



國立臺灣大學管理學院資訊管理研究所  
碩士論文

Department of Information Management  
College of Management  
National Taiwan University  
Master Thesis

大型3D列印結合龍圖兒內結構  
Large-scale 3D-Printing with Zometool Interior Structure

蔡佳昱  
Jia-Yu Tsai

指導教授：陳炳宇 博士  
Advisor: Bing-Yu Chen, Ph.D.

中華民國 105 年 6 月  
June, 2016

國立臺灣大學碩士學位論文  
口試委員會審定書

大型 3D 模型列印結合龍圖兒內結構

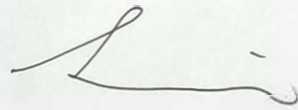
Large-Scale 3D-Printing with Zometool Interior Structure

本論文係蔡佳昱君（學號 R03725048）在國立臺灣大學資訊管理所完成之碩士學位論文，於民國 105 年 6 月 16 日承下列考試委員審查通過及口試及格，特此證明

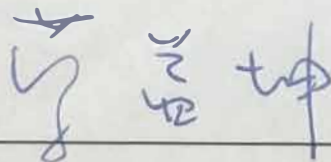
口試委員：



Mong-Te Chi



所 長：





## 中文摘要

積層製造又稱3D列印（3D printing），採用逐層疊加的製造方法，近年來3D列印技術被快速開發，此類型技術大大簡化了對複雜零件的加工製程。其中以熔融擠製成形的加工技術最為簡單、設計最容易，因此坊間許多自組機臺，也是現在桌上型3D列印機中使用最多的技術。我們提出一個新的3D列印方法，能夠比現有方法節省材料並且縮短列印時間，使得3D列印更加經濟。另外，本研究在3D列印的材料外還加入了使用Zometool的複合結構，使得3D列印的結構支撐上可以用一個新的方法代替，同時兼顧經濟，又能有非常良好的抗壓能力。第三個重要突破是，結合Zometool和3D列印，使得超大型的3D列印成品可以透過家用型的打印機，在省時、省材料的情況下實現，並且兼具堅固、耐壓等優點。

**關鍵字:** 3D列印、大型原型生成、龍圖兒結構、非流型結構、3D模型分割。



# Abstract

Additive manufacturing (AM) produces parts by adding material in layers, which significantly simplifies the process of producing complex 3D objects. Recently, AM technologies are well developed and fused deposition modeling is the simplest technique and is widely adopted to develop so-called 3D printers.

3D printers have become popular in recent years and enable fabrication of custom objects for home users. However, the cost of the material used in printing remains high and one of the remaining obstacles to wide-scale adoption is that the object to be printed must fit into the working volume of the 3D printers. In this paper, we present a new solution to design a complex Zometool structure for the purpose of reducing the material cost in printing a given 3D object and to decompose a large 3D object into smaller parts so that each part fits into the printing volume. These object parts can then be assembled with Zometool to form the original object. The *complex Zometool structure* is designed by an optimization scheme which significantly reduces material volume and is guaranteed to be physically solid, geometrically consistent, and printable. Furthermore, the formation of Zometool struts is duplicated repeatedly in the same pattern which is structurally sound. We formulate it as an iterative searching algorithm to find a compromise solution.

**Keywords:** 3D printing, large scale prototyping, Zometool structure, Non-manifold structure, 3D model segmentation.



# Table of Contents

口試委員會審定書	i
中文摘要	ii
Abstract	iii
List of Figures	vi
List of Tables	ix
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Larger, Better, Faster, Stronger . . . . .	2
1.3 Contribution . . . . .	3
<b>Chapter 2 Related Work</b>	<b>5</b>
2.1 Structural Soundness . . . . .	5
2.2 Printing Time Saving . . . . .	6
2.3 Large Scale Model Printing . . . . .	6
<b>Chapter 3 Overview</b>	<b>7</b>
<b>Chapter 4 Zometool Structure and Segment Generator</b>	<b>11</b>
4.1 preprocessing . . . . .	11
4.2 interior structure Generation . . . . .	11
4.2.1 Zometool-formed tetrahedron . . . . .	12
4.2.2 Propagate and pile up tetrahedrons . . . . .	13
4.2.3 Eliminate unnecessary tetrahedrons . . . . .	14
4.2.4 Articulation Points Detection . . . . .	15
4.3 Model to tetrahedron connection construction . . . . .	16
4.3.1 Different types of connection . . . . .	16
4.3.2 Selection of different types . . . . .	18

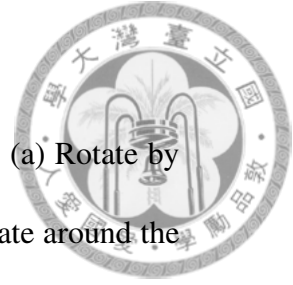


4.4	Additional Zometool struts construction . . . . .	19
4.4.1	Model surface segmentation . . . . .	20
4.4.2	Decision of whether to construct additional Zometool struts . . . . .	21
4.4.3	Additional Zometool member Growing Patterns . . . . .	21
4.5	Model to additional Zometool strut connection construction . . . . .	24
4.5.1	Different types of connection . . . . .	24
4.5.2	Selection of different types . . . . .	25
4.6	Zometool strut force analysis . . . . .	26
4.6.1	Prior knowledge . . . . .	26
4.6.2	The direct stiffness method . . . . .	27
4.7	Model to Zometool Connector Generation . . . . .	28
4.8	Between-Segment Split Surface Generation . . . . .	29
4.8.1	Splitting Segments . . . . .	29
4.8.2	Generate Split Surface . . . . .	29
4.9	Post Processing . . . . .	30
<b>Chapter 5 Implementation and Results</b>		<b>31</b>
5.1	Printed examples . . . . .	31
5.2	comparison . . . . .	33
<b>Chapter 6 conclusion, limitation and future work</b>		<b>40</b>
6.1	Conclusion . . . . .	40
6.2	Limitation . . . . .	40
6.3	Future Work . . . . .	41
<b>Bibliography</b>		<b>42</b>



# List of Figures

3.1	System overview. Given an input model, we will generate a set of printable segments. After that we should assemble them by hands to get the original model. . . . .	9
3.2	Interior structure arranged regularly as tetrahedrons . . . . .	10
4.1	a layer of Zometool structure inside the surface of the model. . . . .	12
4.2	a tetrahedron composed of Zometool components. . . . .	13
4.3	Tetrahedron composed of short struts in different way. . . . .	14
4.4	6 directions of our tetrahedron. . . . .	15
4.5	Eliminate unnecessary tetrahedrons (a) Fill out the entire volume. (b) Leave only a layer. . . . .	16
4.6	Avoid articulation points (a) Structure with articulation points. (b) Fill up tetrahedrons. . . . .	17
4.7	Model to tetrahedron connection. . . . .	18
4.8	Two groups of faces on a tetrahedron . . . . .	19



4.9	Some directions may look the same but actually different. (a) Rotate by 180 degrees around the axis vertical to blue struts. (b) Rotate around the axis along the blue struts by 90 degrees. . . . .	20
4.10	The face indexing of Zometool ball . . . . .	21
4.11	Additional struts construction (a) thin parts that tetrahedron can't be placed in. (b) Additional struts . . . . .	22
4.12	Model surface segmentation (a) Find the nearest point to the end point of connection. (b) Create a set of split plane (c) Leave the faces spreading through . . . . .	23
4.13	Model to additional Zometool strut connections . . . . .	24
4.14	We mark the parts whose tension are more than 200g. . . . .	28
4.15	Connector. (a) Create a columnar body inside the segment. (b) Do Boolean difference to create slots to let zometool insert. . . . .	29
4.16	Postprocessing. (a) Hollow out. (b) Split out (c) Boolean union (d) Boolean difference . . . . .	30
5.1	Kitten example. (a) Segmentation (b) Structure (c) Assembly process (d) Result . . . . .	32
5.2	Ateneav example. (a) Segmentation (b) Structure (c) Assembly process (d) Result . . . . .	33
5.3	Column example. (a) Segmentation (b) Structure (c) Assembly process (d) Result . . . . .	34





5.4	Trimmed ball example. (a) Segmentation (b) Structure (c) Assembly process (d) Result . . . . .	35
5.5	The shape of the models remained the same throughout the whole tests. (a) Our method. (b) Solid parts with glue. . . . .	36
5.6	in shell parts with glue case, the seams would get bigger slowly after the water filled in the bucket. . . . .	37
5.7	Shell parts with glue model collapsed slowly. . . . .	38
5.8	In the shell parts with connectors case, after we filled water into the bucket, the model deformed immediately and kept the shape still. . . . .	39



# List of Tables

4.1	Types of connections. The code means the components type. B:blue, R:red, Y:yellow, S:small, M:median, L:large. The directions can be retrieved in Figure 4.10 . . . . .	19
4.2	Additional Zometool struts types. The code means the components type. B:blue, R:red, Y:yellow, S:small, M:median, L:large. The directions can be retrieved in Figure 4.10 . . . . .	25
5.1	Printed results. In the second part, the first columns mean the components type. B:blue, R:red, Y:yellow, S:small, M:median, L:large . . . . .	35
5.2	Model data and processing results (part 1). . . . .	36



# Chapter 1

## Introduction

### 1.1 Motivation

Additive manufacturing (3D printing) enables fabrication of physical objects from digital models where the printed objects are created by laying down successive layers of material [1, 2]. During the last few years, research on computational techniques of 3D printing has received considerable attention for assisting users to generate desired manufacturable objects [4, 6, 8, 9]

However, reducing the material used in printing, which is an important problem due to its high cost, has not been well studied. The straightforward approach used in commercial printer packages [2] is to uniformly hollow the 3D object by extruding the outer surface and creating a scaled-down version on its inside. The user has to choose a scaling factor (thickness of object) based on experience. A large factor may lead to material waste while a small factor could cause structural stability problem. Thus it is technically nontrivial for the hollowing method to simultaneously match the goals of saving material and maintaining physical solidity in 3D printing. At the same time, the maximum size of an object that



a 3D printer can produce in one pass (the printing volume) is limited by practical considerations. Larger objects must therefore be printed as separate parts and assembled. Existing commercial systems rely on manual partitioning, but this can be tedious if many parts are needed. An optimization of how an object was manufactured is used to determine the precise way in which fabrication devices should be driven to produce an object as close as possible to the goal. The optimization may be formulated to consider visual properties such as scattering [5] or physical properties such as deformation [3], but in this paper we focus purely on 3D shape. We propose that this kind of optimization approach may be applied to the problem of decomposition into print volumes, yielding an automated method that is quicker, easier, and in many instances better than can be accomplished by hand.

## **1.2 Larger, Better, Faster, Stronger**

The key idea is to ‘hollow’ the object by creating an interior structure, made of a mesh of Zometool with separated voids among them inside the object between the Zometool struts, and then separate the hollowed objects to smaller parts to fit in the 3D printers. Complex Zometool structures benefit 3D printing in two aspects. First, the mass of object could be significantly reduced through the use of Complex Zometool structures while maintaining its strength and stiffness. Second, Zometool could be reformed in various figure to approximate the original object’s surface, which makes large objects with various appearance could be formed easily. We develop a specific pattern of Zometool formation to ensure its solidity and printability. Because of the certain pattern of Zometool formation, we avoid the structural optimization problem which is proved to be NP-Hard and



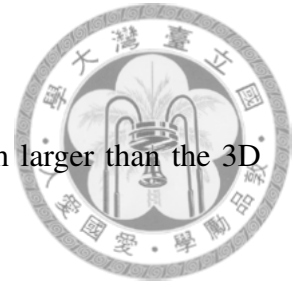
ensure the surface partition could be fit in the 3D Printers. To this end, we formulate it as an iterative searching algorithm to find a compromise solution. The algorithm starts by iteratively construct the inner Zometool structure and computes the offset between the ending points of the Zometool structure and the surface of the original object in different positions and rotations. The computation is to search for the minimized offsets of a way to form the Zometool structure within the original object, which optimizes the marginal adaption between the Zometool structure and the original object's surface. Different from other method, the structure can keep the surface on the correct position.

The complex Zometool structure generated by our algorithm is physically solid especially when the object is large, geometrically consistent after composition of the Zometool Structure and the surface partitions, and printable in 3D printers, formable by hand, which is cost-effective for printing, and exceed the large appearance with structurally solidity.

### **1.3 Contribution**

We propose a novel approach for automatically generating a complex Zometool structure to minimize material usage in 3D printing and enable large objects in 3D printing with structural soundness, taking into account structural solidity, geometrical consistency, formability by hand, and printability of the object. Our contributions are summarized as follows.

- Structural firmness: An marginal adaption optimization is adopted to promote the firmness between the Zometool structure and the the hollowed surface of the original object.



- **Printability:** the volume of the object to print could be much larger than the 3D printer.
- **Assemblability:** we could assemble the Zometool structure and the partition of the hollowed object by hand.
- **Efficiency:** we could avoid to print the inner structure of the object while the thin hollowed outer surface is all we need to print to assemble the object, which saves time and the material.
- **Connectability:** the algorithm optimizes the marginal adaption between the Zometool structure and the hollowed surface of the original object, which make the connection between the Zometool structure's ending points and the parts of the partitioned hollowed outer surface of the original model feasible.
- **Structural soundness:** the specific pattern of the Zometool formation ensure the structural soundness of the whole assembled object even when the object is large.
- **Accuracy:** By the supporting structure, the surface of the model can be on the correct position.



## Chapter 2

### Related Work

Our work has three main goals, which are structural Soundness, Saving Printing Time and Exceeding Large Scale Model Printing. To achieve these goals, there are some previous researches that could give us some information.

#### 2.1 Structural Soundness

With 3D printers being more accessible to everyday users due to their rapid drop in price, more and more attention has been paid to fabrication techniques and applications. Quite a few topics in 3D fabrication, including structural analysis [9, 12], segmentation [6]. Based on finite element analysis of stress, the approach of [9] can strengthen printed objects by hollowing, local thickening, and adding extra struts. Different from the goal of our work, it does not always reduce the volume of printing material. A hollowing optimization algorithm based on the concept of honeycomb-cells structure was proposed to provide maximal strength in tension with minimal material cost [Lu et al. 2014]. A skin-frame structure for the purpose of reducing the material cost in printing a given 3D



object is designed to hollow out the model to be printed, which succeed a very good result [11].

## **2.2 Printing Time Saving**

Printing Time saving has been addressed by [7], where the object is converted into a truss skin-frame structure. PackMerger framework, the newest solution proposed, converts the input 3D watertight mesh into a shell by hollowing its inner parts [10]. The shell is then divided into segments. A skin-frame structure hollow scheme also achieve an excellent result in printing time saving [11].

## **2.3 Large Scale Model Printing**

Similar to some parts of our approach is the work of Chopper [6] where objects larger than the printing volume are divided into pieces and then printed separately. However, their work does not solve the problem of saving the printing material and is aimed at printing large objects.





## Chapter 3

### Overview

Our flow chart is shown in Figure 3.1. First, the user must input a 3D model which is big enough, the program will randomly choose a point inside this model, and generate interior Zometool structure inside the model surface from this point from this point. structure generated are arranged regularly as tetrahedrons( as shown in Figure 3.2 ). , constructed as a layer of tetrahedrons inside the model surface.

Secondly, we would try different ways to connect from each tetrahedron to the model surface. The connection would be in certain pattern which we are going to describe in chapter 3 . We would try to find the most suitable pattern and build the Zometool struts. After that, we would split the model to the segments which connect to the model surface in a point. The segment's center is at this point. We compute the surface area of this segment. when there is a segment whose area is more than the threshold, we would try to build additional struts between the segment and the tetrahedron and split the segment into even smaller segments.

By this step, we are done generating the Zometool structure. In the meanwhile, there



might be a distance between the connection and the model surface. Thus we make a segment grow a columnar body to connect to the Zometool structure. The columnar body will be used for making segments and the Zometool structure fit together.

By now the interior structure is completely generated, we need to divide the model surface into pieces so that the size of each piece of segment can be printed from the 3D printer. Here we apply the most simple approach to split the model surface, that is, cut the model surface through the split plane of two connection points on the surface segmentation model. For each segment we will generate a set of split plane, we can at last split the model surface through the set of split plane to get the segment we want. Now Our program is coming to an end.

Finally, we need to use some other software to do the rest of the process, including hollowing out solid model into shell, as well as using the sets of split plane to split out the segments which is small enough to be printed in a 3D printer.

After all pieces of segment are printed out, we can assemble them together to get the original model.

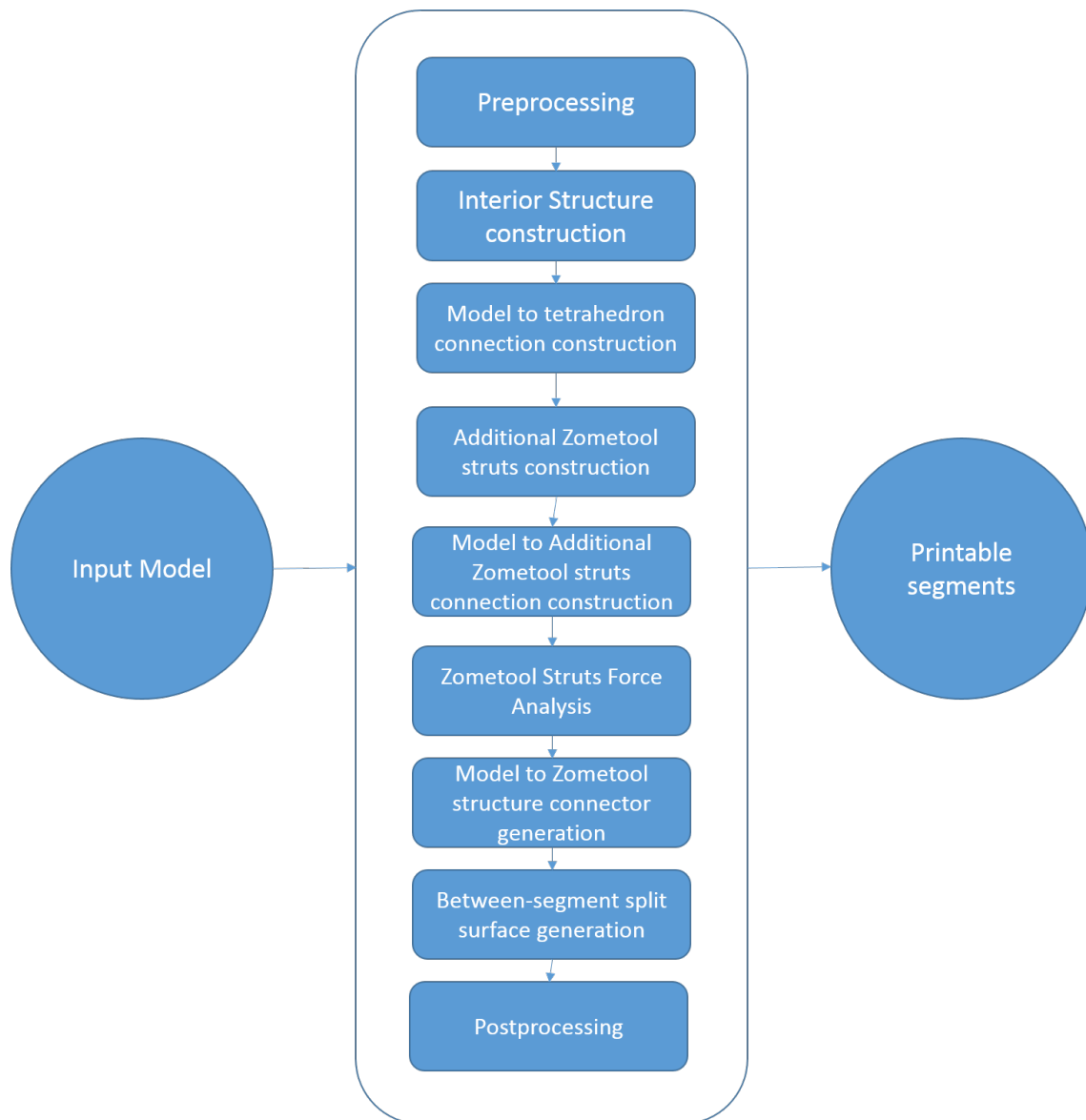


Figure 3.1: System overview. Given an input model, we will generate a set of printable segments. After that we should assemble them by hands to get the original model.

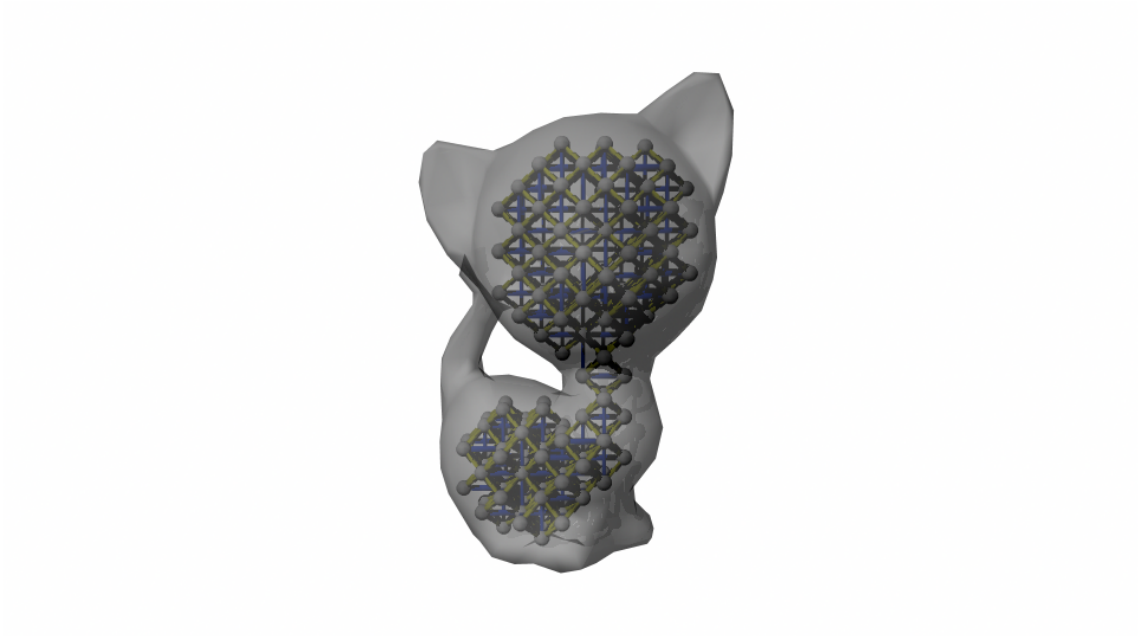


Figure 3.2: Interior structure arranged regularly as tetrahedrons



## Chapter 4

# Zometool Structure and Segment Generator

### 4.1 preprocessing

Our program requires a watertight model, with each face triangular. Thus we will first edit the wrong edges through other software. Furthermore, in order to enhance the efficiency of the program, we reduce the number of faces on model through MeshLab to about 1000 before our operation..

### 4.2 interior structure Generation

Now we need to construct a set of Zometool structure with smaller size than the model surface and strong structure. In earlier studies, such as Zometool Shape Approximation, Large-Scale Rapid-Prototyping with Zometool, use complex algorithms, and analyze if the structure is robust in order to make Zometool closer to the shape of the model. In contrast, we only need Zometool for supporting the model. We don't need Zometool to shape similar to the model surface. As a result, we chose a simple but efficient method is to



place a layer of Zometool structure composed of the same and very strong structure Figure 4.1 on inside the surface of the model, which is a tetrahedron composed of Zometool components. We closely spaced tetrahedrons one by another( as shown in Figure 4.2).

#### 4.2.1 Zometool-formed tetrahedron

Tetrahedron is three-dimensional trusses, which makes the structure of its struts almost exclusively bear the pressure and tension force while the shear force and the moment are negligible. Zometool can form many different shapes of tetrahedron. Among them we choose which consisted of the shortest blue and yellow struts (as in Figure 4.2) which is because through our attempts, we found that if you use medium or long the struts to make tetrahedron, the output would end up too big to print in the home 3D printers. Additionally, we can compose tetrahedron with short struts in two different ways (as shown in Figure 4.3), Among them only the yellow and blue can closely placed one by another

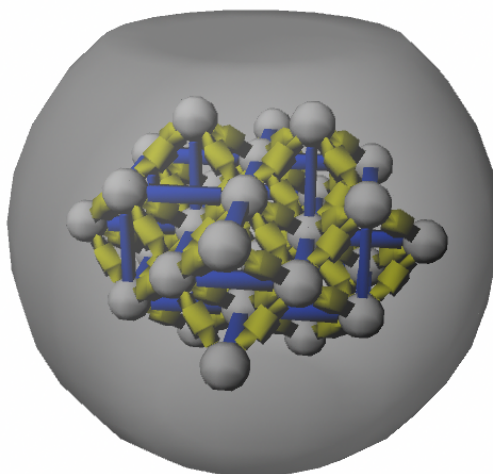


Figure 4.1: a layer of Zometool structure inside the surface of the model.



repeatedly in the space so we choose it as the basic elements of our interior structure.

## 4.2.2 Propagate and pile up tetrahedrons

The tetrahedron we chose has six directions (Figure 4.4 ), and when we want to closely place the tetrahedrons one by one repeatedly, we have to rotate them to an appropriate direction in order to allow the combination of the two adjacent tetrahedrons. We choose a random starting point to build the first tetrahedron, and then keep replicate it, and rotate the replicated tetrahedrons to combine with the existing tetrahedron, and then modify this new tetrahedron with the coordinates right next To the existing tetrahedron.

If a tetrahedron doesn't intersect with the surface of the model, tetrahedron would keep propagating until the new tetrahedron intersect the surface of the model then it will stop

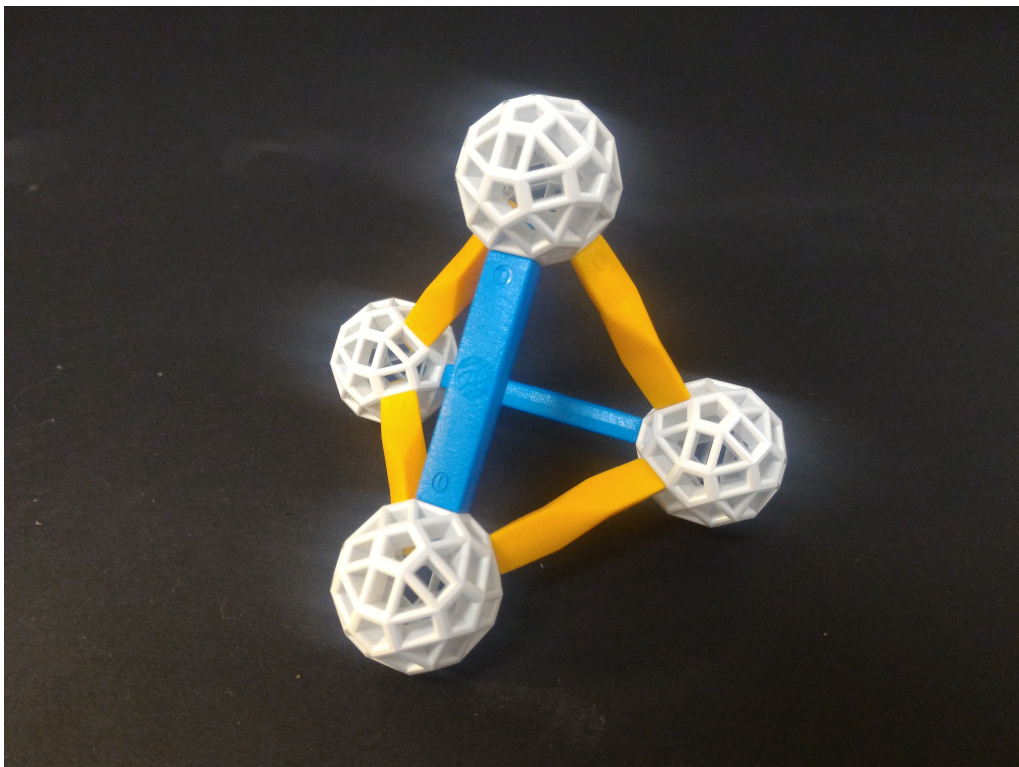


Figure 4.2: a tetrahedron composed of Zometool components.



propagating. In the meantime, the other tetrahedron which doesn't intersect the surface of the model will continue to propagate. The whole process will come to an end while all tetrahedrons have propagated or intersected the surface of the model.

### 4.2.3 Eliminate unnecessary tetrahedrons

After the previous operation, we can get tetrahedrons filling up the entire volume of the model (as shown in Figure 4.5.a), but we only need to create a layer of tetrahedrons right inside the model to get adequate support. Therefore we should eliminate unnecessary tetrahedrons. In order to select the appropriate tetrahedrons to leave and eliminate the other, we use the number of tetrahedrons between tetrahedron to the surface of the model as a basis. Here we select the tetrahedrons whose spacing between itself and the

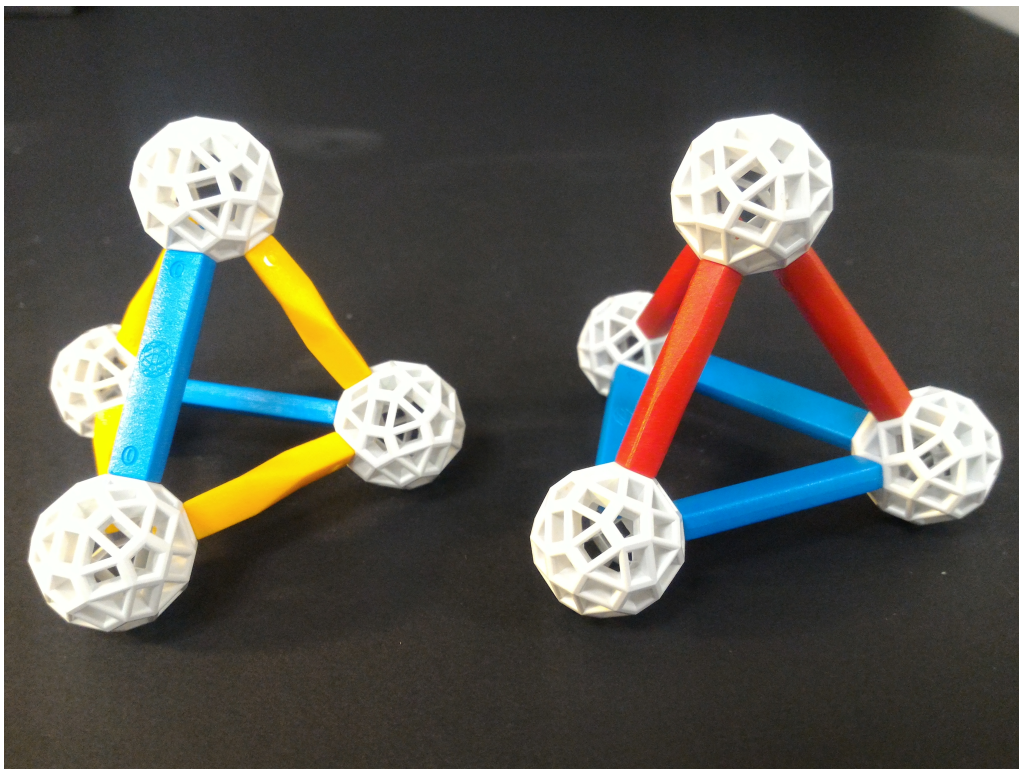


Figure 4.3: Tetrahedron composed of short struts in different way.





tetrahedrons intersect the surface of the model is one tetrahedron to leave, and the other is eliminated (Figure 4.5.b).

#### 4.2.4 Articulation Points Detection

Most of the time, tetrahedron-based structure will be very strong, but sometimes in between the two groups of tetrahedrons only connected through 1 to 2 points. The points will bear shear force and moment (as shown in Figure 4.6.c), and likely to cause damage to the structure. Therefore we must identify the existing Articulation points after eliminating unnecessary tetrahedrons, and must fill up tetrahedrons between the tetrahedrons on both sides of the points (as shown in Figure 4.6.d), to avoid articulation points.

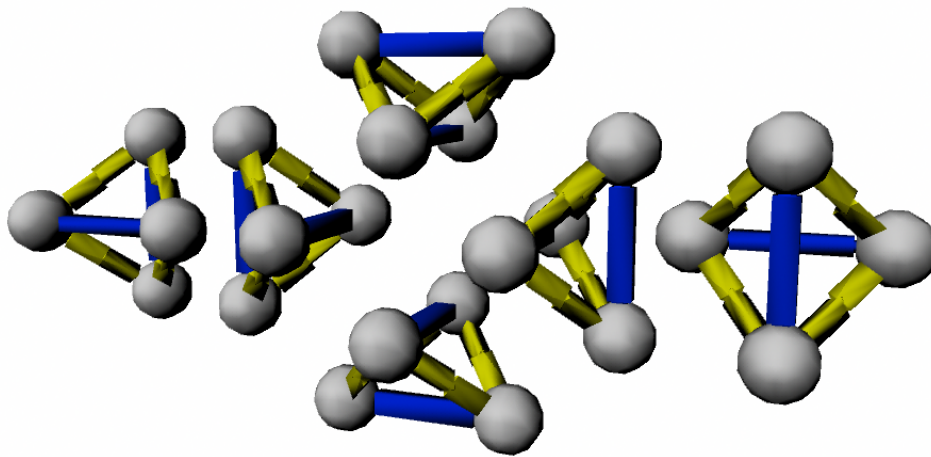


Figure 4.4: 6 directions of our tetrahedron.



### 4.3 Model to tetrahedron connection construction

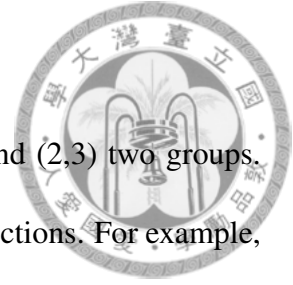
To firmly connect the surface of the model divided into segments with interior structure, the struts must be built between the structure and the surface of the model, as shown in Figure Figure 4.7.a. In order to make the segment is connected firmly enough, we set each segment must be connected to three struts at the same point, as shown in Figure 4.7.b. We chose to use three struts and connected them to the same point because the struts and the tetrahedron connected with also form a tetrahedron. Also, three struts form different directions connected at the same point will lock the rotation of the segment, which makes the segment very stable on the same position and rotation.

#### 4.3.1 Different types of connection

A face on a tetrahedron can grow out a connection. After we test every possible way to make a connection, we found 9 kinds of combinations, as shown in Table 4.1. The tetrahedron connected with has several properties affecting the composition of the connection. we can find it out in the field "Restricted in certain faces on source tetrahedron" and "Restricted faces of sources tetrahedron." This is because of that the four faces on



Figure 4.5: Eliminate unnecessary tetrahedrons (a) Fill out the entire volume. (b) Leave only a layer.



a Tetrahedron (shown in Figure 4.8.a) respectively fall into (0,1) and (2,3) two groups. Each group can be connected with different types of Zometool connections. For example, if we want to connect a tetrahedron with a type-3 connection, its one middle size red truss and two blue trusses can only be connected to face 2 or 3, but can't be connected to face 0 or 1.

According to the types of connections, we build connections on the faces facing the surface of the model on the tetrahedrons. The faces have different direction so we rotate the struts in the connections to fit the faces on the tetrahedrons. In our generator, the six directions are actually extended into 24 directions. This is because some directions may look the same, but the actual in different directions. For example, a tetrahedron which is rotated by 180 degrees around the axis vertical to blue struts looks the same (as shown in Figure Figure 4.9.a) but actually in different directions. Each direction of tetrahedron are then rotate around the axis along the blue struts by 90 degrees and we take them as 2 different direction (Figure 4.9.b), this method allows us to rotate the tetrahedron easily in the previous sections.



Figure 4.6: Avoid articulation points (a) Structure with articulation points. (b) Fill up tetrahedrons.



### 4.3.2 Selection of different types

When creating Connection we will have to try all types of connection, and test if the type of connection intersect the surface of the model, whether intersect other Zometool components. After we make sure this type of connection is available we would measure the distance from the end point of connection to the surface of the model and the distance to the nearest end point of other connection. We then calculate a weighted distance through an equation as follows:

$$E = D + (\max(0, 100 - DP)) \cdot 2/50$$

In addition, if we found that the end point of the type of connection has the nearest distance to other end points less than the threshold (12mm) in the final, the type of connection would be unavailable.

After testing every type, we would choose the type with lowest E as the connection we want to construct. If all Type are unavailable, then we wouldn't construct any connections.

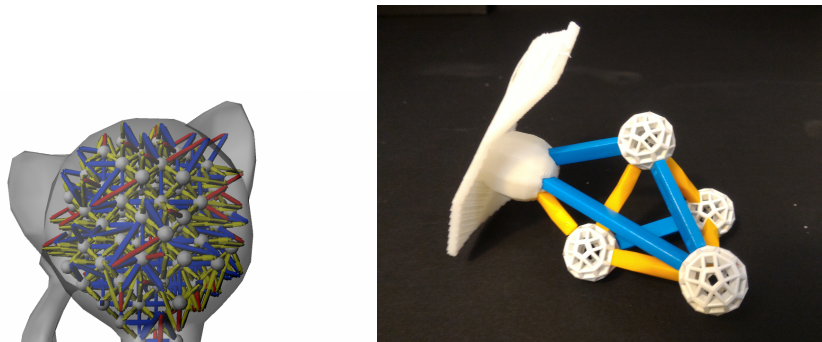


Figure 4.7: Model to tetrahedron connection.



model	Color	Size	Direction	restricted	source face
1	YYB	SSS	5,58,61	No	
2	RYY	SMM	36,49,48	Yes	0,1
3	RBB	MMM	38,7,6	Yes	2,3
4	YRR	MSS	53,41,40	Yes	0,1
5	RBB	MSS	39,25,23	Yes	2,3
6	RBB	MMS	41,12,27	Yes	0,1
7	YBB	SMS	49,19,12	Yes	2,3
8	RBB	MSM	40,26,10	Yes	0,1
9	YBB	SSM	46,13,21	Yes	2,3

Table 4.1: Types of connections. The code means the components type. B:blue, R:red, Y:yellow, S:small, M:median, L:large. The directions can be retrieved in Figure 4.10

#### 4.4 Additional Zometool struts construction

Most Zometool struts will be generated in the previous steps, but sometimes the struts can not be very close to the surface of the model of the surface. It particularly happens near the thin parts (Figure 4.11.a), so we have to construct additional struts to get close to those parts which are too far from the tetrahedrons (Figure 4.11.b). To identify whether

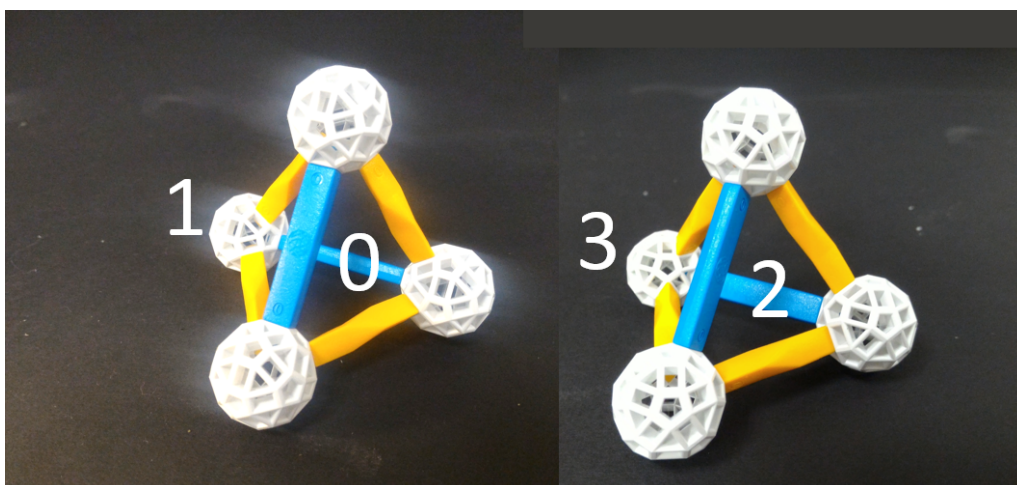
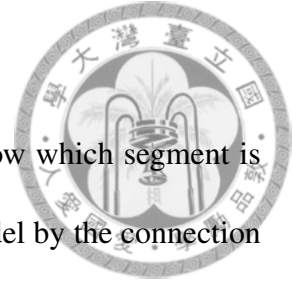


Figure 4.8: Two groups of faces on a tetrahedron



the tetrahedron is too far from the surface of the model, and to know which segment is the tetrahedron connected with, we must split the surface of the model by the connection points distribution. After that, we split out every segment to find out if there is a point on the segment which is too far from the connection point.

#### 4.4.1 Model surface segmentation

In order to obtain the distance from the farthest point on the segment to the connection point of each segment, we must first split the surface of the model into segments. Now we create a segment that connects with a connection. At first we find the nearest point on the segment to the end point of connection as  $N$ . Secondly, we put every other nearest points on the other segments into a set of points, denoted by  $N'$ . We Create a set of split planes  $S$  between  $N$  and every points in  $N'$ . Next, we take  $N$  as the center to spread along the surface of the model, until it touches any split plane in  $S$ . We record all the faces spreading through, and create a polygon when a face intersect a split plane whose shape is the face divided by the split plane. (Figure 4.12.c)

In this step, sometimes because of the unlimited extension of the split plane, in 3D

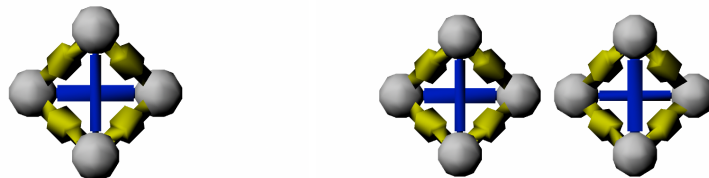


Figure 4.9: Some directions may look the same but actually different. (a) Rotate by 180 degrees around the axis vertical to blue struts. (b) Rotate around the axis along the blue struts by 90 degrees.



space it will split some faces which should not be split. To solve this problem, we would find the faces that not belong to any tetrahedron to combine to the adjacent segment.

#### 4.4.2 Decision of whether to construct additional Zometool struts

Through the above steps, we can obtain the shape of each segment later to calculate the area of these segments. Then we choose the segments whose surface area is so large that the additional struts are needed. We determine whether to

#### 4.4.3 Additional Zometool member Growing Patterns

In order to provide more support, and let the segments divided smaller, we need to construct struts with sufficient supporting force can be effectively approaching the point which is the farthest form N-point. we denote the point by F. The parts which have too large segment is often so narrow that tetrahedrons can't be placed inside it (Figure 4.11.a). Therefore we have to come through these parts with finer patterns (as shown in Fig-

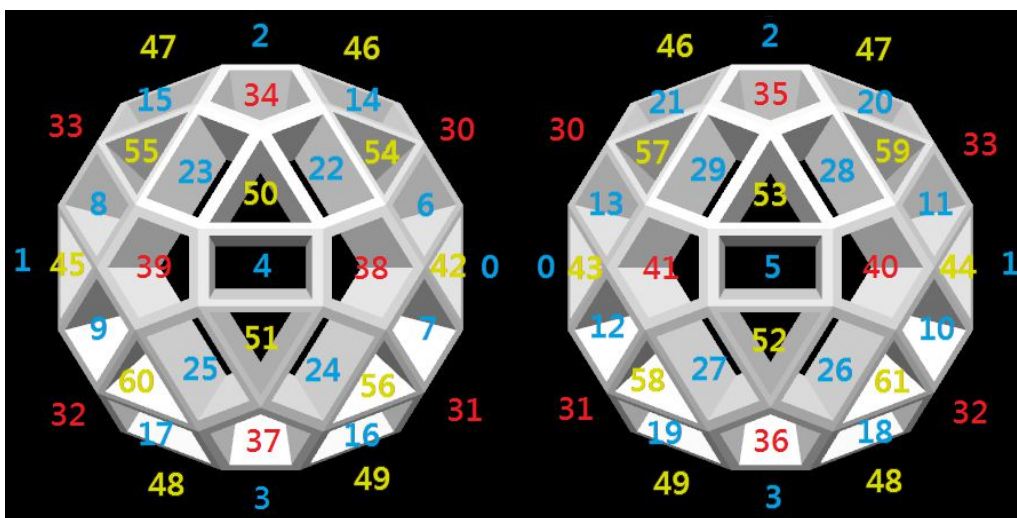


Figure 4.10: The face indexing of Zometool ball



ure 4.11b).

In order to achieve these requirements, we chose the red short Zometool strut. The red bars have greater MOI (Moment of Inertia) so they have greater resistance to bending when subjected to bending moments.

We tetrahedron face to face for the source of their model of the surface, the three vertices are going to try to establish additional zometool member, according to final results established a set of best choice to stay, and the other two groups deleted.

We try to construct the struts from each corner of the face on the tetrahedron, then choose the best result to leave and delete the other two results. The directions can be retrieved in Figure 4.10.

When constructing the struts, we try to put the red strut toward every direction at one node and iteratively try at the next node. Struts created must meet the following conditions, otherwise it would not be successfully constructed:

1.  $FC^2 + relax > TC^2 + TF^2$  FC stands for the distance from tetrahedron face to F-point, relax is used to buffer, letting the struts grow diversely. Here we set to 2000mm<sup>2</sup>. TC stands for the distance from the corner of the tetrahedron face to the strut. TF stands

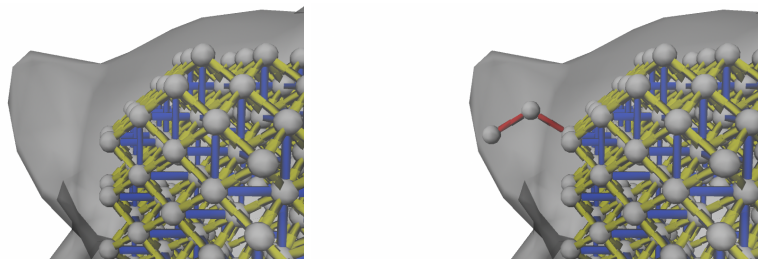


Figure 4.11: Additional struts construction (a) thin parts that tetrahedron can't be placed in. (b) Additional struts





for the distance from the F-point to the strut. This equation is to avoid the struts to grow in wrong direction.

2. The strut does not intersect the surface, and the distance between the surface of the model surface exceeds node radius + thickness.

3. do not intersect with other struts or nodes.

When finished constructing one of the group of struts, we would search for the shortest path between the face on tetrahedron and the nearest strut to F-point. After that we delete the struts which are not on the shortest path. Finally, we compare the three groups with its distance between the nearest strut to F-po

$$D = Nm + D / e$$

Where  $Nm$  is the number of the shortest path struts, and  $D$  is the shortest distance between F-point and the nearest strut.  $e$  is the coefficient, we set it to 5. We will leave the group with minimum  $D$  and delete the remaining two groups.

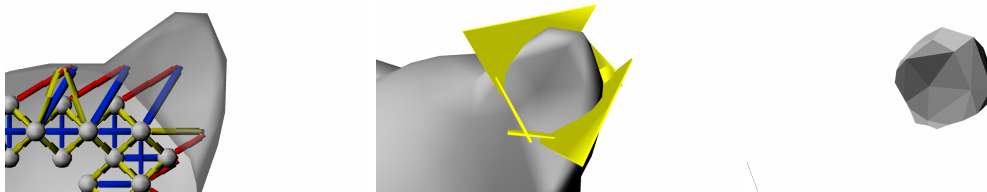


Figure 4.12: Model surface segmentation (a) Find the nearest point to the end point of connection. (b) Create a set of split plane (c) Leave the faces spreading through



## 4.5 Model to additional Zometool strut connection construction

In order to provide more support by additional Zometool struts, we'll build connection from every strut, and connect with the Segment. Similar to the previous "Model to Tetrahedron Connection Generation" approach, we construct connection members from both ends of a strut is connected with the segment at the same point. Two struts connected with the segment will be less robust than three struts connected with the segment, but more fine to put a small part. (Figure 4.13.a)

### 4.5.1 Different types of connection

We found eight ways to construct Additional Zometool struts connection, as shown in Table ???. Connections can be built in 6 directions on Red short strut, so we would try to build connections in six directions and the eight types on every additional Zometool

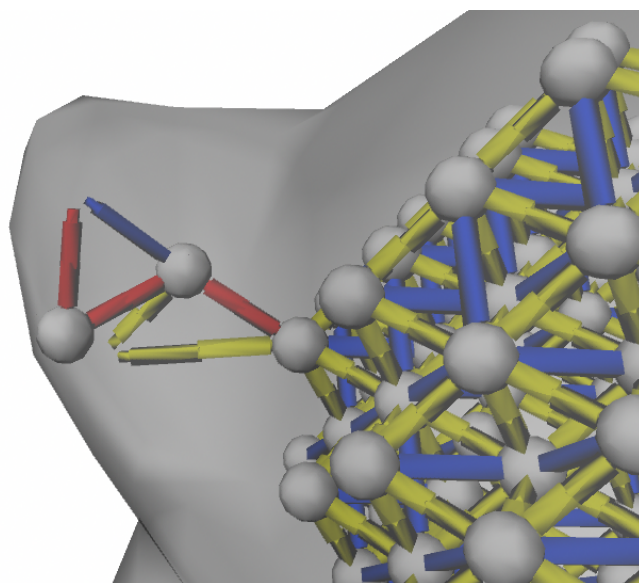


Figure 4.13: Model to additional Zometool strut connections



struts, and finally select the best result to construct.

model	Color	Size	Direction
1	BR	SS	2,33
2	RB	SS	34,8
3	YY	MS	46,47
4	YY	SM	55,45
5	BR	MS	14,34
6	RB	SM	33,1
7	RY	MM	34,55
8	YR	MM	47,33

Table 4.2: Additional Zometool struts types. The code means the components type. B:blue, R:red, Y:yellow, S:small, M:median, L:large. The directions can be retrieved in Figure 4.10

#### 4.5.2 Selection of different types

When creating Connection we would try every type of connection, and testing if each connection intersects the surface of the model, and whether each connection intersect other struts or nodes. After we make sure this type of connection is available we would measure the distance from the end point of connection to the surface of the model and the distance to the nearest end point of other connection. We then calculate a weighted distance through an equation as follows:

$$WD = D + (\max(0, 100 - DP)) / 50$$

In addition, if we found that the end point of the type of connection has the nearest distance to other end points less than the threshold (12mm) in the final, the type of connection would be unavailable.



After testing every type, we would choose the type with lowest E as the connection we want to construct. If all Type are unavailable, then we wouldn't construct any connections.

## 4.6 Zometool strut force analysis

Although the tetrahedron-based structure is very robust, the tensions that Zometool struts could bear is much less than the pressure. Because of that, we have to analyze the force in the structure to know which parts will bear excessive tension, so that we could tie up wire or apply the glue during the assembly to increase the maximum tension it could bear.

General studies use finite element stress analysis method to analyze the whole mesh of an object, but while our structure is truss-based structure which can be analyzed easily by the direct stiffness method to get a result similar to the practice.

### 4.6.1 Prior knowledge

Before analysis, we must first know the maximum tension one strut could bear. After a simple experiment we found the maximum tension which is more than 200g each contact between a strut and a ball. To make the structure safer, we set the maximum tension of the structure to 200g. Based on this constraint, we apply the analyze through the stiffness on each connection point, find out the stiffness matrix to list each equation on different axes and solve the simultaneous equations to get the force applied on each connection point and the displacement on each connection point. We take each segment's weight to be 20g, and we mark the parts (Figure 4.14) while its connection point's tension force is



more than 200g.

#### 4.6.2 The direct stiffness method

All known methods of structural analysis are classified into two distinct groups: (i) force method of analysis and (ii) displacement method of analysis. The direct stiffness method is one of displacement methods. In the displacement method of analysis the equilibrium equations are written by expressing the unknown joint displacements in terms of loads by using load displacement relations. The unknown joint displacements (the degrees of freedom of the structure) are calculated by solving equilibrium equations. After the revolution in computer industry, only direct stiffness method is used in structural analysis area.

Let an elastic body is acted by a force  $F$  and the corresponding displacement be  $u$  in the direction of force. The force ( $F$ ) is related to the displacement ( $u$ ) for the linear elastic material by the relation

$$F = ku$$

where the constant of proportionality  $k$  is defined as the stiffness of the structure and it has units of force per unit elongation.  $k$  can also be calculated by

$$k = (AE)/L$$

where  $A$  is the cross-sectional area of the truss.  $E$  is the elastic module of the truss.  $L$  is the axial directional length of the truss. In a multiple trusses structure we could establish a member stiffness matrix to obtain an equilibrium which was to find the unknown displacements.



$$[p] = [K][u]$$

where  $p$  is the vector of joint loads acting on the truss,  $u$  is the vector of joint displacements and  $K$  is the global stiffness matrix. The above equation is known as the equilibrium equation. It is observed that some joint loads are known and some are unknown. Also some displacements are known due to support conditions and some displacements are unknown. Hence the above equation may be partitioned and solved.

## 4.7 Model to Zometool Connector Generation

Although the connection from tetrahedrons and additional struts are as close to the surface of the model as it could, we still need a connector to make a robust connection between the zometool and the surface of the model surface. We create a columnar body attached inside a segment and connect with the Zometool. (Figure 4.15)

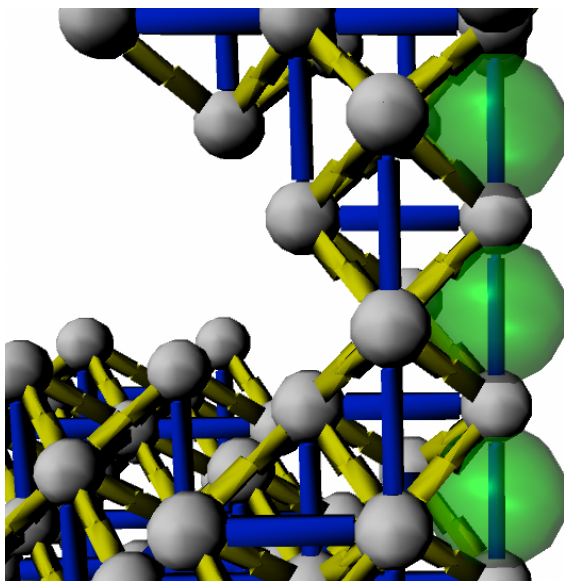


Figure 4.14: We mark the parts whose tension are more than 200g.



## 4.8 Between-Segment Split Surface Generation

For the segment we want to split out, we will generate a group of split plane between F-point and all the other F point on other segments. After the surface of the model is split out by all the split plane, the rest of segment with F-point is the segment we want. Since we still need some other software, we will not directly generate a segment as output, but generate a set of split plane so that we can finally do boolean operations on the surface of the model.

### 4.8.1 Splitting Segments

We use the same method mentioned in 5.4.1 to generate segments, but because we still need some other software, we will not directly generate a segment as output, but leave the segment data for further use.

### 4.8.2 Generate Split Surface

Previously we get a set of split plane and a segment., in order to make the postprocessing easy, we will generate a certain set of split plane specifically for each segment,



Figure 4.15: Connector. (a) Create a columnar body inside the segment. (b) Do Boolean difference to create slots to let zometool insert.



as shown 5.4.2.c. We reserve 2mm of seam between each segment so that the assembly would be easier.

## 4.9 Post Processing

After all the previous generation, we will use some other software (such as blender) to complete the final processing for the results, including hollowing out solid model into shell, split the shell into segments, do boolean union between segments and connectors, and do boolean difference on connector from Zometool struts. Then we can print the final results. (Figure 4.16)

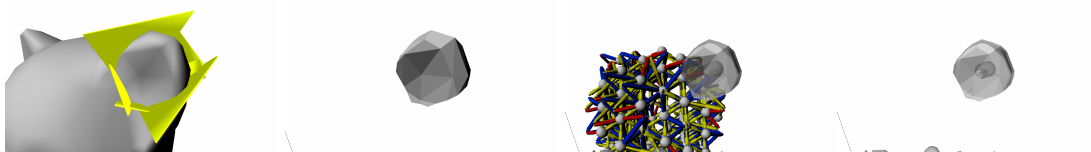


Figure 4.16: Postprocessing. (a) Hollow out. (b) Split out (c) Boolean union (d) Boolean difference





## Chapter 5

# Implementation and Results

The system is implemented in C++. All tests were run on Intel i5 CPU clocked at 1.80 GHZ with 4 GB of memory.

### 5.1 Printed examples

We used UP 3D Printer with ABS material in fast speed and 0.4mm layer thickness. 4 printers print out models at the same time. Since the objects we want to print are very large, printing and assembly time consumes a lot. Thus, we only print 4 results as shown in Table ?? and Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4. Kitten has height of more than 50 cm, divided into 159 parts, total print time is more than 100hr. In order to maintain the connector recesses shape, when we printed the recesses upside, which make a lot of support (Figure 6.1.2.a), so the removal of support took a lot of time (each segment took a couple of minutes). Assembly of interior Zometool structure of Kitten took more than 4 hours, which of Ateneav and Column took about two hours, which of



trimmed ball took more than ten minutes. Connecting the Zometool with the segments is also very time-consuming, assembling kitten took more than a day. Assembling Ateneav and Column and took more than two hours for each. Assembling the trimmed ball took about an hour. We left 2mm seams between segments, after assembly is completed we found that some parts of the seams will be particularly large (Figure 6.1.2.b), we believe that the reasons include: 1. connector deformed during printing so that segment can not connect closely with zometool. 2. When a large-scale model, interior Zometool is hard to connect closely because of the force. 3. segments deformed during printing.

After assembly is completed, the bottom of Kitten will loosen due to the overall weight of oppression. Therefore, we believe the bottom of the structure must be more robust to allow the model to avoid damage.

In our approach, most of the segments will be connected with three Zometool struts to stabilize in the Zometool structure. When the combination of segments and Zometool is coming to the end, there will be full of segments assembled, which makes assembling new segments difficult (Figure 6.1.3.a). Now remove one Zometool strut then it will make



Figure 5.1: Kitten example. (a) Segmentation (b) Structure (c) Assembly process (d) Result



the assembly become very simple.

Zometool Structure usually makes the model very strong, which we can see from the next paragraph of comparative experiments. but sometimes there will be more vulnerable parts, such as the neck parts in Kitten. Although we use tetrahedron-based structure to avoid shear force and bending moment, Zometool can withstand very small tension with respect to the pressure. So according to the result of stress analysis, some parts of Zometool structure bearing strong tension need to be tied wire to keep the structure robust(Figure 6.1.3.b).

## 5.2 comparison

To prove the necessity of Zometool structure in large-scale models, we fabricated Trimmed Ball in different way to measure its print time, aspect, capacity, and assemble time. Former studies about large-scale model 3D printing usually print models in solid or shell, and assemble segments through glue or connectors. Thus, we chose dimensional

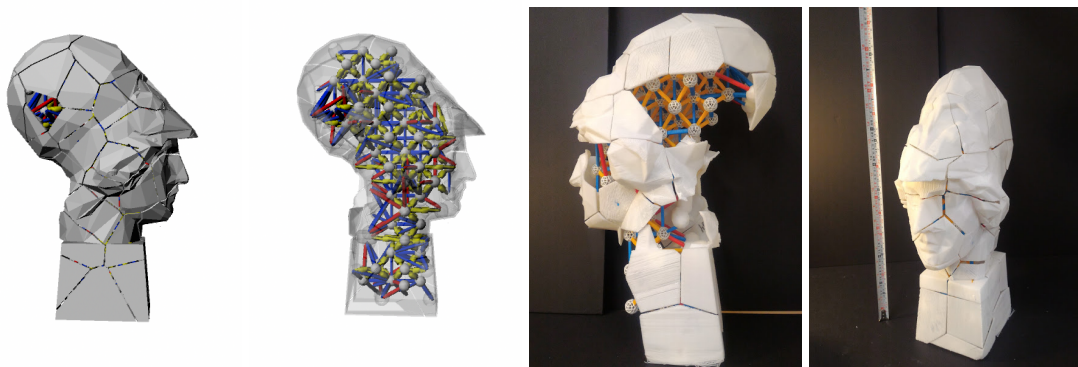


Figure 5.2: Ateneav example. (a) Segmentation (b) Structure (c) Assembly process (d) Result



solid segments with glue, shell segments with glue, shell segments with connectors to compare with our results(see Figure 6.2.1).

We print the solid model very loosely. Solid model can withstand very large weight is very large, but the print time is relatively long. When we need to increase model size, print time would grow in cubic speed, while shell models grow in square speed. The shell parts with glue can be printed and assembled very fast, but the weight they can withstand is small. We predict that when printing a large-scale model, the model will crushed by its own weight. Shell parts with connectors have better supporting force than with glue. Because Connectors will be simultaneously subjected to the normal force, shear and moment, when the size is larger, the model would deform much more. Finally it would be crushed by its weight when the size is very large. In our approach, every segment's recesses have to be facing upward, resulting in each segment would generate a lot of support material. That's the reason why it takes the most time to print in our approach.

We use buckets filled with water put on the model to test the weight models could

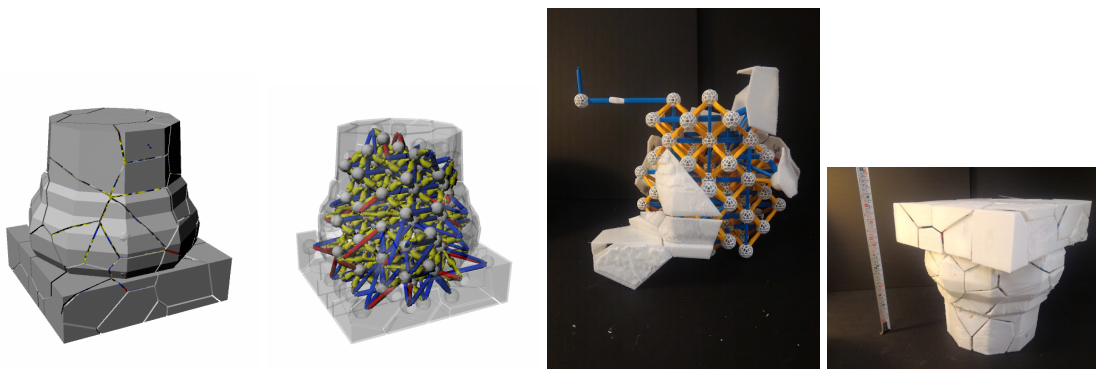
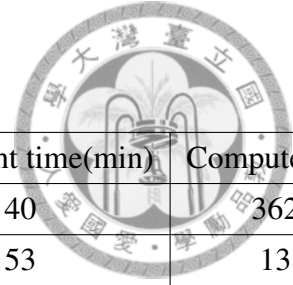


Figure 5.3: Column example. (a) Segmentation (b) Structure (c) Assembly process (d) Result



model	No. of parts	Total print time(hr::min)	Avg. print time(min)	Compute time(sec)
Kitten	159	100:32	40	362+173
Ateneav	61	53:54	53	138+90
Column	61	48:13	47	134+82
Trimmed Ball	33	21:43	31	59+40

model	ball	BS	BM	RS	RM	YS	YM
Kitten	216	320	78	80	21	641	121
Ateneav	77	105	32	34	16	189	34
Column	99	135	35	34	12	280	35
Trimmed Ball	29	52	19	19	7	82	15

model	Zometool assemble time	Segment assemble time	Size(cm)
Kitten	4hr	7hr	36.2*55.2*32.9
Ateneav	2hr	1hr	31*45.2*17.4
Column	1hr	1hr	24.4*24.7*23.3
Trimmed Ball	several minutes	several minutes	16.1*19.9*19.9

Table 5.1: Printed results. In the second part, the first columns mean the components type. B:blue, R:red, Y:yellow, S:small, M:median, L:large

bear. In the first(our method) and second(solid parts with glue) cases, the shape of the models remained the same throughout the whole tests(as in Figure 5.5). In the shell parts with glue case, when the model afforded the weight, the seams would get bigger slowly after the water filled in the bucket(as in Figure 5.6). when the weight came to 4.5kg, the model collapsed slowly until it can't balance the bucket(as in Figure 5.7). On the other

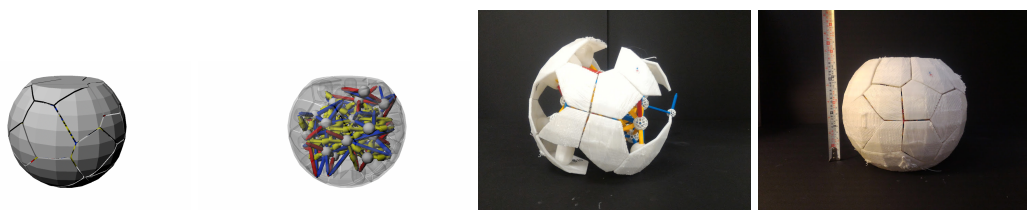
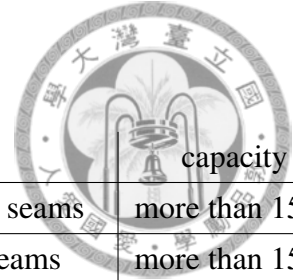


Figure 5.4: Trimmed ball example. (a) Segmentation (b) Structure (c) Assembly process (d) Result



method	print time	aspect	capacity	Assemb
our method	21:43hr	correct shape, some big seams	more than 15kg	several r
solid parts with glue	16:28hr	correct shape, small seams	more than 15kg	; several
Shell parts with glue	8:59hr	deformed shape, some big seams	4.5kg	several r
Shell parts with connectors	14:40hr	deformed shape, small seams	6.6kg	within a

Table 5.2: Model data and processing results (part 1).

hand, in the shell parts with connectors case, after we fill water into the bucket, the model deformed immediately and kept the shape still(as in Figure 5.8). When the model was about to crush, at the moment we filled water into the bucket, the segments exploded right away(as in Figure ??).

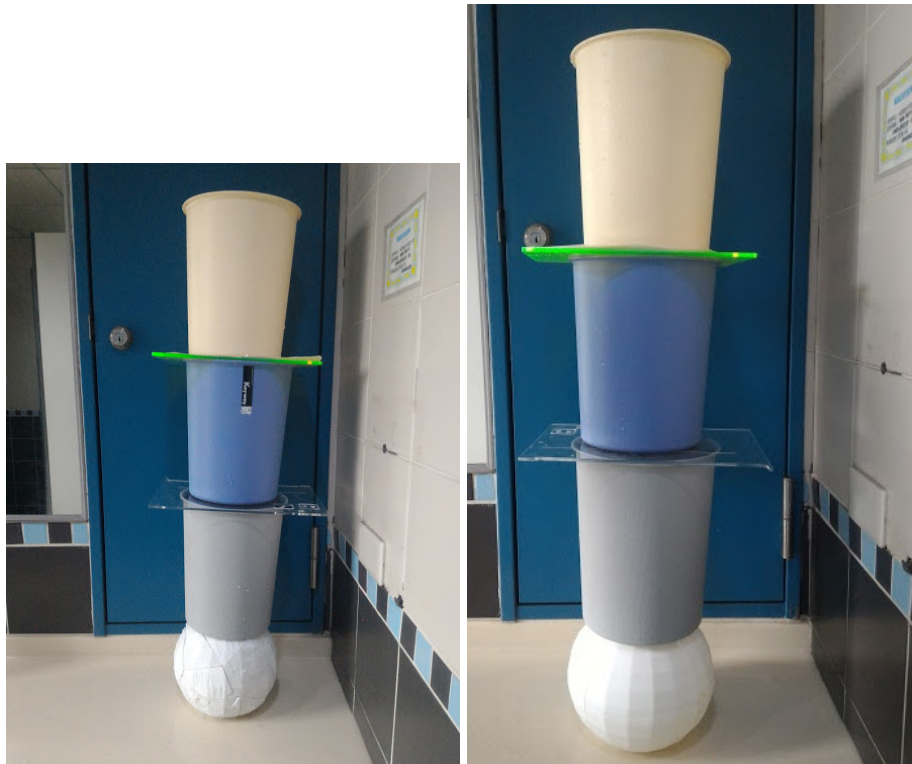


Figure 5.5: The shape of the models remained the same throughout the whole tests. (a) Our method. (b) Solid parts with glue.



Figure 5.6: in shell parts with glue case, the seams would get bigger slowly after the water filled in the bucket.

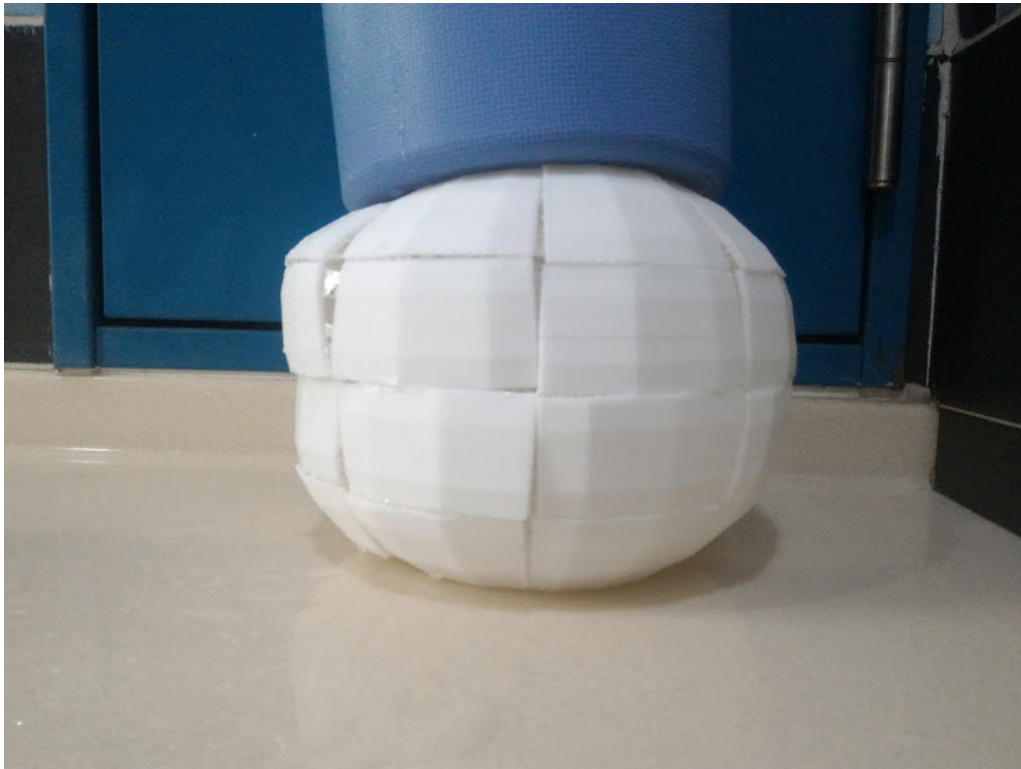


Figure 5.7: Shell parts with glue model collapsed slowly.





Figure 5.8: In the shell parts with connectors case, after we filled water into the bucket, the model deformed immediately and kept the shape still.



## Chapter 6

# conclusion, limitation and future work

### 6.1 Conclusion

We provide a approach to print and assemble large size 3D models effeciently with robustness. We split the large model and use Zometool as the interior structure, the tetrahedron-based structure is so robust that can afford big weight. Although Zometool can withstand large pressure, the tension it can withstand is relatively small. we use the virtual displacement method for structural analysis, and tie wire on the parts which bear a big tension to kepp structural soundness. Our experiments show that our method of implementation of the model can withstand greater weight and avoid bending moments and shear forces on the structure, to achieve maximum support.

### 6.2 Limitation

Our approach can only be applied to a narrow group of object. Only when the model is big enough without prevalent thin parts, we could put Zometool interior structure into the model. If the model is composed of two or more big parts connected on thin parts, we can



not connect the Zometool structure together. In order to maintain the shape of the recesses on the connector, the FDM technology would produce much support which is difficult to remove and wastes a lot of material. Our approach requires a lot of manual assembly, while users prefer higher cost but relatively less manual assembly. Zometool assembly is complex. We need a simple and clear guide unless the user would feel difficult and time consuming when assembling the model.

### **6.3 Future Work**

if models have prevalent thin parts, the structure composed of both Zometool and 3Dprinting could resolve this obstacle. the problem is that we have to keep every recess on a segment face the same or similar direction, which will be the future research direction. Through the results of structural analysis, we should increase the structural robustness at the bottom of the model which make the segments uneasy to loose. Our segmentation is not taking into account visibility of the seams, and for better visual quality we could make cuts along natural seams on the model or use other advanced approaches for higher segmentation quality. Too many small segments are likely to cause the complexity of combinations, so we can combine small segments to make assembly simple and reduce the seams.



## Bibliography

- [1] 3dsystems. <http://www.3dsystems.com>. Accessed: 2016-06-01.
- [2] Tutorial: How to hollow objects for 3d printing. <http://www.shapeways.com/tutorials/creating-hollow-objects>. Accessed: 2010-06-01.
- [3] B. Bickel, M. Bächer, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, and W. Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Trans. Graph.*, 29(4):63:1–63:10, July 2010.
- [4] D. Chen, D. I. W. Levin, P. Didyk, P. Sitthi-Amorn, and W. Matusik. Spec2fab: A reducer-tuner model for translating specifications to 3d prints. *ACM Trans. Graph.*, 32(4):135:1–135:10, July 2013.
- [5] M. Hašan, M. Fuchs, W. Matusik, H. Pfister, and S. Rusinkiewicz. Physical reproduction of materials with specified subsurface scattering. *ACM Trans. Graph.*, 29(4):61:1–61:10, July 2010.
- [6] L. Luo, I. Baran, S. Rusinkiewicz, and W. Matusik. Chopper: Partitioning models into 3d-printable parts. *ACM Trans. Graph.*, 31(6):129:1–129:9, Nov. 2012.
- [7] S. Mueller, S. Im, S. Gurevich, A. Teibrich, L. Pfisterer, F. Guimbretière, and P. Baudisch. Wireprint: 3d printed previews for fast prototyping. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 273–280, New York, NY, USA, 2014. ACM.
- [8] R. Prévost, E. Whiting, S. Lefebvre, and O. Sorkine-Hornung. Make it stand: Balancing shapes for 3d fabrication. *ACM Trans. Graph.*, 32(4):81:1–81:10, July 2013.
- [9] O. Stava, J. Vanek, B. Benes, N. Carr, and R. Měch. Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph.*, 31(4):48:1–48:11, July 2012.



- [10] J. Vanek, J. A. G. Galicia, B. Benes, R. Mech, N. Carr, O. Stava, and G. S. Miller. Pack-Merger: A 3D Print Volume Optimizer. *Computer Graphics Forum*, 2012.
- [11] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, and X. Liu. Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph.*, 32(6):177:1–177:10, Nov. 2013.
- [12] Q. Zhou, J. Panetta, and D. Zorin. Worst-case structural analysis. *ACM Trans. Graph.*, 32(4):137:1–137:12, July 2013.