cells form the basis of finite element models used to determine elastic and plastic behaviors of the so modeled materials [1].

Besides analyzing elastic and plastic properties, researchers also analyze sound absorption [2] and thermal insulation [3] properties of materials using Voronoi-based models. Furthermore, food scientists [4] model fruits to understand their quality and how to preserve them after harvesting. Here, the intercellular transportation of gases ( $O_2$  and  $CO_2$ ) and water is the most important aspect for the preservation of fruits [5]. Intercellular passages are modeled by cutting off some corners and edges of the Voronoi cells.

Current CAD/CAM software has been developed to serve conventional machining operations, such as milling and turning of homogeneous materials. Conventional CAD/CAM systems are therefore poor at designing objects with heterogeneous interior. Moreover, the subtractive methodology of milling and turning is unnatural for fabricating heterogeneity. When the design files are converted to the STL format, only the geometric information of the outer profile of the objects is stored. Although computer scientists already criticized the STL pipeline almost two decades ago [6], it remains a dominant format in 3D printing systems. When it is employed, the properties of the objects (material, color, etc.) defined in the corresponding CAD model are not transferred to the standard STL formats. Instead, users seeking to create parts with inhomogeneous interior are forced to edit their models using the software of the 3D printers, and must make do with the capabilities of the machine being used. In the case of FDM printers, such as the Makerbot, the printer software typically restricts the interior structure to a regular pattern, for instance hexagonal tubes, to be instantiated by a preselection of a few diameter choices. In order to overcome this problem in additive manufacturing, new standards for file systems are needed for assigning structural properties to the objects. Unfortunately these emerging standards are not used by the current commercial 3D printers. The STL format is still de-facto the dominant format in this field.

There have been various studies to fabricate artifacts where spatially varying colors and properties are defined in advance to perform the required functionality [7, 8, 9]. In these works, the interior of the objects are filled with the material that generates the color and functional patterns stipulated for the outer profile. Project Maxwell e.g. [10], is one of the earliest efforts to develop heterogeneous manufacturing paradigms. The project proposed to employ shape optimization techniques on the objects to be printed in order to realize functional properties via

geometrical representations. The problem with this approach is that the current commercial 3D printer technologies are not capable of printing the required interior structures.

Heterogeneous objects could be manufactured after sampling the interior using a voxelization as described in the studies [8, 9, 11]. Since these approaches are not compatible with current CAD software, researchers prefer developing their own application specific software using voxel-based modeling. The proposed 3D editing properties are not as powerful as commercial CAD software. So, the geometry of the objects is modeled in some CAD system and then exported into the research software. Properties are assigned in the research software. In one of these studies, Doubrovski et al. [9] proposed a layered manufacturing paradigm to overcome the limitations of the CAD/CAM tools. They proposed a bitmap-printing method to produce layers of different material composition and demonstrated their approach by printing a customized prosthetic joint socket. The test object incorporated pressure-sensing elements in the interior. Their approach did not compute planar object slices, and instead utilized the default voxel resolution of the 3D printer. They focused on realizing the required material properties through local material composition of the slices.

Another way to fabricate heterogeneous articles is to partition the model so that the generated STL files will represent interior structure. A recent study using this paradigm is done by Prevost et al. [12]. They can shift the center of gravity of the artifacts by designing the interior to consist of cells, some of which are empty, others filled with material. As a result, the fabricated objects can stand in unexpected poses. Depending on the cell configuration and composition, several STL files are sliced and some of the slices printed at the same time. In a different study using separate design files, Lu et al. [13] modeled the interior of the objects with Voronoi cells and hollowed the interior of these cells to increase the strength-to-weight of the objects. It is a common characteristic of these studies [8, 9, 12, 13] that they use resin based 3D printers.

In another remarkable study, Ge et al. [7] introduced the concept of printed active composites, a different type of heterogeneous structure. They utilize a CAD file representing the fiber architecture at the lamina and laminate level. With their approach, they were able to build a self-folding and opening box. They conjecture that their approach can print objects having spatially varying material properties.

Voronoi cells are commonly utilized in material sciences to model polycrystallines, molecular structures and porous structures because Voronoi cells model the internal geometry more realistically than other modeling approaches. With Voronoi-based models, where the smallest spatial elements are represented as single Voronoi cells, researchers try to analyze the behavior of the modeled structures under various physical conditions. For instance, Nygards and Gudmundson [14] modeled the geometry of dual-phase steel using Voronoi cells, and employed uniaxial tensile tests to determine the stress-strain behavior of the models. When compared with the experimental data, their models did well except near the end of the elastic region. In another similar study, Biner [15] utilized Voronoi cells to discretize the functionally graded materials and used this

model to comment on the thermo-elastic behavior of the materials at different temperatures. In another recent study, Katani et al. [16] similarly utilized Voronoi cells to model the microstructure of dual-phase steels. They constructed Voronoi cells from the images of a scanning electron microscope and used a damage model to gain insight into possible failure mechanisms of the materials under different loading conditions. Besides studies of materials using Voronoi cells, porous structures have also been modeled utilizing Voronoi diagrams; e.g., in the study of Kou et al. [17]. After constructing Voronoi tessellations, they merge some of the cells to form concave cells according to the requirements of the application. Furthermore, they use the vertices as the control points and generate B-Spline curves to delimit cells, thereby achieving a more natural appearance.

As we will prove below, there are specific topological changes that the cross section of a 3D Voronoi tessellation undergoes, as the cutting plane sweeps through it. These changes may trigger some events of the specific applications. There are two main types of transitions. The first one is about the appearance or vanishing of cells in the layout and the second one is about the local arrangements between the neighboring cells. In the second event, the number of the edges of the cells changes by a switching operation. At an intermediate position, of the sweep plane, in the case of the second event, there occurs a vertex having four edges in general. These changes also occur in random soap bubble structures [18]. Analyzing multicomponent metallic glass models, Gellatly and Finney [19] utilized *radical planes* to track the distortions in the local structures. Radical planes vary in their normal direction based on cell geometry. In contrast to using radical planes to observe the distortions, Li et al. [20] employed normal planes of the third dimension of the grain boundary network. As in the study of Li et al. [20], we will consider the topological transitions between the adjacent slices (normal to z-axis) of the object to be printed.

Bio-printing is another field of additive manufacturing where heterogeneity is necessary to fabricate personalized tissues and organs. In one of the recent studies in tissue engineering, Khoda and Koc [21] proposed a novel heterogeneous scaffold modeling approach for additive manufacturing processes. They achieve the given heterogeneity by varying the distance between the deposited filaments at the optimum angle in each layer. An important aspect of their approach is that they maintain continuity and connectivity of the extruded filaments. In another study, the main focus is on a medial surface representation of the organ/tissue obtained as a pruned Voronoi diagram of sites densely distributed across the organ surface; [22]. Chow et al. [23] developed a geometric modeling paradigm for porous structures utilizing 2D Voronoi diagrams, to be used especially in tissue engineering applications. They build 3D structures by perturbing the Voronoi sites, moving them along chosen trajectories. The trajectories are then discretized, and consecutive intermediate site positions are used to define layers stacked in the vertical direction. This generation paradigm naturally leads to a 3D printing strategy.

With the above-mentioned models, researchers can design and 3D print artifacts that have superior properties (higher thermal conductivity, better sound absorption, increased stiff-

ness, etc.) and customized geometric shapes as compared to conventional 3D printing. Although 3D printing is ideal for manufacturing solids with heterogeneous interior, the current manufacturer-provided 3D printing software is not capable of producing artifacts having heterogeneous cellular structures. In particular, there are no universal standards representing heterogeneity [17].

Voxel-based representations may be considered samplings of the object interior and are accurate to the density of the sampling [11]. When the density is increased, the number of voxels grows by the third power, thus imposing limits on scaling to high-density object designs. Variable density sampling reintroduces the problem of representing the volume regions in which a sampling has a particular density. Thus, voxel-based representations are of limited applicability and generality.

### 3. Algorithm

#### 3.1. Print Setup

In our experiments, we have decoupled the link between the CAD file and the printer, bypassing the proprietary software that creates the STL file. We generate the printer (machine) code directly and drive the printer from a Raspberry micro-computer that requests the G-code for the STL slices one-by-one, when the printer is ready for the next slice. This setup, although for this paper unnecessary, was chosen in anticipation of future experiments in which the dialogue between printer and CAD system is naturally bidirectional and seeks to explore in greater depth the query-based architecture proposed in [24]. The advantages argued there include latency hiding when client and server are communicating over large geographic distances; coping with very large STL files; as well as semantic integrity; to mention a few.

#### 3.2. Objectives

We distinguish printing head movements, in which material is extruded or solidified, from *fast* movements during which no material is extruded or solidified. In analogy to G-code terminology, we refer to the latter as *fast travel*.

The objectives of this work are two-fold:

1. Eliminate fast travel. This implies that material is to be deposited in a single sweep. A sweep consists of contiguous segments each of which deposits material.
2. Exploit the fact that the topology of consecutive slices changes in many cases only locally or not at all.

We achieve the first objective by constructing, for each slice, an Euler cycle that covers all edges of the slice, appealing to well-known graph-theoretical facts. The cycle traverses each edge of the slice twice without any fast travel.

The second objective is accomplished by restricting the interior structure of the print slice to a spatial Voronoi tessellation. Here, we must analyze how the transition of the slicing plane past a vertex of the 3D Voronoi tessellation changes the slices. By randomly perturbing the sites defining the Voronoi tessellation, by a small distance, we can serialize the change events and

avoid a raft of special cases that can lead to complicated topological changes and, moreover, would invite robustness issues familiar from computational geometry [25].

When printing an object, we restrict to printing the membranes enclosing each Voronoi cell, assuming that these separating surfaces are composed of the same material. As explained before, there are many possibilities for structuring the interior of the Voronoi cells, by specific materials, densities, grain direction, and so on. They are beyond the scope of this paper, however.

### 3.3. Slice Structure and Euler Cycle Determination

The input to the algorithm is a spatial Voronoi diagram and the object boundary. For simplicity we assume that the boundary is polyhedral. We assume that the Voronoi sites are in general position. This assumption can be enforced by randomly perturbing the sites prior to the Voronoi diagram construction. With the positive  $z$  axis the direction of slice deposition, we assume that every plane  $z = k$  intersects the part boundary in a number of disjoint simple polygons. This assumption is natural for 3D printing, and could be removed by adding update events for the exterior boundary, to the sweep algorithm below. Later we will remark further on topological changes in the boundary arcs of the slices. For now, we consider just one simple polygon as part boundary.

Given a connected graph  $G$ , an *Euler cycle* is a cycle of edges that includes every edge exactly once. It is well-known that an undirected graph has an Euler cycle if, and only if, every graph vertex is incident to an even number of edges. If the graph is directed, moreover, then the number of incoming edges should be the same as the number of outgoing edges, for every graph vertex.

Given a graph, a *simple cycle* is a cycle of edges  $(u_k, u_{k+1}), k = 0 \dots m$ , where  $u_m = u_0$  and the vertices  $u_0 \dots u_{m-1}$  are distinct. If the graph is directed, moreover, the cycle edges  $(u_k, u_{k+1})$  are used consistently with their direction unless noted otherwise.

The boundary polygon, and the interior edges of each slice are a planar graph with undirected edges. We replace every undirected graph edge with a pair of directed edges in opposite direction. This results in a graph whose every vertex is incident to an even number of edges, and where the number of incoming edges equals the number of outgoing edges, for each graph vertex. Therefore, the graph has an Euler cycle.

More specifically, since the slice graph is planar, it partitions the plane into cells bounded by simple cycles of directed edges. The outside of the object, moreover, forms one cell bounded entirely by boundary edges. We could lay out the cells one-by-one, each cell boundary printed completely before moving to the next cell. This strategy will generate fast travel as shown in Figure 1. Pick one cycle, cycle 1 for instance, and begin the layout at vertex  $u$ . When finished, we lay out cycle 2, beginning at  $u$  so as to avoid fast travel. This ends again at  $u$ , so we start cycle 3 also at  $u$ , ending again at  $u$ . At that point the head must execute fast travel, since no further cycles share vertex  $u$ .

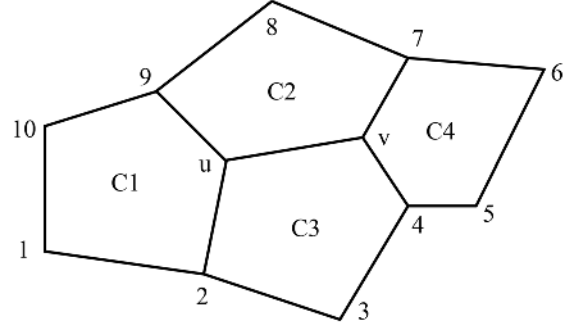


Figure 1: When cycle boundaries are printed one-by-one, the print head must do fast travel some of the time.

A better strategy is to print along an Euler cycle. One such cycle, also beginning at vertex  $u$ , is

$$u, 9, u, v, u, 2, 3, 2, 1, 10, 9, 8, 7, 6, 7, \\ v, 4, 5, 6, 5, 4, 3, 4, v, 7, 8, 9, 10, 1, 2, u$$

The planar slice at  $z = k$  intersects the part boundary and the Voronoi diagram in line segments, each a pair of directed edges as explained before. Each cycle is bounded by a simple loop of edges directed such that the cycle interior is to the left. The directed edge  $(u, v)$  is the *co-edge* of the directed edge  $(v, u)$  and vice-versa. The algorithm for determining an Euler cycle is a straightforward depth-first graph traversal. It recursively builds the Euler cycle  $L$ , a global data structure, and is called with arguments  $C$  and  $e$ , where  $e$  is a directed edge and  $C$  is the simple cycle of the graph bounded by  $e$ .

Initialize  $L$ , to an empty list  
Call Euler( $e, C$ ),  $e$  any edge,  
     $C$  the cycle bounded by  $e$

```
Euler(e,C):
01. mark e and C
02. append e to L
03. let e' be co-edge of e,
    C' the cycle bounded by e':
04. if C' is not marked:
05.   call Euler(e',C')
06.   goto step 08
07. otherwise, as C' is marked:
08.   let f be the next edge bounding C
09.   if f is not marked:
10.     set e to f and go to step 1
11.   else return
```

The three cases that are handled are summarized in Figure 2. If the adjacent cycle  $C'$  is not yet marked, Case (a), the recursion will process  $C'$  before continuing the traversal of  $C$ . If  $C'$  is marked but the edge  $f$  is not, Case (b), we continue to add edges of  $C$  to the Euler cycle  $L$ . Finally, if  $C'$  is marked and all edges of  $C$  have already been processed, Case (c), then the recursive call that initiated processing cycle  $C$  must have entered  $C$  with edge  $f$ , from cycle  $C'$ . In this case we return to cycle  $C'$  and continue to process it.

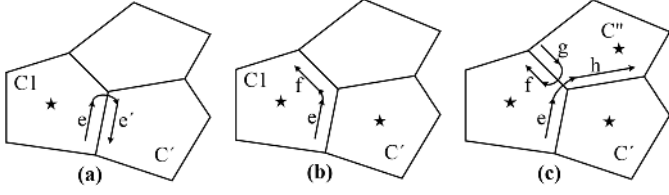


Figure 2: The three situations when computing an Euler cycle. The star indicates that a cell is marked.

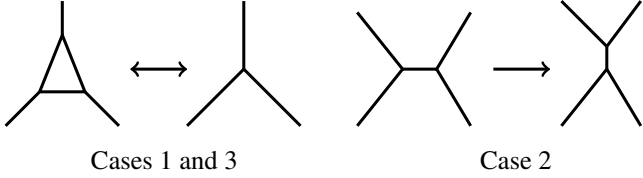


Figure 3: Graph update events.

Correctness of the algorithm is established by a routine induction on the number of cycles. Running time is proportional to the number of graph edges, since the graph was assumed connected. Moreover, the graph is planar, hence the number of edges is linear in the number of graph vertices. Finally, since sites are in general position, interior vertices are of valence 3.

### 3.4. Events

The  $z$ -coordinates of the Voronoi vertices divide the  $z$  axis into intervals. Absent changes in the boundary, the planes  $z = k$  with  $k$  in an interval generate isomorphic graphs. More than that, edge length and slice vertex positions vary linearly with  $k$ . When the plane crosses a Voronoi vertex, the graph changes in a simple manner that is governed by the incident 3D Voronoi edges. We now discuss these changes and how to exploit them.

At the initial position, the slicing plane intersects the part and is below the Voronoi vertices inside the part. The initial graph is obtained by intersecting the initial plane with the facets of the part and of the Voronoi diagram. Its simple cycles are computed with a standard algorithm. Edge  $e$  is linked to its co-edge  $e'$  and is also linked to the next edge in its simple cycle. We construct an Euler cycle via a depth-first traversal, as explained above.

We update the graph and the Euler cycle at the Voronoi vertices. By general position, the Voronoi vertices have distinct  $z$  values, hence no 3D Voronoi edges are perpendicular to the  $z$  axis, and each vertex has four incident edges in 3D-space. There are three cases (Figure 3):

1. If one Voronoi edge lies above the sweep plane, a triangle comprised of three pairs of (slice) graph edges collapses into a vertex.
2. If two Voronoi edges lie above the sweep plane, an oriented pair of graph edges collapses to a vertex and then expands into another pair of edges.
3. If three Voronoi edges lie above the plane, a graph vertex expands into three pairs of edges that form a triangle.

The graph is updated by removing the collapsed edges, adding the new edges, and placing them in clockwise order around the incident vertices.

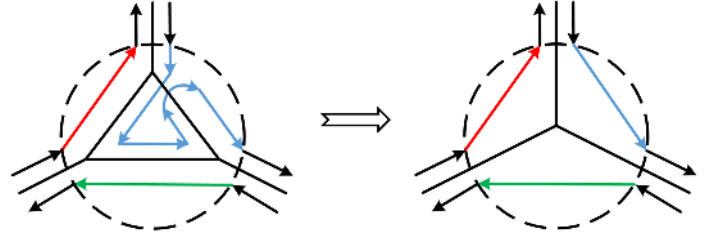


Figure 4: Cycle update for Case 1. Inverse direction shows one of three Case 3 configurations.

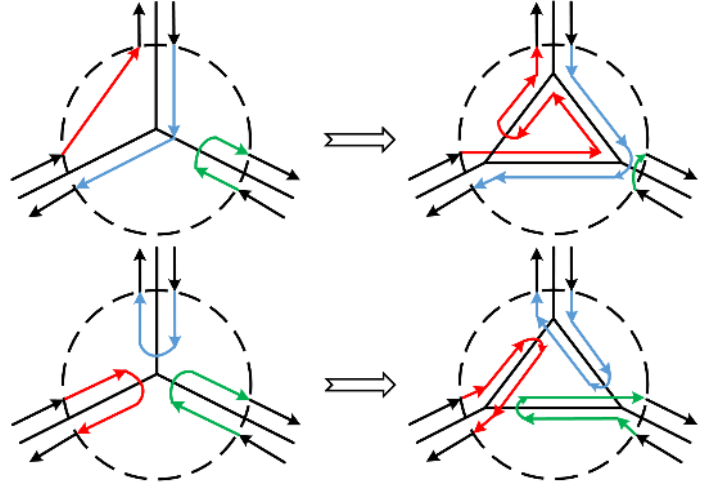


Figure 5: Case 3 cycle updates involving U-turns.

The updates to the Euler cycle are described next. They are complicated by the fact that the path fragments, at the change location, can be distant parts of the Euler cycle, yet must be updated in constant time. Moreover, because of prior events, the local topology of the Euler cycle need not correspond to the depth-first construction of the initial Euler cycle.

In Case 1, the Euler cycle is updated by dropping the edges that enclose the disappearing triangle. Then, the edges that were incident to the collapsing triangle are connected respecting the previous connectivity. See Figure 4. Note that there are several cases how the disappearing triangle connects the black directed edges. Some of them are seen in the illustration of Case 3. Nevertheless, connecting the black edges correctly is straightforward. Furthermore, since the connectivity is kept, the result must be again an Euler cycle. Finally, the connections can be done in constant time if the edge list of the Euler cycle is doubly linked and each directed graph edge is linked to its use in the Euler cycle. The details are routine.

Case 3 is the inverse of Case 1. In Case 3, the six directed edges incident to the vertex that expands into a triangle are connected in one of three ways: No U-turn, one U-turn and three U-turns, at the old edges. Figure 4 shows the case of no U-turn, when transforming the right configuration into the left. The other two cases are shown in Figure 5.

In case 2, the two edges that collapse and leave the graph are replaced by the two edges that enter the graph, as shown in Figure 3. The Euler cycle update proceeds as follows: The four

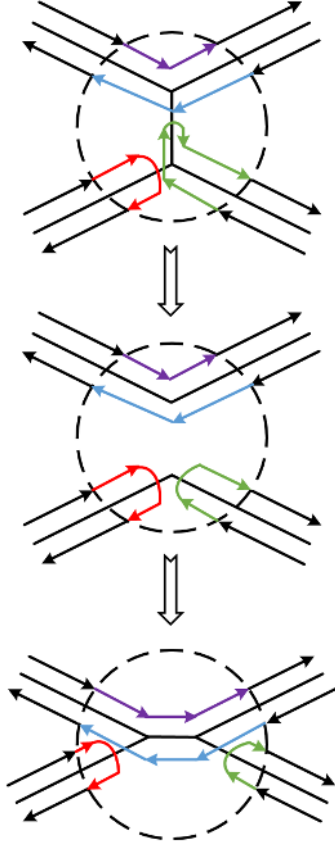


Figure 6: Case 2 update with intermediate step.

directed edges that do not disappear are connected as if they met in a common vertex. This involves retracting or dropping the vanishing edges in the Euler cycle. After the new pair of edges has been added into the graph, they are used in the new Euler cycle connecting the four edges. This may involve extending a U-turn. Figure 6 shows the 2-step method.

### 3.5. Boundary Cycles

When the slicing plane passes a vertex on the boundary the resulting events will have more cases and so will the number of incident graph edges. Here, we cannot directly perturb boundary vertices to reduce the number of cases, but we can employ the same idea that was used in segment intersection to serialize the state changes; e.g., [26]. Briefly, imagine the normal of the slicing plane tilted by an infinitesimal angle against the  $z$ -axis. The effect is that the event-triggering vertices intersect one-by-one, and so are serialized. Therefore, the cycles that need updating can be processed one-at-a-time.

We have analyzed a number of update situations that proceed much like the cases discussed above, but more edges may be involved and Voronoi cells may be divided. A typical case for boundary updates occurs when printing the base of the polycentric joint (Figure 19). Here, we pass from printing a ring-shaped cross section to three ring sectors. Note that there is also the option to recompute an Euler path if the topological change becomes too complicated.

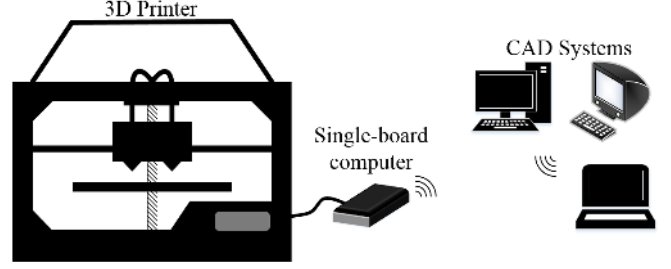


Figure 7: Experimental set-up.

## 4. Experimental evaluation of the manufacturing paradigm

We evaluate the performance of the proposed manufacturing paradigm for 3D printers using a MakerBot Replicator 2X printer. The experimental set-up is illustrated in Figure 7. The printer has two nozzle heads and is directly connected to a single-board computer (Raspberry Pi) through USB. The single-board computer operates as a server sending slice files to the printer upon request, the printer being the client for this interaction. In turn, the raspberry requests the slice files from the CAD system, wirelessly, and in this role the CAD system is server and the raspberry is client. The motion paths for the slices are generated by the CAD system and are conveyed to the raspberry, slice by slice or all at once, depending on the choice of the user.

During the slice by slice protocol, the motion paths are described with the native instructions of the printer driver embedded into the motherboard. When sending all slices at once, we utilize a post-processor to convert the nozzle head motions into a single G code file. This is done because the printer driver is not capable of handling multiple G code files for the same artifact. On the CAD side, we use two different software packages. The geometric modeling of the artifact shape is done in Rhinoceros 3D. The Voronoi tessellation (i.e., the interior structure) and the generation of paths are accomplished in Grasshopper, an add-on visual programming environment for Rhinoceros 3D. In addition to the built-in functional blocks of Grasshopper, we have supplementary Python scripts that implement the algorithms proposed in this paper and also communicate with the raspberry wirelessly. With this strategy, we do not need to convert the design files into STL files before printing: We slice the model and generate motion commands within the same CAD software as opposed to the conventional approaches in 3D printing. This eliminates the interoperability problems between CAD/CAM software. In the following subsections, the performance of the proposed manufacturing paradigm is evaluated from different perspectives.

### 4.1. Printing of a cube

Before printing complex artifacts, we first explore the paradigm on a cube with a size of 50 mm. There are 30 randomly distributed Voronoi sites in the cube. The schematic of the cube and its 3D print version are seen in Figure 8. Here we have divided the cube into 5 equal parts in order to assess the print quality of the interior faces. Among them, the yellow



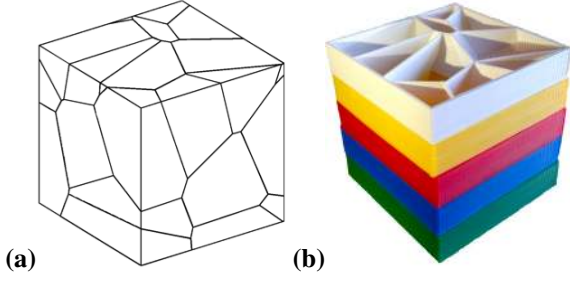


Figure 8: Cube with 30 Voronoi cells.

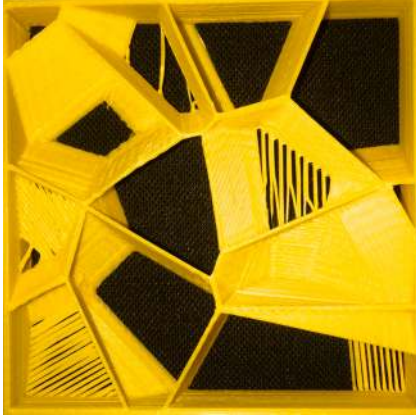


Figure 9: Top view of the yellow segment of the cube shown in Figure 8(b) with partially covered faces.

segment is the worst because some of the faces have shallow angles with the horizontal (slicing) plane. The top view of this yellow part is shown in Figure 9. The reason for this failure is that there are not enough beads to cover the faces at these angles. The lowest possible angle for the faces to get covered depends on the radius of the nozzle extruding the material and the resolution in  $z$  axis of the printer as illustrated in Figure 10. The angle is calculated as  $\alpha = \sin^{-1}(\Delta z/2r)$ . Considering our printer, the lowest angle is ( $\Delta z = 0.1 \text{ mm}$ ,  $r = 0.2 \text{ mm}$ )  $14.5^\circ$ . We assume here that the diameter of the extruded material is equal to the diameter of the nozzle, which gives the lowest achievable angle for the specified printer. However, this is not the case in practice. In order to decrease the fabrication time and print high-quality artifacts with higher resolutions, the nozzle heads move at a speed due to which the printed diameter of the extruded material is less than the diameter of the nozzle. That is, after the extruded material sticks to the table or to the previous layer, it is stretched and extruded at the same time and its diameter is decreased. The ratio between the distance traveled and the amount of the extruded material determines

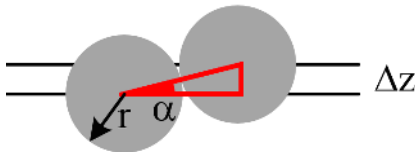


Figure 10: The lowest angle.

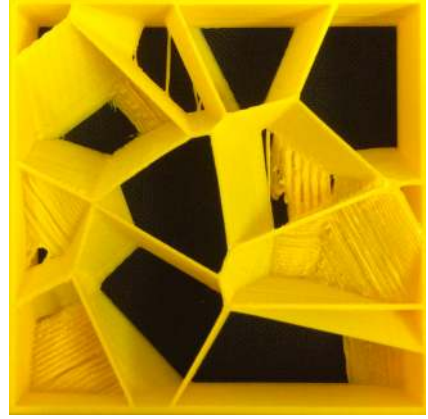


Figure 11: Extruding more material.

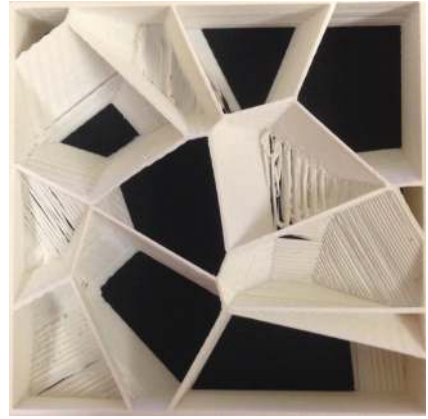


Figure 12: Covering horizontal faces.

the actual width of the edges. As a first attempt to cover these problematic faces, we decreased this ratio and printed the same segment of the cube as again; see Figure 11. As expected, the faces at angles with the horizontal are greater than the lower limit ( $14.5^\circ$ ) of the printer, are covered fully.

As a second experiment, we kept the ratio at its default value and covered the regions between the consecutive edges of the same face, so obtaining a ladder like structure. The same segment of the cube was printed using white filaments with this method and provided in Figure 12. This approach covers the faces better than the first one, but if the ladder beads are longer than approximately  $5 \text{ mm}$ , they will sag and the corresponding faces will not be planar.

We offer three additional ideas to solve this problem without adjusting the printer settings.

1. The first idea is simply to use a 3D printer that has a higher resolution in the  $z$ -direction. That is, the minimum layer height of the printer should be less than the current one, which is  $100 \mu\text{m}$ . In order to justify this solution, we printed the problematic part of the cube with a 3D printer having  $25 \mu\text{m}$  layer resolution as provided in Figure 13. As can be inferred from the figure, the limit angle, defined in the first paragraph of this section, decreased considerably (from  $14.5^\circ$  to  $3.6^\circ$ ), but the horizontal faces are not

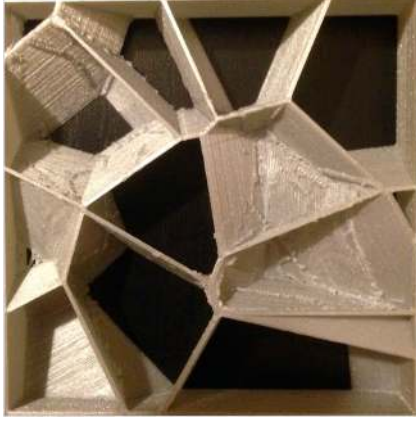


Figure 13: High resolution printer.

covered anyway.

2. The second idea is to stretch the cells along the  $z$  axis such that the slopes of the faces increase. We explored this idea on the same cube by multiplying the slopes with 4 and 2. The resulting quarter and half models of the cube are given in Figure 14. The quarter model on the left of the figure has no coverage problems, but one of the faces of the half model at the top left corner was not printed well. In order to have better results with this solution, the mean of the slopes of the faces should be considered before multiplying.
3. The last solution is to adjust the Voronoi sites such that the resulting 3D diagram will have faces with higher slopes. As an example, the same size of the cube is tessellated with 30 Voronoi sites in different ways and shown in Figure 15. The position of the Voronoi sites are determined randomly in **a)** and as can be seen there are many faces that are difficult to cover completely. In case **b)**, the  $z$  coordinates of the Voronoi sites are assigned to the same value resulting in an extruded 2D Voronoi diagram. This is similar to the conventional interior filling patterns of the current 3D printers, except that it is not uniform. If the application does not require heterogeneity in the  $z$  axis, then this would be an option in order to get higher quality artifacts. In case **c)**, the  $z$  coordinates of the Voronoi sites are restricted to be between  $22.5\text{ mm}$  and  $27.5\text{ mm}$ . As can be seen from the figure that the slopes of the faces of the Voronoi cells are now higher and more printable than the random tessellation approach.

One of the main objectives of this study is to eliminate the fast travels at each slice. The cube given in Figure 8 is used to assess elimination of fast travel. The size of the cube is  $50\text{ mm}$  and there are 30 Voronoi sites randomly distributed. We compare our method with two other methods and, for a fair comparison, we disregard the bottom and top covers of the cube which are both  $1\text{ mm}$  in thickness. As summarized in Table 1, when we use the official software of the printer (Makerbot Desktop) with 13 % infilling and  $0.2\text{ mm}$  layer thickness, there are exactly two separate motions that deposit material continuously, at each slice: one for the outer shells (2 of them) and the other

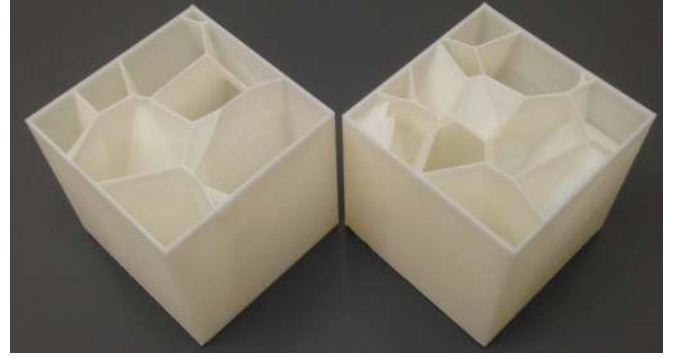


Figure 14: Slope modifications.

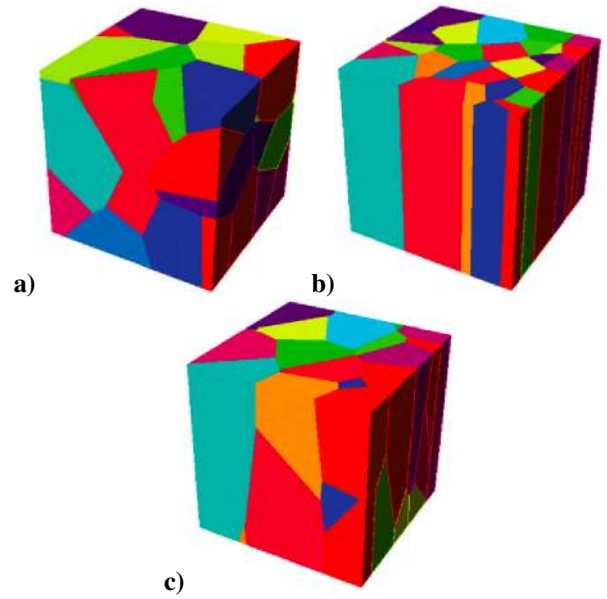


Figure 15: Voronoi tessellations: a) random b) constant  $z$  c) restricted  $z$ .

for the interior filling (hexagons). In order to maintain continuity while extruding hexagons, a full hexagon and 2 extra edges are extruded at the slice  $2n$  and the remaining 4 edges of the second hexagon are deposited at the next slice  $2n+1$ . The second method shown in the table is based on covering each Voronoi cycle one by one and later extruding the outer boundary in order to have all the edges covered twice. As expected, the number of motions with this method is much more than with the other two methods since there is fast travel between the many motions per slice. Although there are 30 cells in the cube, in each slice roughly 14 Voronoi cycles are covered in each slice on average, due to the random 3D tessellation. Table 1 includes in the average the additional motion needed to cover the boundary. The third method is the Euler approach proposed in the paper. It requires a single extruding motion per slice and requires no empty travel. Since the number of motions directly affects the printing time and the quality of the artifact, elimination of the fast travels was one of the objectives of this study. In the G-code files, we need to have 3 different commands between consecutive motions: extraction, fast travel, and restart. Thus, elimination of the fast travels indirectly results in decreased G-code file sizes,



Table 1: Comparison of the number of continuously extruding motions when printing the cube.

Method	Total Number of Motions	Average Number of Motions / Slice
Printer software	480	2
Naive Voronoi	3643	15.18
Euler approach	240	1



Figure 16: Inner and outer paths printed with different colors.

an added benefit.

#### 4.2. Separation of the inner and outer paths

One of the aspects of our paradigm is that the generated continuous Euler loops can easily be divided into inner and outer paths. Depending on the application and the capabilities of the printer, these paths can be utilized to print artifacts having different materials or colors at the interior and at the outer boundaries. We implemented this attribution of the paradigm with our two nozzle headed 3D printer. Black and white filaments were used while printing the sample artifact as shown in Figure 16. There are black scratches on the white faces of the sample due to the difficulty in adjusting the retraction heights in dual printing. If a 3D printer capable of extruding different colors from the same extruder is used, these marks would be absent.

#### 4.3. Interior filling of cells

Another important attribution of the proposed manufacturing paradigm is that the interior of the Voronoi cells can be filled with different patterns using various materials or colors depending on the capability of the printer. The motion paths of the interiors may be merged with the main loop of the slice if they are of the same material/color or they may be generated separately. This aspect of the paradigm was realized on a sample box shape using white and green filaments as shown in Figure 17. In this example, the offsets of the cycles are used to generate the interior pattern. The resulting offsets are then merged together into a spiral for the corresponding nozzle head. Due to the same problem mentioned in the previous subsection, there are undesirable white and green smudges on the filled region. Instead of offsets, other patterns can be used to fill the interior



Figure 17: One of the interior cells filled with offsets.

of the cells. For instance, the beads may be aligned in the directions of the applied forces on the artifacts, so increasing the strength.

#### 4.4. Test cases

We fabricated two different parts with structured interior to illustrate a few functionalities of the proposed paradigm. In the first case, the moving  $P$  of Purdue University is printed with the conventional and the proposed approaches. These printed artifacts are compared in terms of their self-standing ability, print time and material consumption. As a second test case, we printed the stationary part of a three ball polycentric joint [27]. We tessellated Voronoi sites according to the stress distribution in this example.

##### 4.4.1. The $P$

After we designed the  $P$  in Rhinoceros 3D, it was first fabricated using the official software of the printer which requires STL files of the designed artifacts. The design and the printed object are shown on the upper row of Figure 18. Owing to the fact that the interior is filled with hexagons homogeneously, the center of gravity of the artifact is not shifted and it is to the right of the base of the  $P$ . Thus, it is not possible for the  $P$  to stand on its own. In order to make it stand, we increased the density of the Voronoi sites on the left of the object, as shown in the bottom row of the figure. For the tessellation, we followed the framework explained in the appendix. Due to the higher density of the Voronoi edges, more material was extruded on the left section of the  $P$  which shifted the center of gravity. In addition to the functional disadvantage, the  $P$  with a homogeneous interior (13% infill) required more material than the heterogeneous  $P$  and required 38% more time to print according to homogeneous printing with 0.1 mm layer height; see Table 2. We compare the time with the first homogeneous case because we use the same extrusion ratio (extruded material per unit motion, mm/mm) with the high resolution case and we print each edge twice. Note also that the material consumption in the homogeneous cases is slightly different. Theoretically it should be the same since in both cases the volume to be filled is exactly the

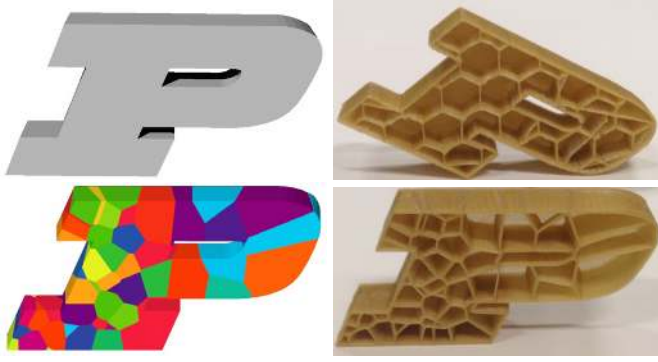


Figure 18: Homogeneous and heterogeneous fillings of the  $P$ . Note the diagonal line extruded in the homogeneous  $P$ , at the upper right. It is due to a bug in the software that comes with the MakerBot.

Table 2: Comparison of the prints of the  $P$ .

Interior	Layer Height	Material	Print Time
Homogeneous	0.1 mm	13.77 g	105'
Homogeneous	0.2 mm	13.01 g	51'
Heterogeneous	0.2 mm	7.05 g	76'

same. This difference is due to the fact that the interior filling patterns of the two cases are not identical.

#### 4.4.2. Three ball polycentric joint

As a second test case, we printed the stationary part of a three ball polycentric joint [27]. After modeling the base part boundary, we distributed the Voronoi sites inside the volume according to the high stress regions pointed out by Alaci et al. [27]. That is, there are more sites at these regions in order to increase the strength of the part. As a part of the design, these Voronoi sites may also be placed such that the manufactured part will have desired directional properties. The sites are located around the mean height of the part, as discussed before in Section 4.1. The designed model, Voronoi tessellation, printed part and the functional joint are shown in Figure 19. We printed the moving part of the joint using a resin based 3D printer along with its conventional fabrication pipeline. Our approach consumed less material and printed in a shorter time as shown in Table 3.

Note that the outer shape of the object is not constant along the  $z$  axis. Three slots are designed for the balls on the moving part of the joint. These topological changes constitute a discontinuity that is addressed by trimming the cells that extend into the slots. It seems simplest to just restart the Euler loops above

Table 3: Comparison of the prints of the joint.

Interior	Layer Height	Material	Print Time
Homogeneous	0.1 mm	74.21 g	568'
Homogeneous	0.2 mm	72.12 g	281'
Heterogeneous	0.2 mm	22.84 g	243'

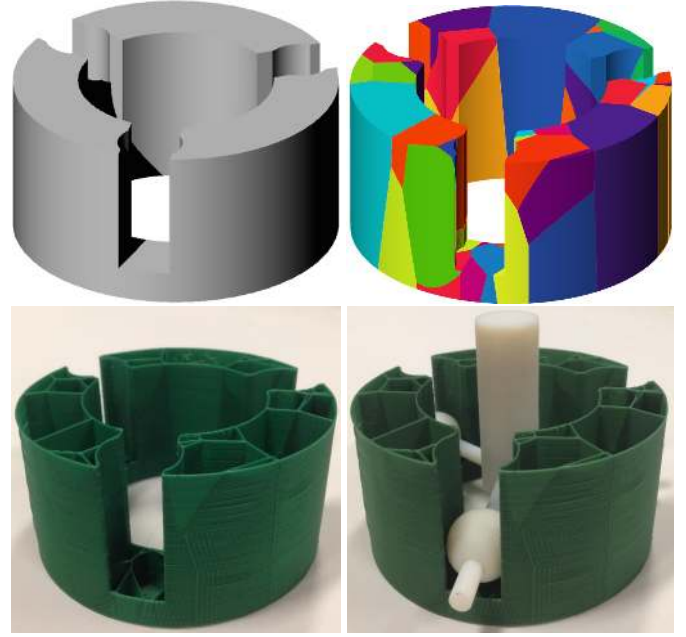


Figure 19: Design and fabrication of the three ball polycentric joint.

Table 4: Extruded material (mg) per motion for different approaches.

Case	Conventional	Voronoi
The $P$	57.4	117.5
The Joint	17.7	35.1

the transition to the slots, but this ignores the fact that sites below impact the cell boundaries above in the continuing interior. Therefore, the lower boundary of each of the three slots, and subsequently the side walls of the slots, contribute to the part boundary and thus change the Euler path at the incipience of the slots. The pruning of the Voronoi cells that must be done is analogous to the pruning that was done below. Topologically, printing proceeds from one to three boundary loops. The general case is a transition from  $m$  to  $n$  boundary loops and requires further research. This issue needs further investigation to better understand how best to transition from  $n$  to  $m$  boundary loops.

We further compared our Voronoi based approach with the conventional homogeneous printing approach in terms of the amount of extruded material per continuous motion. These ratios are presented in Table 4. While calculating them, we assumed that bottom and top surfaces have the same number of motions as the intermediate layers. As expected, our approach extrudes more material per motion even though it has less material consumption overall. This results in more uniform beads, since the continuity of the extrusion process is maintained, and less surface irregularities.

#### 4.4.3. Further cases

In addition to the previous two test cases, we have printed several other objects with our paradigm as shown in Figure



Figure 20: Various artifacts

20. The two examples in the top row show that the proposed manufacturing paradigm is suitable for the construction of difficult structures. With the Voronoi based modelling, mechanical properties of the bricks in the construction could be altered and the multi-directional load bearing capacities could be arranged. Furthermore, as in the case of the keystone in the arch, the interior of some the cells/bricks could be filled with materials that expand over time to create compressive forces on the neighbouring cells/bricks. The examples at the bottom row of Figure 20 are provided to show that the approach has no restriction on the complexity of the boundary faces. The problem we have with the horizontal faces of the cells are also valid with the horizontal boundary faces. In these situations, supporting walls can be used for proper printing as in the conventional approach with FDM.

## 5. Discussion

The traditional approach of treating each slice as a separate structure ignores the fact that, in many cases, there is a high degree of similarity between consecutive slices, geometrically as well as topologically. This is trivially the case when the interior of the object printed is filled with a fixed pattern that has little functional significance. A structure derived from a 3D Voronoi diagram, on the other hand, has functional significance, as evidenced by the extensive literature on the subject. Therefore, the inter-slice coherence and the small number of possible topological changes is significant. Furthermore, the query-based client-server architecture favors incremental slice transmission that increases scalability and can support compact slice encodings.

When printing empty Voronoi cells, faces with a low slope present difficulties. We developed several ways to ameliorate

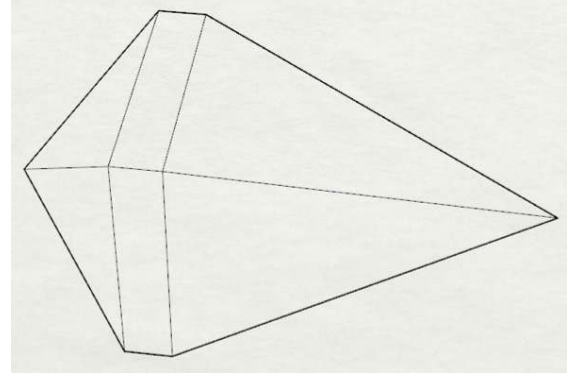


Figure 21: Perspective view of the artifact.

this problem. They deserve to be explored further in application contexts. In certain application contexts Voronoi diagrams are a starting point for modeling a cellular structure. The Voronoi cells are then modified, resulting in the final cell shape. Scaling might be an elegant tool in such domains. Non-empty cells, on the other hand, do not have the difficulty of printing faces with shallow slope. Nevertheless, how to print the cell interior efficiently deserves additional consideration of cell content coherence, density and structure.

## 6. Acknowledgments

The project has been partially supported by the National Science Foundation, CMMI-1361783.

## Appendix. A framework for Voronoi tessellation

We consider shifting the center of gravity (CG) of a prototyped part, using irregular 3D Voronoi tessellation. Determining where to place sites so that the design requirements are satisfied would be tedious if not forbidding. In this example application, the center of gravity of the artifact will be modified by developing an irregular Voronoi tessellation of the interior. Similar approaches can also be employed for different physical properties of the artifacts as mentioned in the introduction and literature review.

The design approach will be elaborated through an example artifact shown in Figure 21. It consists of two pyramids and one box sections. If the object is filled homogeneously, it is obvious that the CG will be to the right of the box section. The aim of the interior design is that the artifact should stand on its own on one of the rectangular faces of the box section. This is done by relocating the CG into the box region.

Before determining the heterogeneous tessellation, we do a regular tessellation (hexagonal) as in Figures 22 - 24 and get the weights of the sites according to the corresponding physical properties. This approach will help us evaluate the heterogeneous Voronoi tessellation without calculating the Voronoi diagram each time.

The weights of the sites will depend on the surface areas of the cells. These weights are shown as a temperature plot in Figure 25. As can be inferred, the weights are higher at the brown

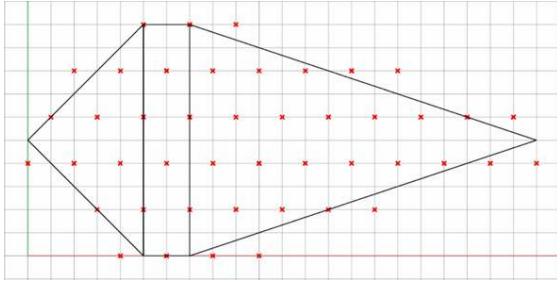


Figure 22: Sites of the regular Voronoi tessellation.

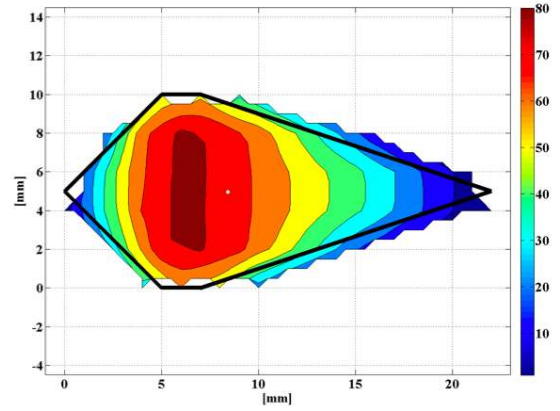


Figure 25: Temperature plot of the weights.

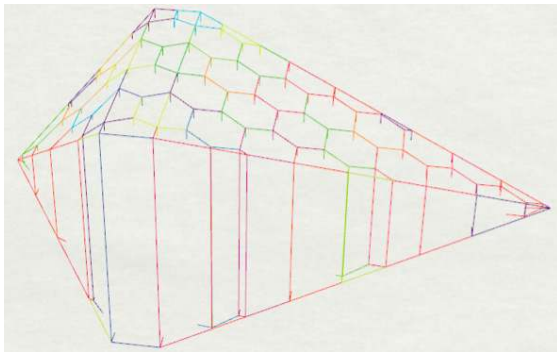


Figure 23: Perspective view of the regular tessellation.

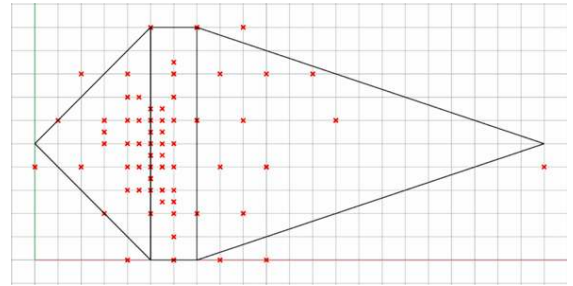


Figure 26: Voronoi tessellation based on the temperature plot.

and red regions; that is, sites in these regions are dominant in the calculation of the CG. We will make use of these regions while tessellating the Voronoi sites to satisfy the design goal. A sample irregular tessellation is provided in Figure 26. It is obtained by adding sites to the high-temperature regions with a regular pattern that allows fast estimation of the resulting shifted position of the CG. This step is repeated until the design goal is reached. Another way to effect site relocation would be to let the user click with the mouse, moving sites from where they are to where they should be. These activities can be guided in detail by updating the temperature plot periodically. A rich case history can be so assembled that would become a knowledge base that allows automating the process in time.

As stated earlier, we do not need to calculate the Voronoi diagram to determine exactly the weights of the sites. Instead, we use the weights obtained from the regular tessellation, thereby decreasing the computation time. If this configuration satisfies the design requirements, then the 2D and 3D Voronoi diagrams can be computed as in Figures 27 - 28. The 3D diagram is obtained by perturbing the z-axis values of the Voronoi sites as discussed in the paper. The method works very well empirically and is validated by an exact calculation of the CG position. The designed artifact is also printed, Figure 29, to further prove that the CG is within in the box region. Since it is almost symmetrical with respect to the mid-plane, only half of it is printed to show the interior.

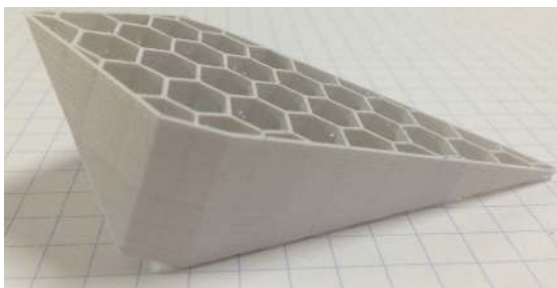


Figure 24: Half of the printed artifact with regular tessellation.



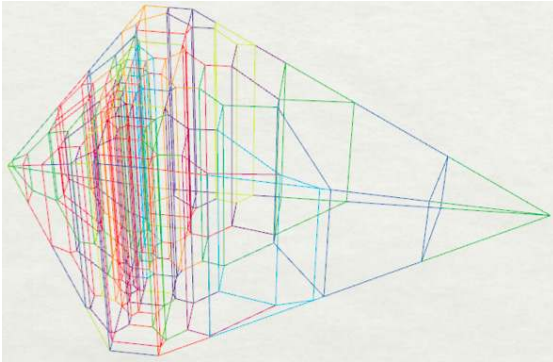


Figure 27: Resulting extruded 2D Voronoi diagram.

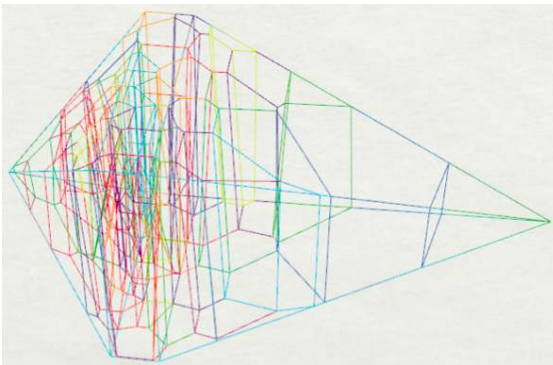


Figure 28: Resulting 3D Voronoi diagram.

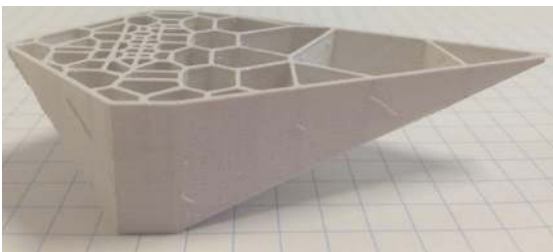


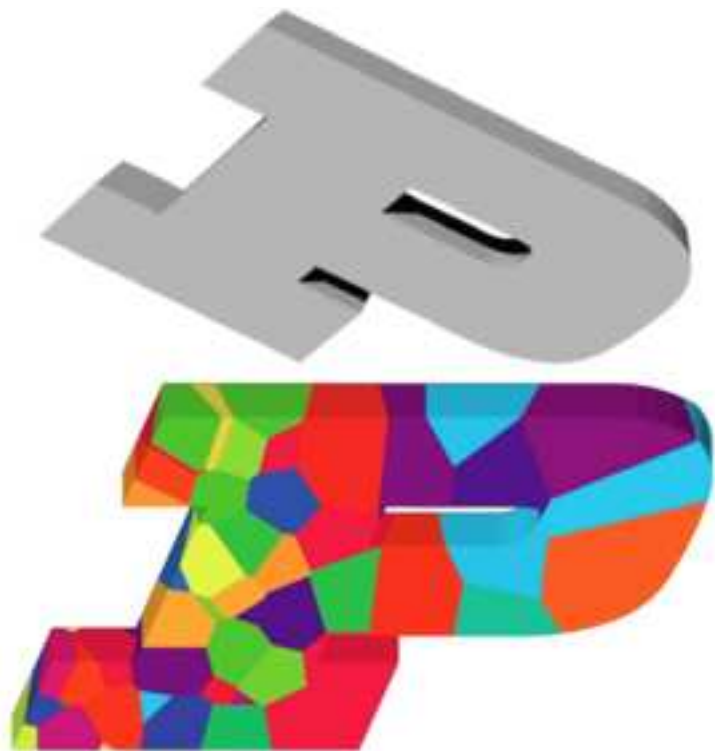
Figure 29: Half of the printed artifact with 3D Voronoi tessellation.

## References

- [1] S. Ghosh, K. Lee, S. Moorthy, Two scale analysis of heterogeneous elastic-plastic materials with asymptotic homogenization and voronoi cell finite element model, *Computer Methods in Applied Mechanics and Engineering* 132 (1) (1996) 63–116.
- [2] L. J. M. Jacobs, K. C. H. Danen, M. F. Kemmere, J. T. F. Keurentjes, Quantitative morphology analysis of polymers foamed with supercritical carbon dioxide using Voronoi diagrams, *Computational Materials Science* 38 (4) (2007) 751–758. doi:10.1016/j.commatsci.2006.05.009.
- [3] M. Y. Ma, H. Ye, An image analysis method to obtain the effective thermal conductivity of metallic foams via a redefined concept of shape factor, *Applied Thermal Engineering* 73 (1) (2014) 1277–1282. doi:10.1016/j.applthermaleng.2014.08.064.
- [4] H. K. Mebatsion, P. Verboven, Q. T. Ho, B. E. Verlinden, B. M. Nicolai, Modelling fruit (micro)structures, why and how?, *Trends in Food Science & Technology* 19 (2) (2008) 59–66. doi:10.1016/j.tifs.2007.10.003.
- [5] M. K. Abera, P. Verboven, E. Herremans, T. Defraeye, S. W. Fanta, Q. T. Ho, J. Carmeliet, B. M. Nicolai, 3D Virtual Pome Fruit Tissue Generation Based on Cell Growth Modeling, *Food and Bioprocess Technology* 7 (2) (2013) 542–555. doi:10.1007/s11947-013-1127-3.
- [6] A. Dolenc, I. Mäkelä, Rapid prototyping from a computer scientist's point of view, *Rapid Prototyping Journal* 2 (2) (1996) 18–25. doi:10.1108/13552549610128198.
- [7] Q. Ge, H. J. Qi, M. L. Dunn, Active materials by four-dimension printing, *Applied Physics Letters* 103 (13) (2013) 131901. doi:10.1063/1.4819837.
- [8] N. Oxman, Variable property rapid prototyping, *Virtual and Physical Prototyping* 6 (1) (2011) 3–31. doi:10.1080/17452759.2011.558588.
- [9] E. L. Doubrovski, E. Y. Tsai, D. Dikovskiy, J. M. P. Geraedts, H. Herr, N. Oxman, Voxel-based fabrication through material property mapping: A design method for bitmap printing, *Computer-Aided Design* (2014) 1–11doi:10.1016/j.cad.2014.05.010.
- [10] D. Dutta, N. Kikuchi, P. Papalmbros, F. Prinz, L. Weiss, Project maxwell: Towards rapid realization of superior products, in: *Solid Freeform Fabrication Symposium, DTIC Document*, 1992, pp. 54–62.
- [11] V. Chandru, S. Manohar, C. E. Prakash, Voxel-based modeling for layered manufacturing, *Computer Graphics and Applications, IEEE* 15 (6) (1995) 42–47.
- [12] R. Prévost, E. Whiting, S. Lefebvre, O. Sorkine-Hornung, Make it stand, *ACM Transactions on Graphics* 32 (4) (2013) 1. doi:10.1145/2461912.2461957.
- [13] L. Lu, B. Chen, A. Sharf, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D. Cohen-Or, Build-to-last, *ACM Transactions on Graphics* 33 (4) (2014) 1–10. doi:10.1145/2601097.2601168.
- [14] M. Nygård, P. Gudmundson, Three-dimensional periodic voronoi grain models and micromechanical fe-simulations of a two-phase steel, *Computational Materials Science* 24 (4) (2002) 513–519.
- [15] S. Biner, Thermo-elastic analysis of functionally graded materials using voronoi elements, *Materials Science and Engineering: A* 315 (1) (2001) 136–146.
- [16] S. Katani, S. Ziaei-Rad, N. Nouri, N. Saeidi, J. Kadkhodapour, N. Torabian, S. Schmauder, Microstructure Modelling of Dual-Phase Steel Using SEM Micrographs and Voronoi Polycrystal Models, *Metallography, Microstructure, and Analysis* 2 (3) (2013) 156–169. doi:10.1007/s13632-013-0075-7.
- [17] X. Y. Kou, S. T. Tan, Heterogeneous object modeling: A review, *Computer-Aided Design* 39 (4) (2007) 284–301. doi:10.1016/j.cad.2006.12.007.
- [18] D. Weaire, N. Rivier, Soap, cells and statistics—random patterns in two dimensions, *Contemporary Physics* 25 (1) (1984) 59–99. doi:10.1080/00107518408210979.
- [19] B. Gellatly, J. Finney, Characterisation of models of multicomponent amorphous metals: the radical alternative to the voronoi polyhedron, *Journal of Non-Crystalline Solids* 50 (3) (1982) 313–329.
- [20] S. F. Li, J. K. Mason, J. Lind, M. Kumar, Quadruple nodes and grain boundary connectivity in three dimensions, *Acta Materialia* 64 (C) (2014) 220–230. doi:10.1016/j.actamat.2013.10.033.
- [21] A. K. M. B. Khoda, B. Koc, Functionally heterogeneous porous scaffold design for tissue engineering, *Computer-Aided Design* 45 (11) (2013) 1276–1293. doi:10.1016/j.cad.2013.05.005.



- [22] M. Näf, G. Székely, R. Kikinis, M. E. Shenton, O. Kübler, 3d voronoi skeletons and their usage for the characterization and recognition of 3d organ shape, *Computer Vision and Image Understanding* 66 (2) (1997) 147–161.
- [23] H. N. Chow, S. T. Tan, W. S. Sze, Layered Modeling of Porous Structures with Voronoi Diagrams, *Computer-Aided Design and Applications* 4 (1-4) (2007) 321–330. doi:10.1080/16864360.2007.10738552.
- [24] C. Hoffmann, V. Shapiro, V. Srinivasan, Geometric interoperability via queries, *Computer-Aided Design* 46 (2014) 148–159. doi:10.1016/j.cad.2013.08.027.
- [25] C. M. Hoffmann, Robustness in geometric computations, *Journal of Computing and Information Science in Engineering* 1 (2) (2001) 143–155.
- [26] M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd edition, Chapter 2.
- [27] S. Alaci, F. Ciornei, C. Filote, Considerations upon a new tripod joint solution, *Mechanics* 19 (5) (2013) 567–574. doi:10.5755/j01.mech.19.5.5539.



Homogeneous

vs.

Heterogeneous