2024

External Support Structures in Fused Deposition Modeling 3D Printing

Adam Mystkowski
amystkowski@gardner-webb.edu

External Support Structures in Fused Deposition Modeling 3D Printing

An Honors Thesis
Presented to
The University Honors Program
Gardner-Webb University
31 March 2024

By

Adam Mystkowski

Accepted by the Honors Faculty

Dr. Miroslaw Mystkowski, Thesis Advisor	Dr. Wilson Hawkins, Honors Director
Dr. Robert Bass, Honors Committee	Dr. Elizabeth Amato, Honors Committee
Dr. Angelina Smith, Honors Committee	Dr. Abby Garlock, Honors Committee

Table of Contents

Chapter 1: Introduction

1.1 3D Printing

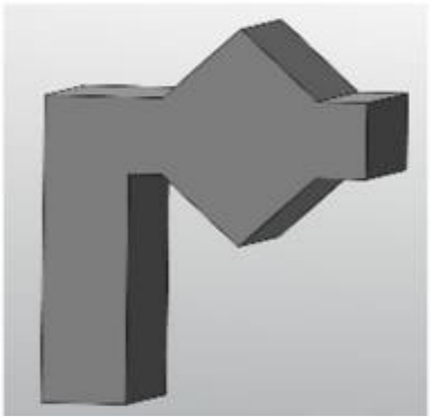
3D Printing, sometimes also referred to as Additive Manufacturing, is a technology that has garnered a lot of attention in the past several years. The concept behind 3D Printing is simple. The name calls back to traditional printers, which print text or images on flat surfaces. The output of these printers can be described as 2D, as it can be described in two axes of space. 3D Printing adds a third axis, allowing it to print objects that "pop out" from the flat surface and occupy space in all three dimensions. While the technology has attracted a significant community of hobbyists, the benefits of the technology have also been recognized in manufacturing. 3D Printing has been a subject of interest in rapid prototyping, as it could print complex shapes at a rate faster than other manufacturing methods.

1.2 FDM Printing

While there are many different types of 3D Printing techniques, the most common type is Fused Deposition Modeling (FDM), mainly due to its relatively low cost compared to other types. FDM printers print objects on a print bed by extruding a melted plastic filament in layers. Before a 3D model is printed it is first processed by a piece of software called a slicer. A slicer, fittingly, slices a 3D model into a series of thin horizontal layers ordered from bottom to top. An FDM printer then traces each of these layers one-by-one, with each consecutive layer above the last. The layers are traced by a movable nozzle. When a layer is finished, either the nozzle shifts slightly up or the print bed shifts slightly down to begin the next layer. As the nozzle traces layers, it extrudes a melted plastic that is formed by gradually melting a plastic filament that is attached to the nozzle from a spool.

1.3 Overhangs

While FDM printers and other similar types are very promising, they also present a significant limitation: they tend to struggle with models that have overhanging parts. When a layer is extruded by an FDM printer, it will only stay in place if there is something for it to rest on from below. Overwise, it will fall onto the print bed due to gravity. In 3D Printing, parts of 3D objects where material isn’t supported from below are referred to as overhangs. Unfortunately, it is not uncommon for 3D models to have one or several overhangs. The most obvious examples of overhangs are features that clearly stick out downwards from an object, such as the arms on a humanoid figure. An example of a simple model that presents this problem is shown in Fig.1(a). Notice the hanging diamond-like shape on the right side of the model. If the model were to be printed as is, the part wouldn't be able to print properly. These sorts of features may be referred to as local minima. However, local minima are not the only type of overhang; upward inclines can also be problematic. To an extent, FDM printers can handle inclines. This depends on the degree of steepness of the incline. The threshold varies from printer-to-printer, but it most often is around 45 degrees. In cases where an incline exceeds this threshold, each consecutive layer tends to sag over the previous layer, ruining the surface quality of the model. The image in Fig.1(b) demonstrates how a printer’s performance is affected by the steepness of inclines. The following shape was oriented upside-down when it was being printed. Notice that the surface quality decreases as the slope gets steeper.



(a) (b)

Figure 1. (a) An example of a simple shape that would be impossible to print without additional support [11]. (b) A 3D printed model showing a printer's ability to print inclines of varying steepness [3]

1.4 External Support Structures

In a lot of cases, the most feasible way to solve the problem of overhangs is through printing external support structures. These are additional structures that are printed along with a model for the purpose of providing support for overhangs. These structures are removed from the model once it is finished printing. The image in Fig.2 presents an example of external support structures. The model is the same as the one shown previously, but with extra structures added so that it could be printed successfully. The process of designing and adding support structures to 3D models can be very time consuming, so many developers have attempted to write programs that generate these structures automatically. The process of generating external support structures presents several challenges, attempting to create supports that are stable, but also use the least material and take the least time to print. The purpose of this paper is to present the challenges that come with designing these programs, and the methods used in the programs to overcome those challenges. External support structures should not be confused with internal support structures. These structures support a model from the inside, instead of the outside. While they also present an interesting challenge, and are generated using similar methods, they are outside the scope of this paper's topic.

Figure 2. The same shape shown in Fig.1(a), but printed with support structures [11].

Chapter 2: Challenges

Designing external support structures introduces many challenges. There are many different variables which have to be considered, and a change to one may affect others. In some cases, variables may be inversely proportional to each other, so a balance has to be found between them. The following few paragraphs will go over several of these variables.

2.1 Stability

One variable that is considered is stability. It describes how effectively the supports support the geometry of the model, and how resistant the supports themselves are to sagging or falling over. Stability is desired in supports, because if they cannot support model, then they are, by definition, useless. Supports have to be strong enough to support the weight imbalance of the subparts being printed above it [4]. Hence, stability is always an important factor taken into consideration when designing support-generation algorithms. Besides the model geometry, support design may also consider weight imbalance and the friction caused by the extrusion nozzle as it brushes against already-printed layers [4]. Stability cannot really be measured, but it can be deduced from a printed result. When a model comes out as intended, the supports could be considered stable. Unstable designs instead may result in poorly printed supports or parts of the model, which often tend to resemble stringy messes such as the one shown in Fig.3. Oftentimes, many of the other variables that are considered when designing support-generation algorithms tend to be inversely proportional to stability, so it is often treated more of as a threshold to reach rather than as a quality to maximize.

Figure 3. An example of a failed print [8].

2.2 Material Usage

Another variable that is considered is the amount of material used. When a model is printed with external supports, additional filament is needed to print the supports in addition to the model. This additional material is considered waste, as it is not reusable and is discarded after printing [12]. Therefore, external support structure-generating algorithms aim to use as little material as possible. This variable can be measured in several different ways, such as volume [10][12], weight [3][5][6], and filament length [4][8]. Material usage can be optimized in multiple ways. One way is by generating supports with geometries that are less dense and thinner. Another way is by putting the base of supports on the model geometry instead of the print bed if it results in a shorter distance [5]. Better material usage usually tends to be inversely proportional to stability, so support-generating algorithms attempt to find a good balance between the two variables. The image in Fig.4 shows three copies of the same model printed with different types of supports. The supports were removed and placed in a pile next to their respective models. This shows how different generation strategies can affect material usage.

Figure 4. Shown are three copies of the same model, each printed with different types of supports. Supports are removed and piled up next to their respective model [8].

2.3 Time

A third variable that is often considered is time. Time optimization is desirable in multiple stages of designing and printing 3D models. Time is relevant in two areas which have to do with support structures: support generation and printing. In relation to support generation, time describes how long it takes for a support-generating algorithm to generate a complete set of supports for a model. High algorithm execution efficiency is a major concern in support-generation algorithms [8], so developers of said algorithms seek methods that yield a low generation time. In some cases, an algorithm may be designed to generate suboptimal supports in order to improve processing time. As a result, processing time may be inversely proportional to material usage and stability depending on the algorithm. Now let's discuss time in regards to printing. Printing time describes how long it takes to print a model with the external supports. When a model is printed with supports, the print time increases to accommodate the supports [12], so support-generation algorithms attempt to generate supports that minimize the additional time. Unlike with processing time, printing time usually correlates positively with material usage. Naturally, when there is less material needed to be deposited, it takes less time to deposit all that material as well. However like processing time, print time also tends to remain inversely proportional to support structure stability.

2.4 Post-production

One last factor that is considered is post-production. In 3D printing, post-production refers to steps that are taken after a 3D model is printed to finish it. A major part of post-production is removing the external supports, so the way in which the supports were generated and printed can significantly affect this step. Therefore, removability is an important variable. External support structure-generating algorithms improve removability of supports by reducing the surface area in which the supports contact the model. This is usually done by shaping the tips of the supports contacting the model into particular shapes, such as spheres or narrow tips [6]. An example of such tips is shown in Fig.5.

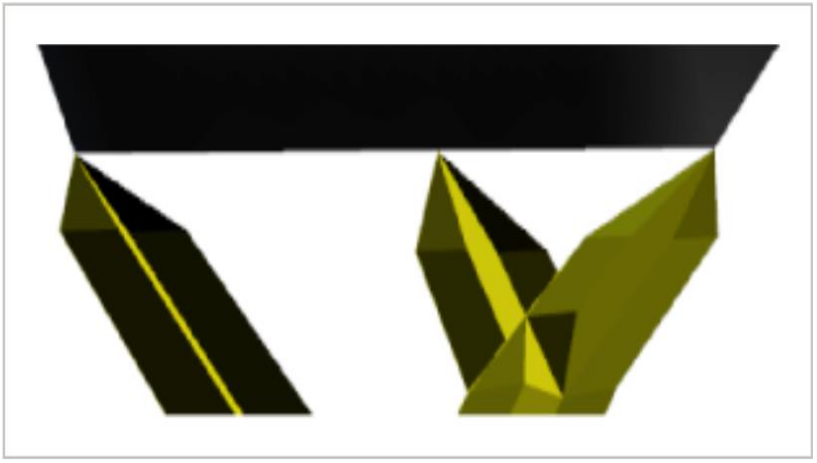


Figure 5. Support structures with narrowed tips [6].

Another significant variable concerning post-production is the surface quality of the model after supports are removed. When support structures are broken off of a model, they may leave behind surface artifacts that may be detrimental to the surface finish [12]. Surfaces that are strongly connected to supports may have visible smudges or rough textures after post-production. Different support methods have varying effects on surface quality [12], though methods with better removability tend to have a lesser effect.

2.4.1 Soluble Support Material

While on the subject of post-processing, it is important to make note of sacrificial or soluble support material. The idea is to print supports with a different material than the model itself, a material that is soluble or sacrificial in some way [12]. While printing, the printer would use the soluble filament for the support structures, and a separate normal filament for the model itself. Then once the model is finished, it is then submerged in a solvent for several minutes. The solvent dissolves the support material while leaving the model unaffected. An example of a model printed with soluble supports is shown below, after printing and after post-production respectively. In FDM printing, common soluble materials are polymer feedstocks, which dissolve in relatively benign solvents such as water and limonene [12]. Soluble support material can significantly improve post-production and negate the demand for supports to be easier to remove. However, it is only usable by printers that support printing with two or more types of filament simultaneously. Even on these printers, soluble support material can limit the number of colors that can be used in models that are intended to be printed in multiple colors.

Figure 6. A model printed with soluble supports, before and after they are removed respectively [3]

Chapter 3: Pre-generation

3.1 Support Optimizations

Before generating external support structures for a model, it may be worth minimizing the need for supports as much as possible. In many cases, it is possible to make slight adjustments to models to significantly reduce the amount of supports required, or even eliminate the need for supports altogether. These practices can be referred to as support optimizations. The next few paragraphs will discuss various support optimizations.

3.1.1 Rotation

One method of support optimization is rotating the model. Oftentimes, the geometry of a model that needs support is significantly affected by its orientation relative to the print bed. Rotating a model can change which parts are overhangs and which ones are not. The images in Fig.7 demonstrate how different orientations of the same object can affect how much support is required.

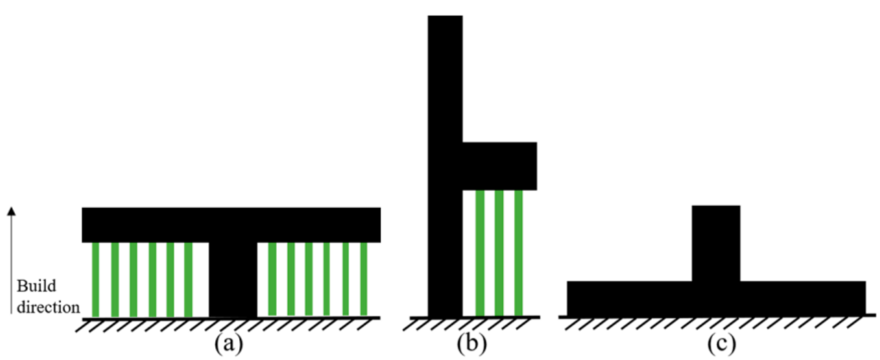


Figure 7. The same shape and the supports required to print it at different orientations [12].

3D models can be rotated in three angles of rotation: roll, pitch, and yaw [1]. Some developers have designed algorithms that search for a combination of these three angles that orient a model in such a way that the amount of geometry that requires support is minimized. A benefit of this method is that it does not alter the shape of the model [12]. This is useful when the shape of the printed object has to be exact to be useful and altering the shape of the model in any way is not applicable. However, the orientation of the model may also affect its structural soundness. Rotating a model affects how the filament is layered, which then affects how the model breaks [12]. In some cases, the rotation that produces the minimum amount of overhanging geometry may reduce the structural stability of the model. The images in Fig.8 demonstrate how different orientations can affect the structural soundness of a model. The model on the left was printed as is while the right one was printed on its side. Notice that the layers are aligned differently as a result, affecting how the object can be fractured.

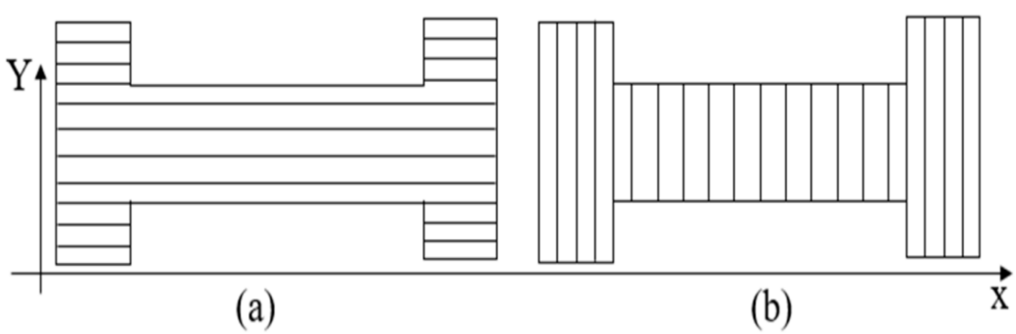


Figure 8. The same shape with different print layer alignments [12].

3.1.2 Modifying Surface

Another method of support optimization is altering the geometry of the model. This involves altering parts of the original model's geometry so that they are more print-friendly [12]. This involves raising points so that they aren't hanging below other geometry, or altering inclines so that they are less steep. Developers have designed algorithms that optimize model geometry by shifting the positions of vertices in problematic locations within a certain level of tolerance [10]. These algorithms allow a tolerance distance to be specified, which describes how far vertices are allowed to be shifted from their original positions [10]. The diagrams presented in Fig.9 shows an example of a process that locates and alters geometry to make a part more print-friendly. Compared to other support optimization methods, this one tends to be more situational, since altering the model's geometry may not be desired in some cases.

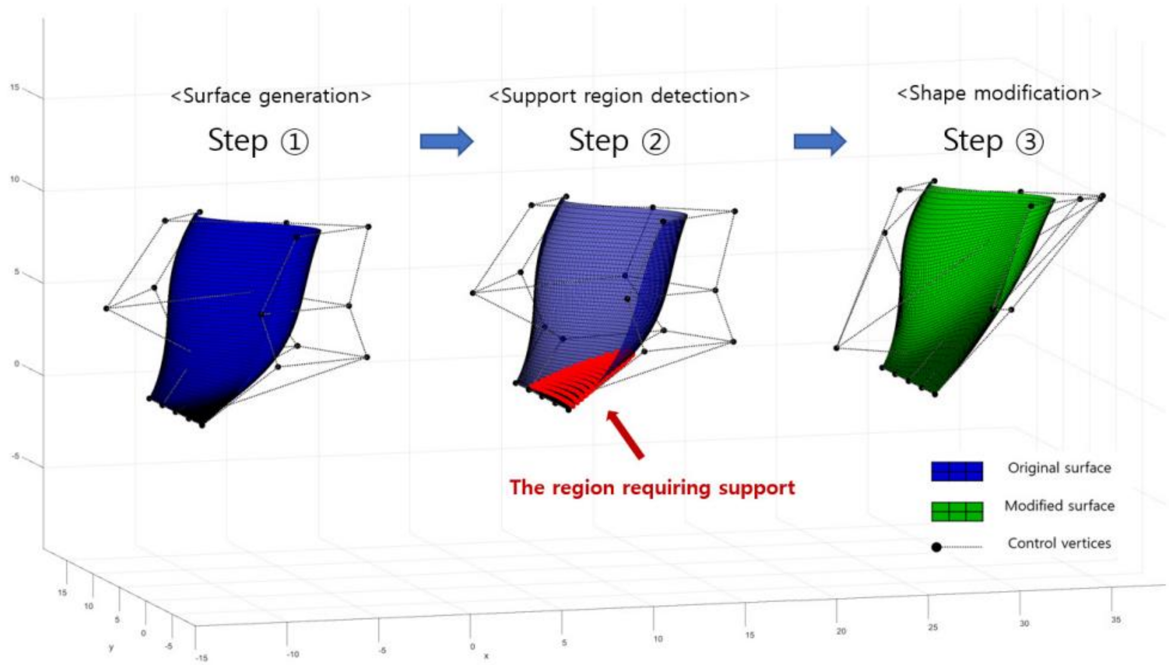


Figure 9. Illustration of a process that alters a model to make it more print-friendly [10].

3.1.3 Dividing Model

A third potential method of support optimization is splitting the model into multiple pieces. The idea here is that segmenting a model into multiple simpler pieces which are then printed separately may require less support than if the whole model were to be printed in one piece. After printing, the parts can then be assembled to get a whole model. This idea is explored in an article, which describes a program that hollows the model out, splits it into smaller pieces, and arranges said pieces as compactly as possible on the print bed [7]. A series of images in Fig.10 shows how an example model is converted into a collection of smaller compactly-packed parts using said program, and its post-production process after being printed. Not only does this method reduce the amount of required supports, but it also reduces the amount of print area required [7], which may be helpful when using smaller printers. Additionally, the printed model's shape is at least in theory unaltered from the original model, similarly to rotation methods. The main disadvantage of this method is that it requires more post-processing, as the model has to be assembled after being printed.

Figure 10. Conversion of a model into tightly-packed pieces, the printed result, and its assembly [7]

3.2 Finding Overhangs

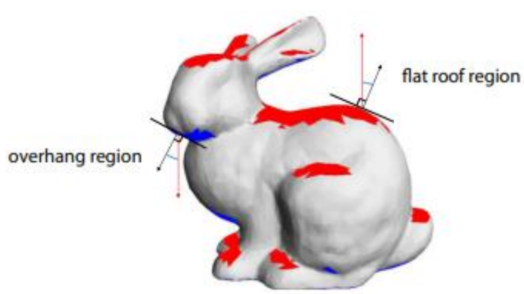
While the previous optimization methods are optional, the one mandatory step that must be taken before external support structures are generated is to locate overhang regions in the model. In order to generate supports, the regions in the geometry that need supports must be identified beforehand. Overhang-searching algorithms fall into two major groups: polygon-based algorithms and slice-based algorithms. Both types of algorithms are discussed in detail in the following paragraphs.

3.2.1 Polygon-based Searching

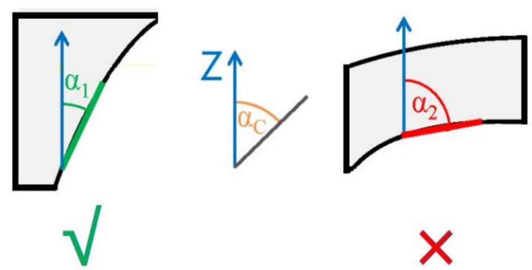
Polygon-based search algorithms find overhangs based on the original 3D polygon model of the object. In digital form, a popular way to describe 3D objects is as a combination of many polygons, often specifically triangles as any polygon can be formed from a combination of triangles. Polygon models are defined using three types of features: faces, edges, and vertices. There are distinct methods for finding overhangs based on each of these three types of features, which are described in the literature [6]. These procedures are repeated for every face, edge and vertex that composes the model, resulting in a complete map of the geometry that requires external support structures.

3.2.1.1 Faces

First, let us discuss faces. Faces can be categorized as overhangs based on its steepness relative to the building direction. To find the “steepness”, algorithms compare a few vectors [6]. In mathematics, vectors are entities that define a specific direction. Usually they are represented using arrows. In 3D, vectors are defined by a coordinate point with three values: x, y, and z. Sometimes, a second coordinate point may be included when the location of a vector is needed, but it is not needed in our case. The search algorithms compare two vectors: one that is vertical, pointed either straight up or straight down, and one that is perpendicular to the surface of the face, often called the normal vector. Alternatively, a vector tangent to the surface [5] may be used instead of the normal, though it doesn't change the procedure by much. Depending on the format of the model, the normal vectors of all polygons may already be given, but it can be computed manually by finding the cross product. The steepness can then be found by computing the angle between the vectors. This can be done by computing the dot product. If the angle exceeds a certain threshold angle, which depends on what is the steepest angle the printer can handle, the face can then be classified as an overhang [6]. The two images in Fig.11 visualize this process using normal vectors and tangent vectors respectively. The former additionally shows how this process can be used to find roof areas as well as overhangs, which while not particularly useful for generating external support structures, are important in the subject of generating internal supports, which as mentioned earlier are outside the scope of this paper.



(a)



(b)

Figure 11. (a) Illustration showing usage of vertical and normal vectors to find overhangs [2]. (b) Illustration showing usage of vertical and tangent vectors to find overhangs [5].

3.2.1.1.1 Narrowing Faces into Points

As mentioned earlier, external support structures are usually designed in such a way that physical contact with the model itself is minimized in order to be easier to remove and to reduce surface artifacts. Therefore, support structures usually don't cover entire faces. Instead, faces that are marked as overhangs are reduced to a collection of points which are later used as connection points for the generated external supports. This process is commonly referred to as sampling. There are multiple sampling methods that algorithms can make use of. Literature which talks about the support-generating program Meshmixer mentions that it uses "Watershed and Poisson surface sampling strategies" in passing without going into detail [3]. While I couldn't find more information about "Watershed" sampling, I did find literature that goes more in-depth on the latter method, which is more often referred to as Poisson Disc Sampling [5].

The method starts with a flat surface, presumably a triangle, on which a single random point is selected. Next, a ring-shaped area is defined around that point, with the inner radius defined as a constant r and the outer radius defined as r doubled. A certain number of random points k is then selected in that region. Next, for each point, the distance between it and each other point is compared. If the distance is less than constant r, then the second point is removed. Once these distance checks are completed for every point in the ring, the original point which the ring was formed around is marked as "inactive". This entire process is then reiterated for every single point in the former ring, as well as for any points that are generated in later iterations. The process ends once all points are marked as "inactive". A visualization of Poisson Disc Sampling is shown in Fig.12, with red dots representing "active" points, black dots representing "inactive" points, and white dots representing newly chosen random points.

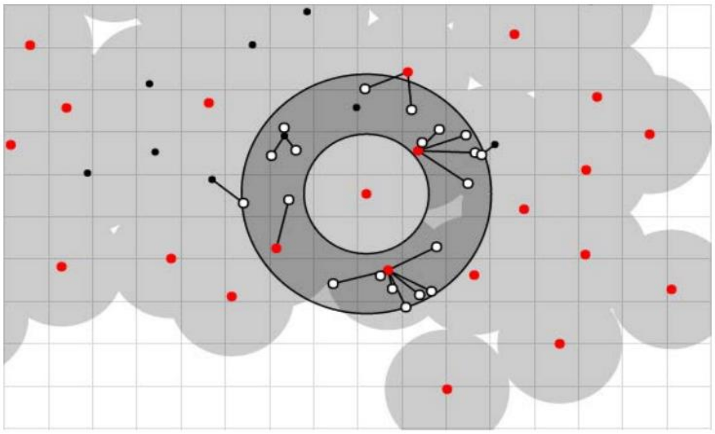


Figure 12. Illustration of Poisson Disc Sampling [5].

In the literature, there are mentions of yet another method called uniform sampling [6][8]. One piece of literature that mentions the method doesn't go into detail on how it works, simply mentioning that a "quick hardware-oriented scanline rasterization algorithm" was used [6]. Other literature goes into a bit more detail. How the method works is that it subdivides a given surface repeatedly until all subdivided sections are small enough, and then the centers of these sections are used as points [8]. The source also features a graphic that illustrates the process [8]. Said graphic is shown in Fig.13.

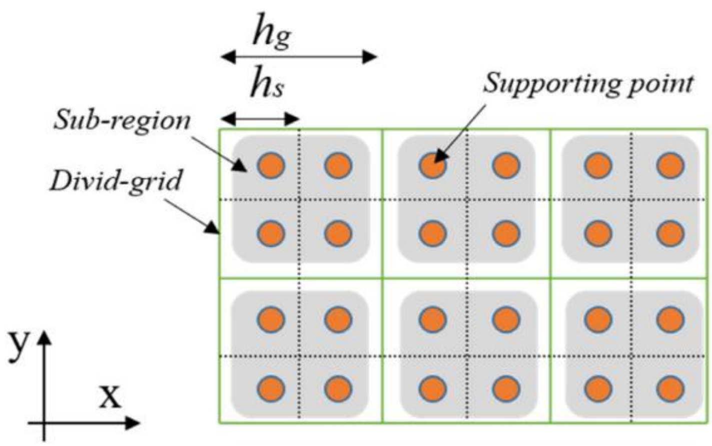


Figure 13. Illustration of Uniform Sampling [8].

3.2.1.2 Edges

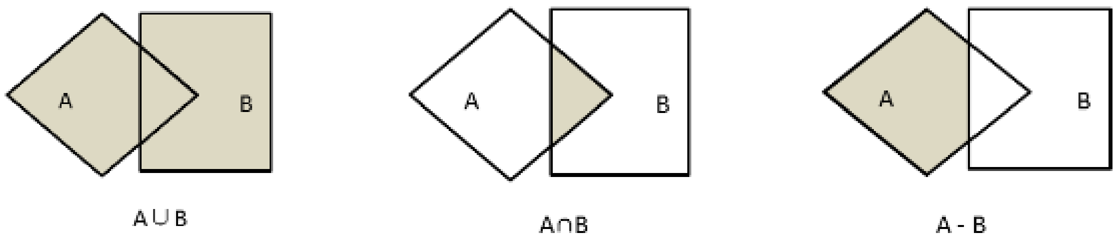
Edges can be identified as overhangs using methods similar to those used for faces [6]. To begin, edges are always adjacent to exactly two faces. The normals of both sides need to be found, which can be done as described previously. Next, a composite vector must be found by finding the average of those two normals. The resulting average vector corresponds to the edge. It can then be used with the vertical vector to find an angle, which can then be compared using the same method discussed previously to decide whether the edge is an overhang or not [6].

3.2.1.3 Vertices

Vertices can be analyzed by comparing it to other neighboring vertices and locating local or global minima [6]. First, all the neighboring vertices must be found which share an edge with the original vertex. Then, the heights relative to the print bed of each vertex must be compared to the original. If any of the vertices is lower than the original, then the original vertex is not an overhang. Otherwise, if the original vertex is lower than all its neighbors, then it is classified as an overhang.

3.2.2 Slice-based Searching

Alternatively, overhangs can be identified by analyzing the model after processing it through slicing software. In sliced form, models are represented by a series of layers, which are 2D shapes representing cross-sections of the model parallel to the printing base. Algorithms that analyze sliced models mainly involve comparing the differences between consecutive layers to see which parts are or are not supported from below [13]. This can be done using Boolean operations for 2D planar simple polygons [9]. Three of these operations are union, intersection, and difference [9]. Union includes all parts of both shapes. Intersection only includes parts that are included in both shapes. Difference removes parts from the first shape that are also part of the second shape. These three operations respectively are illustrated below along with their corresponding notations.



(a) (b) (c)

Figure 14. (a) Union, (b) Intersection, and (c) Difference operations illustrated [9].

The article by Dumas et al. presents a somewhat detailed implementation of slice-based searching [4]. The algorithm takes layers into account by analyzing the exact path that the extrusion nozzle traces over. The path is divided into a set of segments that are within a certain specified length limit. Then, the algorithm looks at each segment and identifies if its two endpoints are supported from below properly enough. When determining support stability, the algorithm considers the width of the extruded filament. Endpoints are treated less as points and more like circles with a radius corresponding to that width. Stability is determined by how much of the endpoint's area is supported from below. If at least 50% of the point's surface is supported from below, then the point is considered stable.

Chapter 4: Support-Generating Algorithms

Now it is finally time to discuss the support-generating algorithms themselves. Once all the points on a 3D model that require support are identified, these algorithms generate the geometry of the structures that are to support all these points. These structures are then added to and printed simultaneously along with the model itself. Support-generating algorithms can be categorized based on the general shape of the structures that they generate.

4.1 Vertical Array

Vertical array structures are arguably the easiest to generate, but the least optimized. Basically, they are shaped like vertical columns which perfectly cover entire overhanging surfaces and head straight down from there. The pillars are loosely filled [8], mainly consisting of simple scaffolding, to provide some level of efficiency. These types of structures are simple enough that algorithms for generating them are often included by default in several common pieces of 3D printer software [8], which makes them easily accessible. Additionally, these structures tend to have very good stability since they have a large contact area with the model and the print bed [8]. This also means that the step of narrowing faces into points is not needed for this type. However, vertical array structures use a significant amount of material, take a lot of time to print, and are difficult to remove; particularly for surfaces with complex geometries [8]. Shown in Fig.15 is an example of a model printed with vertical supports.

Figure 15. A model printed with vertical array supports [5]

4.2 Slopping Wall

Slopping wall structures are very similar to vertical array structures discussed previously, but with one key optimization added. If there is any model geometry close enough to an overhanging region, a sloped pillar with its base connected to that neighboring geometry will be generated instead of a strictly vertical one [8]. This results in pillars that are significantly shorter than ones connected directly to the print bed, reducing the amount of support volume [8]. Compared to vertical array structures, sloping wall structures have improved but still rather poor material usage and print time, while still maintaining good stability. However, the processing time of algorithms that generate such structures may be worse, since additional time must be dedicated for locating neighboring geometry. Shown in Fig.16 is an example of a model printed with sloping wall supports. The model is the same as the one shown on the previous figure.

Figure 16. A model printed with slopping wall supports [5]

4.3 Unit Cell

Unit cell supports are generated by meshing together multiple instances of a specific geometric shape. Examples of applicable shapes include truncated octahedrons and rhombic dodecahedrons, though the literature doesn't rule out the possibility that other shapes could also be usable [12][14]. Examples of a truncated octahedron and a rhombic dodecahedron respectively are shown in Fig.17. The shapes can additionally be hollowed out to varying extents depending on what is more preferred.

(a) (b)

Figure 17. (a) Truncated Octahedron and (b) Rhombic dodecahedron [14].

The support structures are generated by analyzing the original 3D model as a space of voxels of the chosen shape [14]. Once the overhangs are identified, the first step is to add “Interface Unit Cell” voxels to these areas [14]. These series of voxels serve as the contact surfaces with the model itself and hence are designed so that they make minimal contact and are easy to break off. Next, “Target Unit Cell” voxels are added [14] to serve as connecting points between the interface cells and the rest of the soon-to-be-generated support structure. Next, attempts are made to generate some additional cells referred to as “Support Unit Cell” voxels. Under each target unit cell, the algorithm searches each voxel downwards until the print bed is reached or collision with model geometry occurs. If the bottom is successfully reached, then a unit cell is generated on the spot. Afterwards, the algorithm attempts to generate chains of cells in between the pre-existing cells. These chains are generated using Dijkstra’s shortest path algorithm [12] [14]. Beforehand, each voxel in the entire work area is marked with a particular weight value. The algorithm attempts to generate chains that go through the voxels with the lowest weight values. Therefore, higher weights are generally assigned to voxels that the developer doesn’t want chains to be generated though, such as spaces that are occupied by model geometry or spaces that would result in perfectly horizontal chains. This last step is reiterated until there are no cells left that are unsupported.

These structures have lower support volume and support contact area compared to similar structures consisting of traditional columns [12]. But, the variety of shapes that they can form is somewhat restricted by the limited number of geometric shapes that they can be generated with. A diagram presenting an example of a unit cell support structure is shown in Fig.18.

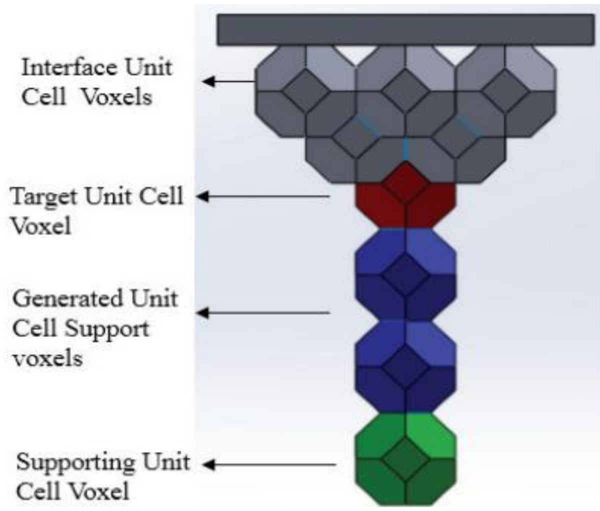


Figure 18. Example of a unit cell support structure [14]

4.4 Tree-Like

One of the most popular types of support structures are tree-like supports. These types of supports are ones that start with one point at the base, and then branch out as they go upward, until every branch ends at an overhanging point to support. The process of designing tree-like supports can be described as a 3D variation of the Euclidean Steiner Minimal Tree Problem [6][8].

Algorithms for generating tree-like structures start from the top and generate trees downwards. Structure generation begins at the overhang points. Under each point, the algorithm looks in a cone-shaped area, with the slope determined by the steepest angle that the printer can tolerate [6][8]. Within that area, the algorithm searches for intersections with either surface geometry or cones under other points. If an intersection with the model geometry is found, then a line is generated from the point to a second point somewhere on the intersecting geometry and the tree is considered finished [6][8]. If an intersection with a second cone is found, then a new point is selected within the intersection, and lines are generated from the points representing the top of the cones to this new point [6][8]. A new cone region is then established for this new point and the process repeats. If neither of the two discussed features are found, then a strictly vertical line is generated from the point down to the print bed. At that point, the tree is also considered finished. The process stops repeating once all trees have an established base. An example visualization of the cone-shaped regions and the support structures that result from them are shown in Fig.19.

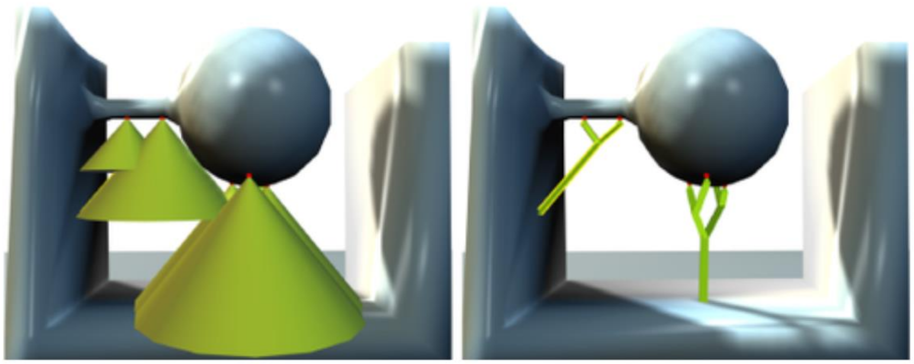


Figure 19. Visualization of cones, and resulting support structures [6].

Compared to other previously discussed structures, tree-like structures can have even less material usage and print time [12]. The trees can be printed with a pretty low width while still maintaining sufficient stability. However, depending on how exactly the tree-like supports are generated, there can be potential issues with stability [8]. Certain tree-like supports may topple due to torque forces caused by the printing nozzle as it extrudes across surfaces [8]. Additionally, tree-like structures may result in longer generation times due to its greater complexity, though some have found ways of mitigating this [6][8]. An example of tree-like support structures is shown in Fig.20.

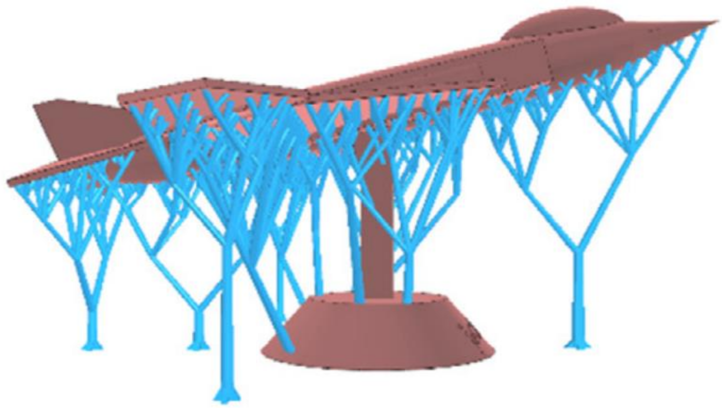


Figure 20. A model with generated tree-like supports [8].

4.5 Space-Efficient Branching

This next type is very similar to tree-like supports discussed previously, but it is considered its own distinct type in some literature [12]. Space-efficient branching supports also have tree-like shapes, but they take advantage of some additional optimizations that aren't used in tree-like structures.

One such optimization is keeping structures close-to and aligned with the model surface [3]. The idea behind this is to think of columns as series of discs rather than as proper cylinders. When support columns are bent closer towards a model's surface, the discs are simply shifted and no new material is actually added. Therefore, although such a change would mean increased support volume in 3D Euclidean distance, there is no difference in practice [3]. Meanwhile, the column's closer proximity to the model means that the printer nozzle now doesn't have to travel as much distance, improving printing time, and the two surfaces can more effectively support each other, improving stability [3].

Compared to traditional Tree-like supports, Space-efficient structures may have even less material usage, faster print time, and potentially better stability. However, these structures may be more complicated to generate due to additional optimizations. An example of Space-efficient branching support structures is shown in Fig.21.

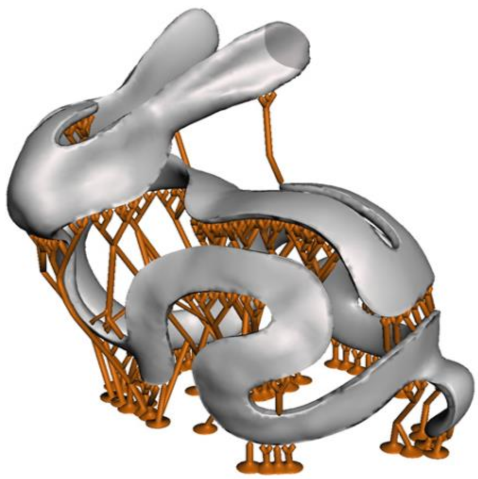


Figure 21. A model with generated space-efficient branching supports [3].

4.6 Bridge

As the name indicates, these support structures are made up of series of horizontal bridges and vertical pillars that support them from below. These structures are inspired by scaffolding used in construction of buildings [4]. The concept here is that when just a few layers of filament are extruded between two elevated endpoints with no other support from below in between, this results in a bridge structure that is surprisingly stable and doesn't have a significantly sagging surface from the top, even when additional weight needs to be supported from above.

An algorithm for generating bridge supports starts by analyzing the 3D model for stability at each layer from bottom to top [4]. For each layer, each separate shape is measured for its center of mass. If a center of mass is found that would make a part of the structure unstable, then a bridge is added below. Each bridge must be supported at its two endpoints. Each endpoint may be supported by adding a pillar below, attaching the endpoint to another bridge, or connecting the endpoint to the model itself [4]. Additionally, bridges must not exceed a predefined maximum length [4].

Next, an iterative series of steps is repeated multiple times to fully link all the bridges and pillars generated in the previous part. During each iteration, the algorithm sweeps through every layer and searches for opportunities to add bridges [4]. The algorithm uses a sophisticated scoring system that considers pre-existing bridges and pillars to figure out how beneficial a new bridge would be in a certain location [4]. If a high-enough score is found, then the bridge is added. Additionally, if the algorithm runs into pre-existing pillars that almost fit certain additional bridges, then it may slightly bend the pillars at the top to make them fit [4].

Compared to Tree-like structures, Bridge structures are more stable at their bases, so they are less prone to falling over while printing [4]. However, these structures may also be affected by a printer's ability to print basic bridge shapes, which may be affected by its abilities and settings. An example of a model printed with bridge supports is shown in Fig.22.

Figure 22. A model printed with bridge supports [4].

4.7 Grain

This type of support structures works best for 3D models which have overhangs that cover very large surface areas and produce many overhanging points.

Once all overhanging points are identified, the algorithm divides these points into sub-sections that cover specific areas, which are referred to as the namesake “grains” [15]. These grains can cover curved as well as flat surfaces. Two methods are used to subdivide the points into smaller regions. The first method separates points based on their height relative to the print bed. This results in ring-shaped regions. The second method follows a more sophisticated set of steps. It starts by locating the two points in the group that are the furthest from each other. Next, the midpoint is found between the two points. Afterwards, the region is split into two along a line that goes through said midpoint. This sequence of steps is then recursively repeated until the resulting regions cover areas that fall under a certain threshold.

Then, each grain is supported from below with a large shape that covers the entire surface area of the grain and resembles an inverted pyramid [15]. Each inverted pyramid is supported from below by a pillar shape which reaches all the way to the print bed, unless the overhang is low enough wherein the supporting pyramid itself already reaches the print bed [15]. In this case, the bottom tip of the inverted pyramid is cut off so that the shape lays flat on the base.

These types of support structures are more situational compared to other types described previously. An example of grain support structures is shown below. The example is intended to support a model consisting of multiple spherical shapes.

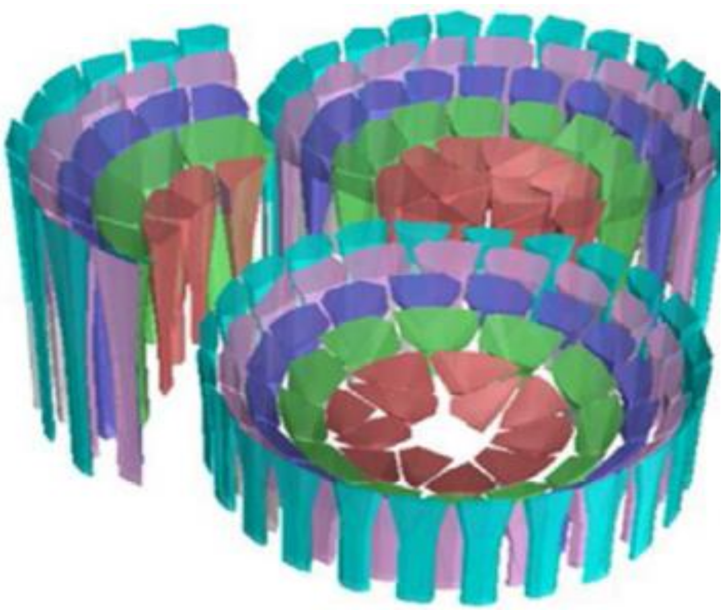


Figure 23. An example of grain support structures [15].

Chapter 5: Summary

The problem of supporting overhangs in 3D printing is a complex problem that has been studied extensively, and many unique solutions have been presented for it. The process of generating external support structures has many detailed steps such as model optimization, locating overhangs, and generating support geometry. Over the years, many optimizations and methods have been theorized and tested, many with good results.

Out of all the methods discussed in this thesis, tree-like supports and the similar space-efficient branching supports appear to be the most popular and extensively explored. This popularity can be seen with Meshmixer, a program that generates these types of supports, as it is commonly referenced in the literature, and the performance of newly proposed programs are often compared to it [4][8]. After these two types, bridge supports seem to come next in terms of popularity, as they seem to be the most often mentioned alternative type [5][8]. An example of software that generates these types of support is IceSL [8]. Despite how unoptimized they are, vertical array supports see much use as they are often the default type generated by software included with 3D printers such as Makerware [8]. Unit cell and grain supports meanwhile seem to be more obscure compared to other types. More research may be worth pursuing to get a better idea of the practicality of those types.

References

[1] Guacheta, Juan C., et al. “3D printing part orientation optimization: discrete approximation of support volume.” *International Journal of Electrical and Computer Engineering*, vol.12, no. 6, 2022, https://doi.org/10.11591/ijece.v12i6.pp5958-5966

[2] Lee, Jusung, and Kunwoo Lee. “Block-based inner support structure generation algorithm for 3D printing using fused deposition modeling.” *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 5-8, 2016, https://doi.org/10.1007/s00170-016-9239-3

[3] Schmidt, Ryan and Nobuyuki Umetani. “Branching Support Structures for 3D Printing.” *ACM Special Interest Group on Computer Graphics and Interactive Techniques*, no. 9, 2014, https://doi-org.ezproxy.gardner-webb.edu/10.1145/2619195.2656293

[4] Dumas, Jérémie, et al. “Bridging the Gap: Automated Steady Scaffoldings for 3D Printing.” *ACM Transactions on Graphics*, vol. 33, no. 4, 2014, https://doi-org.ezproxy.gardner-webb.edu/10.1145/2601097.2601153

[5] Qiu, Jiankang, et al. “A novel supporting structure generation scheme to 3D printing.” *International Conference on Internet Multimedia Computing and Service*, no. 69, 2015, https://doi-org.ezproxy.gardner-webb.edu/10.1145/2808492.2808561

[6] Vanek, J., et al. “Clever Support: Efficient Support Structure Generation for Digital Fabrication.” *Computer Graphics Forum*, vol. 33, no. 5, 2014, https://doi-org.ezproxy.gardner-webb.edu/10.1111/cgf.12437

[7] Vanek, J., et al. “PackMerger: A 3D Print Volume Optimizer.” *Computer Graphics Forum*, vol. 33, no. 6, 2014, https://doi-org.ezproxy.gardner-webb.edu/10.1111/cgf.12353

[8] Zhang, Nan, et al. “Local Barycenter Based Efficient Tree-Support Generation for 3D Printing.” *Computer-Aided Design*, vol. 115, 2019, https://doi.org/10.1016/j.cad.2019.06.004

[9] Shan, Yan. “Design of A Support Automatic Generation Algorithm Based on Layer Information.” *Academic Journal of Science and Technology*. vol. 6, no. 3, 2023, https://doi.org/10.54097/ajst.v6i3.10657

[10] Ahn, Jaeseung, et al. “Knowledge-Based Design Algorithm for Support Reduction in Material Extrusion Additive Manufacturing.” *Micromachines*, vol. 13, no. 10, 2022, https://doi.org/10.3390/mi13101672

[11] Shi, Kanle, et al. “Slicing and support structure generation for 3D printing directly on B-rep models.” *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 3, 2019, https://doi.org/10.1186/s42492-019-0013-x

[12] Jiang, Jingchao, et al. “Support Structures for Additive Manufacturing: A Review.” *Journal of Manufacturing and Materials Processing*, vol. 2, no. 4, 2018, https://doi.org/10.3390/jmmp2040064

[13] Tricard, Thibault, et al. “Ribbed support vaults for 3D printing of hollowed objects.” Computer Graphics Forum, vol. 39, no. 1, 2019, https://doi.org/10.1111/cgf.13750

[14] Vaidya, Rohan and Sam Anand. “Optimum Support Structure Generation for Additive Manufacturing using Unit Cell Structures and Support Removal Constraint.” *Procedia Manufacturing*, vol. 5, 2016, https://doi.org/10.1016/j.promfg.2016.08.072

[15] Habib, Md Ahasan and Bashir Khoda. “Support grain architecture design for additive manufacturing.” *Journal of Manufacturing Processes*, vol. 29, 2017, https://doi.org/10.1016/j.jmapro.2017.08.008