The University of Southern Mississippi

The Aquila Digital Community

Fall 12-1-2017

3D Printing Concrete Structures and Verifying Integrity of their G-Code Instructions: Border Wall a Case Study

Jason Breland
University of Southern Mississippi

Follow this and additional works at: https://aquila.usm.edu/dissertations

Part of the Architectural Technology Commons, and the Other Computer Sciences Commons

Recommended Citation

3D PRINTING CONCRETE STRUCTURES AND VERIFYING INTEGRITY OF

THEIR G-CODE INSTRUCTIONS: BORDER WALL A CASE STUDY

by

Jason Spencer Breland

Dissertation
Submitted to the Graduate School,
the College of Science and Technology,
and the School of Computing
at The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

December 2017

3D PRINTING CONCRETE STRUCTURES AND VERIFYING INTEGRITY OF

THEIR G-CODE INSTRUCTIONS: BORDER WALL A CASE STUDY

by Jason Spencer Breland

December 2017

Approved by:

_____
Dr. Dia L. Ali, Committee Chair
Professor, Computing

_____
Dr. Chaoyang Zhang, Committee Member
Professor, Computing

_____
Dr. Beddhu Murali, Committee Member
Associate Professor, Computing

_____
Dr. Ras B. Pandey, Committee Member
Professor, Physics and Astronomy

_____
Dr. Bikramjit Banerjee, Committee Member
Associate Professor, Computing

_____
Dr. Andrew H. Sung
Director, School of Computing

_____
Dr. Karen S. Coats
Dean of the Graduate School

Published by the Graduate School

ABSTRACT

3D PRINTING CONCRETE STRUCTURES AND VERIFYING INTEGRITY OF

THEIR G-CODE INSTRUCTIONS: BORDER WALL A CASE STUDY

by Jason Spencer Breland

December 2017

Thanks to advances in Additive Manufacturing (AM) technology and continued research by academics and entrepreneurs alike, the ability to "3d print" permanent concrete structures such as homes or offices is now a reality. Generally, AM is the process that allows for a 3d model of an object to be converted into hardware instructions to generate that object layer by layer using a malleable medium such as a plastic. Specifically, large scale concrete AM can now generate a structure, such as a building, layer by layer more quickly and efficiently than traditional construction methods [6, 39]. This innovative, semi-autonomous process promises many improvements over traditional construction methods, but it also introduces new challenges to be overcome. The increased level of automation, the accelerated construction speed, and costly nature of defects are all important factors that emphasize the need for a thorough review of the final hardware instruction sets before production of the project ever begins.

In this research, we propose and explore five methods to help verify model integrity of the print instructions: visual inspection of the design elements, using Fuzzy Logic to predict thermal stress, extrusion end point evaluation, pathing collision checks, and ray tracing for identification and analysis of overhangs. While these methods are not all inclusive, they will help to identify potential defects and high risk design elements in the pre-production phases of a project.

Collectively these five verification methods proposed serve as a starting point for verifying model integrity. These verification methods derive detailed information from the instruction sets, execute various simulations and data analysis, and provide feedback to improve the overall model design and print process. Additionally the simulation process described herein can be built upon to produce other methods of verification. Earthquake or wind resistance tolerances could potentially be verified using existing model data and material data.

Lastly these verification methods will be actively applied across a case study for a proposed wall along the southern border of the United States. This application was selected specifically because Additive Manufacturing should clearly have substantial benefits over traditional hands on construction methods for this project. A concrete wall without any of the intrinsic complications of lived in buildings, may prove to be an outstanding killer application of 3d printing technology. Not only is the border wall used as a test case for the verification methods, but it also serves as a cost analysis to predict the cost benefits of 3d printing simple mostly automated projects. The author does not endorse any political stance by proposing this case study. The case study is purely a scientific endeavor to explore the feasibility of concrete structures outside the scope of traditional buildings. Additional applications of the research could include water levees, dams, and perhaps even bridges. The construction of large scale concrete infrastructure may prove to be an ideal problem domain for Additive Manufacturing.

DEDICATION

I would like to dedicate this work to my wife and daughter, Mary and Ariah, for their continued support and encouragement. You are both my inspiration, and I love each of you deeply. I would also like to thank the rest of my family for their continued support through my academic career including my mom, dad, and brothers, Justin, Travis, Jonathan, and Jim who have always been there regardless. Thank you all for your encouragement and support as always.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

ABS   Acrylonitrile Butadiene Styrene

AM   Additive Manufacturing

ASTM   American Society for Testing and Materials

FDM   Fused Deposition Modeling

MIT   Massachusetts Institute of Technology

PLA   Polylactic Acid

RFP   Request for Proposal

RepRap   Replicating Rapid Platform

SLS   Selective Laser Sintering

SLM   Selective Laser Melting

UV   Ultraviolet

CHAPTER I - INTRODUCTION

1.1 Problem Statement

With Additive Manufacturing, commonly referred to as 3d printing, both the 3d model and the settings of the 3d printer have to be precisely setup to ensure a high quality final product. Even small counter intuitive details within the model or settings can have adverse effects on the final product. Flawed prints and designs can be common with a 3d model which has never been manufactured before or never manufactured before on that specific 3d printer. These flawed prints and the nature of iterative prototyping design processes can result in numerous prints through the design cycle. Creating a 3d print, such as a "custom adaptive" pencil grip for special needs children, took a research team three iterative design cycles for the resulting print to be finalized [46]. Hands on feedback is useful in design, but as print size increases so does time and material cost per iterative cycle.

Obviously, the iterative design cycle can become prohibitively expensive when scaled up to large scale concrete printers. The cost of materials and time to recover the print site are completely incomparable to the small scale 3d printing machines. Similarly late stage design changes and corrections in traditional construction methods are the number one cause for costly rework [47]. One solution to avoid iterative design cycles for 3d printed structures can be the same as traditional construction methods: identifying and correcting issues during the pre-production stages of development. Virtual design review has allowed for corrections and alterations during pre-production stages for a more cost effective iterative design cycle [48]. For large scale concrete structures, pre-production design analysis and verification would likewise reduce costly rework.

1.2 Significance of Research

Early identification and correction of potential issues is key to streamlining the print process. Carefully simulating how the 3d print instructions would be executed provides detailed information about the proposed structure. The information derived from the simulations is analyzed in an attempt to verify model integrity. Five methods of verification are proposed here to aid in data driven decision making during the design process. The goal of the verifications methods described here are to ensure model integrity before the print process ever begins. Every issue identified and eliminated early is time and money saved further down the project timeline.

Three of the five methods of verification were self-evident after working with small scale 3d printers. Firstly, visual inspection of the design elements is obvious. There is no substitute for having an expert designer thoroughly review the final structure to be printed. The goal with the research here is to provide detailed information to the expert designer to aid in the data driven decision making process.

Collision detection and rerouting is also a self-evident problem and solution. A clear print bed is an easy solution for small scale prints. When printing a large concrete structure, trees, existing structures, plumbing, or support structures can all prove to be physical obstacles to the printing extruder. If the locations of the obstruction entities can be predicted with reasonable accuracy, collision prediction and rerouting is possible.

Perhaps the most restrictive limitation of 3d printing is the strict support requirements for any overhang. Of course when viewing the 3d model, most overhangs are obvious, but again the stakes are raised with large scale concrete printers.

Additionally certain overhangs will be within an acceptable tolerance while others are not. If supports are required for the next printed layer, it is absolutely necessary that they be in place before the print process continues. An innovative overhang detection system is proposed that utilizes ray tracing renderings to automatically detect and identify if an overhang is within acceptable tolerances.

The final two verification methods were identified after correspondences with Andrey Rudenko were he expressed common issues he was having with large scale 3d printed concrete structures. Concrete is a fundamentally different material than plastic so it was necessary to discuss real world experience with the printing process. After discussing with Rudenko issues he had experienced, the following two methods were developed in response.

The locations where concrete extrusion begins and ends can often lead to a vertical seam across the print. The effect is mostly visual, but under certain conditions can also be a structural weakness. Extrusion end point tracking is included in order to flag segments that will have a seam before printing.

Long continuous sections of concrete expand and contract with thermal conditions. Not only can cracks form from thermal contraction shortly after printing, but routine contraction and expansion from day, night, and seasonal cycles can also compromise structural integrity as well. Fuzzy Logic is proposed as a method to display predicted thermal stress fractures.

Collectively these five verification methods proposed here serve as a starting point for verifying model integrity. These verification methods derive detailed information from the instruction sets, execute various simulations and data analysis, and

provide feedback to improve the overall model design and print process. The simulation

process described herein can be built upon to produce other methods of verification.

Earthquake or wind resistance tolerances could potentially be verified using existing

model and material data.

Lastly a real world project, the proposed border wall between the United States

and Mexico, is used as a case study. A concrete wall, without any of the intrinsic

complications of a lived in buildings, may prove to be an outstanding application of 3d

printing technology. In fact many structures such as water levees or perhaps bridges and

dams and other infrastructure may one day be produced more cheaply through AM

processes. The border wall will be used as a test case for the verification methods, but it

will also serve as a cost analysis point to predict the cost benefits of 3d printing simple

mostly automated projects.

The remainder of the dissertation is organized as follows:

Chapter II provides background material for Additive Manufacturing historically

and in the context of concrete structures. It will also serve as a brief introduction into

both Fuzzy Logic and Shortest Path algorithms. In depth analysis of how AM works is

delegated to chapter three.

In Chapter III, the Additive Manufacturing process is explored in depth. It begins

with a broad view of the overall art path required to generate the G-Code instruction sets

that are ultimately printed to produce the 3d modeled object physically. Each step along

the way is then discussed in detail from creation of the model, the slicing process that

generates the instruction set from the model, and lastly how the printing process

physically produces the final product.

Chapter IV discusses related works relevant to the research found here. Lee Butler's G-Code importer which was directly built upon is discussed in detail. Slic3r's approach to overhang analysis is reviewed. Lastly tools that help verify model integrity are reviewed and analyzed.

Chapter V outlines the simulation process through Blender that allows for in depth analysis followed by the five verification methods described above: visual inspection of the design elements, using Fuzzy Logic to predict thermal stress, extrusion end point evaluation, G-Code pathing collision checks, and ray tracing for identification and analysis of overhangs.

Chapter VI follows the case study through iterative pre-production design cycles. Firstly the overhang detection and analysis is used to better design the model, then Fuzzy Logic analysis is implemented to further update the project. The final result is then analyzed in depth in an attempt to estimate total project costs.

Lastly in Chapter VII, the conclusion and future work section serve as a brief summary of the results of the research contained herein. Several potential avenues of future work are briefly discussed as well.

CHAPTER II - BACKGROUND

2.1 An Introduction to Additive Manufacturing

Today the terms 3d printing and Additive Manufacturing are used largely interchangeably. When discussing 3d printing, most people today probably envision the desktop sized extrusion based printers common in online communities for producing small plastic novelties and prototypes. Truthfully, there are numerous manufacturing methods which can all be categorized under the term Additive Manufacturing. To properly determine what does or does not qualify, Additive Manufacturing must be clearly defined. The ASTM Standard describes Additive Manufacturing as "the process of joining materials to make objects from 3d model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies, such as traditional machining" [3].

Loughborough University describes the most common methods for 3d printing in the following ways [34]. Material extrusion is the most common method of AM. A material, usually a plastic like PLA or ABS, is extruded one layer at a time starting onto a bed that is often heated. Binder jetting is similar to material extrusion except a powder material is added to the surface after each layer is completed which continually brings the "bed" up to level. A binding agent is applied to create the solid portions of the object. The powder acts as a support material to allow for overhangs in any area. Material jetting drops material much like an ink jet printer to build up the object. Powder bed fusion is similar to binder jetting in that it uses a powder material, but rather than a binding agent, a powerful laser solidifies the powder. Bed fusion manufacturing solidifies powder (usually metal) to create strong components which the other methods cannot. Stereolithography works similar to bed fusion but instead of powder it uses a photo

sensitive polymer liquid which is solidified by ultraviolet laser technology. Also rather than building up the object with powder, the solidified portion is lowered deeper into the liquid so that the new layers are generated at surface level.

Fused Deposition Modeling (FDM) is often called extruder based 3d printing because of the processes involved, and was patented more than twenty years ago by Stratasys [50]. The expiration of some of their key patents is what would ultimately inspire the open-source RepRap movement to create 3d printers for home use based on this technology [2]. Stratasys describes the process as follows: a heated extruder melts a thermoplastic to a "semi-liquid" state. This heated plastic is extruded out on to a platform that is usually also heated but below the melting point of the plastic so that it solidifies but does not shrink overly much. Layer upon layer is built up upon the existing layers. Sacrificial supports may also have to be produced to support overhangs either with the same material or another more easily removed plastic. A second extruder will be required if another material is utilized. Because of the elastic properties of the material used, short "bridge" areas can go unsupported as long as the front and end of the extrusion line is supported [50].

While Material Jetting can refer to a process that uses numerous materials the most common application is wax casting. The jewelry industry has historically used wax castings to model jewelry for decades. This process extrudes heated wax to automate the process. The process is similar to the extrusion method except uses a drop on demand system different than the standard extruder system. Supports are still necessary, but a lower melting point wax is used to provide the supports so that they can be easily removed using a precisely heated bath [34, 49].

2

Anatol Locker from All3DP describes Bed Fusion printing as follows [49]. Bed Fusion is also called Selective Laser Sintering (SLS) or Selective Laser Melting (SLM) depending on the exact process mechanic. SLM is often considered a subset of SLS. A powdered material bed is built up one layer at a time. A powerful laser is used to sinter the powder bed causing the granules to fuse into solid segments. After each layer is finished being fused, another thin powder layer is applied until the object is completed. Two big advantages of this process is that the powder bed provides support during the printing process and the versatility of materials is substantial. Ceramics, nylon, glass and numerous metals including steel, silver, and aluminum can all be printed. The equipment is expensive. While many home users may not own the equipment, there are many companies online which provide the printing services.

Binder Jetting is very similar to the bed fusion method described above. Layer by layer powder is laid out to create a new layer above the previously finished layer. The key difference here is that a liquid binding agent is used to solidify each print segment rather than a laser [34].

Stereolithography which is the oldest method of 3d printing may be a bit more complex than some of the other methods, but despite its age is still in use today for many specific printing processes [34]. The mechanics of the print process are similar Bed Fusion printing but with the key differences noted here [49]. Instead of a powder medium like the Bed Fusion method, a liquid polymer print medium is implemented. Each layer, which still requires support, uses a UV laser to react with the polymer liquid to create solid layers across the topmost surface of the object. The entire object is lowered layer by layer deeper into the liquid as the final object is built up layer by layer. When the object

is finished, it is rinsed with a solvent and is often placed in to finish baking in a UV light environment.

While these are five of the most common methods of Additive Manufacturing, depending on what is considered a separated technique or a subset of an existing technique, up to nine different printing methods could each be considered a different technique [49]. Additive Manufacturing is definitely the subject of intensive research at this time with new applications or advancements being announced regularly.

2.2 A Brief History of Additive Manufacturing

Modern stereolithography was invented during the 1970s by Dr. Hideo Kodama in Japan, but it wasn't until 1986 that Charles Hull patented the technology and began to commercialize it [40]. While revolutionary, stereolithography filled a very niche market and the process proved to be slow and expensive.

A concise timeline, constructed by T. Rowe Price, for some of the more groundbreaking and interesting 3d printing advances can be seen below [38]. It's easy to see the correlation between RepRap's first self-replicating printer in 2008 and the sudden rapid developments in 3d printing in various fields the following years.

- In 1992 the first stereolithographic machines were produced for commercial use by 3d Systems.
- In 2002 A working kidney was 3d printed by combining 3d print technology and organ growing technology developed in 1999.

4

- In 2005 Adrian Bowyer at the University of Bath created the RepRap movement. It is an open source project that designs 3d printers that can make key components to make more 3d printers.

- In 2006 selective laser sintering (SLS) allowed the fusing of materials by laser. This allows for industrial parts to be prototyped and produced in more durable materials.

- In 2008 RepRap created the first self-replicating printer.

- Also in 2008 Shapeways allowed for anyone with a 3d model to order inexpensive 3d real life versions of their own custom creations online.

- Also in 2008 the first person walked with a 3d printed prosthetic. The entire prosthetic including joints was printed in a single structure. No assembly or detailed resizing was required.

- In 2009 do it yourself (DIY) 3d printer kits hit the market for the first time.

- Also in 2009 Dr. Gabor Forgacs printed the first blood vessel

- In 2011 the University of Southampton printed an unmanned "plane" or drone in seven days with a budget of 5000 British Pounds. Its elliptical wing shape would typically be expensive, but 3d printing made them much more affordable.

- Also in 2011 we saw the first 3d Printed Car by Kor Ecologic. The Urbee is an environmentally friendly prototype that gets 100-200 MPG city-highway respectively.

- Also in 2011 3d printing in gold and silver became available bringing AM into the jewelry market.

- In 2012 the first 3d printed prosthetic jaw was implanted into a patient in the Netherlands.

The explosive exposure of 3d printing today can find its roots with the RepRap movement. Adrian Bowyer wanted to create 3d printers that could reproduce the majority of parts for future generations of 3d printers. The entire project is open-source which has kept costs down and made 3d printers more widely available. RepRap as a whole started gaining world-wide popularity, with numerous varieties of printers available, in the early 2010s [2].

Today 3d printing has impacted a wide variety of fields and promises to impact many more. 3d printing of concrete structures looms in the construction industry promising to disrupt the industry. New Zealand recently 3d printed and successfully launched a rocket into a suborbital trajectory [31]. With complex metal alloys, glass, and concrete as potential materials in 3d printing, Additive Manufacturing has potential to impact any given industry with manufactured goods. One bank and financial services company, ING, in Amsterdam is estimating that half of all manufactured goods will be 3d printed by 2060 which would also reduce worldwide trade by 25% [52]. Those estimates may sound farfetched, but just as television is moving toward an on demand streaming industry, car parts, toys, and jewelry may all be printed on demand on site one day.

2.3 Additive Manufacturing of Concrete Structures

There are several commercial entities actively engaged in the early stages of printing full scale usable concrete buildings both on and off site. Even Dr. Behrokh Khoshnevis, who may be the premier researcher in Additive Manufacturing concrete

structures, has created a commercial entity to monetize his research [22]. Within the field of concrete AM, businesses are competing to make both headlines and profits.

In 2014 Andrew Rudenko made headlines when he 3d printed what is possibly the first permanent large scale structure show in Figure 2.1 [32]. Based on the RepRap architecture, he constructed an oversized 3d printer that used concrete instead of plastic. As a proof of concept he printed a castle-like structure for his children. This project does a great job of showcasing how complexity can be easily handled by Additive Manufacturing processes. Because complexity is handled by the pathing of the printer rather than human labor, volume of material becomes the contributing factor to the cost of the project rather than complexity.

Figure 2.1 First 3D Printed Permanent Concrete Structure

Note: Image courtesy of Andrey Rudenko.

In October of 2015 Andrey Rudenko collaborated on a project with Lewis Yakich from the Philippines to construct a 1076 ft^2 full sized hotel suite. The printing process

itself took roughly 48 hours [17]. The finished building can be seen below in Figure 2.2.

The suite includes a giant sized Jacuzzi which was also part of the 3d printing process.

The feat is impressive because it exhibits the versatility of the 3d printing process. Unlike

many proof of concept structures, the hotel suite is designed for actual use. Additionally

Lewis Yakich was approved by the Philippine government as a qualified builder for their

low cost housing program. He seemed posed to revolutionize affordable houses for low

income families. Unfortunately Yakich has been missing since November of 2015 [17].

Figure 2.2 First 3D Printed Hotel Suite, Exterior

Note: Image courtesy of Andrey Rudenko.

In 2016 Dubai became the first country to complete a 3d printed office building.

Now they are boasting a larger project. A project to build a 1,375 foot tall skyscraper is in

the early development stages. This will be possible through the combination of 3d

printing technology and existing precision crane technology. The ruler of Dubai has

announced that by 2030 he plans for 25% of all buildings in his city will be 3d printed [1].

In early 2017 Apis Cor, a startup company based out of the US, has successfully manufactured a 400 ft² fully furnished livable space in Russia [39]. The entire printing process was completed in 24 hours during the Russian winter to demonstrate not only the speed of the technology but the range of operability.

A strong interest in commercializing AM for construction is pushed forward by compelling statistics. Dr. Behrokh Khoshnevis, Lewis Yakich, and Apis Cor all estimate a monetary savings of approximately 60% across an entire project compared to traditional building methods [6, 17, 39]. Besides the financial benefits Dr. Khoshnevis estimates CO_2 emissions can be cut by 75%, energy by 50%, and solid waste by 86% compared to the traditional concrete masonry unit construction method [22].

As promising as AM appears to be for use in construction, there are at least two major drawbacks which may be unavoidable. Up front cost for large scale concrete printers are going to be expensive. As with many emerging technologies initial costs tend to be high (The first cell phone initially cost $3995) before dropping over time [15]. The second most obvious drawback would be a net loss of construction jobs. It seems obvious that as automation increases, fewer manual labor construction jobs will be available. Dr. Khoshnevis has made two compelling points regarding the potentials loss of jobs. Firstly in the year 1900 over 60% of all adult US citizens were farmers. Today only 1.5% of Americans fill the profession. The reduction in the total number of farmers per capita was the natural outcome of automation. Secondly he points out that additional jobs will become available around the new emergent technology. These new jobs can also be filled

by women and aged individuals whereas traditional construction jobs are filled by young to middle aged men [6]. Regardless the construction industry may look drastically different in the decades to come.

Additive Manufacturing is still in the early stages of development. As an emerging technology, it could prove to be a disruptive technology for the construction industry. Numerous commercial and academic entities are actively researching and developing AM within the construction field. It may not be surprising to see 3d printing as a common construction method in five to ten years.

2.4 A Brief History of Concrete Structures

Concrete is a complex material with an equally complex history. By investigating historical structures that utilized concrete, we can gain insight into the design strategies and materials of the structure to see how they have stood the test of time.

The Great Pyramid at Giza was constructed around 2500 BC. Pyramids notwithstanding, bricks of straw and mud were the most commonly used building component in Egypt at this time [16]. This makes the pyramid's structure of quarried stones and 500,000 tons of gypsum and lime mortars for the foundations of the casing stones even more impressive as an architectural feat. Little is usually mentioned about the mortar foundation of the pyramid, but so much raw mortar was used that archeologists are at a loss to explain where the fuel to cook that much mortar could have come from [16]. Prof. Michel Barsoum believes the casing stones may have been cast rather than quarried. While this accusation is hotly debated, he has compelling scientific evidence based on electron microscope imaging [4]. If Barsoum is correct, Egyptians could have

employed complex concrete thousands of years before the Greeks or Romans. Regardless of whether the casing stones were quarried or cast, the solid mortar foundation of the pyramid has definitely performed admirably showing that mortars and concretes can support compressive forces for thousands of years.

When the Greeks first starting mixing pozzolans with existing limestone cement around 600BC, the result was tremendously strong concrete which proved more durable and less reactive to water, especially salt water which quickly erodes most standard building materials [16]. Pozzolans, gaining its name from Pozzuoli (Naples,) is a volcanic ash which as described above helped to create the first hydraulic concrete. Hydraulic concrete, so named for its ability to cure underwater, chemically reacts with water and lime to cure, rather the previous non-hydraulic cements that reacts with the carbon dioxide in the air [16]. While the Greeks did make use of pozzolan concrete, it is more commonly called Roman concrete for good reason. The Romans refined cement into a valuable tool that became a hallmark for some of their most famous designs and infrastructure.

The Romans may be the culture most famous for their concrete structures many of which still stand today. The Roman baths, the Coliseum, the Pantheon, and numerous aqueducts and concrete ports either stand in full or in part to this day [12]. The concrete mixtures which still stand today are only half of what allowed these structures to persist. Circles and arcs were a common design theme as seen below in the coliseum. These design elements, even without modern rebar, have allowed these structures to stand through earthquakes, harsh winters, and storms for thousands of years.

Figure 2.3 The Roman Coliseum

Note: Image courtesy of Mary C. S. Breland.

The Pantheon may not be as famous as the coliseum, but it may surpass the coliseum as an architectural achievement. At least five different concrete mixes were used throughout the height of the Pantheon pictured below to incorporate lighter and thinner materials higher up [26]. The dome shaped structure with the square insets to lighten the load again contain no rebar and is pictured below. Concrete is strong compressively but weak against tension forces, or pulling forces. Structurally the Pantheon had to be designed so that all supporting pressures were completely compressive rather than tensional for the structure to still stand. Despite the fact that virtually all sizable concrete today employs rebar for structural reinforcement, the roman structures by design have stood without it.

Figure 2.4 The Dome of the Roman Pantheon

Note: Image courtesy of Mary C. S. Breland.

The Great Wall was initially constructed with standard lime based mortar, but as the years progressed China developed a superior mortar by combining the inorganic lime with organic rice [44]. Bingjian Zhang, Ph.D., and colleagues identified amylopectin, a complex carbohydrate, as the secret ingredient to make the rice mortar stronger and more resistant to water than traditional mortar. Other structures like tombs, pagodas, and city walls in China used the rice mortar before the Great Wall. The rice mortar is still used for restoration projects.

Exact dates are scarce but most sources agree that the quality of concrete and the secret of Roman hydraulic concrete specifically was lost after the fall of the Roman Empire [16, 11, 20]. While numerous cathedrals and castles were built by exceptional artisans, the technical proficiency of concrete as a material just wasn't matched. For example the Cologne Cathedral in Germany, which began construction in 1248 is now in constant rotating repair in order to replace the weathering historical stones and concrete [51]. Again architectural design was flourishing, but it may not always be obvious which mortars or concretes will last hundreds of years versus those that can last thousands of years.

Hydraulic concrete was rekindled as a building material when Vitruvius' Roman manuscripts outlining concrete and construction were discovered in a Swiss Monastery in 1414 and began to be recirculated [2]. Unfortunately no major breakthrough with concrete really occurred until 1793 when John Smeaton, a Frenchman, discovered a new modern way of creating hydraulic concrete. His breakthrough was immediately used on the Eddystone Lighthouse, a disastrously watery site and the driving force behind his discovery [16]. In 1824 another major breakthrough occurred. An Englishman named Joseph Aspdin invented and patented Portland cement. It is the basis of the modern concrete we use today. Like Roman concrete, it is also hydraulic but it cures faster, harder, and is less water permeable [16, 12].

As a brief aside, Roman concrete may not be completely outperformed by modern concrete as implied. In fact traditional Roman concrete has some less than obvious advantages over modern Portland cement. Today, we use approximately 16 billion tons of Portland concrete every year. It accounts for roughly 7% of all carbon emissions yearly

[41]. Roman concrete only has to be heated to 900° C compared to Portland cement being heated to 1,450° C. Adopting Roman cement recipes in part or full could help reduce total global emissions [41]. Roman concrete despite being more water permeable and less hard appears to stand the test of time better. The combination of volcanic ash and lime is more durable long term than Portland cement [42]. Portland Pozzolana Cement (PPC,) a modern version of Roman concrete, is now available and becoming more popular in modern markets.

The last historical concrete structures that will be investigated have an unlikely source. Thomas Edison investigated the use of concrete in building structures at the beginning of the 1900s [5, 13]. After founding his own Portland Cement Company, Edison began pushing concrete for buildings, sidewalks, roads, and other structures. One of his innovative ideas was casting an entire house in a single pour using concrete. Steel support bars and wood frames would be laid out in their entirety before concrete was funneled in from the top to fill all the forms. The modern hydraulic cement allowed the concrete to cure in the absence of air. The single pour house Edison envisioned did become a reality, but the forms proved more costly than anticipated. An initial investment of $175,000 (inflation not accounted for) proved too costly for the idea to spread. Despite ultimately failing to catch on, the raw materials proved to be much cheaper and far sturdier than traditional wood homes at the time. A wooden home cost $4,000 in total then. Pouring a home would cost about $1,200 in materials and labor. Edison's concrete single pour homes may have been ahead of their time, but could also be seen as a precursor for things to come. 3d printing processes of today overlap heavily with

Edison's vision of the single pour house, but luckily the additive manufacturing process

has greatly reduced the necessity for extensive frames and molds.

2.5 A Brief Introduction to Fuzzy Logic

All humans are indirectly introduced to Fuzzy Sets and Fuzzy Logic when we are

children simply because of how humans abstract data. A child learns their parents are

"tall" while they are "short." They learn that an object close at hand is "near" while an

object a fair distance away is "far" The natural next question is one of degrees. Is the

pencil across the room really "far" compared to Tibet or Mars? It seems that "far" can be

subjective rather than objective. With Fuzzy Sets objective limits are assigned to loose

categories like "tall" in order to derive broad sets or categories. In this case "average

height" could be assigned values from 5'8" to 5'10." "Slightly tall" could be assigned a

value from 5'10" to 6.' "Tall" would be 6' to 6'2" and so forth. While Fuzzy Sets have

probably existed loosely in language as long as language has existed, it was Lofti Zadeh

who proposed their existence mathematically in his paper aptly called "Fuzzy Sets" [53].

Specifically Zadeh defines Fuzzy Sets in his paper as follows:

> "Let X be a space of points, with a generic element of X denoted by x.
>
> Thus X = {x}.
>
> A fuzzy set A in X is characterized by a membership function $fA(x)$ which
>
> associates with each point in X a real number in the interval [0,1], with the
>
> values of $fA(x)$ at x representing the "grade of membership" of x in A.

Thus, the nearer the value of fA(x) to unity, the higher the grade of membership of x in A." [61]

With the above example 6'1" could be assigned a value of "tall" and given any person's height and a membership function it could be objectively determined if the person fit the category or not. Membership functions are basically the subjective factor for determining set membership but defined in a mathematical way [53]. Depending on the guidelines or membership function 6' may classify as tall or it might not.

In the context of this research, Fuzzy Sets will be used to assign broad categories that are easily interpreted qualitative values from continuous data. For example a section of wall may be 9144.36mm long. This section of continuous wall could be considered "fairly long" or likely to have "high levels of thermal stress" in this case. Fuzzy Sets and Fuzzy Logic will be explored as a tool to increase readability of large amounts of data in section 4.3.

2.6 A Brief Introduction to Shortest Path Algorithms

The Shortest Path problem has been explored and optimized with numerous potential solutions. The proper question to ask though isn't which algorithm is best, but rather what does the current problem at hand look like. For each potential algorithm solution that is still in common use, it seems each has its own unique cases where it outperforms or fits the problem scenario better than the others. Four of the most commonly mentioned algorithms will be compared here.

The Shortest Path problem is one that is classically explored for educational purposes. Most of the terms and theories are common knowledge, but Alex Chumbley's online publication "Shortest Path Algorithms" was used as a reference to ensure accuracy of the terms found here [46]. To properly compare algorithms a few terms must be established. A vertices or node is a point within a graph or set of possible positions. An edge is a potential path or move between two vertices. A cost or weight can be associated with each edge to show that certain choices may not be equally costly as others. An undirected graph would be a set of nodes where an edge can be traversed either way. A directed graph would be a set of nodes whose edges are one directional. Two edges could be passed between nodes to allow a directed graph to behave like an undirected one. Lastly the solution or solution plays a huge factor in which algorithm will be optimal for a given problem. Below are four popular algorithms with worst case cost denoted with V for vertices and E for edges.

- Dijkstra's Algorithm - $O(V2)$ – Solves shortest path from a single source. No negative weight values are possible [46].

- A* Pathing [47] - $O(E)$ – Solves single pair shortest path using heuristic method. This is the most popular shortest path search according to Amit at Stanford [58].

- Floyd-Warshall Algorithm - $O(V^3)$ – Finds shortest paths between all vertices. No negative weight values are possible [46].

- Johnson's Algorithm - $O(EV+V2log(V))$ – Finds shortest paths between all vertices. It may be faster than Floyd-Warshall for scenarios where few edges exist between vertices [46].

As described above, context of the problem will define the best algorithm for that problem set. If a solution is needed for a shortest path from every node to every node then Johnson's Algorithm or the Floyd-Warshall Algorithm may be best. In our context, we will be looking for the Shortest Path given a start node and an end node. This would result in the A* Pathing algorithm producing the solution at the lowest cost.

A* Pathing's efficiency is due to it favoring nodes close to the starting vertices and nodes close to the goal node [58]. A* can be adjusted for speed or accuracy based on its heuristic and cost functions [59]. This ability to increase speed at the expense of accuracy can be useful for certain scenarios were the optimal path may not be required, but real time calculations are required. Overall A* can be a powerful tool that quickly solves the Shortest Path algorithm for a given pair of starting and ending vertices.

CHAPTER III - G-CODE GENERATION AND EXECUTION FOR EXTRUSION

BASED ADDITIVE MANUFACTURING

3.1 The Art Path of Designing and Implementing a 3D Print

As an overview, a 3d model is "sliced" into layers by a program such as the

original Slic3r program or a commercial alternative. The process and results are

discussed in depth in Section 3.3. The resulting G-Code is then uploaded to the

corresponding 3d Printer whose inputs were used during the slicing process. After

insuring the printer is properly calibrated, the 3d object can be printed. A visual of the art

path process can be seen summarized below in Figure 3.1.
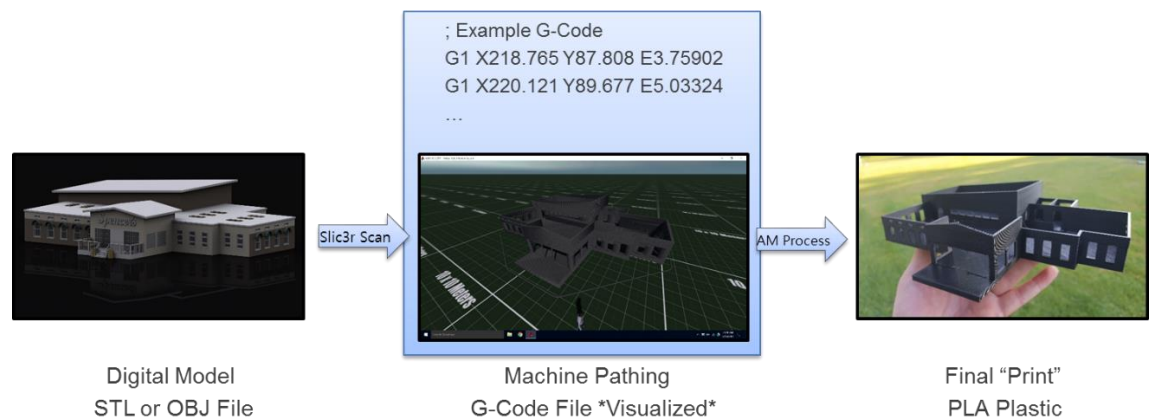
Figure 3.1 Overview of the 3D Printing Process

Every 3d print must first begin as a 3d model. There are numerous online

repositories and communities for sharing 3d models both for free and for purchase.

Unless the current project is a new or unique design, it is often worth it to check and see

if the 3d print is already created as a 3d object and online. A completely custom model

would have to be 3d modeled from scratch. This can be accomplished in any 3d modeling

software, but for these examples Blender will be used. A brief introduction into how a 3d model is created can be found in Section 3.2. Once a 3d model is in hand, it can be processed into G-Code, but it is common that the model will be updated and reprocessed as necessary to fully meet project requirements.

STL and OBJ files are the most common model files that are exported for processing for Slic3r, other formats may be supported as well depending on the exact slicing program being used. By default the G-Code is represented in absolute, positive coordinates but some values like mm of extruded material may be reset to zero during the print process. The absolute coordinates along with the plain text formatting of the G-Code file, make it possible but time consuming to alter the file by hand.

With a small-scale 3d printer using PLA or ABS plastic, it is common, perhaps even the norm, for a print to fail or contain defects. In these cases, the original model is updated, or the settings within Slic3r are changed then another print is attempted. This trial and error cyclical design process may be inefficient for small plastic objects, but would prove disastrous if adopted for large scale concrete printers. The modeling process, Slic3r process, and printing process will all be explored in more depth in the following three sections.

3.2 3D Modeling in the Context of 3D Printing

3d modeling is a complex subject that can take years to learn. This section will introduce some of the more simple concepts of 3d modeling that are relevant to 3d printing. How a model is created, what constitutes a closed mesh, and how overlapping closed meshes can affect 3d printing will be explored.

To begin with, every 3d model is made up of polygons. Each polygon has at least

three edges and three vertices, but may contain more. Vertices are sometimes called

nodes or points. The polygons combine by connected edges and vertices to create an

overall shape or mesh. For the case of 3d printing we are only interested in completely

enclosed 3d meshes. That means that every single edge is connected to a polygon on both

sides. The object should be completely self-contained without any holes or gaps into the

interior of the mesh. While this requirement is not always necessary when modeling it is

required for 3d printing technology to read an inside vs an outside of the object for

printing purposes. Faces, edges, and vertices of a simple cube are highlighted below in

Figure 3.2.
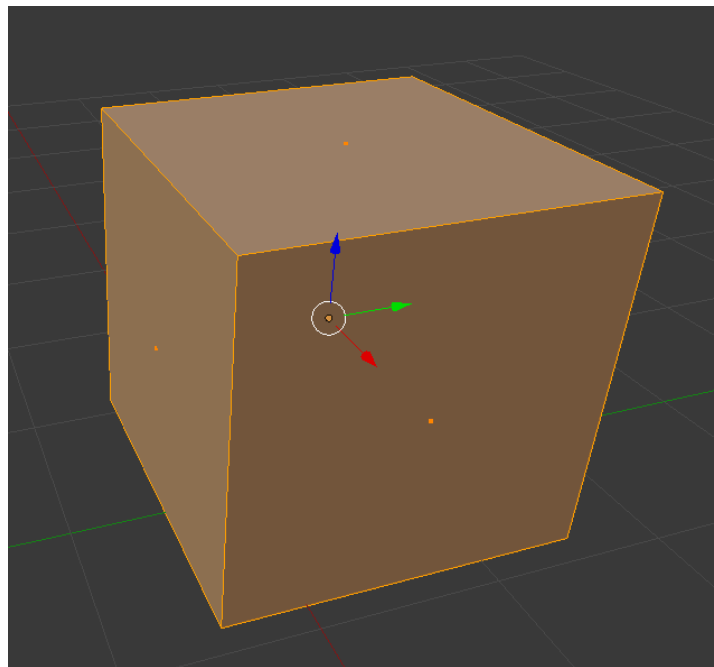
Figure 3.2 Modifiable Points, Edges, and Faces

Each face, vertices, and edge can be moved, split, cut, etc. to create more complex

shapes from an existing mesh. Within 3d modeling software it is the norm for complex

22

"modifiers" to exist. These allow for advanced modifications to be made that would be slow or difficult to perform by hand. For example the extrude command would allow for one or more faces to be extruded directly outward in such a way as to create a new set of edges and vertices to support the face(s) that are now shifted outwards. Boolean addition, subtraction, or intersection are all ways to combine simple meshes into a single more complex mesh. There are numerous styles for modeling meshes and oftentimes many ways to accomplish the same exact task. The ultimate goal here though is to produce a single 3d model/mesh that can be easily interpreted by a slicing program.

Having multiple meshes that overlap can prove to be troublesome for many slicing programs. In Figure 3.3 below, two cubes heavily overlap. When the slicing program hits a polygon face it may assume it is now inside an object. When it hits another face it may assume it is now outside the object. For overlapping meshes this becomes a problem.
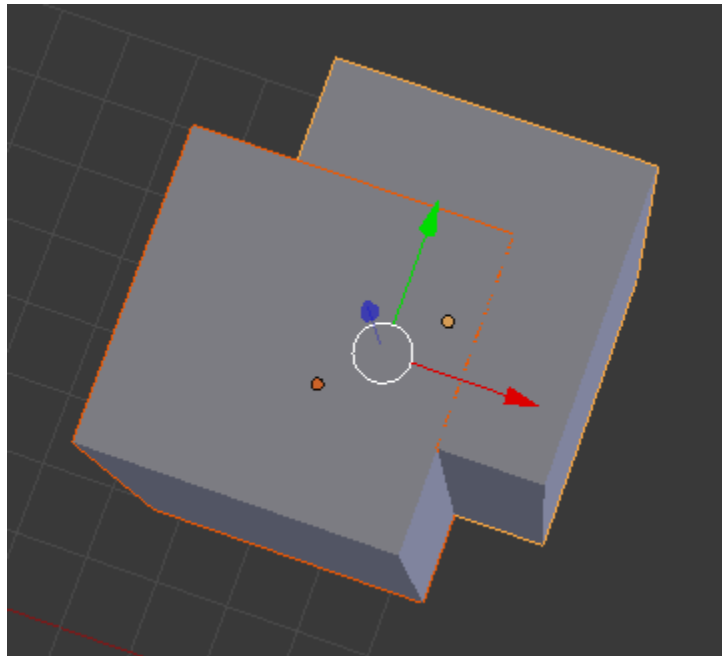
Figure 3.3 Two Overlapping Meshes

Because two polygons overlap in the center of these two cubes, the slicing program may assume that this area is "outside" of a single mesh rather than inside two separate meshes. This can and does result in a faulty print like the one shown below in Figure 3.4. Many slicing programs now include an option to scan and fix for such issues, but this takes additional time and processing power. The safe route is to ensure that the "object" to be printed is not made up of numerous sub-meshes. Joining each object with Boolean addition could be one solution for combining separate overlapping meshes.
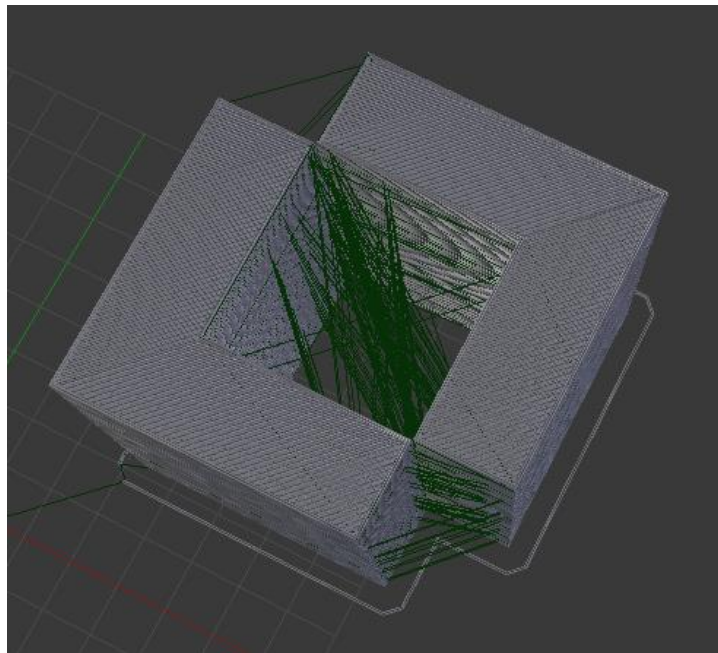
Figure 3.4 Overlapping Meshes after Slicing Process

Once the 3d mesh is completed, it is ready to be passed to through a slicing program to generate the G-Code instructions that will allow it to be printed. This sounds like a simple one way process, but in reality it is common to return to the mesh to make adjustments and alterations. The majority of corrections will happen here at the model

level or with the slicing settings in the next step. It may also be worthwhile to keep

multiple versions of the mesh that is being edited as many alterations may have

unforeseen consequences. In fact often times a mesh may be accidently "broken" through

an export process or a faulty modifier at which point rolling back to the latest stable

version of the mesh is usually the best choice.

3.3 Slic3r's Approach to Generating G-Code

While Slic3r is not the only slicing program, it was the first publicly available

slicing tool and dominated the market for years [56]. Due to its open sourced nature, it is

possible to dissect and evaluate its object processing algorithms to better understand how

G-Code pathing is generated from a 3d model.

As a whole the process flow of Slic3r can be seen below in Figure 3.5. Slic3r

receives input parameters including object file(s.) These inputs may also include

instructions to construct a raft, supports, etc. Typical input variables required to prepare

the instructions for a specific printer can be seen below in Table 3.1.

Once all the inputs have been supplied the slicing process can begin. Firstly

instructions will be output to the G-Code file for the startup script. This is modifiable

script that does basic setup for the printer to ensure proper calibration and temperature

settings. Typical startup instructions include homing all axis, turning on fans, and so

forth.

The core of the Slic3r algorithm is the cross section analysis of the object to be

rendered. Based on the resolution, we step through cross-sections or "slices" of the model

and convert those to extrusion paths. We begin with the bottommost layer.

Output to G-Code file – Ending script (modifiable)

Slicing Program

External Input

Model File

Created by a 3D modeling program. Typically in STL, OBJ, or AMF format. Units can be arbitrary and have to be resized accordingly.

User Inputs Parameters

Detailed information on the filament, bed, extruder, are all required to properly output the instruction set.

Output Startup Instructions

Startup instructions typically include homing all axes, warming up bed, extruder, etc.

Slice Model

Starting at the bottom of the model create a cross section through the x-y plane. Next create instructions to path the extruder around the edge followed by the interior according to infill parameter *Misc build paths included.

Done with model cross-sections?

No

Yes

Output Ending Script

Finalize the G-Code instr. Instructions can include turning off the bed, extruder, returning the nozzel towards home, turning off fan, etc

Output

G-Code Output

Final G-Code Instructions. Can be machine dependent but luckily these have been standardized by the community.

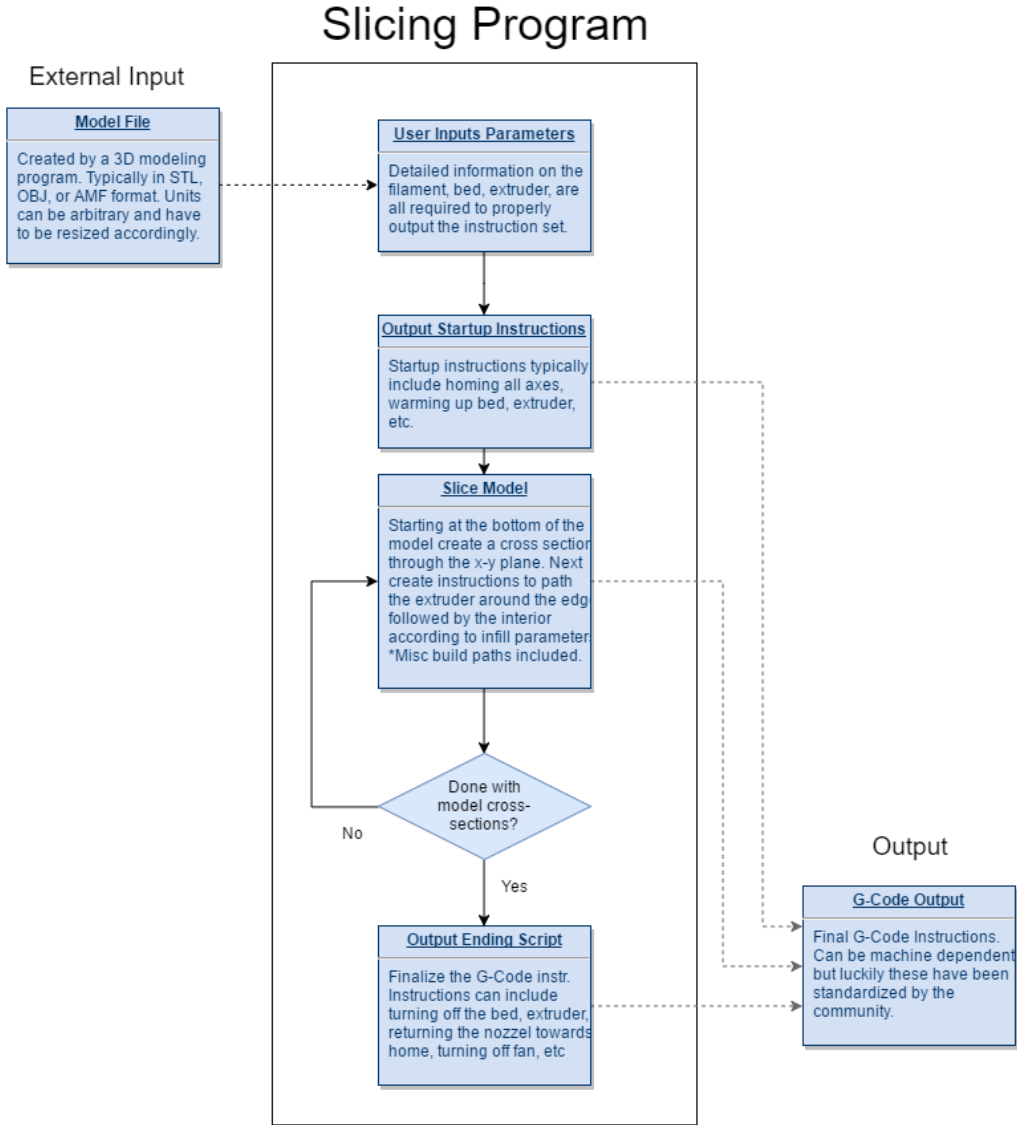Figure 3.5 Slic3r's Object Processing Outline
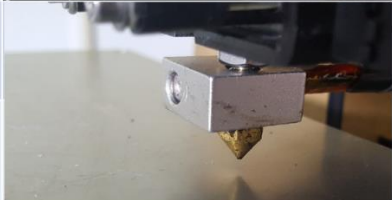
For each cross section an exterior path is created then an infill is applied. The infill will be solid for the top and bottommost layers. After each layer we step up in the z-axis to the next cross-section of the object, and continue scanning and outputting each layer/slice until all cross-sections have been evaluated and the results printed to the G-

Code file. Export to the G-Code file is performed in real time as the object is parsed.

Interruption of the process will result in a partial file.

Table 3.1

Typical Inputs Required for Slicing a 3D Model

	Parameters	Visual
Model	.STL .OBJ or .AMF Model Model Scale Rot. & Pos. Slice Height (Resolution)	
Filament	Material: PLA, ABS, etc Diameter: 1.75mm, 3.0mm	
Extruder	Diameter: 0.4mm Temperature: 190°C	
Bed	X Dimension: 120mm Y Dimension: 120mm Temperature: 60°C	

The slicing process can be difficult to envision. Below in Figure 3.6 is an example of a "slice" through an object that is to be printed. In this case the resulting cross-section would just be a square, but every overhang must be supported so in this case supports will also be included for the cube that is not yet part of the cross-section.
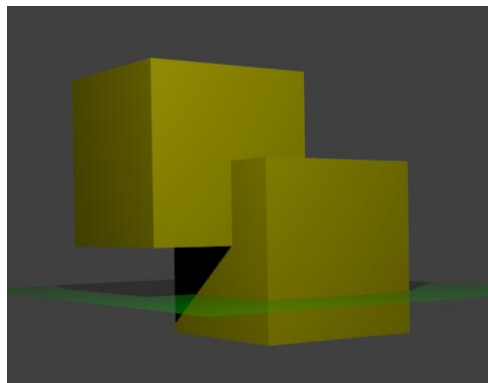
Figure 3.6 Visual of a Plane or Cross Section of an Object

The resulting printed layer of the above cross-section can be seen below in Figure 3.7. The infill type for this object is rectilinear meaning the printed infill paths from top left to bottom right for this layer but in the next layer it will run top right to bottom left in direction to reinforce the interior of the printed object. The resulting combined layers form the body of the print. The final results of the cubes in Figure 3.6 can be seen below in Figure 3.8.
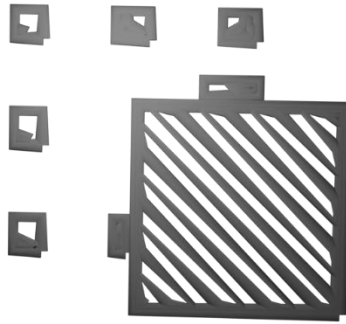
Figure 3.7 Printed Layer from Cross-Section in Figure 3.6
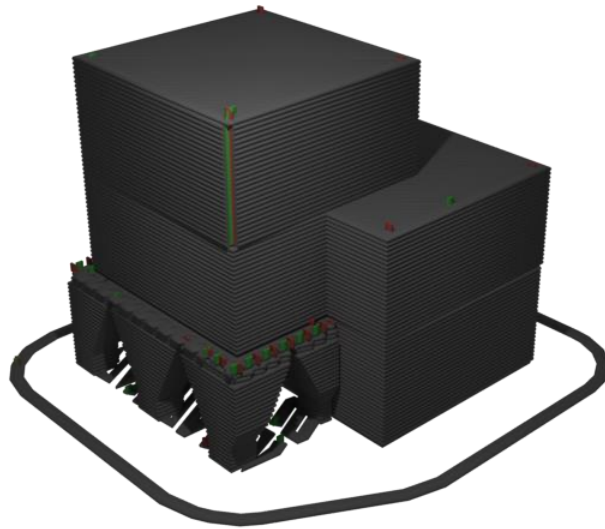
Figure 3.8 Completed Printing Paths for Shape in Figure 3.6

The supports pictured in Figure 3.8 would be sacrificial and should snap off with a little effort. The instructions to print the supports should attempt to reduce adhesion between layer segments for those portions of the print.

Once the body of the print has been completed a script which details the final instructions for the 3d printer will be appended to the existing instructions. These usually include turning off any heated elements or fans and may also home the X axis in case any additional material accidently extrudes or so the heat of the nozzle doesn't melt any nearby printed segments.

3.4 3D Printing Process in Application

At its heart, 3d printing is about carefully controlling variables. The goal with verifying model integrity is to reduce as much risk as possible via preproduction analysis. 3d printing a concrete structure is a complex task that both combines the variable nature

of concrete with the daunting complexity of large scale 3d printing technology. Both of these two disparate worlds combine in a unique and challenging way. While it is tempting to focus solely on the 3d printing aspects of the task, that would underplay the complexity of working with concrete. Each different concrete mix will experience the chemical reaction that cures the material into a solid in its own way, and will also have different properties once hardened. Once mixed on site, a dozen more variables can and will influence the curing process and the final strength of the concrete. For instance adding too much water is a common novice mistake that can adversely affect the final cured strength of the concrete.

Similarly the setup, maintenance, and operation of a large scale Additive Manufacturing machine is a complex undertaking which can require difficult troubleshooting. Even with a desktop sized machine and an experienced user, acquiring a high quality print often requires several iterations of trial and error. Knowing which variables to tweak to optimize the final product can blur the line between science and art. Simplify3D is a commercial grade alternative to Slic3r, the original and perhaps most commonly used slicing program. Simplify3D has approximately 150 adjustable parameters under its advanced settings [43]. Despite, or perhaps because, the complex nature of the 3d printing, several online communities are eager to share information, models, and experience. Even still these online communities make up only a small portion of Additive Manufacturing as a whole.

Regardless of the print medium, certain key processes must always occur. All extruder based 3d printers must adjust the extruder nozzle with precision through the X and Y plane, before the nozzle steps up a layer. Stepper motors are precision electrical

motors that maintain accuracy over the course of operation through the use of magnetic "steps" that help ensure that no cumulative offset occurs [60]. At least one motor is required per axis, but more can be used as mechanically necessary.

Below in Figure 3.9 the three distinct axes are labeled. An X and Y motor each control an axis then a pair of axes step the X axis and extruder up through the Z axis. For this printer the print bed as a whole is moved for stability, but for larger 3d printer it would be necessary to move the frame on both sides through the X axis. This more complicated movement would require two motors rather than the one pictured here. To ensure that the motors know when they have found the origin, or zero position, end stops sensors detect right before each motor would cause a collision along each axis.
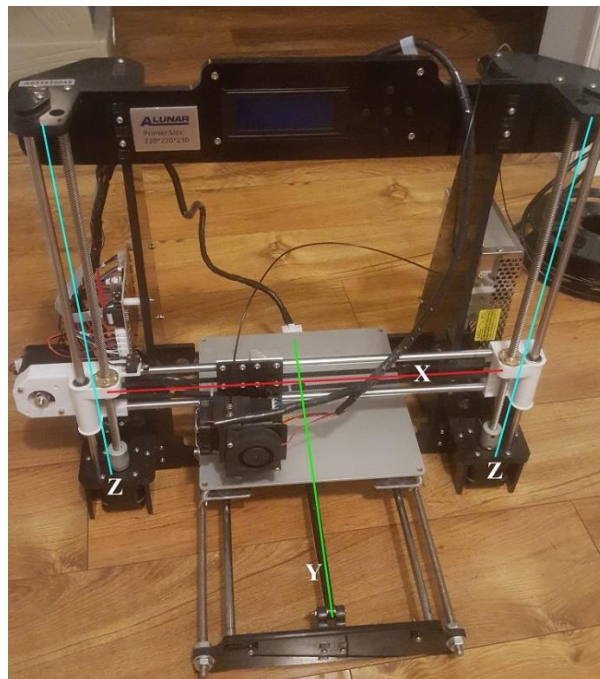
Figure 3.9 Small Scale Prusa i3 3D Printer

There are two ways that the stepper motors transfer rotational force into linear motion. For the X and Y axes, belts with teeth are used to shift the extruder or the print

bed along each respective axis. An example of the belt based stepper motor can be seen
in Figure 3.10 below.

Figure 3.10 Y Axis Motor for Prusa i3 3D Printer

The second method for converting rotational motion into linear motion can be
seen below in Figure 3.11. The threaded rods pictured here work in conjunction with the
brass lock nut shown here to move the X axis mounting bracket. Counter-clockwise
motion moves the mount down in the Z axis while clockwise raises the mount.
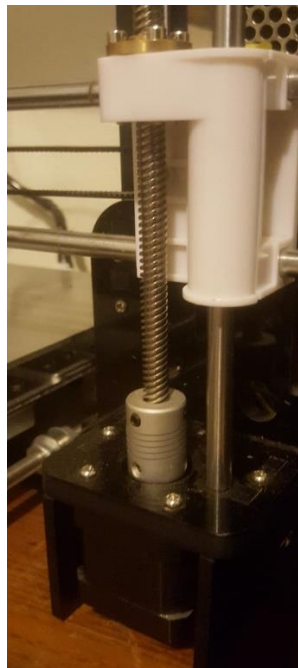
Figure 3.11 Z Axis Motor for Prusa i3 3D Printer

Despite these motors being used for a small scale printers, the concepts are scalable up to NEMA 34 sized stepper motors. While stepper motors are made larger than the NEMA 34, their large size reduces their effectiveness to dissipate heat. Hence the larger stepper motors are not designed for continuous use [60]. Because the mechanics for small scale printers have been scaled up for the basics of large scale 3d printing, many of the potential issues of the emerging technology have already been troubleshot through the small scale printers. The result is a series of lessons learned and optimizations that overlap into the problem domain of large scale 3d printing.

CHAPTER IV – RELATED WORKS

4.1 Lee Butler's G-Code Import Plugin

In 2014 Lee Butler uploaded to GitHub a plugin for Blender that would import G-Code files into Blender version 2.7.0 [25]. His Import plugin was based on a similar G-Code import utility also submitted to GitHub but by Simon Kirkby in 2011 for Blender version 2.5.0 [35]. According to the documentation in Kirkby's upload, his submission was in turn based on a DXF file importer for Blender.

Lee Butler's G-Code importer is a reasonably straight forward process, but the flow chart below in Figure 4.1 has been designed robustly so that additions to the process flow can be easily brought in without reworking the original algorithm's layout.

G-Code Import Process
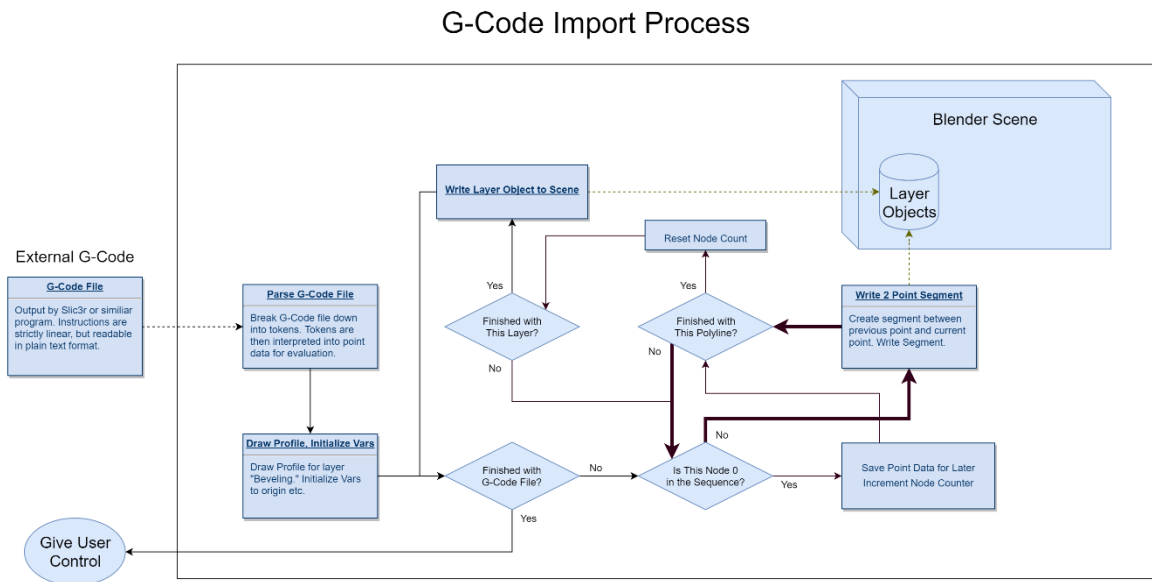
Figure 4.1 Lee Butler's G-Code Import Process

Once a user selects a file to import, the file is parsed in its entirety with each move point broken down by layer, segment, and point. A contour exactly like the one

shown in Figure 4.2 below is drawn to approximate a vertical cross section of each printed segment.
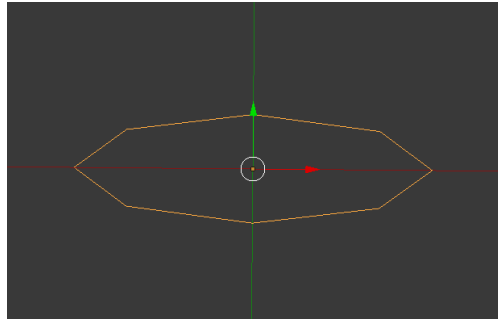
Figure 4.2 Layer Segment Profile

That profile is applied to the pathing that will soon be traversed in full. The resultant "tubes" can be seen below in Figure 4.3. They do not have any true volume as they are not enclosed shapes, and they also remain as polylines rather than meshes which limits how they can be manipulated. For instance materials cannot be applied to the shapes here to change the objects colors or properties.
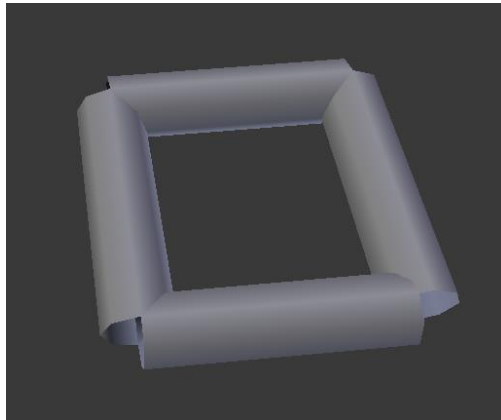
Figure 4.3 Profile "Beveled" Along Path

The bulk of the work is handled by the next portion of the import algorithm. Within each layer, each polyline segment is traversed, and within each polyline segment,

each point is also branched through. For any given point, if that point is the first point in a segment, it is just held on to for later. If it is any other point in the segment, a path is drawn from the previous point to the current point. As shown in Figure 4.3, this unfortunately creates discontinuous segments rather than a continuous multi-node path. The darker directional paths in Figure 4.1 represent this segment creation code which is the most commonly executed portion of the algorithm.

At the end of each segment the node count is reset, and at the end of each layer, all objects for that layer a created as a single object. The result is that if the G-Code file has 43 layers, 44 objects in the scene will be created including the profile object.

While Lee Butler's G-Code imported does create an approximation of the G-Code printing process, his approximation lacks precision at the start and end of every section of every segment. Also for rare occurrences where more than one empty path is taken in a row, the nodes between are discarded. His algorithm does do an excellent job of parsing G-Code with a few objections. The algorithm is prepared for Slic3r G-Code files. For Simplify3D G-Code files, the first node in each segment gets ignored. When the G-Code file is first pulled, all irrelevant data, comments, and blank lines get dropped. This is fine for the parsing process, but revisions cannot be made to the G-Code file to reroute empty paths or document layer number with comments etc. Overall Lee's G-Code importer may be a bit on the primitive side, but it was still a huge step forward for just getting the basic structure together to allow for better approximations and simulations.

4.2 Slic3r's Overhang Detection Process

As previously mentioned, one of the methods proposed for model verification is overhand detection and analysis. While this is not a new method, it is a new way of performing the verification. Slic3r along with most other slicing softwares, detect overhangs so that they can print supports for those sections. Slic3r is specifically being discussed here because of its open source code that allows us to confirm any assumptions about its overhand detection algorithm [56].

Slic3r analyzes potential overhangs based on the angle of overhang of the outer shell pathing of one layer compared to the outer shell pathing of the layer immediately beneath it. Infill can be safely ignored as each infill method ensures that proper support will be available. The key measure here is the angle which seems to work fairly well for small prints. The acceptable angle of tolerance for any given print can be controlled by the user which allows the user to determine the amount of supports necessary for any given 3d printer. Even on the same exact printer, extrusion speed and fan speed can influence what level of overhang is successful.

A print like the one seen below in Figure 4.4 shows various overhang angles and how successful they were over a given series of layers. Note that the elastic nature of heated plastic leads to defects rather than outright failure for this print. Two major factors may play a role in the fact that many overhangs begin within the acceptable tolerance, but as layer builds upon layer, defects begin to emerge. In fact even the 70° overhang section begins stable enough. First the "top" of the object here was the bottom when printing so it was in contact with a heated bed. The heated bed keeps the plastic from shrinking overly much while still allowing it to solidify. The second factor, which mostly likely combines

37

with the first, is cumulative error. Once the plastic begins curling up at the edge, the next layer has a non-leveled layer to print on. This results in the next layer curling up even further with the defect cascading down through all layers printing above the existing defect. The effect begins at 30º, but is more noticeable at 35º and over. While the printer used to produce this test print did have a heated bed, the inclusion of a contained heated print space would certainly improve overhang print quality as temperatures would normalize throughout the enclosure.

Figure 4.4 Overhang Test Print for Increasingly Steep Angles

Using angles to detect overhangs works reasonably well for the context of small plastic prints, but as scale is increased total overhang distance should correlate more strongly with total overhang distance. For example, if each layer for a large concrete print

is one inch tall or two inches tall, the total safe overhang distance shouldn't change at all. Angle prediction should suggest that two inch thick layers would result in a higher total safe overhang distance. Of course with layer height and angle both accounted for, it would be possible to determine total overhang distance, but there are a few more potential issues even if this alternative calculation was used.

All calculations observed within Slic3r rely on the absolute path information. Because complex 3d shapes are actually at play here rather than just the pathing information, the pathing data may prove insufficient to properly measure overhang distances for certain scenarios. Most importantly the entire process is opaque to the user. While it can be implied which sections required support by the placement of the sacrificial scaffolding, a 3d visual is not supplied. All together Slic3r and many other slicing softwares perform admirably for small scale printing, but as prints become increasingly large, like they do with large scale concrete printing, total overhang distance and visibility become increasingly important.

4.3 G-Code Analysis Tools

With many commercial entities pushing to break ground in the field of 3d printing concrete structures, there are certainly verification methods being developed that are not being made public. Currently there is little in the way of verifying model integrity prior to the printing process open to the public at this time. However, Andrej Cupar published an interesting paper back in 2015 for verifying model integrity after the print process is completed. This is accomplished by 3d scanning the printed object and comparing its physical structure back to the original 3d model [54]. While this is an interesting take on

39

model verification, our goal is to verify model integrity before the printing process ever occurs. In this section, we will investigate Simplify3D and GCode Analyzer as tools which allow for detailed visual inspection of the model prior to printing [43, 55].

Despite the fact that Simplify3D costs $150 in a market with over a dozen free slicer softwares, it still remains one of the chief competitors in the field [57]. Besides the fact that it produces excellent G-Code instructions that can improve print quality, it also offers one of the most detailed simulations of what the G-Code instructions will look like before the file is exported. Below in Figure 4.5, detailed pathing for both empty paths and extruded paths can be seen. Each path is color coded by how fast the nozzle is moving giving a detailed breakdown of where the print process is being the most careful.
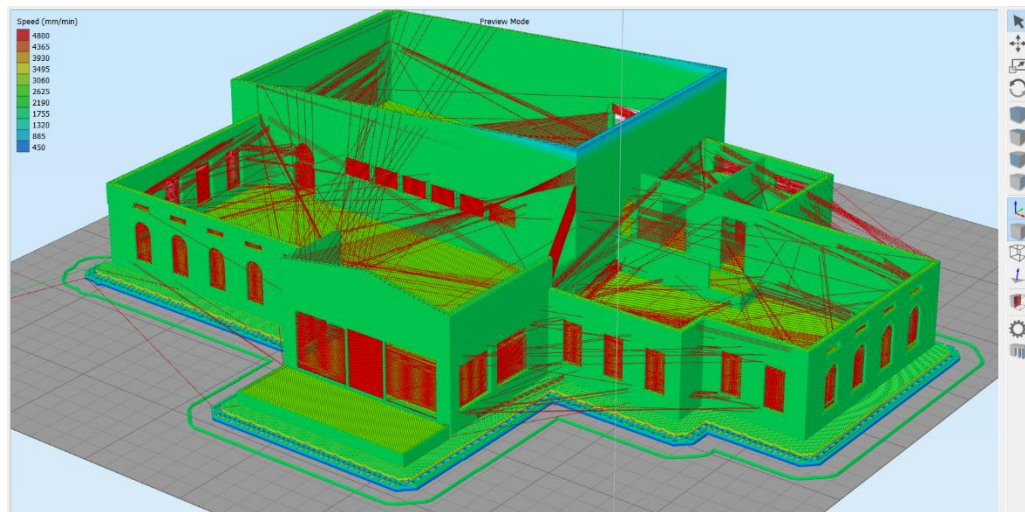
Figure 4.5 3D Visualization by Simplify3D

Besides the detailed pathing, the granularity for the playback is phenomenal. Sliders can be used to hide the beginning or the end of the print and additional step forward or step backward options will allow the user to see each and every point that is being printed one step at a time. A detailed view of a model at roughly 60% completion

can be seen below in Figure 4.6. These print simulations would qualify as a 4d print process that can be evaluated forward and backwards with ease.

As an overall review of the software, models load quickly and G-Code files export quickly. The software works well for what it is designed to do. Unfortunately Simplify3D was designed as a slicing tool and not a verification tool.
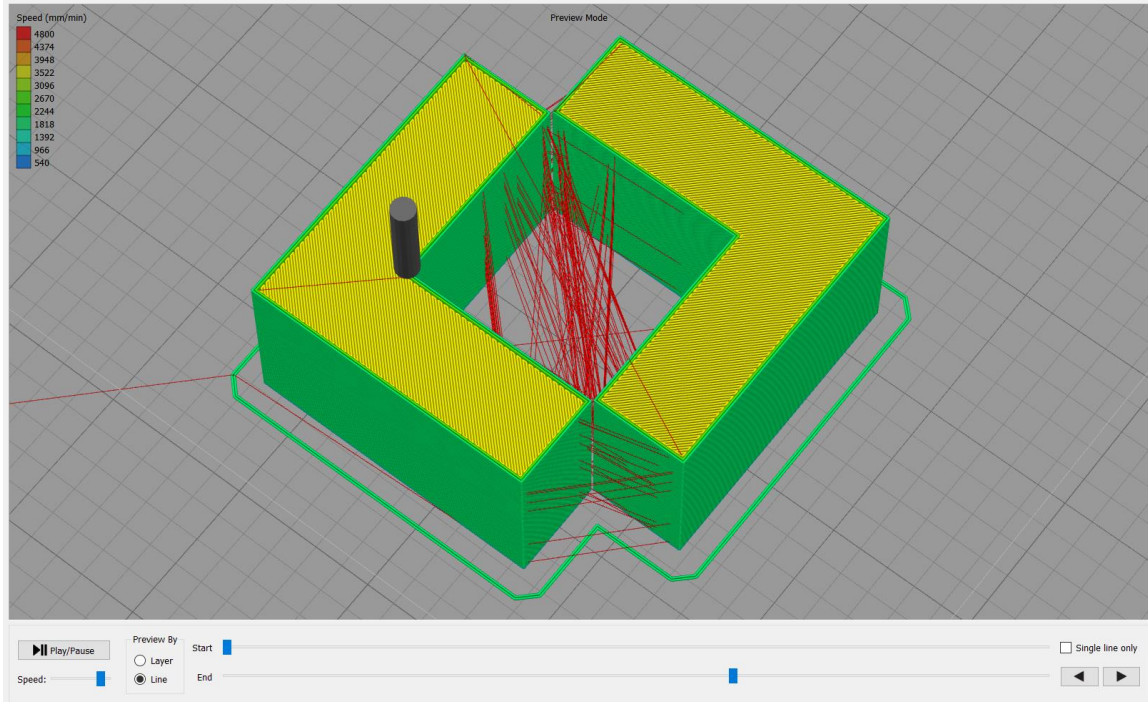
Figure 4.6 Simplify3D's 4D Printing Process

Simplify3D excels as a tool for visual validation and a slicing tool to generate G-Code, but verification is a manual process. Overhangs seem to be calculated in much the same way that Slic3r checks for them, but again no visual feedback is ever supplied to the user. Possible collisions are ignored as the print bed is assumed to be free of obstructions. Potential shrinkage, the plastic equivalent to concretes thermal stresses, are also ignored. It is honestly unfair to judge the software on the basis of what it wasn't designed to do,

but the evaluation is made here on the grounds that no software currently available performs detailed model verification prior to the print process.

Another app called GCode Analyzer is specifically designed to let users review their G-Code files to gain valuable insight into what the final product may look like [55]. GCode Analyzer is a web based app that lets the user upload a file for full analysis. It has both a 2d and a 3d view and also supplies detailed feedback such as dimensions, materials used, and time to complete which are all useful. Below in Figure 4.7, a detailed layer breakdown can be seen. Like Simplify 3d, path speeds are shown by color and the print process can be stepped forwards or backwards with precision. On the right a slider lets the user step up or down through the layers. Unlike with Simplify 3d, playback options are only available for the 2d view and not the 3d view.

Figure 4.7 GCode Analyzer's Layer Print Analysis

The 3d view for GCode Analyzer, shown below in Figure 4.8, is a bit underwhelming after looking at Simplify3D. It shows the paths for all printed segments, but they have no physical volume; the paths are represented by single pixel lines. No print speeds are indicated. No time-lapse or 4d play back is available. While functionality of the 3d view is lacking, it should be mentioned that the software is available for free online compared to Simplify3D's $150 price point.
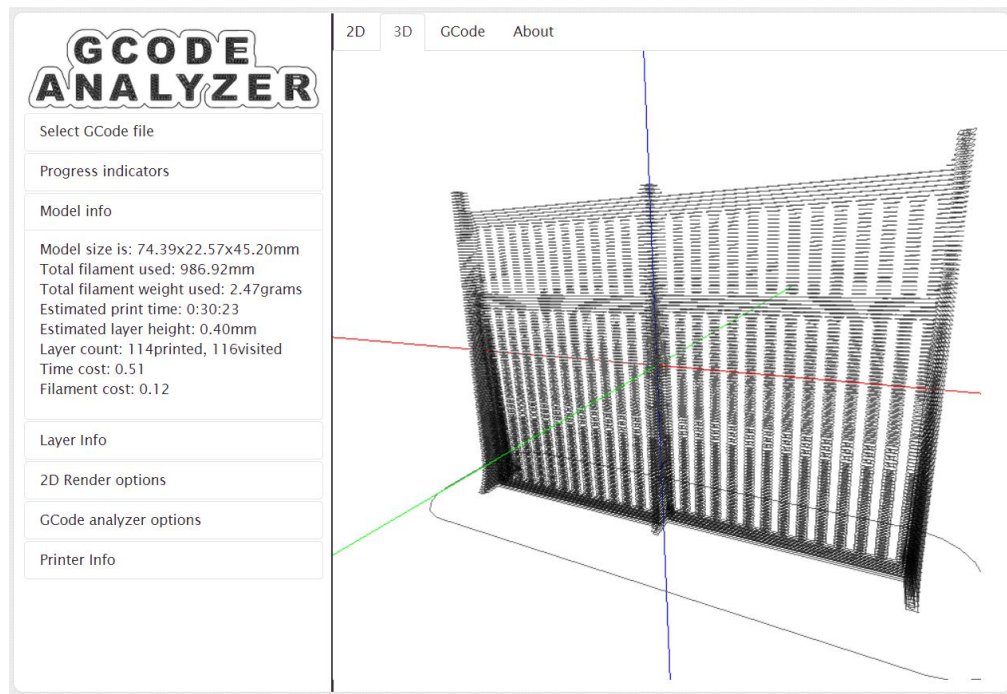
Figure 4.8 GCODE ANALYZER's 3D View

GCode Analyzer does provide simple data for the uploaded model, but lacks any deep analysis. Summing the total filament used or calculating total time based on move speeds is trivial given the information is already contained within the G-Code file.

In conclusion Simplify3D does an excellent job within its designed intentions. GCode Analyzer lets the user review their G-Code file by hand, but "analyzer" is a bit of

a stretch. It can still be a useful free tool to quickly look over a model online. While it may be unfair to evaluate the tools above based on what they don't do, in depth model analysis and verification are both lacking at this time. It seems model verification is still in the early stages of research. This could be due in part to plastic prints being so cheap and easy to produce.

CHAPTER V - PROPOSED METHODS FOR VERIFYING MODEL INTEGRITY USING DATA-CENTRIC ANALYSIS AND SIMULATIONS

Blender, which is an open source 3d modeling program, is built on Python code which can be entered into the console or used to run code outright. This makes for an excellent platform to import, manipulate, and export both models and renderings. Simon Kirkby originally wrote a G-Code importer for Blender which would take the G-Code absolute coordinates from the text file and create corresponding path lines. Additional tweaking and cleanup was performed on the open source project. Kirkby's importer served as the initial code base for simulating G-Code pathing to predict the results of the execution of said instructions. As research and simulations expanded and become more complex, extensive rewrites were necessary to accommodate the change requirements.

Five methods of verifying G-Code instructions have been proposed, but only difficulties relating to three of those were self-evident. Visually inspecting the design elements goes without saying. There is no replacement for having an expert review a design. Likewise collisions with existing objects, plumbing, or supports is an obvious concern. The difficulties regarding overhangs and the extrusion based AM methods only become obvious once a person is familiar with the 3d printing process and its limitations. Certain specific problems only become apparent when dealing with large scale concrete AM processes. Andrey Rudenko is responsible for pointing out the issue of thermal stress forces especially along long walls. He is also responsible for pointing out the when extrusion end points align vertically they can cause a "seam" that can be a weak point or visual defect. Based on the additional information of the problems Rudenko had encountered, the last two methods for verifying model integrity were developed.

5.1 Virtual Simulation of the Printing Process

G-Code instructions lack any abstract hierarchy. That is to say, the instructions are not organized into blocks or have headings to hint at where each instruction may fit into the print process as a whole. The file is a series of instructions which will be executed in a strictly linear fashion. For detailed analysis, it is preferable to reintroduce the hierarchy of layers and segments to the existing point data found in the G-Code file. Simon Kirkby did exactly that with the G-Code import code he wrote for Blender in 2011 and uploaded to GitHub.com [35]. It was updated by several other contributors before it was eventually left abandoned. In like manner, Lee Butler used a matching organizational structure for his version of the G-Code import plugin for Blender [25]. Layers are broken down by polygons, which are in turn broken down by points.

It was Lee Butler's import plugin that acted as a foundation for simulating the print process. While the import plugin did create a reasonable approximation of the printed segments, numerous improvements have been made. The most notable alterations will be discussed here. This modified G-Code import plugin for Blender is the foundation for additional calculations and simulations for verifying model integrity.

The process flow for the modified G-Code algorithm can be seen below in Figure 5.1. It builds directly upon Lee Butler's G-Code importer with numerous alterations. Firstly the layer by layer printing approximations have been drastically improved more closely follow a 3d printer's extruding process. Secondly many additional objects and paths have been incorporated to allow for more complex analysis. The additional scene objects include empty paths, start and end points for pathing, overhang layer paths, and overhang areas for the start and end caps as well.
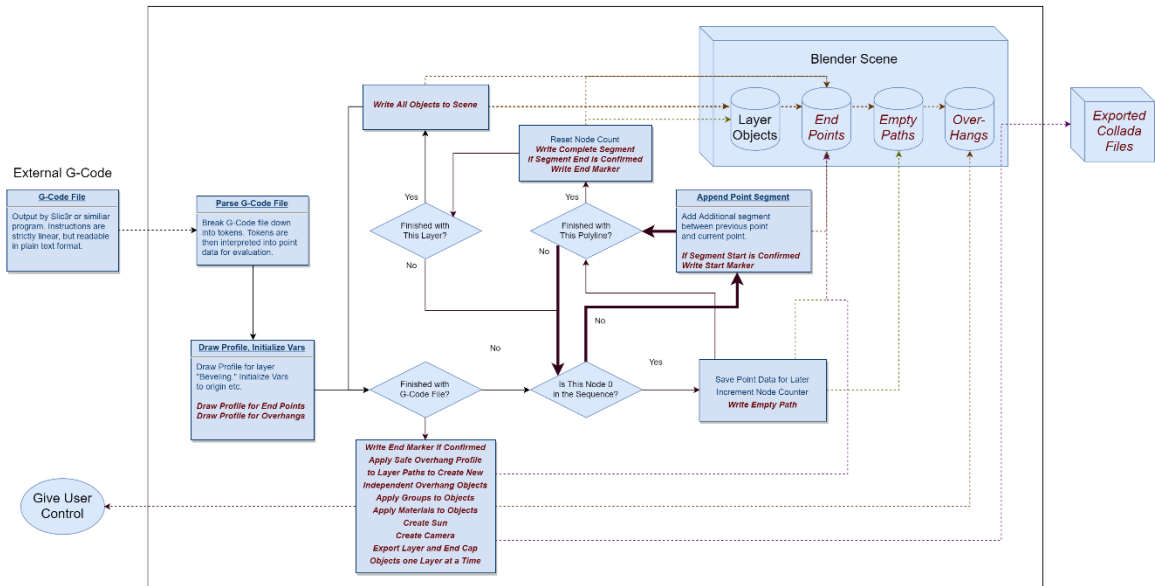
Figure 5.1 Modified G-Code Import Process

An improved profile was created to better fit the contour of printed concrete. The new profile can be seen below in Figure 5.2. While it is just a rough approximation, it is much more accurate than the previous profile. Even for similar mixtures of concrete, the typical profile can change depending on materials and mixture ratios. Only eight points are used for the profile here, but more could be used. Additional points will also increase the polygon count for the entire scene potentially causing more slowdown.

Figure 5.2 Profile for Concrete Material Approximation

When the new profile is "beveled" along a path it creates a 3d object that more closely approximates the shape of slumping concrete. The sudden cutoff at the beginning and end of each segment is still unrealistic but will addressed shortly.
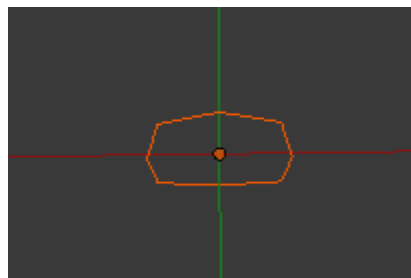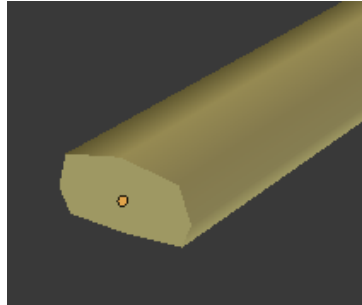

Figure 5.3 New Profile "Beveled" Along Path

With the original import plugin, the print segments were "broken" at each mid-point, meaning each path between any two points was its own complete path unconnected from the next in the sequence. For example a continuously printed segment from A to B to C was originally represented by two separate segments of A to B and B to C. Segments are now continuous to more accurately reflect the printing process and increase visual accuracy. Each "elbow" or joint creates a smooth visual bend. As an example, Figure 5.4 represents the original geometry created by the parsing process and Figure 5.5 illustrates the cleaned up simulations which better approximates the actual print geometry.
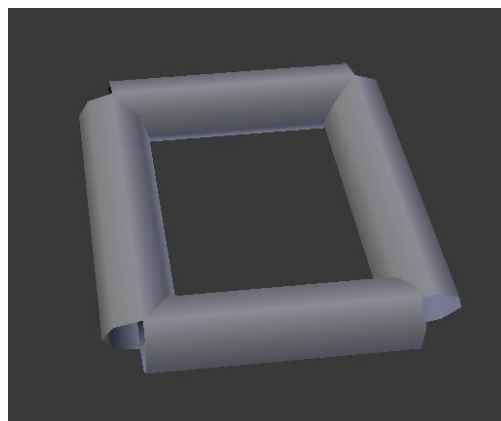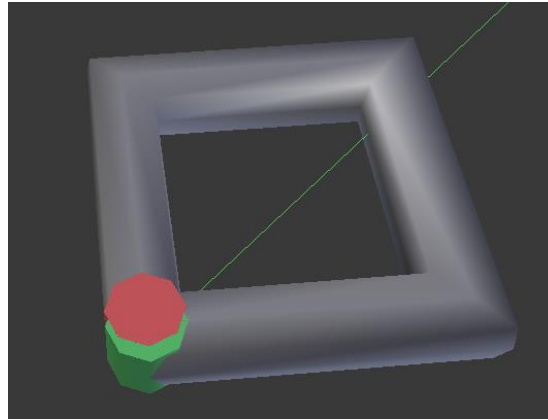
Figure 5.5 Improved Segment Simulation

The green and red octagons represent the start and end points of the printed

segment. The green line represents the empty path of the extruder leading up to printing

the object. The corners of the actual print would be more rounded than the sharp edged

corners shown here so the approximation is not yet perfect.

Having smooth rounded joints that properly represent the product of a round

extruder nozzle greatly increased the polygon count, and overall proved too

computationally costly for such a small improvement. Below in Figure 5.6, the improved

layer objects for a section of concrete wall can be seen. These layers represent the bulk of

what would actually be printed.
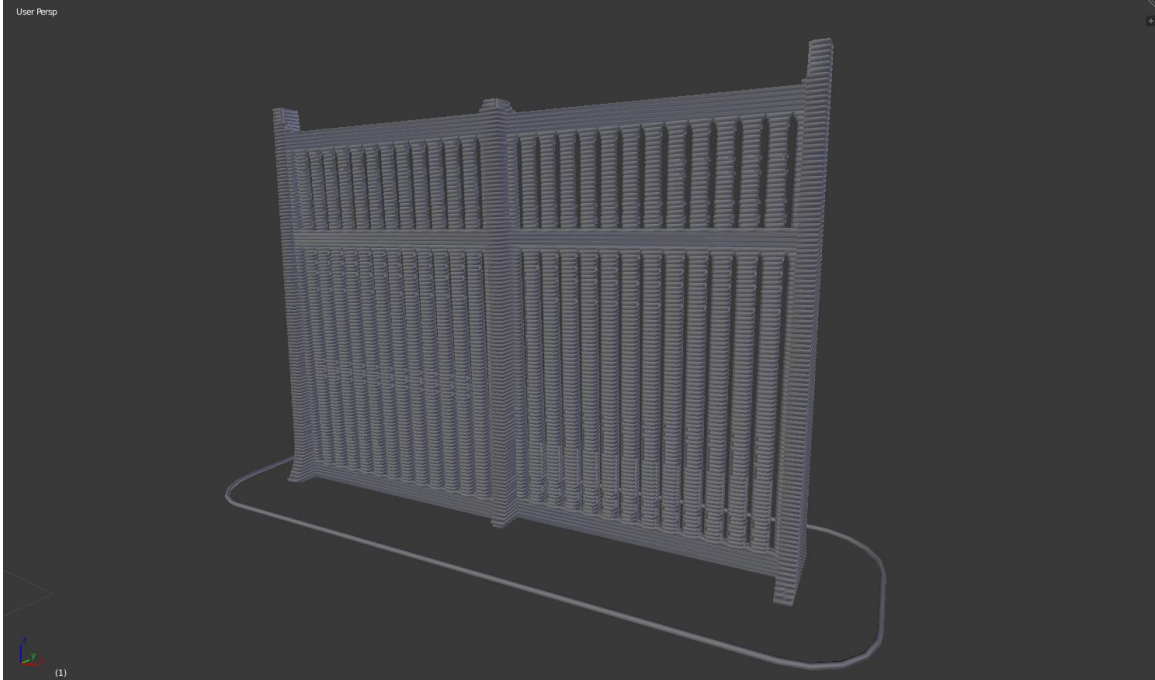
Figure 5.6 Simulation of Printed Layers

Several additional visual elements have been introduced to provide further information and simulation assets. Most extruders have a circular nozzle. The original pathing would terminate at the center of the nozzle for each path effectively subtracting half a nozzle width at the beginning and end of each segment. Start and end octagons were created as approximations of the round nozzle. It can be seen in Figure 5.7, they are color coded red and green for the end and start points respectively. During rendering of design documents their materials would typically be set to the default grey like the rest of the layers during rendering. As mentioned previously, they are representative of the

50

physically printed path ends.
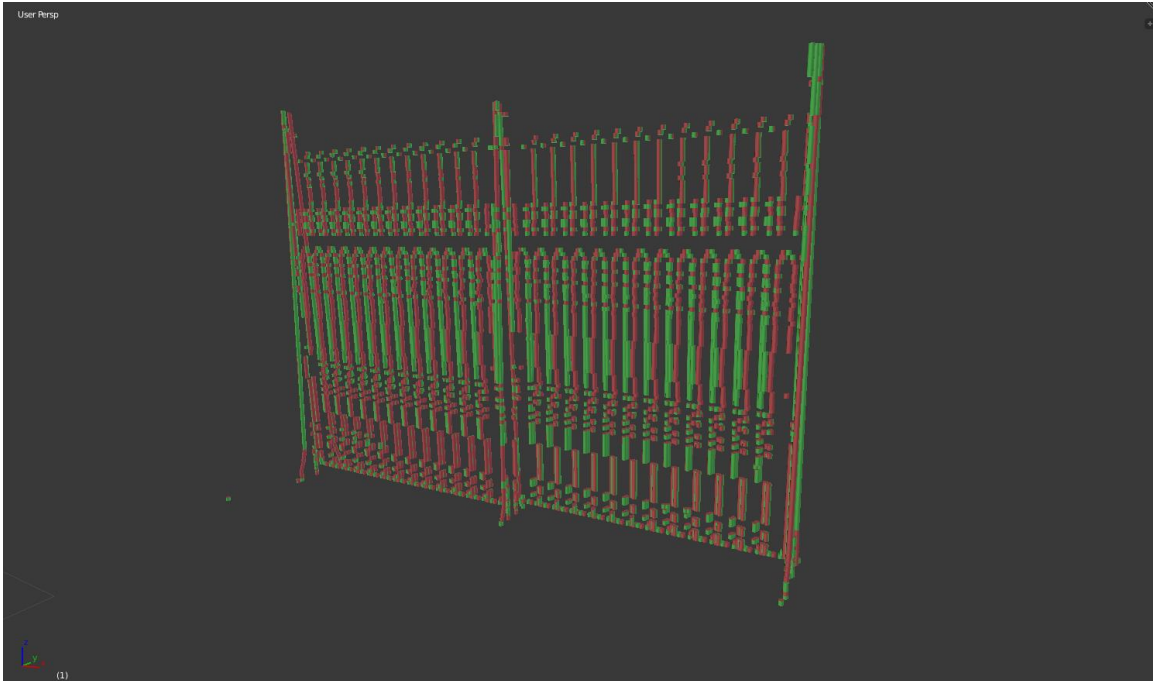
Figure 5.7 Start and Stop Markers for Print Paths

Empty paths were added to show locations the extruder moves through when not printing. Not only is this information important for detecting potential collisions which will be discussed later, it also helps establish print order of segments. Because no geometry is attached to these lines, they are not visible during renderings, but can be seen during evaluation within Blender. The empty paths may not be visible as collada objects if exported and imported by another program. Despite how the empty paths appear below in Figure 5.8, these lines have no volume or depth or materials. Because the paths have

no mass, each line is represented by roughly a one pixel wide line.
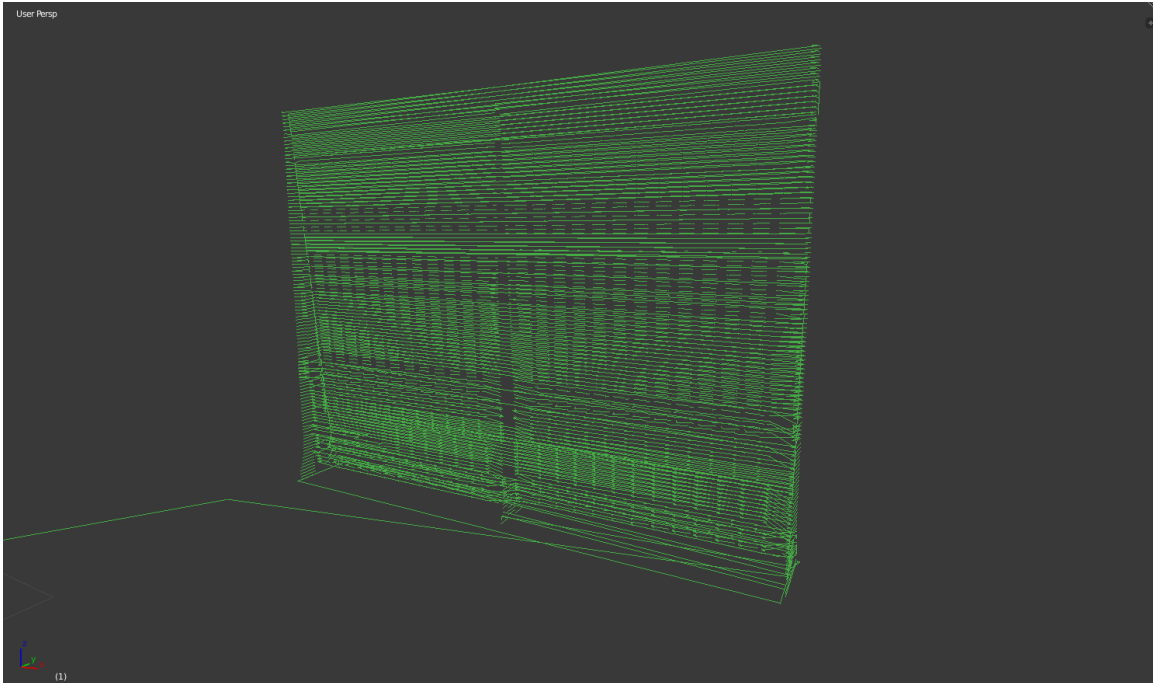
Figure 5.8 Empty Paths Traveled by Extruder

When combined into a composite scene like the one illustrated below in Figure 5.9, a four dimensional printing process can begin to be interpreted. An empty path moves in from the origin. A brim is printed in like every other segment. The green octagon represents the start location followed by the body of the print segment and finally capped with an end marker. All printed segments will follow this pattern. By following each path in the simulation one layer at a time, the exact path the printer will take can be traced out. When all object types are combined in a scene, the result is the composite image pictured below. Because so many segments and details are partially or fully obstructed by other layers, a time lapse animation can be implemented in order to provide

a detailed unobstructed view of the process as a whole.
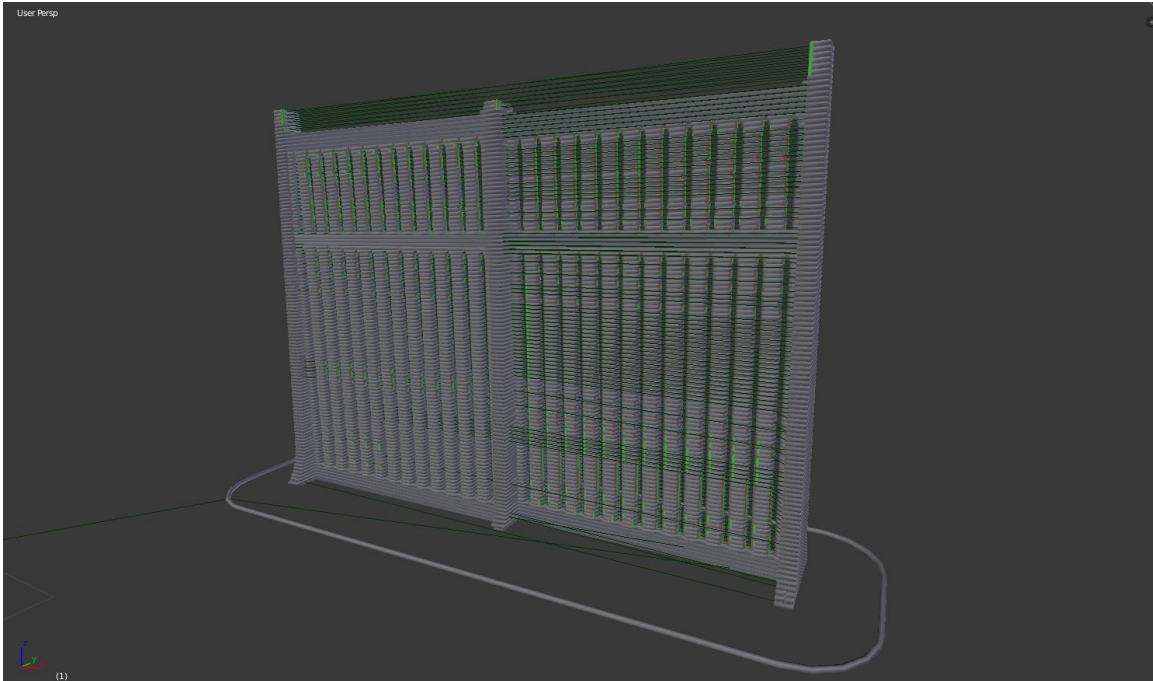
Figure 5.9 Composite Scene

The last design element of geometry introduced is created based on the same pathing as the layer objects. Whatever the safe overhang distance of material being worked with is added to the width of each segment. This safe overhang distance will be user defined based on the exact properties of the printing materials. The safe overhang distance creates an approximation of available print area for the layer immediately above each overhang object. This process will be discussed in further detail in section 3.6.

With so many potential objects entering a scene, Blender can struggle to process all of the data depending on how each object is defined. For small quantities of objects, creation time is roughly linear, but as the number of objects continues to double, total time to parse all objects becomes far greater than the linear 200%. In Table 5.1 nine different quantities are examined over two trials. Additionally the data for both trials is

53

graphed out in Table 5.2. The exponential growth to time as the number of objects

increases restricts the total number of objects that can reasonably be loaded into the scene

at any given time.

Table 5.1

Total Time to Create Objects for Various Quantities

Object Count	Trial Avg. Time	Percent Increase
16	0.025	NA
32	0.046	180%
64	0.093	201%
128	0.201	215%
256	0.541	276%
512	1.865	342%
1024	9.680	510%
2048	63.605	665%
4096	638.915	946%

While each print segment or polyline can be treated as its own object, this level of

granularity is not necessary unless data is required for a time lapse playback for each and

every node for each layer. Within Blender, a more appropriate method for grouping

segments into a single parent object is by layer. Layer by layer granularity can still allow

for a 4d playback as well as overhang detection as described in Section 5.6. Grouping all

layers into a single larger object has shown very little benefits, but could prove useful for exceptionally tall geometries.

Table 5.2

Time to Create Objects Based on Object Count
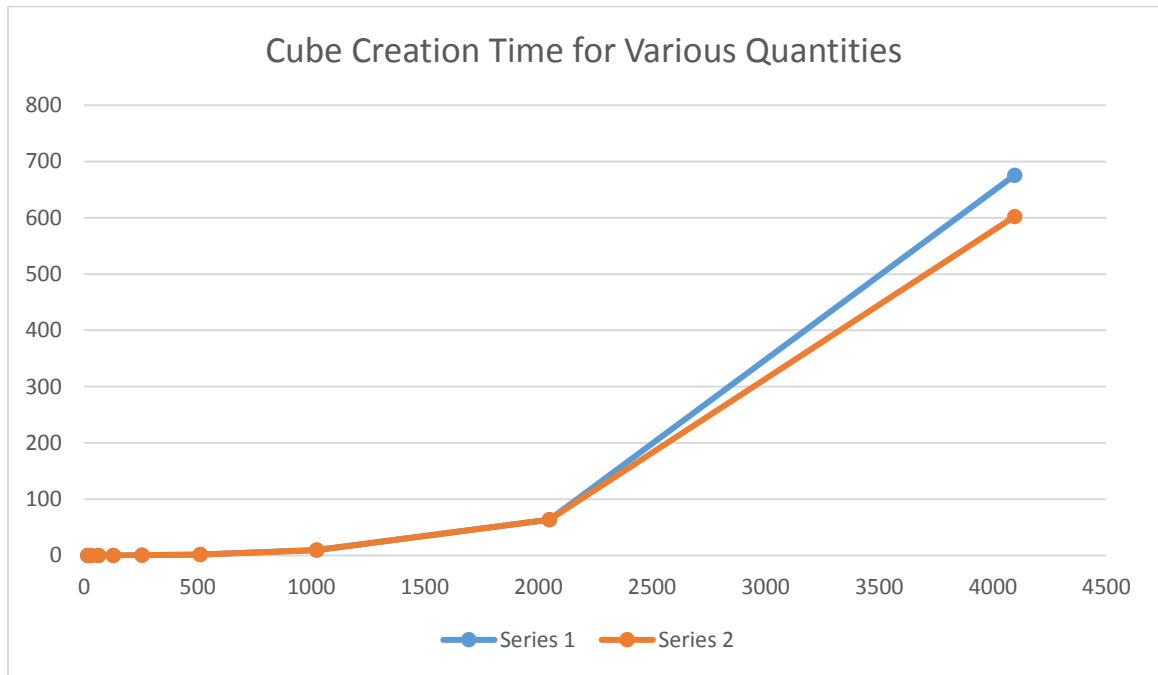
The two trials in Table 5.2 were roughly identical for both datasets except the last and slowest time. The graph shows the beginnings of an exponential function which could easily take several hours or days to complete if the number of objects were doubled a few more times. The trials involved very simple cubes as the base object, but object complexity will of course factor heavily into Blender's limitations.

The overall optimal object granularity has proven to be at the layer level so that the total number of objects should rarely exceed 1000 total objects. Besides object

granularity there are a few additional tweaks to the parsing algorithm that have proven to be useful.

Grouping like objects has also proven to be a large time saver. By placing all overhang objects in a group, the entire group can be hidden or deleted at once if need be. When dealing with hundreds or thousands of objects, quickly modifying large groups can be a huge timesaver. Likewise naming objects starting with their layer number keeps objects physically located on the same layer together in scene graph. Including leading zeros insures that layer two does not come after layer ten.

One inaccuracy which still exists in the simulation process is the nozzle location in relation to the print segments. In reality the extruder or nozzle is immediately above each printed segment. In the simulation, the extruder path actually is pathed along the middle of printed segment. This is a side-effect of extruding a profile along the path lines. It is not a serious defect, but the half layer of thickness must be considered when calculating the exact bottom of the extruder. Because every single path is "off" by half a layer, the uniform inaccuracy can easily be accounted for when checking for collisions.

All of the elements or actors described above act as a structural simulation of the proposed 3d printed object for further calculations and algorithmic analysis. There are five methods of validation that will be explored, each of which utilizes the objects created in this section as described in the above steps.

5.2 Visual Inspection of Design Elements

G-Code in and of itself can be read, but it's impossible to visualize a complex object that would be created by that code without help. Using the G-Code to simulate

how the finished print would look is the base for all methods of verifying model integrity. Visually inspecting the design elements is obvious, but such a necessary and powerful tool that it can't be overlooked. As stated before, there is no substitute for an expert looking over a design. Often times even a casual observer can easily identify defects or issues with the instruction set.

"Spencer's" was a restaurant designed by the author many years ago and acted as a test model throughout the research. Below in Figure 5.10, some steps were never fully pathed out during the slicing process. Permanent solid steps should be a perfect candidate for 3d printing, but the steps pictured here turned out to hollow. Removing the steps from the original model fixed the issue, but replacing the steps with solid ones would have also worked.
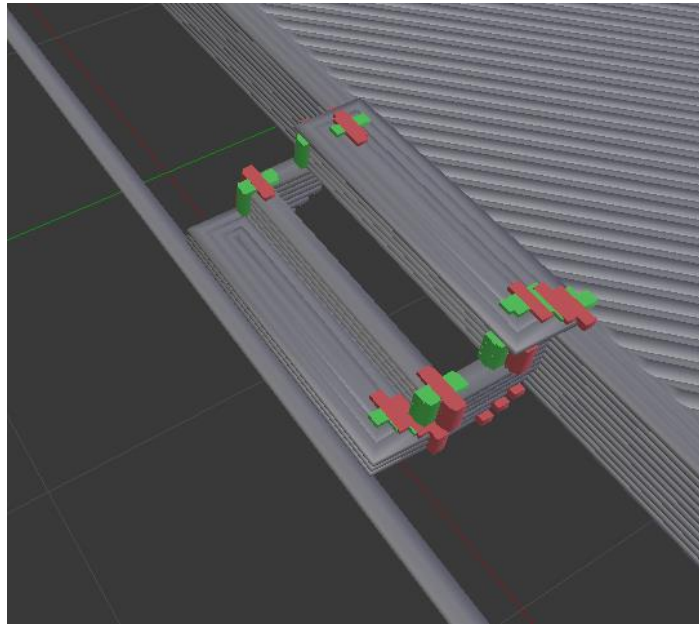
Figure 5.10 Failed Inspection Element 1

Another obvious issue, which can be seen below if Figure 5.11 was the support

structure for the front of the building. The original design called for full sized windows

on either side of the doors. The resulting sliced model attempted to place two concrete

support pillars on either side of the door. This could create a dangerous scenario if the

frames for the windows and door were not designed to be load bearing. Replacing the

concrete pillars with permanent metal supports would guarantee front visage would be

fully supported.
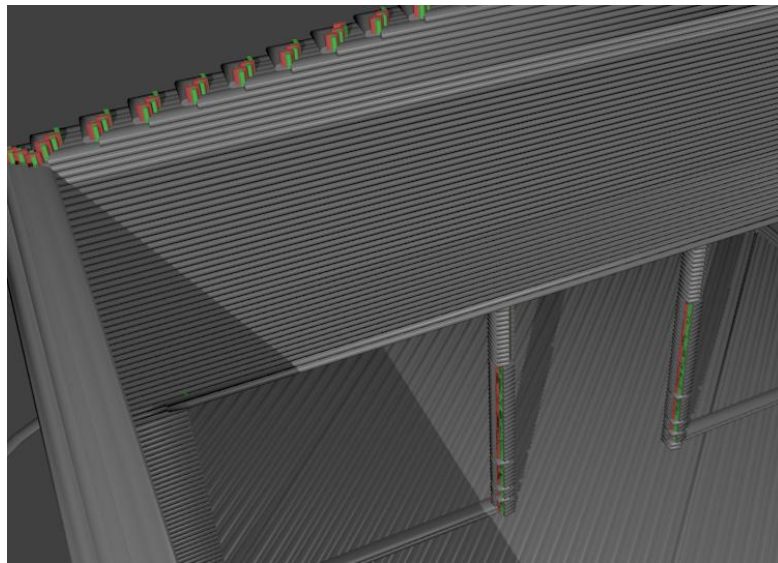
Figure 5.11 Failed Inspection Element 2

Most issues identified through the visual inspection method would be addressed

by directly modifying the original model. Because of the diverse nature of potential

problems that can be identified, modifying the G-Code directly would be difficult and

time consuming.

5.3 Using Fuzzy Logic to Predict Thermal Stress Fractures

G-Code files contain specific data down to the mm level for positioning of every

node along its pathing. The sheer volume and detail of this data can be difficult to

impossible to analyze by hand depending on the project size. However by using "Fuzzy

Sets" to approximate the general risk of stress fracture, the segments can be grouped into

categories based on their general risk as defined by broad categories. The precision data

of the model segment lengths is used to group each segment into a set based on general

risk. How the logic of these Fuzzy Sets falls into the import process can be seen below in

Figure 5.12.
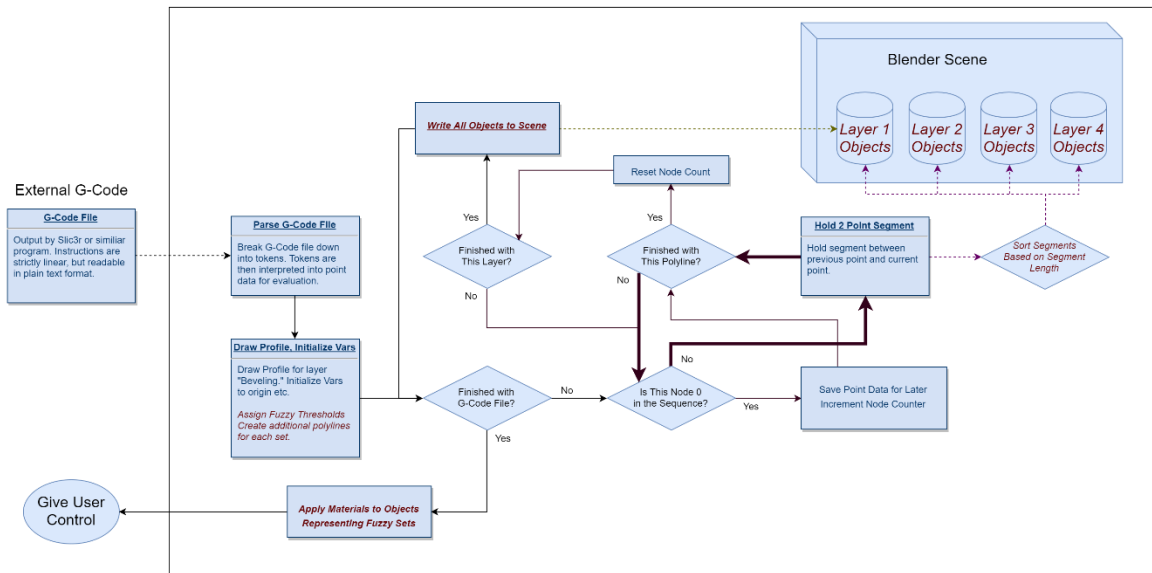
Figure 5.12 Logic Flow to Import Object Within Fuzzy Sets

The algorithm based on Lee Bulter's import process has several key additions.

The core addition is the implementation of Fuzzy Sets. Up front at the beginning of

execution, different paths have to be defined for each Fuzzy Set. In this case, we have

four total sets. Each and every segment is evaluated by the distance formula below. Based

on the results of the evaluations each segment is placed into its appropriate Fuzzy Set. At

the end of each layer, all segments within a given Fuzzy Set are combined into a single

object for each set. The combination of all segments into macro objects is necessary to

avoid the slowdown from creating too many objects as described in Section 5.1. The

numerical improvement can be seen at the end of this section in Table 5.3. Lastly

materials are assigned to every Fuzzy Set group to visually show which segment is in

each Fuzzy Set.

Each segment is color coded according to its corresponding fuzzy set so that

thousands of segments can quickly and easily be appraised at a glance when rendered in a

3d environment. Deriving the distance of each section is fairly trivial. Using the

Pythagorean Theorem and the points from the pathing data, the distance can be easily

calculated. In this case,

$$Distance = \sqrt{(X_n - X_{n-1})^2 + (Y_n - Y_{n-1})^2}$$

where n is the line number of a pathing instruction where the extrusion value is greater

than zero. The X and Y values represent the corresponding X and Y coordinate values

from the G-Code instructions.

Finally each segment can have a material applied to it that corresponds with each

fuzzy set. The result would be thousands of lines of instructions with detailed information

used to derive a version of the model that encapsulates valued knowledge. Colorization of

the model segments makes cursory evaluation easy as can be seen below in Figure 5.13.
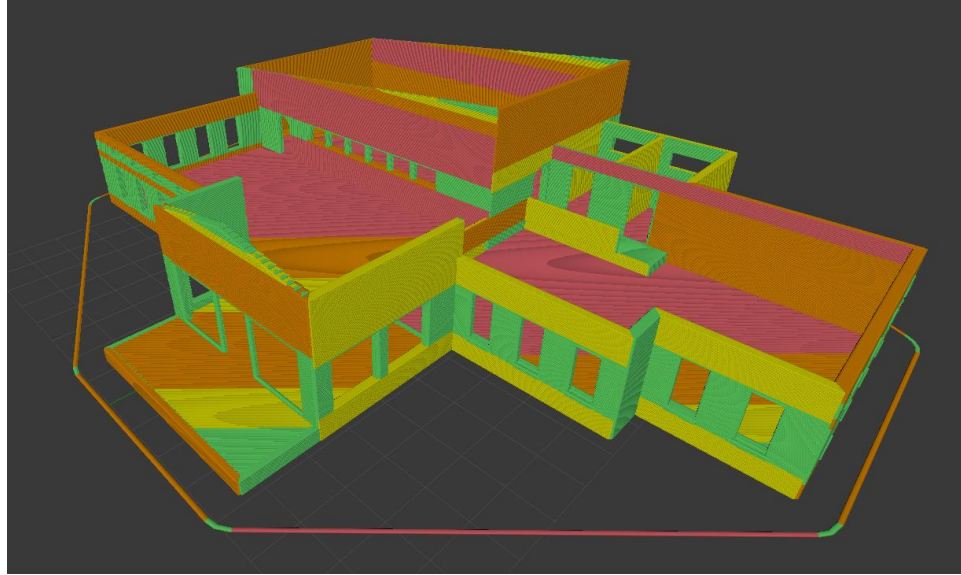
Figure 5.13 Fuzzy Sets Visualized via Colored Materials

The majority of thermal stress within a structure's walls will align with its pathing direction, but for the slab thermal stress as well as stress from the slab settling is multi-dimensional and thus only partially addressed by this method.

Table 5.3

Total Object Created Compared to Time to Process

Granularity	Objects	Time (Seconds)
Segment	53869	1555.986
Layer	553	0.827
Complete	5	0.429

Optimizing the import algorithm proved to be necessary for any reasonably sized object to parse within acceptable time limits. The 53,869 objects in the Figure 5.13 took 1556 seconds, or just under 26 minutes, to completely analyze and load. Even loading the

saved scene (no parsing) takes several minutes. Previous testing with object counts in Section 5.1 has shown that using separate objects for separate segments becomes costly. Grouping like objects by fuzzy set into "macro" objects greatly improved efficiency. In this case, the total number of objects can potentially be reduced down to the number of total layers times the number of fuzzy sets, or approximately 552 objects for this scene. While the 552 object load tremendously more quickly at .8 seconds, this does not continue as a linear trend. If only 4 objects are used with all fuzzy sets combined into one macro object a total loading time of .4 seconds is the result.

In conclusion, Fuzzy Sets, provide excellent visual feedback for straight linear sections like walls, but fail to fully convey tension for planer sections like a slab. In order to achieve reasonable parsing rates, proper granularity of print segments must be considered when processing the G-Code data.

5.4 Extrusion End Point Evaluation

"Seams" created by vertically aligned extrusion end points can create potential weak points in the structure or visual defects in the surface as shown below in Figure 5.14. Severe or not, there is an easy solution that exists within most slicing softwares. Rather than taking the shortest path when pathing through an object, random starting locations can be used instead.

Figure 5.14 Extrusion End Points in Aligned Pattern

Note: Image Courtesy of Andrey Rudenko.

Figure 5.15 below is similar to Figure 5.14, but rather than being a real world example it is based on the simulation of the G-Code instructions for printing a cylinder. Start and end markers have been flagged in green and red respectively to depict a seam that otherwise would be difficult to detect. The long continuous seam could be identified as defect or structural weak point in the print. The start and end points could be randomized as represented in Figure 5.16 to eliminate the uniform seam.
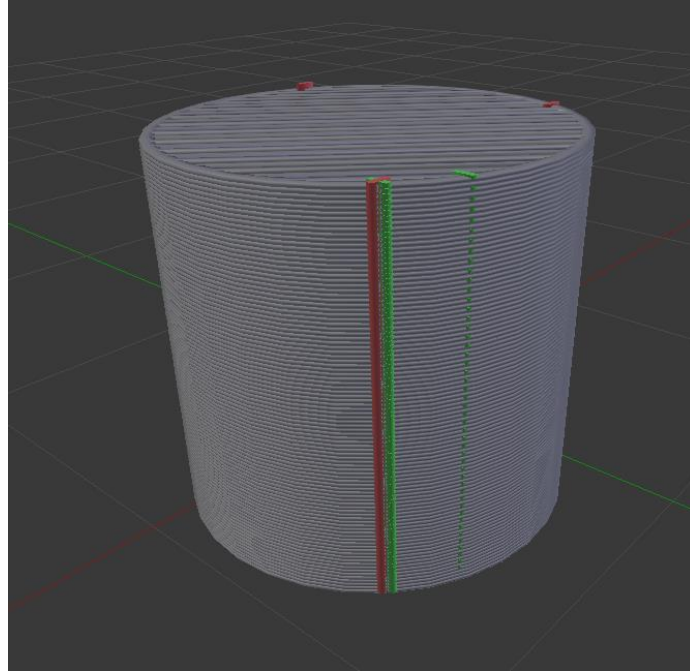
Figure 5.15 Extrusion End Points Aligned in a G-Code Simulation

There is an important drawback to randomizing the start location for each

segment. Because the optimal path is no longer used, additional time is necessary. Across

thousands of segments, the cumulative time can be substantial. Luckily paths that lack a

positive extrusion value, or "empty" paths, are usually much faster than paths were actual

printing occurs.
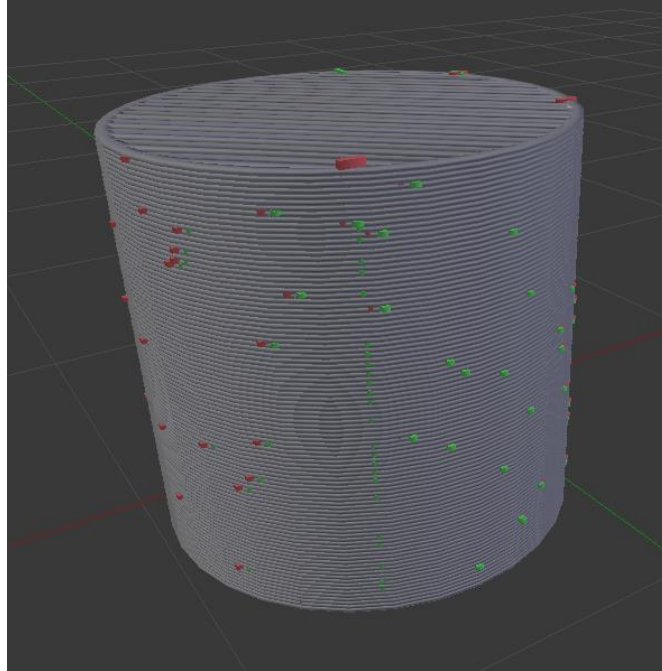
Figure 5.16 Extrusion End Points Randomized in a G-Code Simulation

5.5 G-Code Pathing Collision Checks

The premise of the G-Code pathing collision check system summarized here is pretty straight forward. If the exact pathing of the 3d printer's extruder is known and the relative location and dimensions of any object that can be collided with is known, a collision should be predictable and even preventable for empty paths. Collisions while the extruder is active become more complicated as defects or gaps in the original model may occur. At this time collisions while printing can produce an error, but do not automatically reroute.

To begin with the testing for the collision of two 3d objects as they move along a path could prove computationally expensive and a daunting algorithm to calculate. Perhaps key areas around the contour of the first object could be designated as important and check those paths for collisions with the second object. Fortunately the exact

dimensions of the extruder nozzle will be known for any given 3d printer. Because the

dimensions are static, the problem can be simplified by adding the radius of the nozzle to

any objects that could obstruct the path. In Image 5.17 below, the nozzle on the left is

prepared to move down the green path next to the cube on the right. The path is a near

miss, but the width of the nozzle looks like it will cause a collision regardless. In Figure

5.18, the dimensions of the nozzle have been added to the cube. The path now intersects

the box.
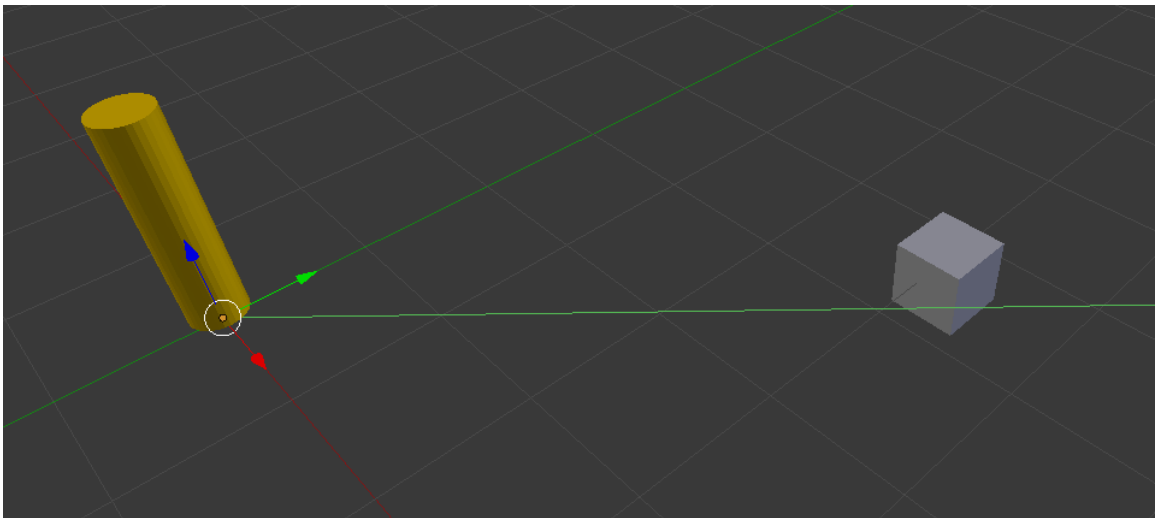
Figure 5.17 Simplified Real World Collision Scenario
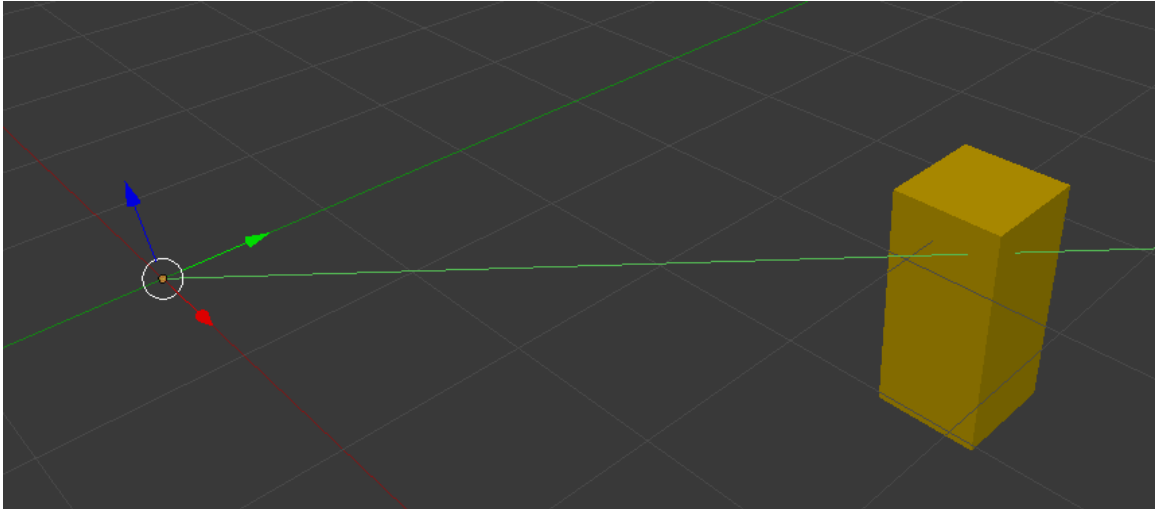
Figure 5.18 Mathematically Simplified Collision Scenario

While the method of adding the dimensions of the extruder to collisionable

objects reduces a complex collision checks considerably, it works best with simple

primitive objects such as boxes and cylinders. Complex objects would need to be

approximated. When collision boxes are initially being created is the best time to include

the extra collision dimensions. Automatic resizing of objects has yet to be implemented

assuming it is possible. Checking every single path like in the G-Code file shown in

Figure 5.19 could prove expensive even with only a single path being checked because a

different check has to be run for every single collisionable object.
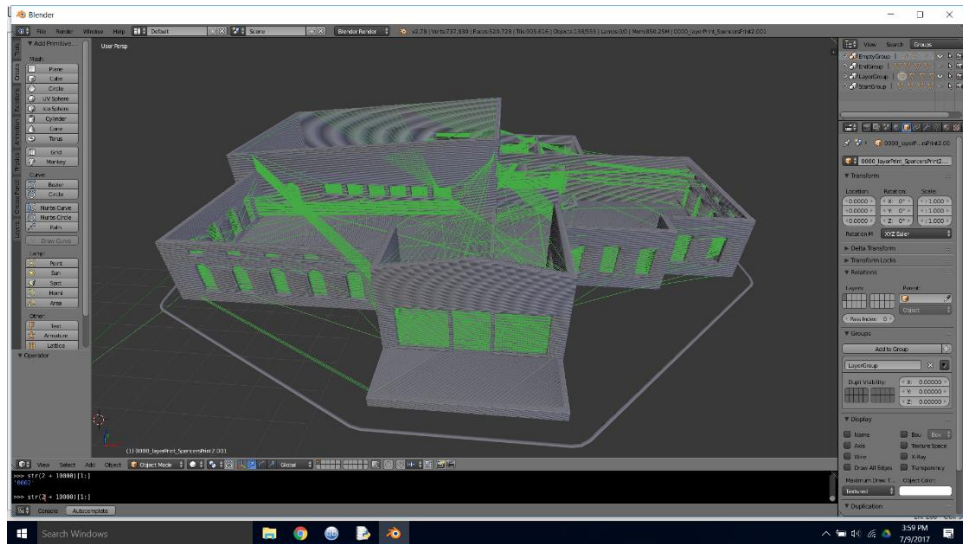
Figure 5.19 Empty Paths Visualization

Collision checks are constrained by the Blender engine. Having to check each object one at a time is a prime example of a constraint placed on us by the Blender engine. To check for collisions within Blender, the *ray_cast* function is called on the mesh object that needs to be checked. The call is made in the following manner:

current_object.ray_cast(position, direction, distance)

Both the *position* and *direction* are (x, y, z) vector coordinates. The *position* parameter is object relative. That is to say that the object is treated as if it were at (0, 0, 0) or the origin. The solution is to subtract the object's position from the start and end node's positions. To get direction from points, the end point can be subtracted from the start point and the new value normalized. Just like in Section 3.3 distance can easily be calculated given two points. Now that collisions can be checked, there is a minor inconsistency to address.

Very rarely if the collision path ends exactly at the exact edge of an object, results can be inconsistent. From one direction the result may flag as a *True* collision, but if the

68

exact scenario occurs from the other side, results may consistently be *False*. Regardless the precision necessary to cause this issue should be exceptionally rare.

Below in Figure 5.20, an empty path lies between two printed segments. This is a typical of the G-Code printing process. If an object were detected between the two segments, the G-Code file can be corrected to accommodate accordingly.
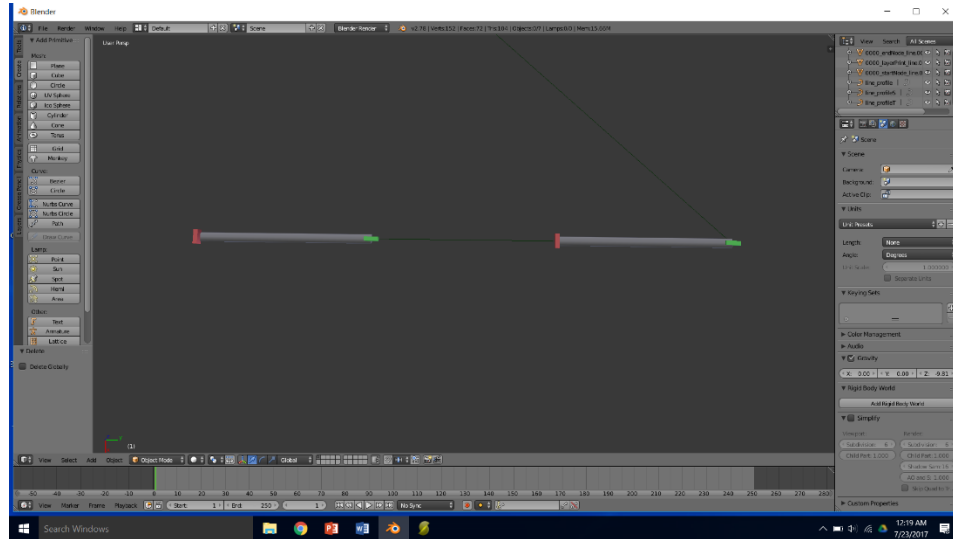
Figure 5.20 Typical Empty Pathing Between Extrusion Segments

The G-Code file uses absolute coordinates. As such between the node that starts the empty path and the node that ends the empty path, as many new nodes as need be can be introduced to create a new unobstructed path. An example of a modified G-Code file can be seen below in Table 5.4 and the visual representation of the changes can be seen below in Figure 5.21.

Table 5.4

Modifying G-Code Paths

Direct Path Between Two Segments	Modified Pathing to Avoid Obstacle
G1 X110.000 Y119.194 F7800.000 ;Empty path G1 E2.00000 F2400.00000 G1 F600 G1 X110.000 Y113.194 E2.51303 ;Print segment 1 G1 E0.51303 F2400.00000 G92 E0 **G1 X110.000 Y106.806 F7800.000 ;Empty path** G1 E2.00000 F2400.00000 G1 F600 G1 X110.000 Y100.806 E2.51303 ;Print segment 2	G1 X110.000 Y119.194 F7800.000 ;Empty path G1 E2.00000 F2400.00000 G1 F600 G1 X110.000 Y113.194 E2.51303 ;Print segment 1 G1 E0.51303 F2400.00000 G92 E0 **G1 X112.000 Y113.194 F7800.000 ;Empty path 1** **G1 X112.000 Y106.806 F7800.000 ;Empty path 2** **G1 X110.000 Y106.806 F7800.000 ;Empty path 3** G1 E2.00000 F2400.00000 G1 F600 G1 X110.000 Y100.806 E2.51303 ;Print segment 2
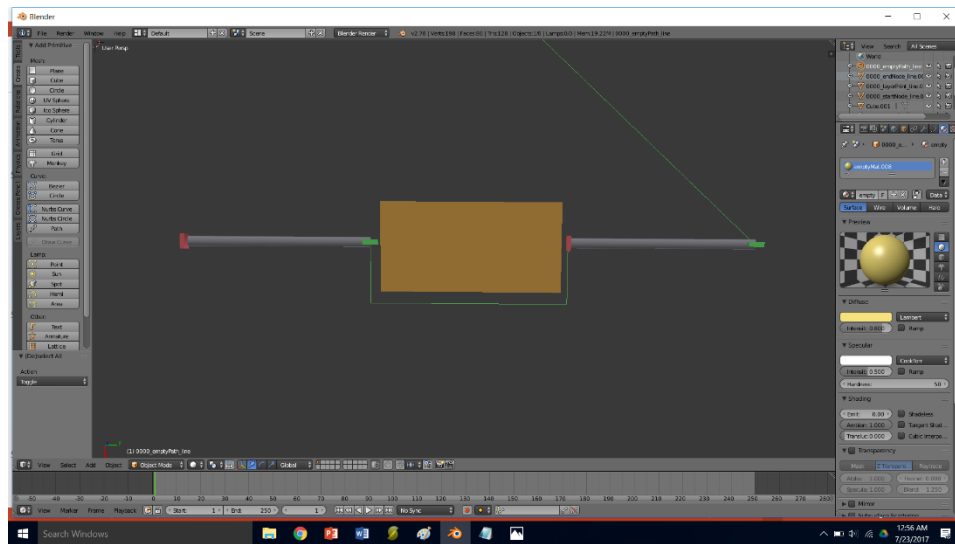
Figure 5.21 Modified Pathing to Avoid Collision

The problem domain has been reduced to rerouting a single point to another point

with linear collision checks. The path can be checked at will to see if the path is clear

between any two points. A 2d or 3d grid could easily be constructed to show available

70

and blocked paths between nodes. This reduced problem has suddenly become a common topic with dozens of potential solutions.

Finding the Shortest Path between the two points is a classic problem with numerous potential solutions. At this time, the best possible algorithm, be it Dijkstra's, A Star, Breadth first, or Depth first, has yet to be determined with empirical trials. Theoretically, A Star pathing should prove to be the most cost effective because of its high efficiency with single node to single node path finding, but additional in depth analysis would be necessary to prove this for this particular problem domain.

5.6 Ray Tracing for Identification and Analysis of Overhangs

Additive Manufacturing is undoubtedly a powerful tool. Despite its versatile nature, Additive Manufacturing must function within a strict set of limitations. The additive nature of the printing process necessitates that each new layer be supported by the previous layer. If a significant portion of the print is unsupported, some sort of additional support must be created either automatically or manually. Slic3r provides options to generate sacrificial supports, and with a large scale 3d printed structure, it would be easy enough to manually place supports sacrificial or otherwise. A key danger arises when an overhang is overlooked, but the print process continues. Additionally certain overhangs may fall within an acceptable tolerance depending on the viscosity of the medium. Small overhangs are acceptable as both plastic and concrete are viscous enough to hold together over small gaps or overhangs. Identifying when those overhangs overextend beyond the acceptable tolerance can also prove useful. The following is a proposed original algorithm for not only determining areas that require additional

support, but the algorithm also makes the distinction between overhangs within acceptable tolerances.

The core principle of the algorithm revolves around moving through the model two layers at a time with a "sun" light casting parallel light rays directly up the z axis. The bottommost layer via Blender settings can be made invisible to the camera, but flagged to still cast a shadow. The topmost layer is treated like a regular object and rendered normally. Placing the camera below the layers and pointed it directly up completes the minimal setup. This configuration can be seen in Figure 5.22. Setting the camera to Orthographic mode will result the camera acting more like a scanner than a traditional camera. That is to say perspective will not slant the resulting image if all layers are square with the camera rendering lens. It should be noted that images rendered from this camera will be each layer viewed from the bottom. To adjust for a traditional top down perspective, the scale for the inverted axis of the camera object should be given a negative value. During the rendering process, the bottommost layer will obstruct light casting a perfectly black shadow on the topmost layer. The areas on the topmost layer that are not supported will subsequently light up. The bright colors in the image will quickly and definitively shows unsupported areas such as the overhang areas of doors and windows illustrated in Figure 5.23.
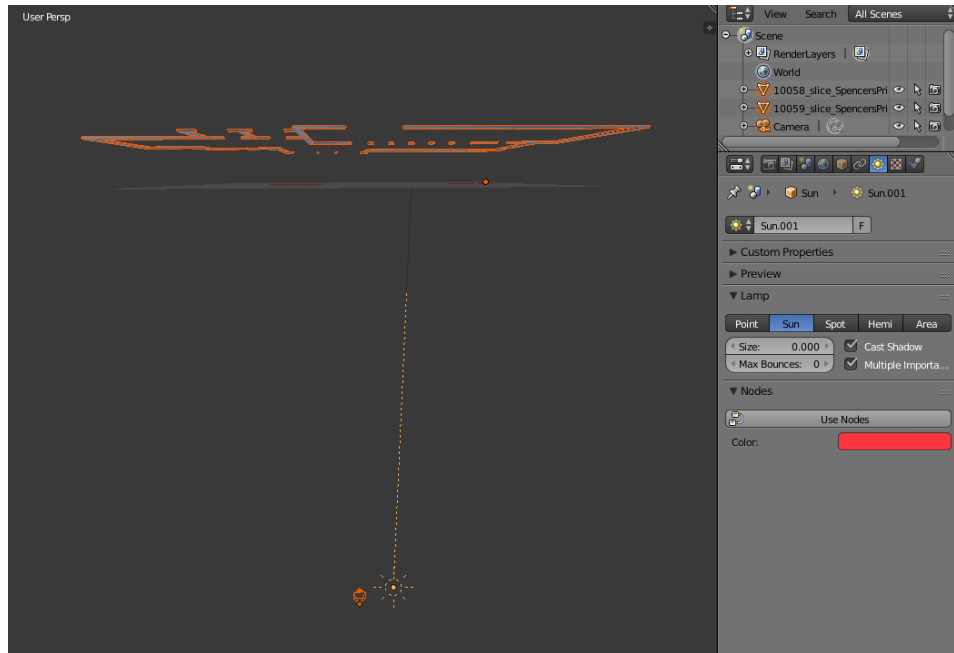
Figure 5.22 Overhang Detection Configuration
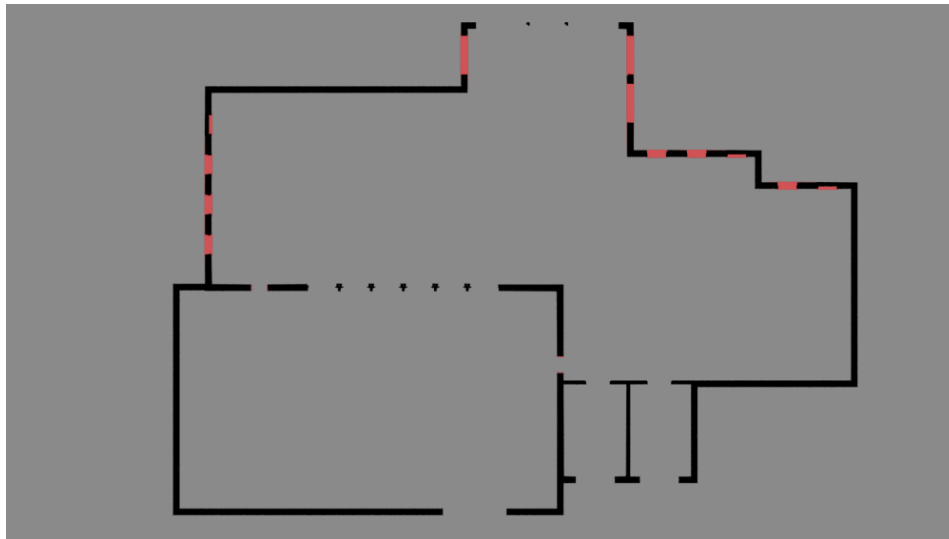
Figure 5.23 Overhang Detection Rendering

Prior to the rendering process. The scene should be loaded in as explained in

Section 5.1. After that the rendering process flowchart can begin. The outline for

rendering using the ray tracing algorithm can be seen in the flowchart below in Figure

73

5.24. As an overview, the scene is dressed in preparation for the printing. All renderable objects are hidden from the camera and their shadows turned off then layer by layer an image is rendered. Finally control is given back to the user.
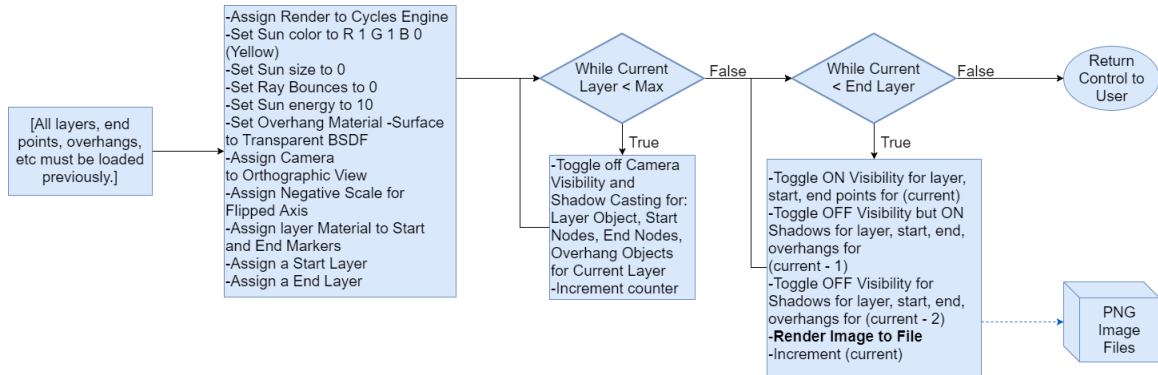
Figure 5.24 Process for Rendering Overhang Image Data

Firstly all layers, extrusion start and end points, and safe overhangs for the layers and start and end points should be loaded in as explained in section 3.1. For ease of stepping through each layer, all like objects on each layer should be combined into a single object if they are not already. Rendering with the cycles ray tracing rendering engine is required. Setting the energy level to 10 from 1 for the sun will result in brighter overhangs. Setting the sun "size" to zero will reduce any hazy edges and reducing the number of possible bounces to zero should as well. In this case, "size" denotes the diameter of the sun. A larger diameter sun will result in shadow edges being less sharp. Any RGB value can be used for the sun but in this example R 255 G 255 and B 0 has been used to create a bright yellow coloration. Segment start and end points should be assigned the same neutral material as the primary layers to avoid any color conflicts.

Overhangs are assigned a "Transparent BSDF" material. In this case a green color has been applied to the material. The transparent material will tint any light that passes

74

through it. In this case, the yellow light will be tinted green for safe overhang diameters

as show in Figure 5.25 below. In the illustration, concrete columns meet a horizontal

overhang. The green areas, tinted by the overhang objects, predict safe overhang lengths

while the yellow represents areas in need of additional support. Because the cycles

renderer with Blender strives to be accurate to life, tinted materials can only reduce the

existing wavelengths of light. It cannot introduce additional colors. In this instance, Red

and green (yellow) can be reduced down to only green, but could not be colored blue.

Likewise a purely red light could not be reduced down to green. The resultant color after

exiting the tinted material will contain an RGB value less than or equal to that of the

incoming light.
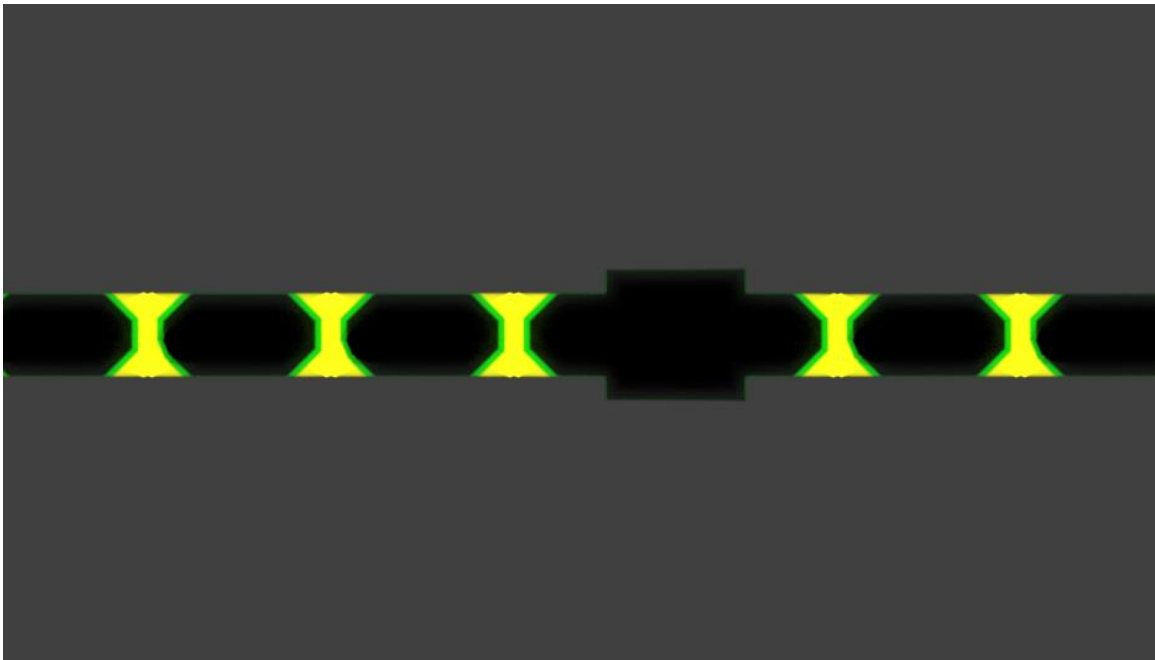
Figure 5.25 Overhang Distances Colorized by Safety Tolerance

Now that all the actors are in place within the scene two options are available for

rendering. Layers can be selected by hand if only one or two sections need to be

analyzed, or a series of layers can automatically be stepped through and rendered for analysis. Regardless of whether the instruction steps are executed manually or automatically, the instructions will remain largely the same.

Firstly all renderable objects should be hidden and their ability to cast shadows turned off. For any given layer n that needs to render, the layer object, start points, and end points for that layer should all be flagged as visible. Secondly for the n-1 layer, shadows should be turned on for the layer object, start and end points, but also the overhang objects. To clarify layer numbers start at zero for the bottommost layer and increment by one for each level progressing upward. When stepping through multiple layers at a time, all n-2 layers should be flagged as unable to cast shadows. In this way n can be incremented to continuously move up from a starting layer to any given layer m which still contains objects. As each layer is stepped through, any perspective camera should also be raised one layer of height to ensure uniform imaging. Because it acts more like a scanner than a camera, an orthographic camera should render based on its scale rather than distance to the object.

To create comprehensive documentations, each render should be exported as its own image file. By default Blender exports files as PNG files, but additional options are available. Standardizing the naming convention may also prove useful. Leading the filename with the layer number padded by zeros will result in all image files being in alphabetical order. For instance using a naming convention like "0001Print.png" will organizationally keep all exported files or rendered objects for each layer together when sorted alphabetically. While this naming convention isn't necessary, many prints contain

over 100 layers and keeping files and objects orderly becomes an increasingly important task as project scale increases.

The end result of the algorithm outlined here will be one or more rendered images. Each rendered image will show a cross-section analysis for each layer. The layer will be black or darkened out where it is completely supported. It will be green in areas that are close enough to a supporting area that no additional support should be required. Lastly areas in yellow will require additional support. While most overhangs can be easily spotted when viewing the full print, predicting which upcoming layer will have an overhang is a much more difficult task. Having an automated process that includes the layer count should greatly reduce the possibility of human error and save analysis time. Attempting to print with insufficient support would prove costly and messy. The methods outlined here should reduce said risks.

CHAPTER VI - A CASE STUDY FOR A PROPOSED BORDER WALL

On March 17th 2017 the U.S. Customs and Border Protection agency published

two request for proposals pertaining to building a physical wall along the southern border

of the United States [10]. Among the requirements in the RFPs were the following items.

The project would span roughly 1000 miles of the overall border. Although the border is

longer, only 1000 miles will be addressed at this time. The wall will be at least 18 feet tall

with an average expected height of 30 feet. Visibility through the wall is not required but

would be "operationally advantageous." The wall must be aesthetically pleasing on the

north side with color specifically mentioned. The wall must not be easy to climb. The

wall must not be easy to dig under with six feet of underground deterrent recommended.

A border wall would make an ideal candidate for large scale 3d printing for a

number of reasons. Firstly it is a simple structure. Homes and offices have much potential

as candidates for 3d printing, but there are numerous obstacles that must be addressed

along the way. By contrast a border wall won't have to account for internal electrical

wiring, plumbing, insulation, or roofing. The low cost of the simple building materials

and large size of the project should result in time and labor being one of the most costly

project expenditures. Of course AM excels in reducing overall labor costs. Lastly the

repetitive nature of a wall is ideal for consecutive identical printed segments.

6.1 Original Design for Border Wall

Besides the requirements described previously, the additive nature of the building

process greatly influenced the design. Version one of the proposed design can be seen

below in Figure 6.1. The large support columns taper inwards in such a way that they

would be difficult to create using traditional construction techniques. The primary wall that is featured in the frontal view is likewise one foot wide at the bottom, but tapers down to five inches at the top. The reduced mass at the top increases stability by supplying a low center of gravity, but the large base sections provides ample support and resistance to collisions or damage. Overall each section or panel of wall is 40 feet wide and 30 feet tall. The side "columns" flare out to five feet at the bottom, while the center support only flanges out three feet.
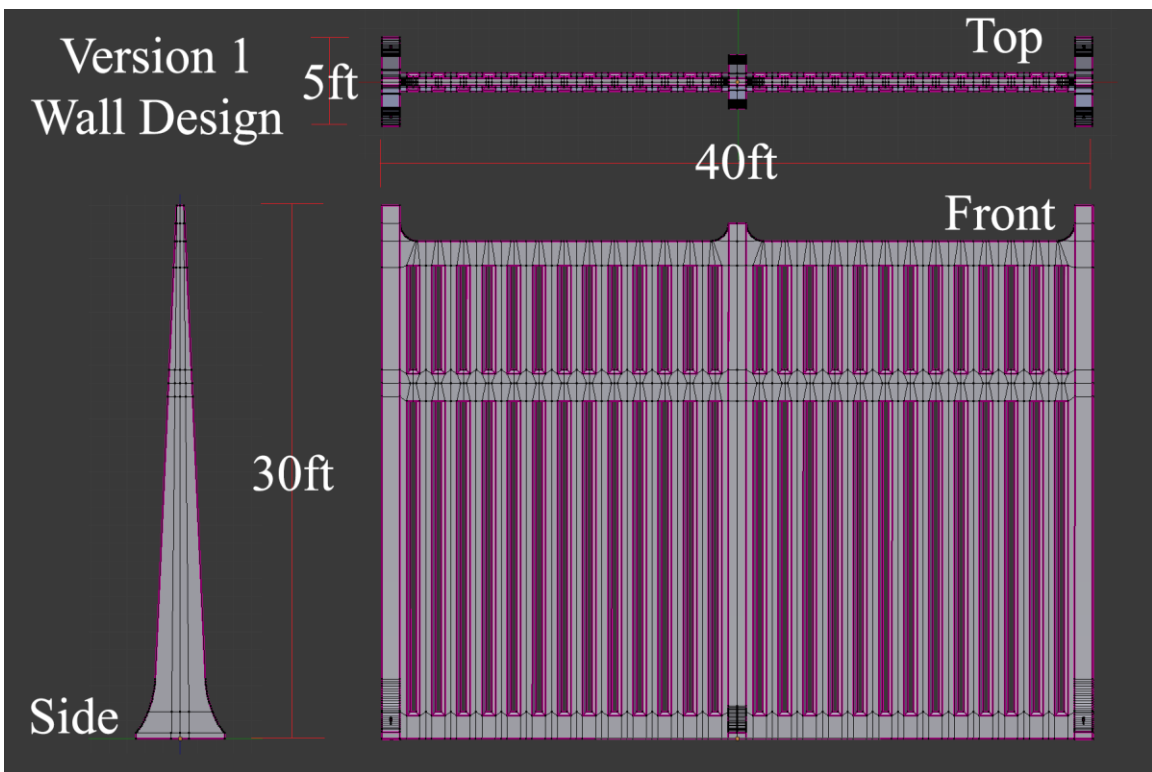

Figure 6.1 Version 1of the Proposed Border Wall Design

Overall three variations of this design will be explored. The second design will incorporate elements based of overhang analysis in Section 6.2. Specifically the overhangs will be removed by introducing a tapered design for the viewing slits. The

third design will attempt to alleviate thermal stress across the length of the wall as outlined in Section 6.3. This will be accomplished by offsetting the primary wall sections so that they do not fall in a single continuous line.

The four inch gaps allow for visibility and reduce the overall volume of the wall which will also reduce cost. Using traditional construction methods the inclusion of these view gaps would increase the cost because of the additional work, waste, and time associated with complexity. The wall is beveled around the gaps to increase the view range angle. A solid cross section was included to reinforce the concrete columns which might could become unstable during the print process.

In this version of the design, the tops of each viewing gap are intentionally left as flat overhangs. As discussed in Section 5.6, all overhangs must be fully supported. While supports could be placed to implement this design, the flat overhangs are an intentional design flaw. This intentional defect is shown here to contrast with a more cost effective and fully automatic solution discussed below in Section 6.2.

6.2 Overhang Analysis

The flat overhangs serve no design purpose with this wall the way they might in a building design with a door or window. In fact the flat overhangs can be completely removed without consequence. One simple solution to remove said overhangs is to taper the view gaps gradually in so that they naturally meet after several small safe overhangs. The current safe overhang distance has been set to half an inch, but as more detailed information for the printing mediums becomes available this could be increased or decreased. Adjusting the taper angel at the top of each view slot down or up in the z axis

direction easily shrinks or expands the overhang distance per layer. This adjustment would prove useful if each layer was just over the allotted tolerance. Reducing the layer thickness will reduce total overhang distance which should result in safer overhang distances. The new version two design can be seen below in Figure 6.2.

In traditional construction, these tapered slits would probably increase the cost. It would be one more feature to build into wall either in a form that the wall is being poured into or by hand if the wall is being manually constructed. With AM the taper slightly reduces materials compared to the version one design. This actually saves materials and thus cost.
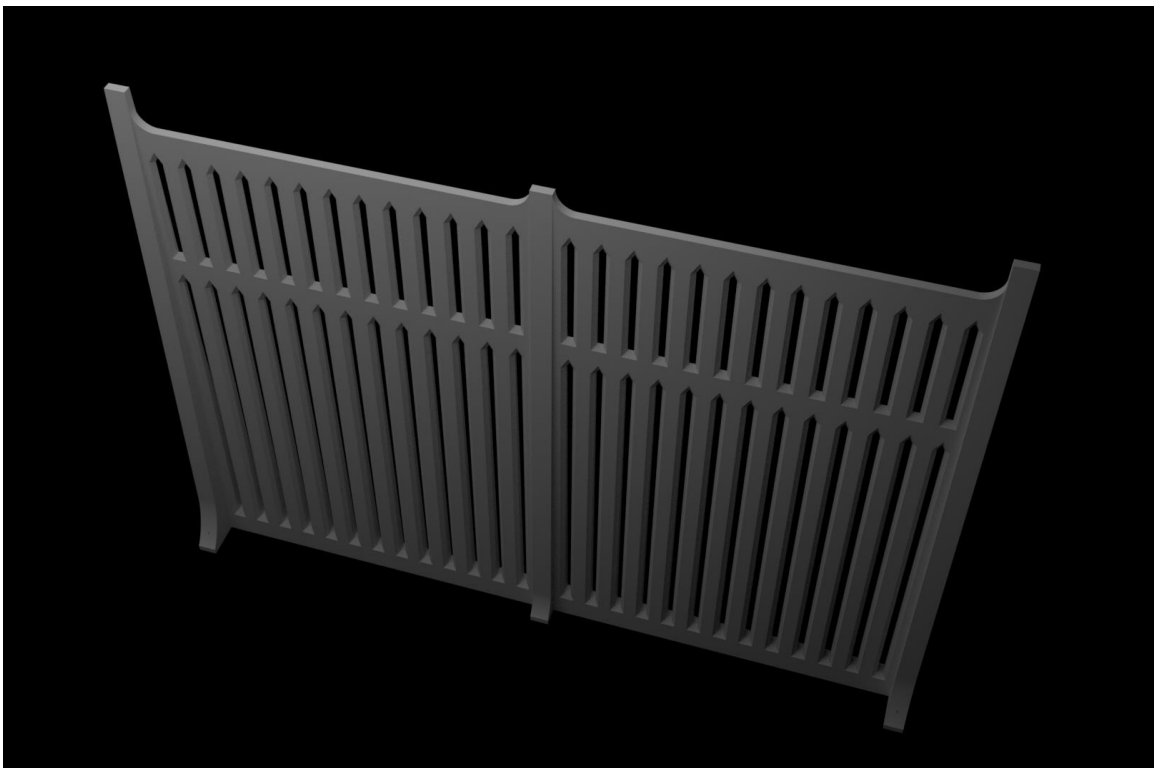
Figure 6.2 Version 2 of the Proposed Border Wall Design

Below in Figure 6.3 is the overhang analysis discussed in Section 5.6 for version one of the border wall prototype. The layer analysis is for the overhanging layer

immediately above the lowest level of columns. As expected the central overhang area falls clearly outside the acceptable overhang range. Figure 6.4 analyses the second version of the build design with the tapered slit tops. This layer falls within the safe overhang tolerance, but to be safe each layer above this one should be checked until the wall is fully joined.

The comparison between version one and version two of the wall design is to make a point rather than to earnestly show a true revision. In contrast the modifications in Section 6.3 are legitimate design updates.
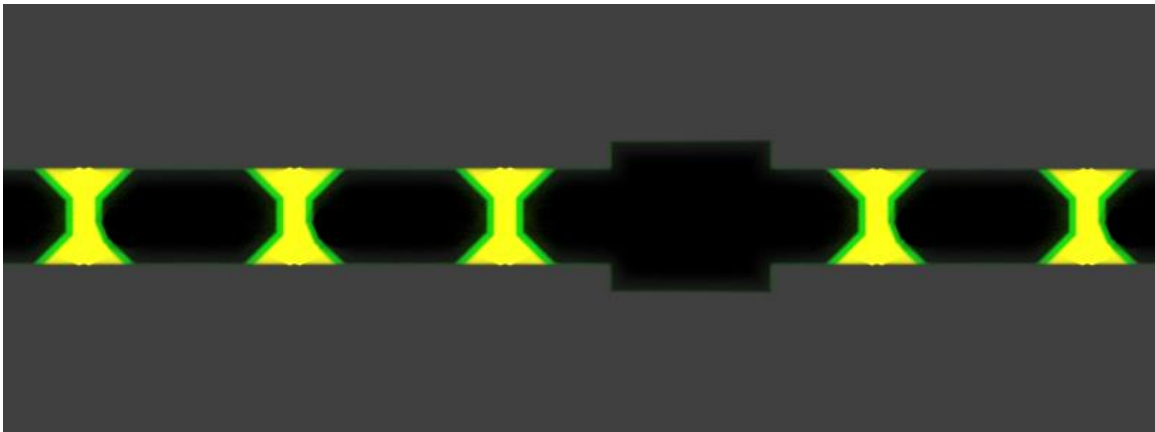
Figure 6.3 Unsafe Overhangs Detected in Design Version 1

Figure 6.4 No Unsafe Overhangs Detected in Design Version 2

6.3 Modified Design to Reduce Thermal Stress

      Even after advocating thorough analysis of a G-Code file before, version two of the border wall was not passed through the thermal stress prediction algorithm described in Section 3.3. It was assumed that the view slits would alleviate any lateral stress through the forty foot wall section. When printing a scale PLA model for the wall, the bottom section cooled enough to dislodge a portion of the wall from the heat bed. Upon further investigation, the continuous forty foot base would almost certainly develop cracks especially given the high temperatures along the southern border. While the areas of high potential stress are narrow in Image 6.5, they are still significant. The lowest band in particular would be the most at risk since most stress seems to typically begin at the ground level and spread up.
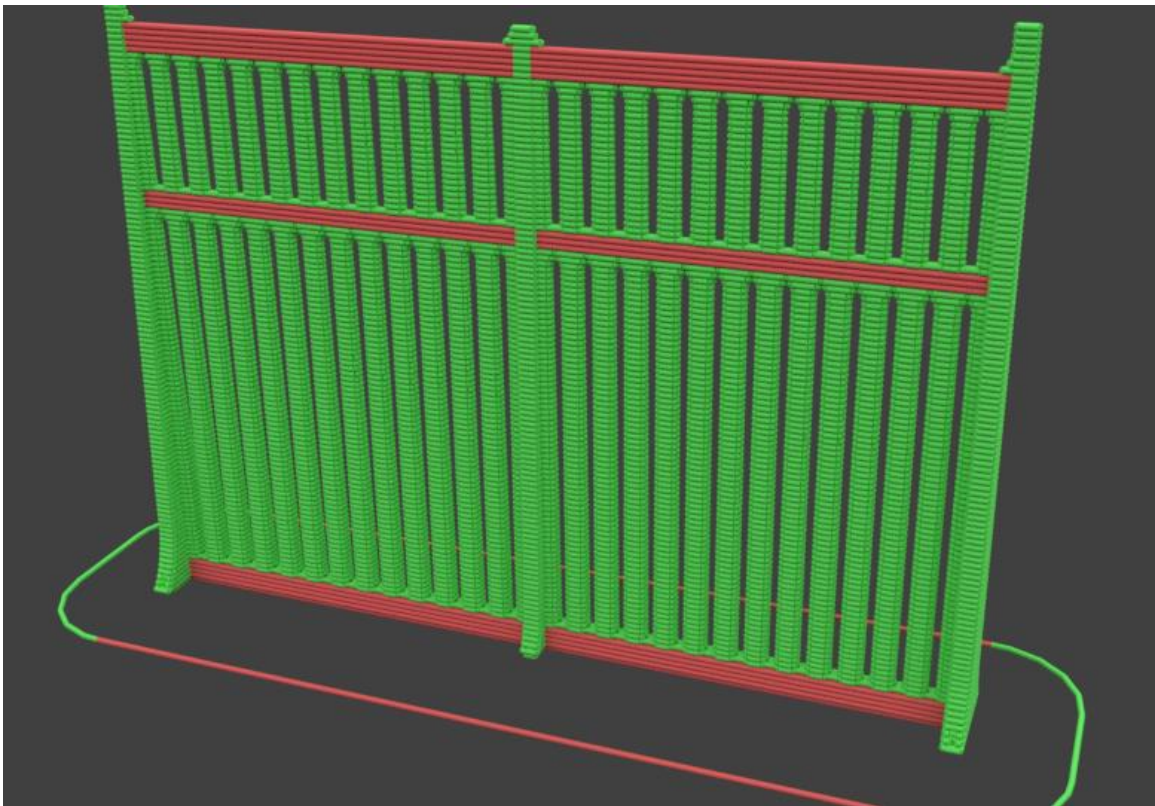
Figure 6.5 High Potential Thermal Stress in Design Version 2

The third and final version of the border wall design can be seen below in Figure 6.6. Both center panels of the wall have been moved out in opposite directions six inches. This means that the one foot wide bases are now offset one base wall width. The three major columns have also been thickened up to accommodate for the offset. Aside from reducing stress along the forty foot length of wall, this design also increases front to back stability with wider columns.

Figure 6.6 Design Version 3 with Center Walls Offset

Lastly the design is reevaluated for thermal stress in Figure 6.7. The new twenty foot sections still hold quite a bit of potential stress, but it is a significant improvement over the solid 40 foot sections. If the design were extrapolated out to share end columns

this would help tremendously. The longest continuous section would be only 20 feet

while version two of the design would have hundreds or thousands of feet of continuous.

With version two of the design, it would be preferable to place sections column to

column with a full air gap up the entire length of the wall to account for expansion during

the heat of the day. With version three, the end columns could be shared to save on

materials and to help reinforce the wall as a whole.
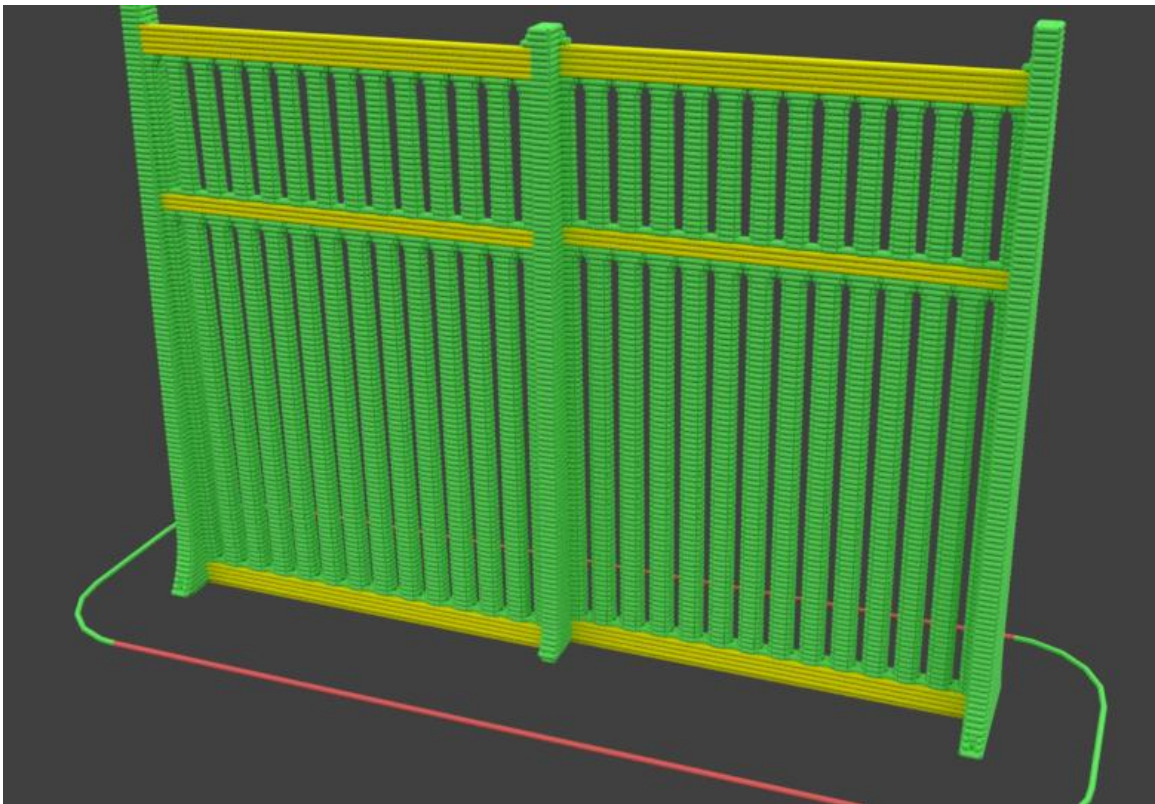
Figure 6.7 Reduced Potential Stress in Version 3

Because Additive Manufacturing is not bound by traditional rectilinear

constraints, a forth version of the wall could be proposed that adopts a slight sine wave

by having one section of wall slightly curved outward to the north and the next section

curved slightly out to the south. This would greatly increase front to back stability. It

would also reduce any remaining thermal stress significantly. There would be a few drawbacks to this design. Visibility through the gaps would no longer be uniform because of the changing angle. Rebar placement laterally through the structure would also become much more difficult. Material required per section would increase. The total width of the wall would increase require more land depending on how pronounced the sine wave was. Overall it might be an improvement, but may also needlessly complicate the theoretical project.

6.4 Estimated Project Costs

There are several factors that will contribute to the overall project costs. Firstly the cost of purchasing, clearing, and leveling the land will be a major factor. The cost of the 3d printing machinery must be considered, but may be the least confident estimate. Next the cost of materials for the wall as well as labor per hour of operation must certainly be accounted for. Lastly the total time to complete the project will be evaluated. Completion time will be based on the number of 3d printers and volume each of those printers can generate. Total time to complete the project will be inversely proportionate to the number of 3d printers increasing costs for that factor.

Estimating the cost to acquire and prepare a continuous path along the border, is obviously a complicated and costly endeavor. To approximate the cost of creating a continuous foundation, the cost of a two lane road will be used as an estimate. A road like path would even out the terrain for more level ground and provide a solid foundation to build upon. According to the American Road and Transportation Builder's Association, the cost to build a two lane road is roughly 2 to 3 million for rural areas and 3 to 5 million

in urban areas per mile [14]. Due to the rigid pathing requirements, the more aggressive

estimates of 3 and 5 million dollars per mile will be used for estimation. Estimating

Urban vs Rural locations is more complex than expected. Data for states and counties is

readily available from the Census Bureau, but exact population density data down to a

precise geographic scale is more difficult to come by. Using Census Bureau data and

light pollution levels from NASA, it can be determined with reasonable accuracy that

90% of the border is rural [30]. Coupled with the 1000 mile project scope in the RFPs, an

estimate of 2.7 billion dollars for the 900 miles of rural terrain and 500 million for the

100 miles of urban terrain can be determined. The resulting total is 3.2 billion dollars to

acquire the land and prepare a reasonable foundation for the printing process. This step

will prove to be the most costly factor in the case study. It should be noted that acquiring

the land and preparing it will be mandatory for all proposed border wall production

methods. It is an unavoidable cost.

The author has detailed cost data for building a 3d concrete printer with a print

area of about ten feet in all axis directions. Those numbers are shown in Table 6.1 below.

These numbers do not include labor for assembly and furthermore would not be directly

applicable to a much larger printer, but regardless an aggressive estimate can be made for

each 3d printer by multiplying the values show here by ten. The steel frame that supports

the printer as it moves about would be much more costly with a larger printer, even if the

motors remain sufficiently large enough to power it. Additionally the price can be

roughly doubled to account for assembly. If mass produced the cost may drop as lows at

ten thousand a unit, but at this time a more aggressive estimate of thirty thousand a unit

will be used. As stated the $30,000 price tag estimate is per unit. Because completion

time of the project will be tied to number of units in production, the total price for

machinery can vary dramatically. For the final calculations 100 total 3d printers will be

assumed across the entire project.

Table 6.1

Estimating Machinery Costs for a Single Unit

Item	Cost	Count	Item Total
Arduino Mega 2560 w/ Ramps	40	1	40
Power Supply	25	1	25
Gecko Stepper Drivers	60	5	300
Motor NEMA 34	100	5	500
NEMA 34 Mounts	10	5	50
NEMA 34 Gecko PSU	25	5	125
Home Sensors	8	1	8
Wiring (Stage 1)	25	1	25
Roller Chain (100 ft)	99	1	99
Roller Chain Connectors	8	1	8
Roller Chain Breaker	12	1	12
Roller Chain Sprocket	10	3	30
Idler Sprocket	15	2	30
10ft 3/4" threaded rod (Lead Screw)	26	2	52
Pillow Block Bearing	10	4	40
Guide Rail System	347	1	347
Cement Pump	480	1	480
Aluminum/Steel framing	650	1	650

	Grand Total:	*2821*

Total material cost is much easier to calculate with confidence. Each forty foot

section of wall contains approximately 684 cubic feet of concrete. Using Lowe's bulk

business prices shown in Table 6.2 quick curing concrete with a 4000 psi compression

strength costs $3.55 per 80 pound bag [27]. Each bag produces six tenths of a cubic foot.

Twenty feet of rebar likewise costs $3.37 per unit. Each forty foot section of wall should take approximately 50 sections of rebar. When totaled out, the cost of a forty foot section of wall is about $4216 in materials. This extrapolates out to $556,512 per mile or $640 million across the project accounting for 15% potential waste. With tax free pricing and a sizable government contract, material prices could be significantly reduced. Also the 15% waste estimate is overly aggressive considering that the Additive Manufacturing process generates very little waste compared to traditional building methods.

Table 6.2

Estimating Machinery Costs for a Single Unit

	Units	Full Price	Bulk Discount	Bulk Rate	volume	Cost per cubic ft
Quikrete	80 lbs	$ 4.44	20%	$ 3.55	.60 cubic ft	$ 5.92
Rebar	20 ft	$ 4.82	30%	$ 3.37		

A 684 cubic foot wall would take approximately 6 hours to print at 15 gallons a minute which is the maximum output of the pump priced in Table 6.1. Unfortunately a thirty foot tall wall will not have sufficiently cured lower layers within a six hour time frame to finish the topmost layers. The solution is to print multiple wall sections at once so that they can all be curing at once. There is even a small scale plastic 3d printer called the Black Belt available now that prints at an angle with a conveyor belt [7]. The end result is that in a single direction, the maximum print area is only limited by strength of the material and how will it is supported once it leaves the conveyor belt. Similarly the proposed 3d printers could use tracks or treads to continually keep moving down the length of the wall. This would result in continuous angled printing rather than "batch"

wall sections being produced. Even with mostly automated machines running around the clock, a mile of wall would take about a month to complete.

Total operational time comes out to approximately 792,000 operational hours. With one hundred machines printing around the clock the project could be completed across the entire scope of the project within a year. This time frame is unrealistic since it fails to account for rain, setbacks, and down time. A more realistic timeline would be two years of printing, or more probably three years with a gradual roll out of 3d printers. Ideally a trail of a few prototype printers would get most of the bugs out of the machinery before mass productions of the entire fleet would begin.

Based on the previously calculated operational hours hourly operational costs can be calculated by estimating labor and expendables such as gas. Even though the process is largely automated, a four man team per machine is the basis for the following calculations. $75 an hour for labor and gas and such across 792,000 hours yields approximately 60 million in cost. Doubling that as previously mentioned would account for setbacks, adverse weather conditions, and general setbacks and malfunctions. This yields us a total labor budget of 120 million which should be an order of magnitude smaller than traditional construction methods. This part of the budget is actually where the majority of savings will be found.

The final budget yields about 3.2 billion for acquisition, cut and fill, and foundation preparation for the site. Raw materials account for 640 million in cost. Equipment calculates out to roughly 3 million for 100 full sized 3d printers. Spare parts and repairs are not accounted for. Labor will total out to approximately 120 million across the life span of the project. The result is a budget of about 4 billion dollars. Even

though aggressive estimates were chosen, the estimate may still be low. Even if the budget is arbitrarily doubled to 8 billion, it is still far below the estimate of 21.6 Billion by Homeland Security or the estimate from MIT for 38 billion [8, 23].

Regardless the estimates shown here do appear to line up with estimates on project costs for homes and offices. If 3d printing can save 60% of costs for a home as Dr. Behrokh Khoshnevis estimates, the saving should be much higher for a simple wall [6]. The reasoning here is that for a home doors, windows, electrical, plumbing, tile, painting, etc must all still be done by hand. With the border wall project, almost no additional systems like lighting are involved. In fact the only process that may require manual attention would be rebar placement, and that too could likely be automated with additional time and research.

Despite the benefits discussed here, there are several limitations and obstacles to be overcome as well. The ridges created by the printing process could create a surface that is easier to climb than a smooth surface. Apis Cor incorporates smoothing fins that create a level surface during the print process [39]. Something similar would have to be incorporated into the 3d printer design to avoid costly manual intervention. Curing time is a major restriction that would have to be worked around.

In summary a 4 billion dollar budget is shockingly low compared to other estimates with traditional construction means. As low as the estimate is, others have estimated a 60% savings across entire projects that are much more sophisticated than this application [6, 17, 39]. The border wall project estimate falls around the 80% savings mark. This is possible because of the low complexity of a wall compared to a livable building.

CHAPTER VII - CONCLUSION AND FUTURE WORK

7.1 Conclusion

Currently five methods for verifying model integrity have been identified. In practice two of those methods "extrusion end point evaluation" and "visual inspection" can be combined. While more work can always be performed, automatically identifying overhangs is a critically important method considering the constraints placed on Additive Manufacturing by overhangs and unsupported layers. Predicting thermal stress fractures appears to be an excellent opportunity for implementing Fuzzy Logic. The Fuzzy Logic renderings help break massive amounts of data into a format easily understood by a human audience. Collison prediction is more of a practical approach to the printing process itself, but proves valuable considering the difficulties and costs that can arise by damaging the Additive Manufacturing machine or the structure being printed.

Because of the simple structure that makes up the border wall and an almost entirely automated process, a border wall could save up to 80% across the project. For comparison 60% savings across a project has been confirmed for much more complex projects by Dr.Behrokh Khoshnevis, Andrey Rudenko, and Apis Cor which are much overall a less automated process [6, 17, 39]. The border wall, although a bit simple in structure, acted as an opportunity to implement the proposed verification methods that were developed and outlined in Chapter V. Despite the model and proposed structure being very simple in design, the proposed verification methods still allowed for detailed analysis and modification of the structure in the early design phases.

In conclusion, the verification methods and simulation groundwork proposed here may not only prove valuable in and of themselves, but could prove to be a launching

point for more complex verification methods and techniques. Additionally the case study shows the high potential of Additive Manufacturing of simpler concrete structures. The increase in project automation over other habitable structures proved to greatly reduce the project cost. Rather than the estimated 60% savings for more complex projects, a savings of approximately 80% could be achievable for projects with higher levels of automation.

7.2 Future Work

Overall there is much more that can be done besides these few methods identified in order to verify model integrity. If structural stresses can be predicted based on the model data available, then optimal rebar placement may be predicted. Furthermore it would be a huge step forward if hurricane and earthquake simulations could be applied to a structure based solely on the G-Code instructions before it was ever even built. If G-Code instructions could one day act as a universal building structure format, hurricane and earthquake simulations could be standardized to an unprecedented degree. A comprehensive rating system for earthquakes and hurricanes could be established for dozens of potential natural disaster scenarios.

Additive Manufacturing of concrete structures besides the more complex homes or offices can also be more fully explored in future. Water levees, storm shelters, even bridges, dams, and other infrastructure may prove to be an ideal problem domain for Additive Manufacturing.

Lastly as large scale 3d printers become more readily available for printing applications, the exploration of various printing materials and techniques can be more easily explored. Various combinations of concrete or even concrete alternatives could

93

prove to be a better material choice for any given structure. Certain pathing routines or cure times between layers may prove to have higher compression strength or thermal resistance.

As a whole, 3d printing concrete structures is a field full of potential. AM will almost certainly have a significant and lasting impact on the construction industry as a whole. As with many emergent technologies, different avenues of research abound, and potential applications remain plentiful and promising.

REFERENCES

[1] A. (2016, May 14). Dubai opens world's first 3D printed 'Office of the Future',

completed in just 17 days. Retrieved October 5, 2017, from

http://www.3ders.org/articles/20160524-dubai-opens-3d-printed-office-of-the-

future-completed-in-just-17-days.html

[2] Bowyer, D., & Olliver, V. (2016, April 08). The Official History of the RepRap

Project. Retrieved October 5, 2017, from https://all3dp.com/history-of-the-reprap-

project/

[3] ASTM Standards. Standard terminology for additive manufacturing technologies, vol.

10.04.

[4] Barsoum, M. W., Ganguly, A. and Hug, G. (2006), Microstructural Evidence of

Reconstituted Limestone Blocks in the Great Pyramids of Egypt. Journal of the

American Ceramic Society, 89: 3788–3796. doi:10.1111/j.1551-

2916.2006.01308.x

[5] Beckett, C. A. (2012, April 02). Thomas Edison's Beautiful Failure. Retrieved

October 5, 2017, from https://christineadamsbeckett.com/2012/04/03/thomas-

edisons-beautiful-failure/

[6] Khoshnevis, B. (2016, July 08). Transcript: Behrokh Khoshnevis on Contour

Crafting: Automated Construction. Retrieved October 5, 2017, from

https://singjupost.com/transcript-behrokh-khoshnevis-on-contour-crafting-

automated-construction/2/

[7] BlackBelt 3D Printer. (n.d.). Retrieved October 5, 2017, from https://blackbelt-

3d.com/

[8] Carranza, R. (2017, February 11). Border wall would cost $21.6B, nearly double Trump's estimate. Retrieved October 5, 2017, from http://www.usatoday.com/story/news/nation-now/2017/02/10/border-wall-would-cost-216-billion-homeland-security-report-says/97769162/

[9] Carriere, E. (2016, July 2). 25% of Dubai Buildings Should Be 3D Printed by 2030, Says Ruler. Retrieved October 5, 2017, from http://www.3dprintpulse.com/2030/?open-article-id=4988527&article-title=25--of-dubai-buildings-should-be-3d-printed-by-2030--says-ruler&blog-domain=3dprinter.net&blog-title=3d-printer

[10] CBP Awards Contracts for Border Wall Prototypes. (2017, August 31). Retrieved October 5, 2017, from https://www.cbp.gov/newsroom/national-media-release/cbp-awards-contracts-border-wall-prototypes

[11] W. (n.d.). Cement history. Retrieved October 5, 2017, from http://www.understanding-cement.com/history.html#

[12] The History of Concrete:. (n.d.). Retrieved October 5, 2017, from http://matse1.matse.illinois.edu/concrete/hist.html

[13] Edison Portland Cement Company. (2017, October 09). Retrieved October 5, 2017, from https://en.wikipedia.org/wiki/Edison_Portland_Cement_Company

[14] Frequently Asked Questions (FAQs). (n.d.). Retrieved October 5, 2017, from http://www.artba.org/about/faq/#6fe848d0c884f851d

[15] Goodwin, R. (2017, March 06). The History of Mobile Phones From 1973 To 2008: The Handsets That Made It ALL Happen. Retrieved October 5, 2017, from

http://www.knowyourmobile.com/nokia/nokia-3310/19848/history-mobile-phones-1973-2008-handsets-made-it-all-happen

[16] Gromiko, N., & Shepard, K. (2017, April 22). Introduction. Retrieved October 5, 2017, from https://prezi.com/9hc6x0a5zrj8/introduction/

[17] Grunewald, S. J. (2016, September 19). Where is Lewis Yakich? The Man Behind the World's First 3D Printed Hotel Suite Vanished a Year Ago Without a Trace. Retrieved October 5, 2017, from https://3dprint.com/149554/where-is-lewis-yakich/

[18] Henshaw, S. (2015, September 16). How to Choose an Infill for your 3D prints. Retrieved October 5, 2017, from https://3dprinting.com/tips-tricks/how-to-choose-an-infill-for-your-3d-prints/

[19] History.com Staff. (2010). Great Wall of China. Retrieved October 5, 2017, from http://www.history.com/topics/great-wall-of-china

[20] The History of Concrete and the Nabataeans. (n.d.). Retrieved October 5, 2017, from http://nabataea.net/cement.html

[21] History of lighthouses. (n.d.). Retrieved October 5, 2017, from http://www.wikiwand.com/en/History_of_lighthouses

[22] Home. (n.d.). Retrieved October 5, 2017, from http://contourcrafting.com/

[23] Kakaes, K. (2017, February 02). Play with the numbers yourself to see why Trump is wrong about the cost of a border wall. Retrieved October 5, 2017, from https://www.technologyreview.com/s/602494/bad-math-props-up-trumps-border-wall/

[24] Kerns , J. (2015, July 17). What's the Difference Between Stereolithography and

    Selective Laser Sintering? Retrieved October 5, 2017, from

    http://www.machinedesign.com/3d-printing/what-s-difference-between-

    stereolithography-and-selective-laser-sintering

[25] I. (2015, January 13). Iraytrace/BlenderGcodeImport. Retrieved October 5, 2017,

    from https://github.com/iraytrace/BlenderGcodeImport

[26] Low, K. N. (2011, September 18). Engineering the Pantheon - Architectural,

    Construction, & Structural Analysis. Retrieved October 5, 2017, from

    https://engineeringrome.wikispaces.com/Engineering the Pantheon -

    Architectural, Construction, & Structural

    Analysis?responseToken=6c1b92617ae7585212b5594e8b2489fe

[27] Lowe's ProServices: Buy in Bulk, (n.d) Commercial Services Buy in

    Bulk_577249990_. Retrieved October 5, 2017, from www.lowes.com/cd_Lowes

[28] Majewski, C. (2017, June 06). A 3D-printed rocket engine just launched a new era

    of space exploration. Retrieved October 5, 2017, from

    http://www.independent.co.uk/news/science/a-3d-printed-rocket-engine-just-

    launched-a-new-era-of-space-exploration-a7765496.html

[29] The Nano Revolution Against Concrete Degradation. (n.d.). Retrieved October 5,

    2017, from http://www.edencrete.com/2017/03/20/the-nano-revolution/

[30] S. (n.d.). Light Pollution and You [Photograph found in NASA Earth Observatory,

    United States]. Retrieved October 5, 2017, from S. (n.d.). [Photograph found in

    NASA, Space]. Retrieved October 5, 2017, from

https://nightsky.jpl.nasa.gov/news-display.cfm?News_ID=745 (Originally

photographed 2012) (Originally photographed 2012)

[31] R. (2017, May 25). New Zealand launches into space race with 3D-printed rocket.

Retrieved October 5, 2017, from

https://www.theguardian.com/world/2017/may/25/new-zealand-launches-space-

race-3d-printed-rocket

[32] Rudenko, A. (n.d.). 3D Castle Completed. Retrieved October 5, 2017, from

http://www.totalkustom.com/3d-castle-completed.html

[33] Sepulcre-Aguilar, A., & Hernandez-Olivares, F. (2012). Assessment of phase

formation in lime-based mortars with added metakaolin, Portland cement and

sepiolite, for grouting of historic masonry. Http://isrctn.org/>, 40(1), 66-76.

doi:10.1186/isrctn06195297

[34] The 7 Categories of Additive Manufacturing. (n.d.). Retrieved October 5, 2017, from

http://www.lboro.ac.uk/research/amrg/about/the7categoriesofadditivemanufacturi

ng/

[35] Z. (2016, October 16). Zignig/blender-gcode-reader. Retrieved October 5, 2017,

from https://github.com/zignig/blender-gcode-reader

[36] Sosnowski, P. (2017, June 12). How to use computer aided engineering to improve a

3D printer. Retrieved October 5, 2017, from

https://3dprintingindustry.com/news/use-computer-aided-engineering-improve-

3d-printer-115641/

[37] Sulleyman, A. (2017, March 14). World's first 3D-printed skyscraper to be built in

UAE. Retrieved October 5, 2017, from http://www.independent.co.uk/life-

style/gadgets-and-tech/news/3d-printed-skyscraper-worlds-first-built-uae-united-
arab-emirates-cazza-crane-printing-a7629416.html

[38] T. Rowe Price Associates (n.d.) "A BRIEF HISTORY OF 3D PRINTING." T. Rowe
Price Associates, Retrieved October 5, 2017, from
https://individual.troweprice.com/staticFiles/Retail/Shared/PDFs/3D_Printing_Inf
ographic_FINAL.pdf

[39] 3D printer. (n.d.). Retrieved October 5, 2017, from http://apis-cor.com/en/3d-printer

[40] "The Ultimate Guide to Stereolithography (SLA) 3D Printing." (2017 March)
FormLabs , FormLabs, Retrieved October 5, 2017, from
formlabs.com/media/upload/SLA_Guide.pdf.

[41] Preuss, P. (2015, June 01). Roman Seawater Concrete Holds the Secret to Cutting
Carbon Emissions | Berkeley Lab. Retrieved October 5, 2017, from
http://newscenter.lbl.gov/2013/06/04/roman-concrete/

[42] Pruitt, S. (2013, June 21). The Secrets of Ancient Roman Concrete. Retrieved
October 5, 2017, from http://www.history.com/news/the-secrets-of-ancient-
roman-concrete

[43] What's New. (n.d.). Retrieved October 5, 2017, from
https://www.simplify3d.com/software/release-notes/version-4-0-0/

[44] Yang, F., Zhang, B., & Ma, Q. (2010). Study of Sticky Rice−Lime Mortar
Technology for the Restoration of Historical Masonry Construction. Accounts of
Chemical Research, 43(6), 936-944. doi:10.1021/ar9001944

[45] Zimmer, L. (2015, January 25). 12,000-square-foot 3D-printed mansion pops up in
China. Retrieved October 5, 2017, from https://inhabitat.com/12000-square-foot-

3d-printed-mansion-pops-up-in-china/ [46] Buehler, E., Kane, S. K., & Hurst, A.

(2014). ABC and 3D: opportunities and obstacles to 3D printing in special

education environments. ASSETS '14 Proceedings of the 16th international ACM

SIGACCESS conference on Computers & accessibility, 107-114.

doi:10.1145/2661334.2661365

[47] Love, P. E., & Li, H. (2000). Quantifying the causes and costs of rework in

construction. Construction Management and Economics, 18(4), 479-490.

doi:10.1080/01446190050024897

[48] Curran, D. (2016, December 9). How virtual reality can have an impact on

manufacturing. Retrieved October 5, 2017, from https://www.virtualreality-

news.net/news/2016/dec/09/how-virtual-reality-can-have-impact-manufacturing/

[49] Locker, A. (2017, August). 9 Basic Types of 3D Printers – 3D Printing Technology

Guide. Retrieved October 5, 2017, from https://all3dp.com/1/types-of-3d-printers-

3d-printing-technology/

[50] How FDM 3D Printing Works. (n.d.). Retrieved October 5, 2017, from

http://www.stratasys.com/3d-printers/technologies/fdm-technology

[51] Berger, A. (2013, July 23). Cologne Cathedral - the eternal construction site | All

media content | DW | 23.07.2013. Retrieved October 5, 2017, from

http://www.dw.com/en/cologne-cathedral-the-eternal-construction-site/g-

16967900

[52] B. (2017, October 3). ING says 3D printing could account for half of manufactured

goods by 2060, wiping out 25% of global trade. Retrieved October 5, 2017, from

https://www.3ders.org/articles/20171003-ing-says-3d-printing-could-account-for-half-of-manufactured-goods-by-2060-wiping-out-25-percent-of-global-trade.html

[53] Klingenberg, B. (n.d.). Defining Fuzzy Sets. Retrieved October 5, 2017, from https://www.calvin.edu/~pribeiro/othrlnks/Fuzzy/fuzzysets.htm

[54] Cupar, A., Pogačar, V., & Stjepanovič, Z. (2015). Shape Verification of Fused Deposition Modelling 3D Prints. International Journal of Information and Computer Science, 4(0), 1. doi:10.14355/ijics.2015.04.001

[55] A. (n.d.). GCodeViewer - online gcode viewer and analyzer! Retrieved October 5, 2017, from http://gcode.ws/

[56] Stay tuned for 1.3.0! (n.d.). Retrieved October 5, 2017, from http://slic3r.org/

[57] Thread. (n.d.). Retrieved October 5, 2017, from http://www.print3dforum.com/showthread.php/339-Which-softwares-do-you-use-daily-or-more-often

[58] Amit's A* Pages   From Red Blob Games. (n.d.). Retrieved October 5, 2017, from http://theory.stanford.edu/~amitp/GameProgramming/

[59] Heuristics   From Amit's Thoughts on Pathfinding. (n.d.). Retrieved October 5, 2017, from http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html

[60] Woodford, C. (2017, January 21). How do stepper motors work? Retrieved October 5, 2017, from http://www.explainthatstuff.com/how-stepper-motors-work.html

[61] Zadeh, L. A. (1965). Fuzzy Logic. Information and Control, 8(3), 338-353. doi:https://doi.org/10.1016/S0019-9958(65)90241-X