

PAIRWISE COMPARISONS AND VISUAL PERCEPTIONS OF 3D  
SHAPE VOLUME ESTIMATION

by

Wenjun Wan

A thesis submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science (MSc) in Computational Sciences

The Office of Graduate Studies  
Laurentian University  
Sudbury, Ontario, Canada

©Wenjun Wan 2022



## **Abstract**

Using pairwise comparisons for estimations increases accuracy. At present, scholars use the pairwise comparisons method to make subjective comparison between one-dimensional image and two-dimensional images. This research is about the subjective comparison of three-dimensional images. We first sets a fixed object volume and then uses the random method to generate multiple three-dimensional objects with different shapes and then scale them to our designed volume values. This study also virtualizes and binarizes the image and prints the actual object in the way of 3D printing for respondents to observe. Thirty-two respondents used the direct and pairwise comparisons methods to rate the volume of five randomly generated 3D shapes. It is found that using the direct method, the observer's estimation errors is higher (in average) than when the paired comparisons method is used. The pairwise comparisons method can improve the accuracy of estimating the volume of random objects.

## **Keywords**

Pairwise comparisons method, Direct method, Randomly generated 3D shapes, Python, 3D print

## Acknowledgment

As time flies, the two-year master's study is coming to an end. Looking back on the two years of study, I have benefited greatly. Every bit of progress made during my studies could not have been possible without the help and encouragement of Prof. Koczkodaj, as well as the support of family and friends, and I would like to express my sincere thanks to you all.

This thesis was completed under the guidance of Professor Koczkodaj; from the topic selection of the thesis to the determination of the experimental plan until the writing and finalization of the thesis, Prof. Koczkodaj has always given me patient instruction and advice. You insisted on reviewing my paper even when you were sick. I would like to express my sincere gratitude to you.

I would like to extend my sincere thanks to the review committee member Dr. Xixiong Guo for his time and for reviewing and advising my thesis.

Finally, I would also like to thank my family and my girlfriend Fan Zhang; because of their support and encouragement, my two years of study have been very smooth.

Wenjun Wan

2022

# Contents

Abstract . . . . .	ii
Keywords . . . . .	ii
Acknowledgment . . . . .	iii
<b>1 Introduction</b>	<b>1</b>
1.1 Multi-Criteria Descision Making . . . . .	1
<b>2 Pairwise Comparisons</b>	<b>4</b>
2.1 Background . . . . .	4
2.2 Pairwise Comparisons Matrix . . . . .	6
2.2.1 Constructing PC Matrix . . . . .	7
2.2.2 Consistency and Inconsistency of PC Matrix . . . . .	8
2.3 Pairwise Comparisons Method vs Directly Method . . . . .	12
2.3.1 One-dimension . . . . .	13
2.3.2 Two-dimension . . . . .	15
2.3.3 Three-dimension . . . . .	17
<b>3 3D Shapes Design and Generation</b>	<b>20</b>
3.1 Rationale . . . . .	20
3.2 Gaussian Blur . . . . .	24
3.2.1 Introduction of Gauss Blur . . . . .	24

3.2.2	Importance of Gauss Blur . . . . .	26
3.2.3	Fuzzy Algorithm based on Gaussian Filter . . . . .	27
3.3	Smoothing . . . . .	29
3.3.1	Threshold Method . . . . .	30
3.4	Abaqus . . . . .	35
<b>4</b>	<b>Python Implementation</b>	<b>38</b>
<b>5</b>	<b>Experiment</b>	<b>46</b>
5.1	Preparation Phase . . . . .	47
5.2	Experiment Phase . . . . .	51
5.3	Analysis Phase . . . . .	53
<b>6</b>	<b>Conclusion and Future Work</b>	<b>56</b>
6.1	Conclusion . . . . .	56
6.2	Future Work . . . . .	57
	<b>References</b>	<b>59</b>
	Appendix . . . . .	64
A1.	Generate 3D Shapes in Python . . . . .	64
A2.	PC Matrix . . . . .	71
A3.	Abaqus Source Code . . . . .	72
A4.	Data . . . . .	74

# List of Tables

2.1	The Scale and Its Deception — (R. W. Saaty, 1987) . . . . .	7
5.1	Direct Method . . . . .	54
5.2	Pairwise Comparisons Method . . . . .	55

# List of Figures

2.1	PC Matrix Example . . . . .	10
2.2	Inconsistency Example . . . . .	11
2.3	2D Experiment Result — (Adamic et al., 2009) . . . . .	17
3.1	Randomly Generated 3D Shape . . . . .	22
3.2	Flow Chart: Generate 3D shapes . . . . .	23
3.3	Normal Distribution — (Distribution, n.d.) . . . . .	28
3.4	3D Shape with Gaussian Blur . . . . .	29
3.5	3D Shape with Smoothing . . . . .	35
3.6	3D Shape in Abaqus CAE . . . . .	37
5.1	Unit Cube: $64cm^3$ . . . . .	48
5.2	Shape A: $58cm^3$ . . . . .	49
5.3	Shape B: $72cm^3$ . . . . .	49
5.4	Shape C: $90cm^3$ . . . . .	50
5.5	Shape D: $113cm^3$ . . . . .	50
5.6	Shape E: $142cm^3$ . . . . .	51
5.7	Google Form: Volume Estimation . . . . .	53
1	Direct Method . . . . .	74
2	Pairwise Comparisons Method . . . . .	75



3	Direct Method . . . . .	76
4	Pairwise Comparisons Method . . . . .	77

# Chapter 1

## Introduction

### 1.1 Multi-Criteria Decision Making

Facing the rapidly changing economic situation and increasingly fierce competitive environment, the decision-making problems faced by various enterprises and organizations are becoming more and more complex. Modern decision-making is not only based on a single criterion but also becomes more complex. Many interrelated and mutually restrictive factors (criteria) have to be considered. This kind of decision-making problem is called multiple criteria decision making (MCDM). The decision-making faced by decision-makers not only needs to comprehensively consider the influence of multiple interrelated and mutually restrictive factors, but also mixed with the subjective preferences of people participating in decision-making, which mainly come from the decision-maker's life background, social values and thinking mode (Zavadskas & Turskis, 2011). In other words, due to the problem of using this method for decision-making and judgment, the problem itself is a fuzzy concept. The information of decision-making may be incomplete, and the decision-maker can not make a complete and

rational judgment and decision-making under the condition of uncertainty (Carlsson & Fullér, 1996). As an essential branch of uncertain multi-criteria decision-making, stochastic multi-criteria decision-making refers to the various natural States faced by decision-makers. It is a kind of uncertain decision-making problem that occurs randomly. It is mainly reflected in the uncertainty or incompleteness of parameter information, such as criteria value is a random variable, criteria weight or the preference of the decision-making subject is uncertain or incomplete. It has a wide application prospect in many fields such as economy, management, engineering design and military.

It is a research problem with important practical background. In practical problems, it is difficult for a single decision-maker to make correct decisions due to the limitations of dry knowledge and cognition. In order to avoid the loss caused by personal decision-making mistakes, enterprises and organizations often organize multiple experts to participate in the decision-making process (Molotsky, 2011). The MCS problem in the group decision-making environment is much more complex. Firstly, due to different professional backgrounds and perspectives of thinking, there are often significant differences or even conflicts between the information provided by decision-makers. It is necessary to gather the preference information of decision-makers according to the rules and produce consistent group classification results. Secondly, The social status, rights and responsibilities of decision-makers are different. In the decision-making process, the impact of different decision-makers on the intermediary results should be considered to reflect its importance.

At present, there are many methods to solve some real-life MCDM problems, such as technique for order of preference by similarity to ideal solution (TOPSIS), analytic hierarchy process (AHP), VIKOR, etc. However, these methods have their own strengths and weaknesses in solving different MCDM problems, and no one method can perfectly solve all MCDM problems (Percoco et al., 2020). Therefore, it is necessary to conduct in-depth research on different methods.

Our research focuses on pairwise comparisons method, which is one of the techniques derived from the AHP method. In this thesis, 3D shapes are randomly generated, and the pairwise comparisons method is used to allow the experimenter to make a subjective volume estimation. Combined with pairwise comparisons, this thesis holds that any different objective stimulus will lead to different subjective processes. With the existence of individual differences (such as clarity, surrounding environment, psychological state, etc.), even the same stimulus will lead to different subjective processes in different people.

# Chapter 2

## Pairwise Comparisons

### 2.1 Background

By reviewing the whole of mankind's history, it can be discovered that people frequently presented results by using comparisons; for instance, the results of sports games are commonly expressed as Player/Team A defeated Player/Team B. When people try to make a decision, they tend to use comparisons to list all the pros and cons, and decide to go forward for the better one. An interesting fact is that the world composites countless pairwise comparisons, as thinking that choosing one of the two movies can be considered one pair, and a bigger collection of movies can be considered multiple pairwise comparisons.

In the thirteenth century, scholar Ramon Llull, who proposed a binary electoral system, first invented the use of the pairwise comparisons for the decision procedure as a formal basis. Afterwards, the method was rediscovered by Condorcet and kept the method as binary, which means the result can be in a state of one of two alternatives (Ramík, 2020). Later,

the pairwise comparisons was improved by Thurstone in a more generalized, and quantitative way (Ramík, 2020). “Since then, the result of the single pairwise comparisons can be identified with a real positive number where the values greater than 1 mean the degree to which the first alternative won, and likewise, values smaller than 1 mean the degree to which the second alternative won” (Ramík, 2020). Pairwise comparisons refer to any process in that two objects in pairs are compared, and judge which of these two is regarded preferred, or has a more significant amount in terms of quantitative nature, or if these two can be found identical or indifferent (Ramík, 2020). “The method of pairwise comparisons is used in the scientific study of preferences, attitudes, voting systems, social choice, public choice, requirements engineering, and multi-agent artificial intelligence (AI) systems” (Ramík, 2020).

Based on the pairwise comparisons, decision-making problems can be resolved by establishing the procedures to combine opinions about alternatives connected to different points of view, using the sense of pairwise comparisons that take the preference of one option over another into account (Ramík, 2020). Also, the selection of the other points of view can effectively eliminate the potential risks of bias caused by using a single criterion or point of view from one person (Ramík, 2020). This mindset is applied to decision-making in various fields of evaluation, selection and prioritization (Ramík, 2020). “According to the nature of the information expressed by the DM, for every pair of alternatives different representation formats can be used to express preferences, e.g., multiplicative preference relations, additive preference relations, fuzzy preference relations, interval-valued pref-

erence relations, and also linguistic preference relations” (Ramík, 2020).

## 2.2 Pairwise Comparisons Matrix

The method of using pairwise comparison matrix (PC matrix) to determine the preference of decision makers was first proposed by turnstone in literature in 1927 (T. L. Saaty, 1991), but this method did not receive corresponding attention at that time due to the lack of consistency test index (Ramík et al., 2017). Saaty perfected this method of pairwise comparisons in the 1970s, and finally applied this method in analytic hierarchy process. Saaty believed that when the consistency ratio index (CR) of the PC matrix was greater than 0.1 (R. W. Saaty, 1987), it showed that the consistency of the PC matrix needed to be adjusted. Later, Saaty thought that a more strict consistency index critical value should be adopted for the low-order matrix. Some international scholars have proposed several mathematical methods for consistency adjustment of PC matrix since the 1980s (Bozóki & Rapcsák, 2008), and some scholars have used similar ideas to study compatibility correction of PC matrix (Fedrizzi & Giove, 2007) and complementary and language PC matrix consistency correction, and obtained a large number of useful conclusions, the consistency adjustment of PC matrix must be based on the rationality of the consistency test. However, it has been pointed out that the critical value setting of proportional index (CR) in consistency test lacks theoretical basis (Koczkodaj, 1993). Therefore, before we delve into the consistency adjustment of PC matrix, we need to understand the logic and basic concepts of constructing a PC matrix.

### 2.2.1 Constructing PC Matrix

When constructing the PC matrix, it is necessary to ask an experienced, authoritative person to judge the relative importance of each element by pairwise comparison. Then we can get the judgment result according to Saaty's basic measurement table of absolute numbers and obtain the PC matrix (R. W. Saaty, 1987).

Table 2.1: The Scale and Its Deception — (R. W. Saaty, 1987)

Intensity of importance	Definition	Explanation
1*	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgment slightly favor one activity over another
5	Essential or strong importance	Experience and judgment slightly favor one activity over another
7	Demonstrated importance	An activity is strongly favored and its dominance is demonstrated in practice
9	Absolute importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2, 4, 6, 8	Intermediate values between the two adjacent judgments	When compromise is needed
Reciprocals of above nonzero	If activity $i$ has one of the above nonzero numbers assigned to it when compared with activity $j$ , then $j$ has the reciprocal value when compared with $i$	
Rationals	Ratios arising from the scale	IF consistency were to be forced by obtaining $n$ numerical values to span the matrix

\* On occasion in 2 by 2 problems, we have used  $1 + \epsilon$ ,  $0 < \epsilon \leq \frac{1}{2}$  to indicate very slight dominance between two nearly equal activities



Assuming that a total of  $n$  elements need to be compared, use  $m_{ij}$  (for every  $m_{ij} > 0$ ) to represent the relative preference between two elements (intensity with numbers), and finally construct the general formula of PC matrix (Koczkodaj, Kułakowski, & Ligęza, 2014).

$$M = \begin{bmatrix} 1 & m_{12} & \dots & m_{1n} \\ \frac{1}{m_{12}} & 1 & \dots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m_{1n}} & \frac{1}{m_{2n}} & \dots & 1 \end{bmatrix} \quad (2.1)$$

Since the numbers for estimating item weight ratios are judged and determined by decision-makers, it is essential to ensure the reliability of number assignments when creating the PC matrix, as different choices of numbers can lead to significantly different results. However, due to the limitation of knowledge, it is difficult for decision-makers to provide reliable ratios objectively and accurately (Wang, Peng, & Kou, 2021). Therefore, it is necessary to check whether the PC matrix is reliable.

### 2.2.2 Consistency and Inconsistency of PC Matrix

After the construction is completed, it is indispensable to judge the consistency of the PC matrix, because only the consistent PC matrix can provide more reliable results, and the results obtained by the inconsistent PC matrix will be questioned (Kułakowski, Mazurek, & Strada, 2021). Many scholars have conducted extensive research on PC matrix consistency in recent decades. There are many ideas for the description and calculation

method of PC matrix consistency, of which two are the most important (Kou, Ergu, Lin, & Chen, 2016). One is the eigenvalue-based description proposed by Saaty, and the other is the maximum error-based description proposed by Prof. Koczkodaj. The former describes its inconsistency from the perspective of the PC matrix as a whole, and the latter describes its inconsistency from the perspective of the error degree of each element in the PC matrix.

Koczkodaj introduced a concept, triad, when judging the consistency of the PC matrix. In the upper-right part of the matrix, any  $m_{ij}$ ,  $m_{ik}$ ,  $m_{jk}$  (for every  $i, j, k = 1, 2, \dots, n$ ) can form a triad of the PC matrix. When all triads in the PC matrix satisfy  $m_{ij} \times m_{jk} = m_{ik}$  (for every  $i, j, k = 1, 2, \dots, n$ ), the PC matrix is consistent (Koczkodaj et al., 2014). If this equation does not hold, that is, the condition  $m_{ij} \times m_{jk} = m_{ik}$  (for every  $i, j, k = 1, 2, \dots, n$ ) is not satisfied, then the PC matrix is inconsistent. The following table shows an example of consistent PC matrix.

According to the table, this is a  $3 \times 3$  matrix, suppose  $i = 1, j = 2, k = 3$ , then  $m_{ij} = m_{12} = 2$ ,  $m_{jk} = m_{23} = 3$ ,  $m_{ik} = m_{13} = 6$ . We can easily find that  $m_{ij} \times m_{jk} = m_{ik}$ , which is  $2 \times 3 = 6$ ; furthermore, this PC matrix only has one triad of  $(m_{ij}, m_{ik}, m_{jk})$ . Therefore, it can be determined that this PC matrix is consistent.

However, in most cases, the PC matrix is inconsistent. Consider another case, as shown in the following image (note: the symbol "/" does not mean division here, but means that two objects are compared). Here, we need to

	A	B	C
A	1	2	6
B	1/2	1	3
C	1/6	1/3	1

Figure 2.1: PC Matrix Example

compare polyhedra A, B, C (Inconsistency only occurs when we compare with at least 3 objects). Assuming that A has 2 times the volume of B, and B has 3 times the volume of C, then obviously, A should be 6 times the volume of C. However, here it is stated that A is 5 times the volume of C instead of 6 times, then the case of inconsistency arises.

At this point we are not sure which input is incorrect, 2 maybe correct or wrong (it can be 1.5), 3 also could be wrong, and it can be 2.5. In order to avoid this confusion, and to make the results more reliable (Kułakowski et al., 2021), it is necessary to study how to indicate and reduce inconsistency. There are many studies have been done in this field, and one of the most famous methods for judging inconsistency is the "Koczkodaj inconsistency indicator" (Kii), which is proposed and named by Koczkodaj (Koczkodaj, n.d.). The Kii general formula of a triad  $(m_{ij}, m_{ik}, m_{jk})$  is:

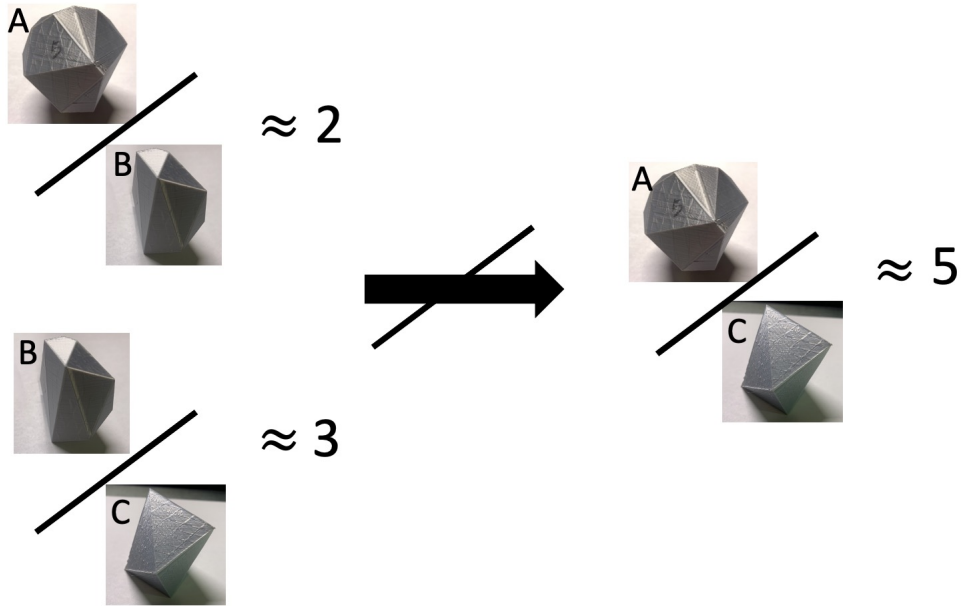


Figure 2.2: Inconsistency Example

$$Kii(m_{ij}, m_{jk}, m_{ik}) = \min(|1 - \frac{m_{ik}}{m_{ij} \cdot m_{jk}}|, |1 - \frac{m_{ij} \cdot m_{jk}}{m_{ik}}|) \quad (2.2)$$

Then, Koczkodaj proposed a concept called inconsistency tolerance. As mentioned earlier, PC matrices are inconsistent in most cases in the real world, and it is impossible to improve these PC matrices from inconsistency to consistency. Therefore, we need to set an inconsistency tolerance to help us decide what inconsistent PC matrix is acceptable. Usually, this value is set to  $\frac{1}{3}$ , or some other value in specific cases (Kakiashvili, Koczkodaj, Montgomery, Passi, & Tadeusiewicz, 2008).

After having the inconsistency tolerance, we first need to calculate the Kii of all triads of the PC matrix. When the maximum Kii value is less than

the inconsistency tolerance, we can consider the inconsistency of this PC matrix to be acceptable; otherwise, this PC matrix needs to be improved. The core of the improvement is to reduce the inconsistency. Koczkodaj proposed a distance-based inconsistency improvement method, assuming that the existing triad is  $(m_{ij}, m_{ik}, m_{jk})$ , and  $m_{ij} \times m_{jk} \neq m_{ik}$ , then we replace  $m_{ij}, m_{ik}, m_{jk}$  with  $\tilde{m}_{ij}, \tilde{m}_{ik}, \tilde{m}_{jk}$ , we can get the reduction inconsistency.  $\tilde{m}_{ij}, \tilde{m}_{ik}, \tilde{m}_{jk}$  can be generated with the following formula (Koczkodaj et al., 2014):

$$\begin{aligned}\tilde{m}_{ij} &= \tilde{m}_{ij}^{\frac{2}{3}} \cdot \tilde{m}_{jk}^{\frac{-1}{3}} \cdot \tilde{m}_{ik}^{\frac{1}{3}} \\ \tilde{m}_{ik} &= \tilde{m}_{ij}^{\frac{1}{3}} \cdot \tilde{m}_{jk}^{\frac{1}{3}} \cdot \tilde{m}_{ik}^{\frac{2}{3}} \\ \tilde{m}_{jk} &= \tilde{m}_{ij}^{\frac{-1}{3}} \cdot \tilde{m}_{jk}^{\frac{2}{3}} \cdot \tilde{m}_{ik}^{\frac{1}{3}}\end{aligned}\tag{2.3}$$

After replacing the three elements in the triad with the largest  $Kii$  value with  $\tilde{m}_{ij}, \tilde{m}_{ik}, \tilde{m}_{jk}$ , recalculate the  $Kii$  value of all triads. If the largest  $Kii$  value is still greater than the inconsistency tolerance, continue this process. Until the largest  $Kii$  value is less than the inconsistency tolerance, we can consider the PC matrix to be consistent with an acceptable inconsistency threshold (Koczkodaj, Kosiek, Szybowski, & Xu, 2015).

## 2.3 Pairwise Comparisons Method vs Directly Method

Pairwise comparisons have been proved to be a superior method capable of yielding more precision in the estimation of areas and lengths, compared to the direct method, which purely relies on visual perceptions. The demonstration has been accomplished by the experiments conducted by

Koczkodaj based on both levels of one-dimension and two-dimension comparisons (Koczkodaj, 1996). Koczkodaj's experiments first introduced the use of computer technologies, which managed to address the defects in past experiments (Koczkodaj, 1996). Instead of using the traditional method as hard copy questionnaires, Koczkodaj implemented a computer program in C with the proper graphical interface on a portable computer, which can lift the restrictions of the number of respondents and the data was easy to collect as it has defaulted to the data bank (Koczkodaj, 1996). In the remainder of this chapter, we will detail the pairwise comparisons method for experiments in one-dimension and two-dimension, as well as our study in the 3-dimension case.

### **2.3.1 One-dimension**

The one-dimension experiment was designed based on avoiding bias toward pairwise comparisons and direct rating and took the human's optical illusion into account. Hence, the experiment decided to generate random and straightforward shapes. "The collected and analyzed sample size was as follows, 92 examples for  $n = 3$ , 90 for  $n = 4$ , 88 for  $n = 5$ , and 84 for  $n = 6$ , (where  $n$  is the number of bars)" (Koczkodaj, 1996). The program generated each combination of three to six bars of random lengths and stored them in internal tables (Koczkodaj, 1996). It excluded the combinations of bars with ratios of lengths greater than five since they are too small to display on the screen and lack practical meaning (Koczkodaj, 1996). Firstly, during the test, respondents were asked to estimate the lengths of the randomly generated combinations using eyes, which is the direct method. Beginning with three bars,  $n$  varied from three to six, and

all  $n$  bars are displayed on the screen for the respondent to estimate the lengths of each one by answering the question, “What percent (out of 100%) would you assign to each bar?” (Koczkodaj, 1996). Afterwards, the same bars were presented again but only showing in one pair at one time for the respondent to answer, “How many times is bar A longer than bar B?” (Koczkodaj, 1996). This process was repeated until all  $\frac{n(n-1)}{2}$  combinations of pairs had been displayed (Koczkodaj, 1996). The program can store all the replies. It built a pairwise comparisons matrix and was able to calculate the lengths of the  $n$  bars as the eigenvector corresponding to the highest eigenvalue of the matrix following Saaty’s algorithm (Koczkodaj, 1996). The system stored the objective measurements for each bar for calculating the error for both pairwise and direct methods. The results are shown in the following figure, displaying the comparison of the two methods. The histograms in each group ( $n = 3, 4, 5, \text{ and } 6$ ) show a similar shape (Koczkodaj, 1996). However, the average error for the direct method is greater than the average error for using pairwise comparisons.

And to summarize the result, by comparing the population’s mean value as it describes the population’s center, the pairwise method was around 4% versus 12% for the direct rating method (Koczkodaj, 1996). The variance for the pairwise comparisons method was about 3% versus 9% for the direct approach (Koczkodaj, 1996). That also proved that the pairwise comparisons method was more accurate and generated more reliable results. Plus, by reviewing the standard deviation, the pairwise comparisons showed a substantially smaller value, which enhanced the credibility of the results (Koczkodaj, 1996). From an analytical view, the pairwise compar-

isons method has proved to be superior to the direct rating method on a one-dimension level.

### 2.3.2 Two-dimension

The same theory applied to a two-dimension level that using pairwise comparisons can yield more precision in the estimation of the areas has been demonstrated in Koczkodaj's experiment. The experiment was designed by computer technologies to compare the accuracy of area assessments by using the direct method and pairwise comparisons method. In total, 179 university students participated the experiment at an average age of 20 (Adamic et al., 2009). The design of the experiments was divided into two parts. In the first part, the respondents were asked to estimate the five given shapes (with the same area size, but students were not informed) shown in one spare unit area (Adamic et al., 2009). Each figure was given 10 to 15 seconds to estimate (Adamic et al., 2009). In the second part, respondents need to estimate the shapes shown in a pair side by side (Adamic et al., 2009). Totally, ten comparisons are displayed as no more distinct pairs can be generated (Adamic et al., 2009). Respondents can estimate it by selecting one of the options of L/R/E (Left/Right/Equal) to answer the bigger size they think, and they need to respond by what factor the larger one was bigger than the smaller one (Adamic et al., 2009). For example, 1.4 means the larger one had a 40% larger area than the smaller one. For the same reason as mentioned in the one-dimension experiment, considering that the human eyes are easier to estimate areas for simple and common shapes, as people are more used to these shapes, to avoid any bias in both methods, the selection of the shapes should not be too tricky to estimate,



such as a sun with many rays (Adamic et al., 2009). Still, it should not be overly easy, like a circle or square, to estimate as well. Therefore, for generating the shapes, the experiment used a convex-bottom method to use polygonizations of point sets in the plane and randomly generate shapes by changing the number of vertices, and the shapes were scrutinized by a scatter diagram with the data points connected by lines afterwards (Adamic et al., 2009). All the random shapes were decided to be random polygons with a chosen number of vertices between three to eight (Adamic et al., 2009). The generation of random numbers, including the coordinates, was finished using Microsoft Excel and a JAVA polygon-drawing program (Adamic et al., 2009). As a result, each randomly generated polygon consists of a combination of triangles and sometimes quadrilaterals (Adamic et al., 2009). The GNU Image Manipulation Program further processed the drawing of shapes, which can resize the shapes to around 60,000 pixels (Adamic et al., 2009). In the end, the five shapes with equal areas can be displayed in the same plane without rescaling. By implementing Satty's eigenvector solution, the matrix of pairwise comparisons, which represents the ratio estimations between pairs of shapes can be achieved (Adamic et al., 2009). However, due to the inconsistency of estimation in the real life, the results generated by the eigenvector solution can only approach the facts unlimitedly as an approximation. The results are shown in the following image (Adamic et al., 2009).

TABLE 1  
STATISTICAL RESULTS FROM EXPERIMENT: PERCENT

Estimation Method	Shape 1	Shape 2	Shape 3	Shape 4	Shape 5	All
Direct						
Error	25.12	26.05	29.19	25.38	23.01	25.75
<i>M</i>	9.89	10.12	8.51	9.75	9.61	9.58
<i>Mdn</i>	10.00	10.00	8.00	9.50	9.00	9.30
<i>SD</i>	3.21	3.30	3.23	3.06	2.92	3.14
Pairwise						
Error	6.43	5.23	5.17	6.03	4.67	5.51
<i>M</i>	9.93	10.07	9.94	10.19	9.88	10.00
<i>Mdn</i>	9.99	10.00	10.00	10.00	9.98	9.99
<i>SD</i>	1.02	0.84	0.79	1.27	0.88	0.96

*Note.*—Estimates of areas for all figures differed significantly between methods,  $p < .001$ .

Figure 2.3: 2D Experiment Result — (Adamic et al., 2009)

To summarize, the estimation errors for two-dimensional shapes were similar to the results of the one-dimension experiment. Hence, for the two-dimension experiment, the average error was 25.75% using the direct method, but the error was only 5.51% for using the pairwise comparisons (Adamic et al., 2009). And compared to the one-dimension experiment, the overall estimated errors were more significant since, generally speaking, humans are easier to estimate one-dimension objects. Overall, the pairwise comparisons method has been proved to be a more accurate method for the estimation of lengths or areas than the direct rating method.

### 2.3.3 Three-dimension

The study of applying pairwise comparisons to 1D and 2D shapes has been described in the previous article; apparently, pairwise comparisons can improve the estimation accuracy. However, there is no study that can prove that pairwise comparisons can also improve the accuracy in the 3-

dimensional (3D) case. Therefore, our study is to fill this gap, by applying pairwise comparisons to 3D shapes to estimate volume, to prove whether pairwise comparisons can also improve the accuracy. Due to the development of society, the rise of 3D printing technology has made printing 3D shapes inaccessible. No longer possible. Therefore, in order to better conduct experiments and allow the experimenter to observe the 3D shape more clearly, we purchased a 3D printer; the model is *Easythreed K7 Desktop Mini 3D Printer*, which is used to print the generated random 3D shapes for the experimenter to estimate the volume.

This thesis is based on the study of applying pairwise comparisons to random 2D shapes and extending it to 3D space to conduct volume estimation studies on randomly generated 3D shapes, which can be used to calculate materials and other fields. Because 3D objects are entirely random, we cannot judge the quality of our generated objects through data; however, we can look at whether our generated objects are complete and comfortable from the perspective of human vision. On the surface, generating a 3D object is to create several coordinates and then connect them. However, it is more complex than 2D, and many situations need to be considered. These will be described in detail in the algorithm part. There have been some studies in this field, such as randomly generating 2D models, and the results will be used to test the accuracy of pairwise comparisons using 2D random but unequal shapes. Before that, koczodaj's research conducted the world's first statistically correct research from a one-dimensional perspective. In the subsequent study, scholars randomly generate 2D images by determining the equal area and then let humans compare them from

the perspective of vision after Gaussian processing. It is found that many people believe that the area of these images are not equal, which proves the significance of these random objects, that is, through the change of images in the fields such as games or architecture; while keeping the area unchanged, it makes people feel huge.

Our study will come from the perspective of 3D space generating multiple 3D shapes with the different volume and utterly random shape. Then use the 3D printer to print these shapes, and apply the pairwise comparisons method to estimate their volume. Finally, analyzing experimental data infers whether pairwise comparisons method for 3D case can also lead to cognitive conclusions similar to 2D studies.

# Chapter 3

## 3D Shapes Design and Generation

### 3.1 Rationale

There are many studies on generating random polyhedra, but it is not easy to generate a nice shape. First of all, we need to know what kind of shape is a more suitable shape. Obviously, the starfish shape is not a nice shape; although it looks beautiful, it is not helpful for the experimenter to estimate its volume. Similarly, the concave 3D shape also needs to be excluded, and the concave part may cause visual deception, resulting in an increase in the estimation error. Therefore, the 3D shape we generate will be based on the theory of acceptable shapes proposed by Prof. W.W.Koczkodaj (in 2009): an acceptable shape cannot be too complex to be estimated, such as radial shapes, nor too simple, such as squares and circles; a completely random polygon is not desirable, and it requires a more holistic approach to constructing a nice shape (Adamic et al., 2009).

Therefore, the existing algorithms for generating 3D shapes are not suitable for our research, although they are all excellent algorithms, such as the use of point clouds to form 3D shapes proposed by Huang, Peng, Hu, and Du in 2017, this algorithm can restore objects in real life very well, and can clearly present the details of objects (Huang et al., 2017). However, as the theory mentioned above, complex 3D shapes are not desirable; thus, we opted for a very primitive method to generate a normal but very nice shape.

Our 3D shape starts by generating random vertices coordinates. Randomly place vertices in 3D space. However, 3D shapes cannot be placed too random; otherwise, the generated shapes are likely to be very extreme, such as large values in the XY plane and small values in the z-axis. This kind of shape, although random, usually leads to an optical illusion trap, which is not conducive to subject comparison. Thus, we restrict the vertices of the 3D shape on a sphere. Setting limits will result in a loss of randomness, but will ensure that the shapes we generate are not too exaggerated and affect the experimental results. Moreover, only restricting the vertices to be on the sphere is not enough because two vertices may be very close to each other when randomly generating vertices. The 3D shape, in this case, is not nice enough, so it is necessary to judge the distance between each vertex when generating vertices, and discard the vertex that are too close. Since the 3D space has a z-axis dimension compared to the 2D plane, the generated points cannot be simply connected directly. It is likely to be connected into a concave 3D shape, which will greatly increase the difficulty of estimating the volume. A nice shape should allow the experimenter to estimate the volume by observing. Therefore, we also need to make a con-

vexity judgment when connecting each vertex. We refer to a convex hull algorithm proposed in the *Engineering Mechanics*. After connecting any three points to generate a surface, it is determined whether the remaining vertices are on the same side of the surface. If there is at least one point on different sides, the surface does not meet the conditional and should be discard (Qin, Jin-hua, Yi, & Ya-dong, 2013). Thus, connecting all the points in this way ensures that the 3D shape is convex.

The image below shows an example of a randomly generated 3D shape, an initial 3D shape that has not been processed

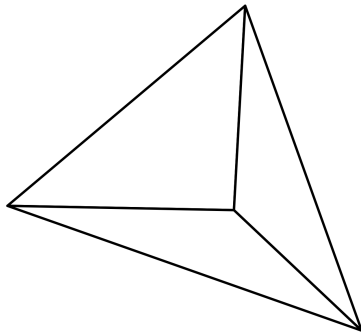


Figure 3.1: Randomly Generated 3D Shape

The flow chart followed briefly illustrated the whole process of generating a 3D shape

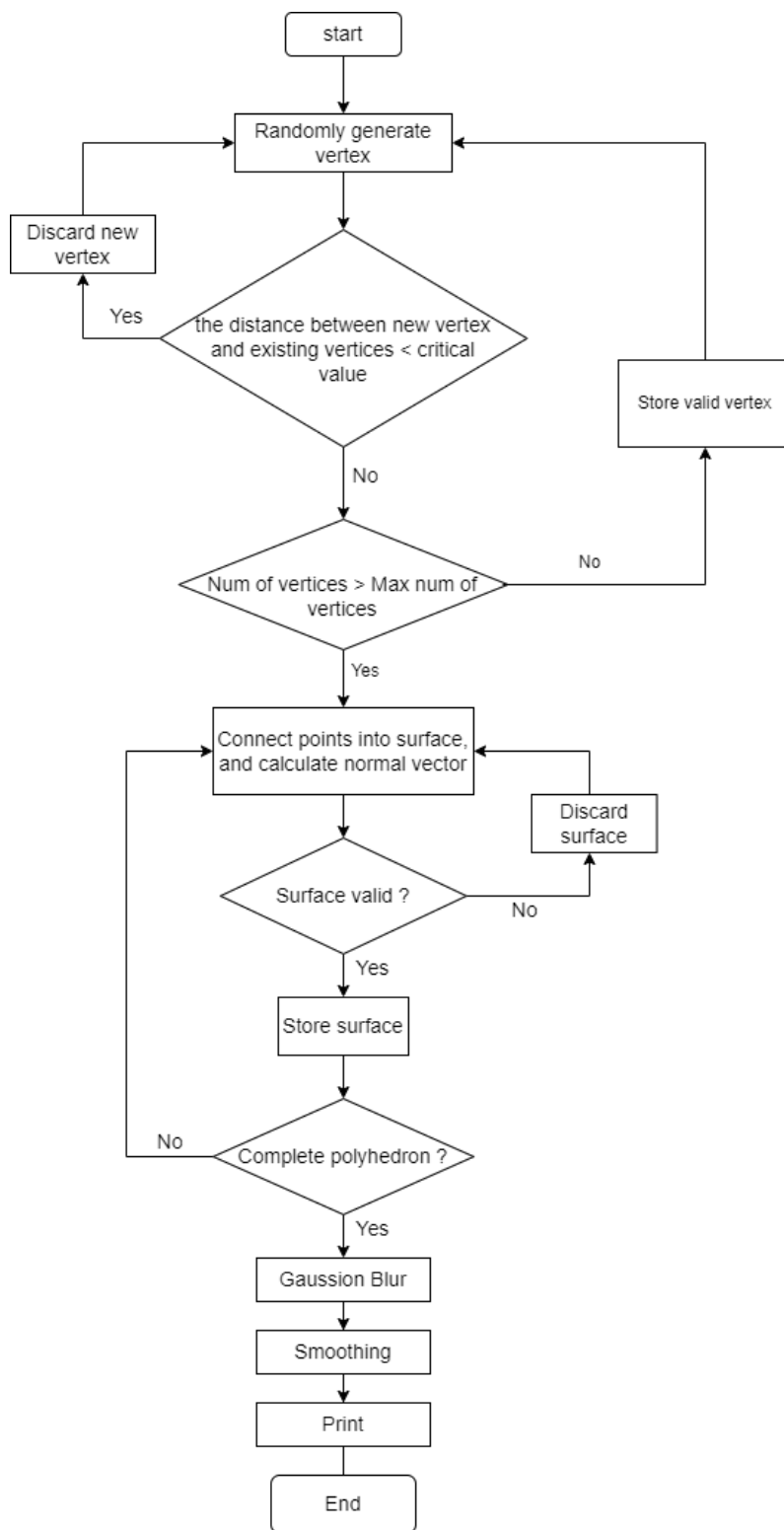


Figure 3.2: Flow Chart: Generate 3D shapes



## 3.2 Gaussian Blur

### 3.2.1 Introduction of Gauss Blur

In image representation, it is a very important process to separate the target from the background. For human visual system, this process is easy to do, but when the computer wants to simulate this process, there will be many problems. Due to the existence of noise and the quantization problem of the original image, in edge detection, it is likely to regard the points that are not edges as edge points. Similarly, it is also possible to lose the existing edge points. The performance of an edge detector depends on its ability to accurately locate the edge. Edge detection is one of the basic subjects of image processing and understanding. Its basic requirements are high detection accuracy, strong noise resistance, no omission of actual edges, no false edges, and high positioning accuracy. In some applications, the accuracy of positioning is even sub-pixel level. However, the positioning accuracy of traditional edge detection methods can only reach pixel level. Therefore, it is necessary to study a good sub-pixel edge location method. However, it is difficult to do this because the actual images contain noise, and the distribution and variance of noise are unknown, and the noise and edge are high-frequency signals. The presence of noise may make the detected edge deviate from its true position. In the image edge detection with noise, it can make the detected edge as close to the real edge as possible, which is an important criterion to evaluate the performance of edge detection operator. Moreover, due to physical and lighting reasons, the edges in the actual image often occur in different scale ranges, and the scale information of each edge pixel is unknown. Edge location is a problem

encountered in the process of edge detection. Another problem that is easy to exist in edge detection is the sharp change in the brightness of the image edge caused by high-frequency components. Therefore, the linear shallow wave and smooth wave used to suppress noise will also blur the degree area. However, smoothing is very necessary, because the process of edge detection depends on the difference of image function, which will amplify the high-frequency components of all components of the signal, including noise. Low pass filter is the most widely used smoothing filter in edge detection. The degree of smoothing is determined by the size or scale of smoothing operator. Generally speaking, for a small scale, the smoothing operator extracts the detailed information of image brightness change, but it is also more sensitive to noise. For a large scale, it can extract the rough information of brightness change, but the detected edge may also have a certain positioning error. Therefore, it is important to choose the optimal scale of the flat edge wavemaker. In noisy images, edge detection needs to smooth and denoise the image first, but when smoothing noise, it is easy to lose the high-frequency information of the image, and the processing effect is not ideal. Theoretically, there is a mutual restriction and mutual variation relationship between detection accuracy and anti noise performance, that is, the principle of uncertainty. Uncertainty principle is an important law in the study of quantum mechanics. From the perspective of information processing, this principle can be expressed as: a signal cannot be highly concentrated in the time domain and frequency domain, so the positioning accuracy and anti noise ability of edge continuation cannot be improved infinitely at the same time. General edge detection methods are difficult to achieve a good compromise between detection accuracy and anti-noise

performance. Gaussian filter is the best filter to meet this condition.

### 3.2.2 Importance of Gauss Blur

In image processing, the most widely used smoothing filter is Gaussian filter, which plays a very important role in edge detection and line detection. Gaussian filter is a kind of linear flat edge full wave filter which selects the weighted value according to the shape of Gaussian function. Gaussian smoothing wave splitter is very effective for removing noise that obeys normal distribution. Babaud proved that when one-dimensional signals are smoothed by Gaussian waveguides, it can be seen from the scale space representation of their second derivative that when the scale of waveguides changes from large to small, the original zero crossings slowly disappear. But there will be no new zero crossing. They also proved that among many types of signals, Gaussian function is the only streamer with this feature. This unique characteristic extends the ability of the Gaussian wave device to locate the zero crossing point when the scale changes, and also has the ability to recover the signal at a sufficiently small scale. Gaussian filter has some important properties, which makes it widely used in image processing. Firstly, the Gaussian filter is a number of single values, which shows that the Gaussian filter uses the weighted average of the pixel neighborhood to replace the pixel value of the point, and the weighted value of the pixel in each neighborhood is easy to monotonically decrease with the distance between the point and the center point. This property is very important because the edge is a local feature of the image. If the smoothing operator still has a great impact on the pixels far from the center of the smoothing operator, the image will be distorted after smoothing.

### 3.2.3 Fuzzy Algorithm based on Gaussian Filter

In product photography, to guide the audience's eyes to a specific part of the image, people can apply Gaussian blur to all other parts of the image to achieve this purpose. People's eyes will naturally move to the clearest place. People can also use this blur effect to hide the features of people, license plates or brand signs you don't have permission to use. In this project, we just don't want the observer's eyes to focus on a specific position. So Gaussian blur is needed. Slight blur can narrow the gap between the midpoints of halftone images, while large blur can turn almost any image into color spots with shape contours. A simpler explanation is that Gaussian blur is a technique for smoothing and desalting parts of an image, usually used to reduce noise or add focus to the foreground of the image. The variation of blur may be very different, but the most common is that people can see the outline of the shape in the blur. This is enough to create visual interest, but not enough to spend time trying to find out what's in the background. This is a good way to deal with this trend. Without enough blur, people will only see an image that is difficult to see in their eyes. Too much blur and color and shape fade away.

From an academic point of view, the so-called "blur" principle can be understood as that each pixel takes the average value of the surrounding pixels. Theoretically, because the image is continuous, the closer the points are, the less the impact of the distant points should be. Therefore, the weighted average is more reasonable. The closer the point is, the greater the weight is, and the farther the point is, the smaller the weight is. According to the normal distribution, it is a bell curve. The closer it is to the center, the

greater the value, and the farther away it is from the center, the smaller the value. When calculating the average value, you only need to take the "center point" as the origin and assign weights to other points according to their positions on the normal curve to obtain a weighted average value. In the image, we can get the weight coefficient of each point around the center point by using two-dimensional Gaussian function.

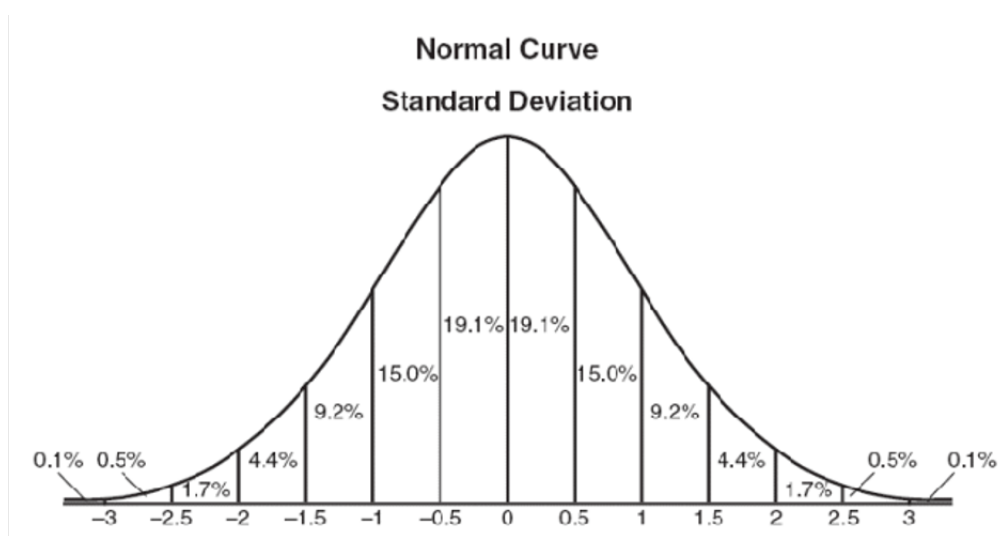


Figure 3.3: Normal Distribution — (Distribution, n.d.)

In general, with Gaussian function, we can calculate the weight of each point. Assuming that the blur radius is 1, construct a  $3 \times 3$  matrix and assume the Gaussian function  $\sigma$  is 1.5, the weight value of each point is calculated according to the coordinate value of  $XY$ , and then the sum of the weight values of all points should be 1, so the above calculated values are normalized. With the normalized weight matrix, it is used as the convolution kernel to convolute with the original image to obtain the blurred value. Gaussian blur is a low-pass filter, which filters out the high-frequency

signal, leaves the low-frequency signal, and removes the boundary of the image content to realize blur.

The image below shows an example of 3D 3D shape with Gaussian Blur



Figure 3.4: 3D Shape with Gaussian Blur

### 3.3 Smoothing

In addition to Gaussian blur, this thesis also binarizes the image. Since the advantages and disadvantages of the image obtained after binarization of the initial input text image will directly impact the later image processing, description and recognition, thus, the research on image binarization is particularly important. The image binarization method needs to be constantly updated and optimized so that the binary image obtained after the binary processing of the original image can still display the original image

information as clearly and completely as possible. One of the most famous images binarization methods is the threshold method (Kavallieratou & Stathis, 2006).

### 3.3.1 Threshold Method

At present, most of the existing image binarization methods are based on the idea of the threshold. First, find a certain value through a certain method. The number of the value is uncertain, which may be one or more. Then classify the gray value of the gray image pixels to be processed according to the found value. Then, the target area and background area are assigned two different values, respectively, to generate the binary image. The above is the general flow of the image binarization method based on a threshold. The value sought at the beginning of this process is called threshold value. Threshold value is the key to this kind of image binarization method. It is an important ruler to separate the target and background in the gray final image. There are two principles for determining threshold value. On one hand, it is to retain the structure information of the original drawing to the greatest extent; on the other hand, it is to avoid wrong segmentation as far as possible; the target area and background area are wrongly divided. Image binarization methods based on threshold can be roughly divided into three categories: global threshold method (Pavlidis, 1993), local adaptive threshold method (Bernsen, 1986), and hybrid threshold method (Kavallieratou & Stamatatos, 2006). Among them, the optimal threshold of the global threshold method is derived from the overall information of the original gray image. In addition, according to the size of the support domain, this method can be subdivided into point-based

global threshold methods and region-based global threshold methods. The famous Otsu method proposed by Japanese scholar Otsu in the 1980s is the point-based global threshold method, also known as Otsu method or maximum inter-class variance method. Its main principles are least-square and discriminant analysis. The threshold of this method is determined according to the maximum variance of the target and background area. However, the application scope of Otsu method is narrow. It is suitable for the situation where the area of the target area and the background area of the image are the same. When there is a great difference between the two areas, the binarization result of Otsu method will not be ideal. For the region-based global threshold method, we have less contact. Here we introduce a gray histogram transformation method (Mason, Lauder, Rutowitz, & Spowart, 1975). Although a one-dimensional gray histogram can well reflect the distribution information of gray image pixels, it ignores any spatial position information of pixels. The two-dimensional gray histogram can reflect the gray distribution information of pixels and increase the average gray information of the neighborhood of pixels, so it reflects the spatial information of pixels to a certain extent.

The gray histogram transformation method is to transform the gray histogram to make the histogram have a deeper trough and sharper peak and then use the familiar bimodal method to determine the best threshold. The transformation of the gray histogram by this method is related to the local characteristics of image pixels, and its purpose is to increase or decrease the gray value of pixels. The precondition of image binarization using the gray histogram transformation method is that the gray image



is only composed of the target area and background area. The gray histograms of pixels in each area are unimodal distribution histograms. The edge operator method is a commonly used gray histogram transformation method. This method generally uses Laplace (Van Vliet, Young, & Beckers, 1989) or Sobel (Sobel, 2014) operators to transform the gray value of image pixels. The advantage of the global threshold method is that the algorithm is simple, and the result is stable. However, it is not suitable to use the global threshold method to binarize all images. The gray-scale histogram distribution of daily pixel points presents a bimodal effect for the gray-scale image with obvious separation between the target area and the background area. When the global or value method is used for binarization, a good binary image can be obtained, However, for gray-scale images that are not clearly distinguished between the target area and the background area caused by uneven illumination or noise, or the gray-scale distribution histogram of pixels does not show bimodal effect, when the global threshold method is used for binarization, there will be binary images with poor effect that some areas are too bright and some areas are too dark. The global threshold method has poor resistance to interference factors such as uneven illumination and noise pollution. Local adaptive threshold method is a method to solve the threshold according to the local gray value information of the pixels of the original image and its surrounding pixels. This kind of method generally divides the gray image into several parts, and then uses the global threshold method to find the best threshold of this part in each part. For different parts, different thresholds must be determined according to the spatial position relationship of pixels in this part, then binarize each part according to the obtained or value, and fi-

nally splice the binarization results of each part to realize the binarization of the whole image. In 1986, Bernsen proposed a very classic local adaptive threshold method (Bernsen, 1986), which usually divides the gray image into  $5 \times 5$   $9 \times 9$  image blocks, and then selects the maximum and minimum gray values of each pixel in each image block for average operation and the obtained value is determined as the threshold of the central pixel of the image block. This method of selecting the threshold of each image block has good performance, but if the image contains a lot of noise, it needs to be improved. For example, the average value can be obtained by averaging the second largest and second smallest values of the gray value of each pixel in the image block. The application range of the local adaptive or value method is much wider than that of the global or value method. It can be used for binarization of gray-scale images with serious interference and poor quality, such as uneven illumination and noise pollution. However, the local adaptive threshold method also has some disadvantages, such as time-consuming and memory consuming in the implementation process, unable to guarantee the stroke continuity of characters, and there are many artifacts. In addition, here we introduce a new local adaptive threshold method proposed in recent years, which is a local adaptive threshold method based on learning rules proposed by Chou et al. (Chou, Lin, & Chang, 2010) in 2010. This method first divides the original gray image into several image blocks, then defines four alternative operations on each image block, and then uses the information features extracted from each image block to learn a decision function, which determines which alternative operation the image block uses to calculate the threshold of the image block. The threshold method has a good binarization effect for the image generated by

the camera affected by the shooting environment and lighting conditions, and the experimental speed is also fast. In the processing process of the local adaptive threshold method, since the original gray winter image is divided into several parts, the gray value of pixel points at the junction of each two parts will show discontinuous characteristics. Therefore, it is necessary to eliminate this discontinuity through image smoothing technology. When determining the national value, the mixed value method considers the overall gray information of the original gray end image and uses the local gray information in the neighborhood of the pixel. This kind of method combines the advantages of the global or value method and the local adaptive threshold method while avoiding the disadvantages of slow and time-consuming implementation of the local adaptive threshold method, and has good practicability. The essential difference of the image binarization method based on the threshold is that the image information used to determine the threshold is different. Each threshold method has its own gray image type that is good at processing. At present, no threshold method can effectively binarize any gray image. Various threshold methods have advantages and disadvantages. In the practical application of image binarization, it is necessary to determine the binarization method according to the specific characteristics of gray image.

The image below shows an example of 3D shape after smoothing.

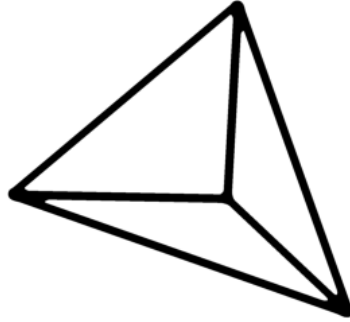


Figure 3.5: 3D Shape with Smoothing

## 3.4 Abaqus

*ABAQUS/CAE* is a front and rear processor of ABAQUS. It has the functions of geometric modeling, model assembly, defining material properties, defining constraints and contacts, meshing, post-processing process automation and so on. It has the functions of special pre and post-processing programs such as Patran and hepermesh. In addition, it is convenient to develop a process environment. *ABAQUS/cm* is a professional composite material modeling tool, which can consider the process performance of the composite layer in the initial stage of modeling to ensure the feasibility of the composite layer in the process. Avoid the increased cost of redesign in the later *R&D* cycle. *ABAQUS/Python* is the most suitable tool for interdisciplinary software development. Using Python for the secondary development and batch processing of ABAQUS can very efficiently complete

the development based on the user's specific needs. Common methods of ABAQUS software executing Python:

- 1) File → Run script... Then select the python program file to run
- 2) Write the program directly in the macro program, and then call the macro Manager (file → macro Manager...) to execute the program
- 3) If only a small amount of code is written, it can be executed in the kernel command line interface at the bottom of ABAQUS software

Our study uses the first approach to visualize 3D shapes by running a python script in Abaqus CAE. The core method is to import the Abaqus library in the python code to turn the 3D shape into a model, then visualize the 3D shape model after running the python script in the Abaqus CAE. Moreover, export the model as a .obj file, which can be processed with Cura and turned into a file that can be recognized by a 3D printer.

The image followed shows a 3D shape in Abaqus CAE

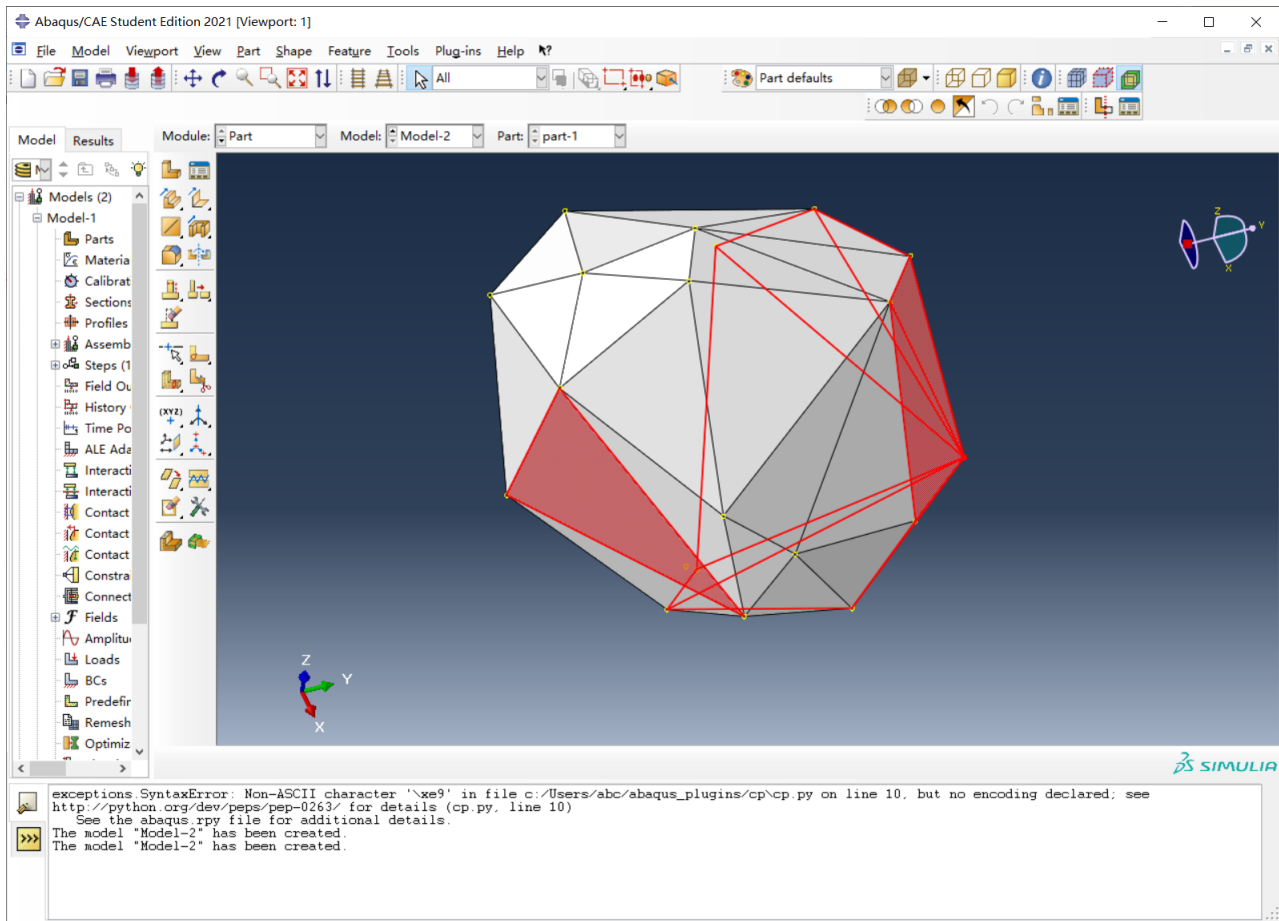


Figure 3.6: 3D Shape in Abaqus CAE

# Chapter 4

## Python Implementation

a. Before we generate 3D shapes, we need to determine the volume of the shape through the PC matrix. Since the experiment needs to be as random as possible, the volume of our 3D shape is also a randomly generated number. However, a too random volume is also undesirable, as it is almost impossible to observe if it is too small, and it cannot be printed if it is too large. Therefore, we first set a random base number as the minimum volume value and then scale it by several fixed ratios. After many experiments, we selected the base volume as 4 and the ratio as 1.2 as parameters. In the PC matrix generated under these parameters, we removed outliers and finally obtained 5 reasonable values as the volumes of the 5 shapes to be generated later.

---

```
size = 5
mat = np.ones((size, size))
n_base = random.randint(1, 5)
ratio = 1.2
n_check = np.zeros(16)
for i in range(size):
```

```

for j in range(size):
    if i <= j:
        Flag = True
        while Flag:
            n = random.randint(1, 15)
            if n_check[n] == 0:
                n_check[n] = 1
                Flag = False
            else:
                continue
        mat[i, j] = 1 if i == j else n_base*ratio*n
    else:
        mat[i, j] = 1/mat[j, i]
print(mat)
print(n_base)

```

---

b. Then, we need to generate 3D shapes. Firstly, random vertices need to be created. Since this project requires shapes which are somehow "nice" for the pairwise comparisons by volume. It excludes harp edges, deep valleys and more. Evidently, no "niceness" definition exists, but we can make a few reasonable assumptions for the generated 3D shapes still having randomness implemented as much as possible.

---

```

vertices = [] # store vertices coord
while True:
    angle1 = np.random.uniform(0, pi*2)
    angle2 = np.random.uniform(0, pi*2)
    # randomly generate vertices coord: x, y, z
    z = radius*cos(angle1)

```



```

x = radius*sin(angle1)*cos(angle2)
y = radius*sin(angle1)*sin(angle2)
if len(vertices):
    if checkVertex(radius, [x, y, z], vertices):
        vertices.append([x, y, z])
    else:
        pass
else:
    vertices.append([x, y, z])

if len(vertices) >= verticeNumber:
    break

```

---

Secondly, Since the randomly generated vertex coordinates are likely to be very close, it is necessary to calculate the distance between the newly generated vertex and other vertices coordinates during regeneration.

---

```

def checkVertex(radius, vertex, vertices):
    # vertex -- newly generated vertex coord
    # vertices -- valid vertices coord

    sign = True
    for point in vertices:
        x1, y1, z1 = point[0], point[1], point[2]
        x2, y2, z2 = vertex[0], vertex[1], vertex[2]
        distance = np.sqrt((x1-x2)**2+(y1-y2)**2+(z1-z2)**2)
        if distance<0.3*radius:
            sign = False
            break
    return sign

```

---

c. After generating the vertices, we need to connect them into convex polyhedra (concave polyhedra may overlap and not even be connected to form a complete 3D shape). In this project, we generate a convex 3D shape by judging the relative position of each surface (a surface formed by connecting any 3 points) and all other vertices; that is, when all other vertices are on the same side of the target surface, it is a qualified surface; otherwise, the surface does not meet the conditions and needs to be discarded.

Therefore, after we generate a surface, we need to calculate its normal vector

---

```

coord1, coord2, coord3 = coords[0], coords[1], coords[2]
x1, y1, z1 = coord1[0], coord1[1], coord1[2]
x2, y2, z2 = coord2[0], coord2[1], coord2[2]
x3, y3, z3 = coord3[0], coord3[1], coord3[2]

a = (y2-y1)*(z3-z1) - (y3-y1)*(z2-z1)
b = (z2-z1)*(x3-x1) - (z3-z1)*(x2-x1)
c = (x2-x1)*(y3-y1) - (x3-x1)*(y2-y1)
n = np.array([a, b, c])
n = n/np.linalg.norm(n)
d = arccos(np.dot(n, coord1)/np.linalg.norm(coord1))
if d>pi/2: # keep normal vector outward
    n = -n
return n

```

---

Then, obtaining the normal vector  $n$ , traverse all the remaining vertices and connect the vertices to any point on the surface to obtain the direction

vector  $V$  from the vertex to the surface. If  $n \cdot V$  is less than 0, it means that the angle between the two vectors is an obtuse angle, which means that the point is on the other side of the surface; that is, the face does not meet the conditions and needs to be discarded

---

```
vertices = [np.array(vertice) for vertice in vertices]
threePoints = list(itertools.combinations(vertices, 3))
chosenSurface = []
for threePoint in threePoints:
    n = nmlVector(threePoint)
    sign = True
    for vertex in vertices:
        vector1 = threePoint[0]-vertex
        result = np.dot(vector1, n)
        if result < -1e-5:
            sign = False
            break
    if sign:
        chosenSurface.append(list(threePoint))
return chosenSurface
```

---

d. After generating all the surfaces that meet the standard, a randomly generated convex 3D shape has been theoretically obtained, and then the 3D shape needs to be visualized

---

```
# Render 3D shapes
ax = a3.Axes3D(pl.figure(figsize=(6,6), dpi=600))
for i in range(len(draw)):
    vtx = draw[i]
    tri = a3.art3d.Poly3DCollection([vtx])
```

```

tri.set_color(colors.rgb2hex([1,1,1])) # face color

tri.set_edgecolor('k')

ax.add_collection3d(tri)

# change axis

ax.set_xlim(-1,1)
ax.set_ylim(-1,1)
ax.set_zlim(-1,1)

ax.grid(False)
ax.set_xticks([])
ax.set_yticks([])
ax.set_zticks([])
plt.axis('off')

plt.gca().set_box_aspect((1,1,1))

# Save image
plt.savefig("./3d_aggregate-{}.png".format(figN))

```

---

e. Apply Gaussian blur to the generated 3D shape. Then set a threshold, change the pixels above the threshold to white, change the pixels below the threshold to black to make the whole 3D shape look smoother, and finally get a nice 3D shape

---

```

# Load image - GaussianBlur

image = cv2.imread("./3d_aggregate-{}.png".format(figN))
result = cv2.GaussianBlur(image, (65, 65), 15)
cv2.imwrite("./3d_aggregate-{}-Gauss.png".format(figN), result)

```

```

# Set Th - modify img
thred = 210 # np.random.uniform(0, 255)
print("Th=", thred)
imageNum =
    np.array(Image.open("./3d_aggregate-{}-Gauss.png".format(figN)).convert("L"))

imageNum[np.where(imageNum > thred)] = 255
imageNum[np.where(imageNum <= thred)] = 0

plt.figure()
plt.imshow(imageNum, cmap='gray')
plt.grid(False)
plt.xticks([])
plt.yticks([])
plt.axis('off')

plt.savefig("./3d_aggregate-{}-thred.png".format(figN))

```

---

f. Since it may cause the shape to exceed the coordinate system after processing, we set a threshold to ensure that the shape is in the coordinate system, which can display the smoothed shape more clearly for the experimenter's reference. Meanwhile, calculate its volume so that the error does not exceed 0.001.

---

```

chosenSurface = np.array(chosenSurface)
upper, lower = 5, 0
while True:
    center = (upper+lower)/2.

```

```
changeSurface = chosenSurface * center
volume = calVolume(changeSurface)
if volume < user_volume:
    lower = center
else:
    upper = center

if abs(volume-user_volume)/user_volume < 0.001:
    print(f"center={center},volume={volume}")
    break

return changeSurface
```

---

# Chapter 5

## Experiment

Ethics is also an essential part of the experiment, especially the subjective judgment of the respondent needs to be used in this project. First, when making 3D models, researchers absolutely guarantee that 3D objects are generated randomly and will not customize specific shapes to meet the experimental results. Secondly, respondents will not be informed of the model information in advance before the experiment to ensure that all respondents are unknown about the model to be estimated before the experiment. In addition, for the final outspoken results, even if the results are inconsistent with the expected results according to the literature review, the data will not be modified to fit the specific conclusions. In addition, the researchers fully recognize that the conclusions in the literature belong to the research results of other scholars and cannot be directly used in this study. This study will fully respect the data conclusions. Finally, the model generated in this study will not have implications, ethics and other issues, will not involve children, mental health and other sensitive information, and will not cause discomfort to respondents through the model.

Our entire experiment will be carried out in strict compliance with the requirements of the ethics section above. Our experiment is divided into three main phases, namely the preparation phase, the experimental phase and the data analysis phase.

## 5.1 Preparation Phase

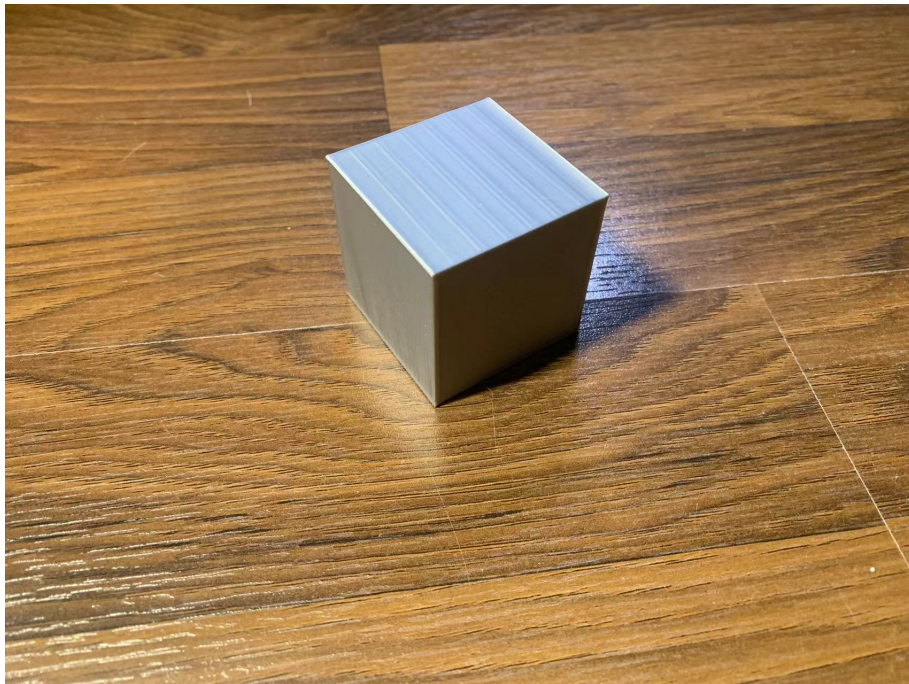
After the code part is ready, the equipment for 3D printing needs to be prepared. The 3D printer used in this experiment is the *Easythreed K7 Desktop Mini 3D Printer* purchased on Taobao (a Chinese online shopping platform), and it can also be purchased on Amazon. *EasyThreed K7 Mini* is an entry-level 3D printer, its nozzle diameter is  $0.4mm$ , the printing speed is set to  $30mm$  per second, the maximum size that can be printed is  $100 \times 100 \times 100 mm$ . The temperature during printing is room temperature, the temperature of the extruder is  $200^{\circ}C$ , the printing material used is PLA, the diameter of the material is  $1.75mm$  and the error is  $\pm 0.02mm$ . The reason we choose this 3D printer is that it is affordable and can meet our basic needs, the disadvantage is that the quality of the parts of the machine are very fragile and often needs to be adjusted; besides, the printing accuracy is lower than other more expensive printers, but basically has no effect on the test.

In order to make the experimental results more accurate, we printed five 3D shapes with different volumes and different shapes. We used these five 3D shapes to conduct experiments with respondents. It is worth mentioning that the precision of the 3D printer we use is not very high, and the material itself has volume. Therefore, the real volume of the printed 3D shapes can-



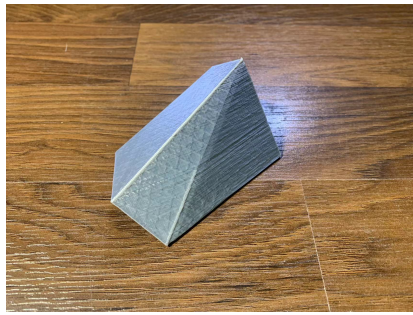
not be exactly the same as the set value, so the error is inevitable. However, the error will be controlled within an acceptable range, that is, the difference that cannot be distinguished by the human eye. Our experimental method uses the pairwise comparisons method with the direct method proposed by Prof. W.W.Koczkodaj in 1998 (Koczkodaj, 1998). This method allows the experimenter to estimate the volume of the object only by visual observation, so the error that cannot be observed by the human eye is allowed.

The images shown below are the 3D shapes that we are using in the experiment:

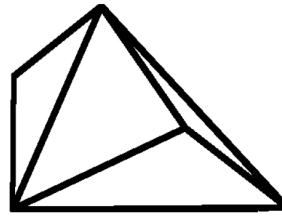


(a) 3D Shape

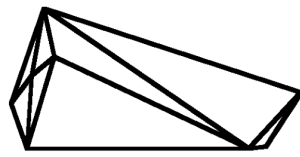
Figure 5.1: Unit Cube:  $64cm^3$



(a) 3D Shape



(b) Front

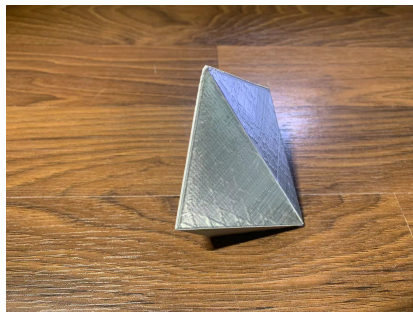


(c) Left



(d) Top

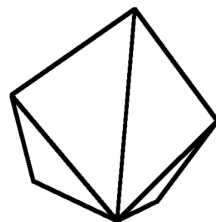
Figure 5.2: Shape A:  $58cm^3$



(a) 3D Shape



(b) Front



(c) Left

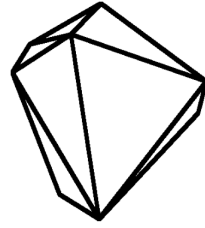


(d) Top

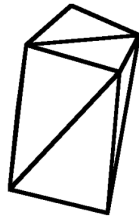
Figure 5.3: Shape B:  $72cm^3$



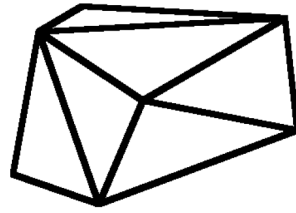
(a) 3D Shape



(b) Front

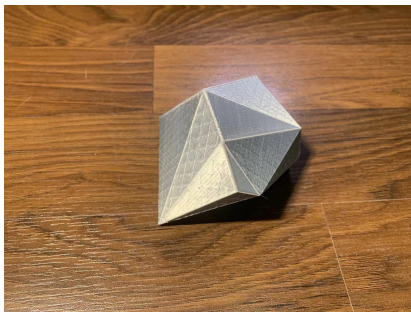


(c) Left

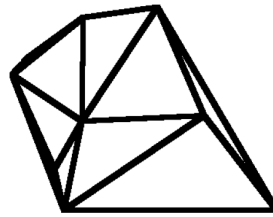


(d) Top

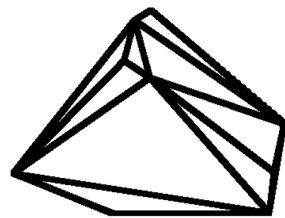
Figure 5.4: Shape C:  $90\text{cm}^3$



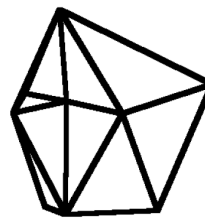
(a) 3D Shape



(b) Front

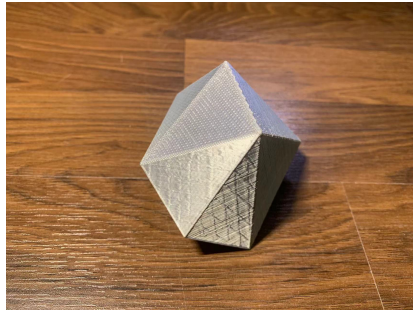


(c) Left

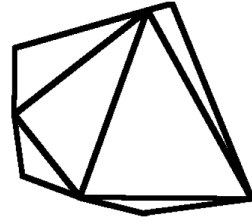


(d) Top

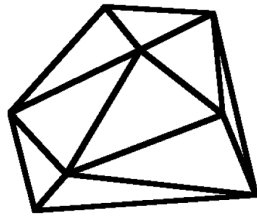
Figure 5.5: Shape D:  $113\text{cm}^3$



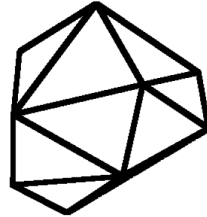
(a) 3D Shape



(b) Front



(c) Left



(d) Top

Figure 5.6: Shape E:  $142cm^3$

## 5.2 Experiment Phase

In our study, 32 respondents estimated the volume of 3D shapes. Twenty-eight of them are from China, three from Canada and one from India. Since most of the respondents were students, twenty-three were between the ages of 20 and 30, and the remaining nine were over 30, but the exact ages were not recorded. In addition, women make up about 40% of the total.

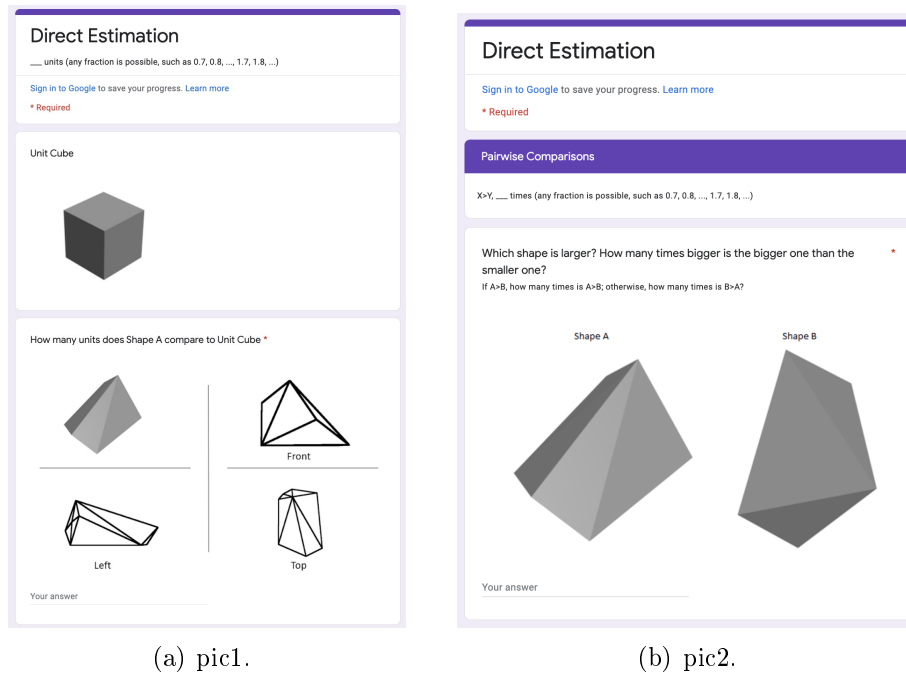
We refer to the 2D case to conduct the experiment and divide the experiment into two parts. In the first part, in order to avoid estimation errors caused by visual bias due to the absence of reference objects, we print a cube with a volume of  $64cm^3$  as a unit cube. Then respondents need to answer how many unit cubes each 3D shape is equal to, and we

also mentioned that any fraction is possible (such as 0.7 unit, 1.2 unit, etc.). Due to the Omicron variant of COVID-19, it is impossible for us to have respondents directly touch the 3D shapes for observation. Therefore, we can only allow the respondents to observe 3D shapes and estimate the volume through online meetings and recorded videos. We display five 3D shapes for all respondents, and each 3D shape is slowly rotated  $360^\circ$  on the  $x - axis$  and  $z - axis$  respectively, so that each face of the 3D shape is displayed to the respondents.

In the second part of the experiment, pairs of five 3D shapes will be presented to respondents. All five 3D shapes are combined in pairs until there are no different combinations, a total of 10 groups. Likewise, the 3D shapes in each group are also rotated and displayed to respondents. Then they need to answer which 3D shape in each group is more significant, and they can also answer that the volumes are equal. In addition, the volume ratio of the two 3D shapes needs to be answered. For example, if *Shape A* is estimated to be 10% larger than *Shape B*, then answer A is 1.1 times larger than B.

To make it easier for respondents to answer and for us to collect data. We made a questionnaire with *Google Form*, which listed all our questions, as well as pictures of all the 3D shapes involved in each question. The Google Form is shown in the following image. In addition, since the respondents in China could not access the Google Form, we also made a *Tencent Questionnaire* for them, which is exactly the same as Google Form. After all respondents submitted their answer, we collected and integrated the data

into Excel for analysis.



(a) pic1.

(b) pic2.

Figure 5.7: Google Form: Volume Estimation

### 5.3 Analysis Phase

Each respondent is required to answer 15 questions (5 questions in the first part, ten questions in the second part) to estimate the volume of 3D shapes by the direct method and the pairwise comparisons method, resulting in 160 answers for the first part and 320 answers for the second part. We tabulated the responses to the two sections and grouped them by each question for analysis. First, we calculate the absolute value of the difference between each response and the correct value. Then the median, mean and standard deviation of the differences in each group were calculated. The following tables illustrate the result.

	Shape A	Shape B	Shape C	Shape D	Shape E	ALL
Mdn	0.16	0.20	0.23	0.27	0.28	0.22
M	0.1527	0.1765	0.2173	0.3070	0.4781	0.2663
SD	0.0988	0.1094	0.1227	0.2129	0.3575	0.2366

Table 5.1: Direct Method

	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pair 8	Pair 9	Pair 10	ALL
Mdn	0.09	0.15	0.15	0.27	0.05	0.10	0.13	0.06	0.15	0.11	0.13
M	0.1166	0.1921	0.2256	0.3072	0.0894	0.1545	0.2047	0.1027	0.2000	0.1109	0.1704
SD	0.0839	0.1378	0.2216	0.2419	0.0689	0.1420	0.1991	0.1300	0.1472	0.0691	0.1684

Table 5.2: Pairwise Comparisons Method



# Chapter 6

## Conclusion and Future Work

### 6.1 Conclusion

Due to all shapes being randomly generated, the errors in estimating volume with these two methods are similar, so the results can be considered to be related only to the experimental method and the respondents. The mean error using the direct method is 0.2663 (26.63%), which is larger than the 15.42% reported by Prof. Koczkodaj in the one-dimensional experiments (Koczkodaj, 1996) and 25.75% reported in the two-dimensional experiments (Adamic et al., 2009). The mean error of the Pairwise comparisons method is 0.1704 (17.04%), which is also more significant than the 3.64% reported by the one-dimensional experiments (Koczkodaj, 1996) and the 5.51% reported by the two-dimensional experiments (Adamic et al., 2009).

This result is reasonable because, from the perspective of the respondents, 3D shapes are more complex than 1D and 2D shapes, and more factors need to be considered when estimating. Moreover, most respondents were

unable to observe 3D shapes as close as they could in the 1D and 2D experiments, thus giving a larger error rate than 1D and 2D cases. Nevertheless, the mean error using the pairwise comparisons method is still much smaller than the direct method. In addition, the standard deviation of the error of the pairwise comparisons method is 0.1684 (16.84%), which is also smaller than the direct observation of 0.2366 (23.66%).

It is not difficult to find from this result that although the error obtained by the two methods is much larger than that in the 1D and 2D cases, the overall error range using the pairwise comparisons method is smaller, and the result is closer to the correct answer. In conclusion, from the above analysis results It can be inferred that the pairwise comparison method can improve the accuracy in estimating the volume of 3D shapes.

## 6.2 Future Work

Since we are the first experiment to apply pairwise comparisons to estimating 3D shapes, it has a lot of imperfections and a lot of room for improvement. First, the logic and algorithms we use to generate 3D shapes are not guaranteed to be optimal because there are no similar experiments for our reference. Second, the 3D printers we use are entry-level machines, and the printing accuracy is not the highest, and the volume of the printed 3D shape cannot exceed  $1000cm^3$ . Due to the randomness of the 3D shape, the actual size that can be printed cannot even exceed  $500cm^3$ . Therefore, we need to use more advanced 3D printers to verify whether the pairwise comparisons method can improve accuracy when estimating larger volumes of 3D shapes. Third, due to the impact of COVID, respondents are unable

to observe 3D shapes up close, so we will need to do more experiments in the future to allow respondents to touch 3D shapes and estimate volumes directly. Finally, with the improvement of the experiment, after we obtain more data, we will seek help from statistical experts by using more scientific statistical methods to obtain more reliable results. We predict that the mean error and standard deviation obtained using the pairwise comparisons method will be smaller than the current results after the experiment is improved.

# References

- Adamic, P., Kakiashvili, T., Koczkodaj, W., Babiy, V., Janicki, R., & Tadeusiewicz, R. (2009). Pairwise comparisons and visual perceptions of equal area polygons. *Perceptual and motor skills*, *108*(1), 37–42.
- Bernsen, J. (1986). Dynamic thresholding of gray level image. *Proceedings of International Conference on Pattern Recognition (ICPR '86)*, 1251–1255.
- Bozóki, S., & Rapcsák, T. (2008). On saaty's and koczkodaj's inconsistencies of pairwise comparison matrices. *Journal of Global Optimization*, *42*(2), 157–175.
- Carlsson, C., & Fullér, R. (1996). Fuzzy multiple criteria decision making: Recent developments. *Fuzzy sets and systems*, *78*(2), 139–153.
- Chou, C.-H., Lin, W.-H., & Chang, F. (2010). A binarization method with learning-built rules for document images produced by cameras. *Pattern Recognition*, *43*(4), 1518–1530.
- Distribution, N. (n.d.). *A plus topper*.  
<https://www.aplustopper.com/normal-distribution/>.  
(Accessed March 26, 2022)
- Fedrizzi, M., & Giove, S. (2007). Incomplete pairwise comparison and consistency optimization. *European journal of operational research*,

183(1), 303–313.

- Huang, J., Peng, Q., Hu, X., & Du, Y. (2017). A combined-alpha-shape-implicit-surface approach to generate 3d random concrete mesostructures via digital image processing, spectral representation, and point cloud. *Construction and Building Materials*, 143, 330–365.
- Kakiashvili, T., Koczkodaj, W. W., Montgomery, P., Passi, K., & Tadeusiewicz, R. (2008). Assessing the properties of the world health organization's quality of life index. In *2008 international multiconference on computer science and information technology* (pp. 151–154).
- Kavallieratou, E., & Stamatatos, E. (2006). Improving the quality of degraded document images. In *Second international conference on document image analysis for libraries (dial'06)* (pp. 10–pp).
- Kavallieratou, E., & Stathis, S. (2006). Adaptive binarization of historical document images. In *18th international conference on pattern recognition (icpr'06)* (Vol. 3, pp. 742–745).
- Koczkodaj, W. W. (n.d.). *The key properties of inconsistency indicators for a triad in pairwise comparison matrices*.  
[https://www.academia.edu/73616651/The\\_key\\_properties\\_of\\_inconsistency\\_indicators\\_for\\_a\\_triad\\_in\\_pairwise\\_comparison\\_matrices](https://www.academia.edu/73616651/The_key_properties_of_inconsistency_indicators_for_a_triad_in_pairwise_comparison_matrices). (Accessed April 2, 2022)
- Koczkodaj, W. W. (1993). A new definition of consistency of pairwise comparisons. *Mathematical and computer modelling*, 18(7), 79–84.
- Koczkodaj, W. W. (1996). Statistically accurate evidence of improved

- error rate by pairwise comparisons. *Perceptual and Motor Skills*, 82(1), 43–48.
- Koczkodaj, W. W. (1998). Testing the accuracy enhancement of pairwise comparisons by a monte carlo experiment. *Journal of Statistical Planning and Inference*, 69(1), 21–31.
- Koczkodaj, W. W., Kosiek, M., Szybowski, J., & Xu, D. (2015). Fast convergence of distance-based inconsistency in pairwise comparisons. *Fundamenta Informaticae*, 137(3), 355–367.
- Koczkodaj, W. W., Kułakowski, K., & Ligęza, A. (2014). On the quality evaluation of scientific entities in poland supported by consistency-driven pairwise comparisons method. *Scientometrics*, 99(3), 911–926.
- Kou, G., Ergu, D., Lin, C., & Chen, Y. (2016). Pairwise comparison matrix in multiple criteria decision making. *Technological and economic development of economy*, 22(5), 738–765.
- Kułakowski, K., Mazurek, J., & Strada, M. (2021). On the similarity between ranking vectors in the pairwise comparison method. *Journal of the Operational Research Society*, 1–10.
- Mason, D., Lauder, I., Rutovitz, D., & Spowart, G. (1975). Measurement of c-bands in human chromosomes. *Computers in biology and medicine*, 5(3), 179–201.
- Molotsky, S. (2011). *What is the decision making process of entrepreneurs' implementation of mcs software, and how are external consultants engaged and used in the process? how does this engagement affect an ongoing relationship between the two?: an exploratory study,[a thesis presented in partial fulfilment of the*

*requirements for the degree of master of business studies in accountancy at massey university, palmerston north, new zealand* (Unpublished doctoral dissertation). Massey University.

Pavlidis, T. (1993). Threshold selection using second derivatives of the gray scale image. In *Proceedings of 2nd international conference on document analysis and recognition (icdar'93)* (pp. 274–277).

Percoco, G., Ghaleb, A. M., Kaid, H., Alsamhan, A., Mian, S. H., & Hidri, L. (2020). Assessment and comparison of various mcdm approaches in the selection of manufacturing process. *Advances in Materials Science and Engineering, 2020*, 4039253. Retrieved from <https://doi.org/10.1155/2020/4039253> doi: 10.1155/2020/4039253

Qin, F., Jin-hua, Z., Yi, H., & Ya-dong, Z. (2013). *The investigation into three-dimensional mesoscale modelling of fully-graded concrete* (Vol. 30) (No. 1000-4750(2013)01-0014-08). Retrieved from <http://engineeringmechanics.cn/en/article/doi/10.6052/j.issn.1000-4750.2011.06.0398> doi: 10.6052/j.issn.1000-4750.2011.06.0398

Ramík, J., et al. (2017). Ranking alternatives by pairwise comparisons matrix and priority vector. *Scientific Annals of Economics and Business, 64* (Special), 85–95.

Ramík, J. (2020). *Pairwise comparisons method*. doi: <https://doi-org.librweb.laurentian.ca/10.1007/978-3-030-39891-0>

Saaty, R. W. (1987). The analytic hierarchy process—what it is and how it is used. *Mathematical modelling, 9*(3-5), 161–176.

Saaty, T. L. (1991). Some mathematical concepts of the analytic

- hierarchy process. *Behaviormetrika*, 18(29), 1–9.
- Sobel, I. (2014). History and definition of the sobel operator. *Retrieved from the World Wide Web*, 1505.
- Van Vliet, L. J., Young, I. T., & Beckers, G. L. (1989). A nonlinear laplace operator as edge detector in noisy images. *Computer vision, graphics, and image processing*, 45(2), 167–195.
- Wang, H., Peng, Y., & Kou, G. (2021). A two-stage ranking method to minimize ordinal violation for pairwise comparisons. *Applied Soft Computing*, 106, 107287.
- Zavadskas, E. K., & Turskis, Z. (2011). Multiple criteria decision making (mcdm) methods in economics: an overview. *Technological and economic development of economy*, 17(2), 397–427.



# Appendix

## A1. Generate 3D Shapes in Python

---

```
# import

import numpy as np

import matplotlib.pyplot as plt

from numpy import pi,sin,cos,arccos

import itertools

import mpl_toolkits.mplot3d as a3

import matplotlib.colors as colors

import pylab as pl

import cv2

from PIL import Image

# *****

#     user Defined parameter

# *****

faceNumber = np.random.uniform(20, 30)

figNumber = 2

volume = 10

# *****

#             function

# *****

#

def cacu_Radius(volume):

    radius = (volume*3/(np.pi*4))**(1/3)

    # print(f"Radius of base:{radius}")
```

```

return radius

# randomly generate vertex
def createVertices(radius=1, faceNumber=4):
    '''
    :param verticeNumber:number of vertices
    :return: coordinates of vertices
    '''
    verticeNumber = int((faceNumber-4)/2)+4
    def checkVertex(radius, vertex, vertices):
        # vertex -- randomly generate vertec coord
        # vertices -- selected vertices coord
        sign = True
        for point in vertices:
            x1, y1, z1 = point[0], point[1], point[2]
            x2, y2, z2 = vertex[0], vertex[1], vertex[2]
            distance = np.sqrt((x1-x2)**2+(y1-y2)**2+(z1-z2)**2)
            if distance<0.3*radius:
                sign = False
                break
        return sign

    vertices = [] # save vertex coord
    while True:
        angle1 = np.random.uniform(0, pi*2)
        angle2 = np.random.uniform(0, pi*2)
        # randomly generate x,y,z coord
        z = radius*cos(angle1)

```

```

x = radius*sin(angle1)*cos(angle2)
y = radius*sin(angle1)*sin(angle2)
if len(vertices):
    if checkVertex(radius, [x, y, z], vertices):
        vertices.append([x, y, z])
    else:
        pass
else:
    vertices.append([x, y, z])

if len(vertices) >= verticeNumber:
    break
return vertices

# calculate normal vector
def nmlVector(coords):
    coord1, coord2, coord3 = coords[0], coords[1], coords[2]
    x1, y1, z1 = coord1[0], coord1[1], coord1[2]
    x2, y2, z2 = coord2[0], coord2[1], coord2[2]
    x3, y3, z3 = coord3[0], coord3[1], coord3[2]

    a = (y2-y1)*(z3-z1) - (y3-y1)*(z2-z1)
    b = (z2-z1)*(x3-x1) - (z3-z1)*(x2-x1)
    c = (x2-x1)*(y3-y1) - (x3-x1)*(y2-y1)
    n = np.array([a, b, c])
    n = n/np.linalg.norm(n)
    d = arccos(np.dot(n, coord1)/np.linalg.norm(coord1))
    if d>pi/2: # keep normal vector outward

```

```

        n = -n

    return n

# surface check
def validSurface(vertices):
    vertices = [np.array(vertice) for vertice in vertices]
    threePoints = list(itertools.combinations(vertices, 3))
    chosenSurface = []

    for threePoint in threePoints:
        n = nmlVector(threePoint)
        sign = True

        for vertex in vertices:
            vector1 = threePoint[0]-vertex
            result = np.dot(vector1, n)

            if result<-1e-5:
                sign = False
                break

        if sign:
            chosenSurface.append(list(threePoint))

    return chosenSurface

# Draw and save
def drawPicture(draw, figN=0, thred=1):
    # Render 3D shapes
    ax = a3.Axes3D(pl.figure(figsize=(3, 3), dpi=600))

    for i in range(len(draw)):
        vtx = draw[i]
        tri = a3.art3d.Poly3DCollection([vtx])

```

```

tri.set_color(colors.rgb2hex([1, 1, 1]))

tri.set_edgecolor('k')

ax.add_collection3d(tri)

# change axis
ax.set_xlim(-1, 1)
ax.set_ylim(-1, 1)
ax.set_zlim(-1, 1)

#
ax.grid(False)
ax.set_xticks([])
ax.set_yticks([])
ax.set_zticks([])
plt.axis('off')

plt.gca().set_box_aspect((1, 1, 1))

# Save image
plt.savefig("./3d_aggregate-{}.png".format(figN))

# plt.show()

# load image - Gaussian Blur
image = cv2.imread("./3d_aggregate-{}.png".format(figN))
# result = cv2.GaussianBlur(image, (120, 100), 75)
result = cv2.GaussianBlur(image, (45, 45), 15)
cv2.imwrite("./3d_aggregate-{}-Gauss.png".format(figN),
            result)

# set Th - modify image

```

```

imageNum =
    np.array(Image.open("./3d_aggregate-{}-Gauss.png".format(figN)).convert("L"))

imageNum[np.where(imageNum > thred)] = 255
imageNum[np.where(imageNum <= thred)] = 0

plt.figure()
plt.imshow(imageNum, cmap='gray')
plt.grid(False)
plt.xticks([])
plt.yticks([])
plt.axis('off')

# plt.show()
plt.savefig("./3d_aggregate-{}-thred.png".format(figN))

# calculate volume
def calVolume(chosenSurface):
    volume = 0.0
    def volume_single(points):
        a, b, c = points[0], points[1], points[2]
        v1 = abs(np.dot(np.cross(a, b), c))/6
        return v1

    for points in chosenSurface:
        volume += volume_single(points)

```

```

    return volume

# renew surface coord
def changeVertices(chosenSurface, user_volume):
    chosenSurface = np.array(chosenSurface)
    upper, lower = 5, 0
    while True:
        center = (upper+lower)/2.
        changeSurface = chosenSurface * center
        volume = calVolume(changeSurface)
        if volume < user_volume:
            lower = center
        else:
            upper = center

        if abs(volume-user_volume)/user_volume < 0.001:
            print(f"center={center}, volume={volume}")
            break

    return changeSurface

# *****
#             Main code
# *****
thred = 215 # np.random.uniform(0, 255)
for i in range(figNumber):
    radius = cacu_Radius(volume)

```

```

vertices = createVertices(radius=radius,
    faceNumber=faceNumber)
surface = validSurface(vertices)
changeSurface = changeVertices(chosenSurface=surface,
    user_volume=volume)
drawPicture(changeSurface, i, thred)

```

---

## A2. PC Matrix

---

```

import random
import numpy as np
from fractions import Fraction

np.set_printoptions(formatter={'all':lambda x:
    str(Fraction(x).limit_denominator())}) # use fraction format
np.set_printoptions(suppress=True)

# Matrix size
size = 5
mat = np.ones((size, size))
n_base = random.randint(1, 10)
ratio = 1.25
n_check = np.zeros(16)
for i in range(size):
    for j in range(size):
        if i <= j:
            Flag = True
            while Flag:

```



```

n = random.randint(1, 15)

if n_check[n] == 0:
    n_check[n] = 1
    Flag = False
else:
    continue

mat[i, j] = 1 if i == j else n_base*pow(ratio, n)
else:
    mat[i, j] = 1/mat[j, i]

print(mat)
print(n_base)

```

---

### A3. Abaqus Source Code

---

```

if mdb.models.has_key("Model-2"):
    del mdb.models["Model-2"]

model = mdb.Model(name="Model-2", modelType=STANDARD_EXPLICIT)

part = model.Part(name="part-1", dimensionality=THREE_D,
    type=DEFORMABLE_BODY)

vertices = createVertices(verticeNumber=20)
chosePlane = chosenPlane(vertices)

# create point
for vertice in vertices:
    part.DatumPointByCoordinate(coords=tuple(vertice))

```

```
# create plane
for coords in chosePlane:
    coords.append(coords[0])
    wire = part.WirePolyLine(mergeType=SEPARATE, meshable=ON,
        points=(coords))
    face_edge = part.getFeatureEdges(name=wire.name)
    part.CoverEdges(edgeList = face_edge, tryAnalytical=True)
```

---

## A4. Data







No.	Unit Cube	Direct				
		Shape A	Shape B	Shape C	Shape D	Shape E
						
1		0.85	1.15	1.25	1.4	1.5
2		1	1.1	1.6	1.7	1.8
3		0.8	1.1	1.5	2	2.3
4		0.7	0.9	1.3	1.7	2
5		1.1	1.2	1.3	1.6	1.8
6		0.9	1.03	1.4	1.8	2
7		1.2	0.8	1.26	1.9	2
8		1.3	1.5	1.6	1.8	2
9		0.6	0.9	1	1.9	2.3
10		0.8	0.85	1.05	2.1	2.2
11		0.75	1	1.15	2.1	2.3
12		0.7	0.9	1.4	1.7	2
13		0.65	0.9	1.4	1.75	1.95
14		0.75	1.07	1.45	1.95	2
15		0.9	0.8	1.1	1.2	1.3
16		0.75	0.9	1.1	1.2	1.3
17		0.75	0.9	1.1	1.2	1.3
18		0.75	0.9	1.1	1.2	1.3
19		0.7	0.9	1.1	1.2	1.3
20		0.7	0.9	1.1	1.2	2
21		0.8	0.9	1.15	1.3	2
22		0.8	1	1.15	1.3	2
23		1.2	1	1.15	1.3	2
24		1.2	1	1	1.8	1.5
25		0.9	1	1	1.8	1.5
26		0.9	1.1	1.25	1.5	1.1
27		1	1.1	1.25	1.5	1.2
28		1	1.1	1.5	1.25	1.8
29		0.85	0.7	1.5	1.25	1.4
30		0.85	0.8	1.05	2	2.1
31		0.65	1.3	1.2	2	1
32		1	0.95	1.8	1	2.5

Figure 1: Direct Method

No.	Pairwise									
	Pair1	Pair2	Pair3	Pair4	Pair5	Pair6	Pair7	Pair8	Pair9	Pair10
	Shape A, B (B:A)	Shape A, C (C:A)	Shape A, D (D:A)	Shape A, E (E:A)	Shape B, C (C:B)	Shape B, D (D:B)	Shape B, E (E:B)	Shape C, D (D:C)	Shape C, E (E:C)	Shape D, E (E:D)
1	1.2	1.35	2	2	1.1	1.5	1.5	1.2	1.2	1
2	1.1	1.4	1.5	2.1	1.3	1.4	1.8	1.2	1.3	1.1
3	1.1	1.45	1.7	2.5	1.2	1.6	2	1.1	1.45	1.35
4	1.23	1.3	1.8	2.6	1.2	1.6	2.2	1.18	1.4	1.29
5	1.2	1.4	1.8	2	1.2	1.5	1.8	1.3	1.5	1.2
6	1.15	1.3	1.7	2.55	1.26	1.5	2	1.3	1.5	1.2
7	1.1	1.5	2	2.4	1.3	1.7	2.1	1.05	1.6	1.2
8	1.2	1.3	1.6	1.8	1.3	1.5	2	1.3	1.5	1.2
9	1.3	1.5	1.8	2	1.3	1.7	2.2	1.3	1.45	1.25
10	1.15	1.3	1.7	2.5	1.5	1.8	2	1.15	1.3	1.15
11	1.2	1.45	1.8	2.6	1.15	1.7	2.1	1.2	1.45	1.2
12	1.2	1.45	1.85	2.6	1.3	1.65	2	1.3	1.7	1.2
13	1.25	1.6	1.95	2.4	1.15	1.55	1.85	1.2	1.65	1.25
14	1.3	1.55	2	3	1.35	1.8	2.1	1.3	1.8	1.4
15	1.1	1.4	1.8	2.1	1.2	1.6	2.1	1.2	1.6	1.1
16	1	1.5	2	2.2	1.1	1.5	2	1.2	1.4	1.1
17	1	1.5	1.8	2.2	1.1	1.5	1.8	1.25	1.4	1.1
18	1.1	1.5	1.8	2	1.1	1.5	1.9	1.15	1.5	1.1
19	1.2	1.3	1.8	2	1.1	1.55	2.1	1.1	1.5	1.2
20	1.1	1.3	1.8	1.2	1.3	1.55	1.75	1.2	1.5	1.2
21	1.2	1.4	2	2.1	1.3	1.45	2	1.25	1.5	1.15
22	1	1.35	1.7	2.5	1.25	1.1	1.8	1.1	2	1.15
23	1	1.45	1.7	2.2	1.15	1.3	1.9	1.5	1.1	1.15
24	1.15	1.2	1.5	2	1.2	1.55	1.6	1.1	1.3	1.2
25	1.3	1.1	2	2.1	1	1.1	1.3	1.3	1.2	1.05
26	0.9	1.5	1.3	2.05	1.15	1.25	1.3	1.3	1.3	1.1
27	1.1	1.4	1.1	2.3	1.1	1.8	1.5	1.05	1.55	1
28	1.1	1	1.75	2.2	1.3	1.4	2.1	1.25	1.8	1.3
29	1.25	1.3	1.7	2.4	1.25	1.9	1.2	1.15	1.15	1.2
30	1.05	1.25	1.9	2.5	1.25	1	1.7	1.2	1.2	1.1
31	1	1.8	1.9	3	1	1.5	1.75	1.2	1.5	1
32	1.15	1	1	2.15	1.2	1.4	2	2	1	1.1

Figure 2: Pairwise Comparisons Method

No.	Shape A	Shape B	Shape C	Shape D	Shape E
1	0.06	0.02	0.16	0.37	0.72
2	0.09	0.02	0.19	0.07	0.42
3	0.11	0.02	0.09	0.23	0.08
4	0.21	0.23	0.11	0.07	0.22
5	0.19	0.08	0.11	0.17	0.42
6	0.01	0.10	0.01	0.03	0.22
7	0.29	0.33	0.15	0.13	0.22
8	0.39	0.38	0.19	0.03	0.22
9	0.31	0.23	0.41	0.13	0.08
10	0.11	0.28	0.36	0.33	0.02
11	0.16	0.13	0.26	0.33	0.08
12	0.21	0.23	0.01	0.07	0.22
13	0.26	0.23	0.01	0.02	0.27
14	0.16	0.05	0.04	0.18	0.22
15	0.01	0.33	0.31	0.57	0.92
16	0.16	0.23	0.31	0.57	0.92
17	0.16	0.23	0.31	0.57	0.92
18	0.16	0.23	0.31	0.57	0.92
19	0.21	0.23	0.31	0.57	0.92
20	0.21	0.23	0.31	0.57	0.22
21	0.11	0.23	0.26	0.47	0.22
22	0.11	0.13	0.26	0.47	0.22
23	0.29	0.13	0.26	0.47	0.22
24	0.29	0.13	0.41	0.03	0.72
25	0.01	0.13	0.41	0.03	0.72
26	0.01	0.02	0.16	0.27	1.12
27	0.09	0.02	0.16	0.27	1.02
28	0.09	0.02	0.09	0.52	0.42
29	0.06	0.43	0.09	0.52	0.82
30	0.06	0.33	0.36	0.23	0.12
31	0.26	0.18	0.21	0.23	1.22
32	0.09	0.18	0.39	0.77	0.28

Figure 3: Direct Method

No.	Pair1	Pair2	Pair3	Pair4	Pair5	Pair6	Pair7	Pair8	Pair9	Pair10
1	0.04		0.20	0.05						
2	0.14	0.15		0.45	0.35	0.05	0.05	0.17	0.06	0.38
3	0.14	0.10	0.25	0.05	0.05	0.03	0.03	0.03	0.16	0.28
4	0.01	0.25	0.15	0.15	0.05	0.03	0.23	0.23	0.08	0.13
5	0.04	0.15	0.15	0.45	0.05	0.07	0.17	0.17	0.04	0.18
6	0.09	0.25	0.25	0.10	0.01	0.07	0.03	0.03	0.04	0.08
7	0.14	0.05	0.05	0.05	0.05	0.13	0.13	0.13	0.21	0.08
8	0.04	0.25	0.35	0.65	0.05	0.07	0.03	0.03	0.04	0.02
9	0.06	0.05	0.15	0.45	0.05	0.13	0.23	0.23	0.04	0.08
10	0.09	0.25	0.25	0.05	0.25	0.23	0.03	0.03	0.11	0.13
11	0.04	0.10	0.15	0.15	0.10	0.13	0.13		0.06	0.28
12	0.04	0.10	0.10	0.15	0.05	0.08	0.03	0.03	0.06	0.13
13	0.01	0.05	0.00	0.05	0.10	0.02	0.12	0.12	0.04	0.12
14	0.06	0.00	0.05	0.55	0.10	0.23	0.13	0.13	0.06	0.07
15	0.14	0.15	0.15	0.35	0.05	0.03	0.13	0.13	0.06	0.22
16	0.24	0.05	0.05	0.25	0.15	0.07	0.03	0.03	0.06	0.02
17	0.24	0.05	0.15	0.25	0.15	0.07	0.17	0.17	0.06	0.18
18	0.14	0.05	0.15	0.45	0.15	0.07	0.07	0.07	0.01	0.08
19	0.04	0.25	0.15	0.45	0.15	0.02	0.13	0.13	0.16	0.08
20	0.14	0.25	0.15	1.25	0.05	0.02	0.22	0.22	0.06	0.08
21	0.04	0.15	0.05	0.35	0.05	0.12	0.03	0.03	0.01	0.08
22	0.24	0.20	0.25	0.05	0.00	0.47	0.17	0.17	0.16	0.08
23	0.24	0.10	0.25	0.25	0.10	0.27	0.07	0.07	0.24	0.42
24	0.09	0.35	0.45	0.45	0.05	0.02	0.37	0.37	0.16	0.48
25	0.06	0.45	0.05	0.35	0.25	0.47	0.67	0.67	0.04	0.28
26	0.34	0.05	0.65	0.40	0.10	0.32	0.67	0.67	0.04	0.03
27	0.14	0.15	0.85	0.15	0.15	0.23	0.47	0.47	0.21	0.03
28	0.14	0.55	0.20	0.25	0.05	0.17	0.13	0.13	0.01	0.22
29	0.01	0.25	0.25	0.05	0.00	0.33	0.77	0.77	0.11	0.43
30	0.19	0.30	0.05	0.05	0.00	0.57	0.27	0.27	0.06	0.43
31	0.24	0.25	0.05	0.55	0.25	0.07	0.22	0.22	0.06	0.38
32	0.09	0.55	0.95	0.30	0.05	0.17	0.03	0.03	0.74	0.08
										0.58
										0.16

Figure 4: Pairwise Comparisons Method