ISSN No. 0976-5697

Volume 6, No. 6, July - August 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Parametric Tuning of the Gielis Superformula for Non-Target Based Automated Evolution of 3D Printable Objects

Jia Hui Ong
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Sabah, Malaysia

Jason Teo
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Sabah, Malaysia

Abstract: 3D printing is an emerging trend fuelled by the rapid technology advancements in 3D printing technology. Printing out 3D designs is something new and interesting but the process of designing 3D objects is far from effortless. Researchers have recently forged ahead in conducting numerous studies on using mathematical formulas to create objects and shapes in 3D space. A mathematical encoding for geometric shapes called the Superformula was proposed by Johan Geilis through the generalization of the Supereclipse formula to generate 3D shapes and objects by extending its spherical products. The focus of this study is to investigate the ideal range of parametric values supplied to the Superformula in order to automatically generate 3D shapes and objects through the use of Evolution Algorithms (EAs). Thus, Evolutionary Programming was used as the EA in this study which serves as the main evolution component that uses a fitness function tailored in a way that it is able to evaluate the 3D objects and shapes generated by the Superformula. The values require by the Superformula to generate 3D objects or shapes are $\mathbf{m_1}$, $\mathbf{m_2}, \mathbf{n_{1,1}}, \mathbf{n_{1,2}}, \mathbf{n_{1,3}}, \mathbf{n_{2,1}}, \mathbf{n_{2,2}}$, **and** $\mathbf{n_{2,3}}$. To obtain the ideal range of values for the afore mentioned parameters, five different sets of experiments were carried out within the range set of {0 - 20}, {0 - 40}, {0 - 60}, {0 - 120}, and {0 - 240}.Each range set of numbers will be tested five times and the final objects from each of the runs were then analysed. From the observations obtained, the range set of {0- 20}, {0-60}, and {0- 120} shows the most promising results as the final objects produced were unique and it was surmised that within these range of numbers contain highly unique and novel 3D objects and shapes.

Keywords: Automatic 3D shape evolution, Parameter Tuning, Evolutionary art, Gielis Superformula, Evolutionary Programming (EP), Evolutionary Algorithm (EA), 3D printing

I. INTRODUCTION

Recent advancements in 3D printing machines have attracted significant numbers of everyday people to get involved in designing 3D objects and shapes, and it said to be the next big industrial revolution [12]. 3D printers enable the possibility of producing goods at a low cost in small quantities [12] even at the convenience of your own home. However, designing 3D objects is a complex and time consuming process that requires a combination of skills and hours of work to complete a simple design. Knowledge of how to use 3D-object design software is a must for those who intend to dive into 3D object printing. Even with the help of the available software, designing a 3D object is far from an effortless task even for the design of simple shapes, what more for complex 3D objects or shapes. This paves the way for numerous researchers to attempt generation of 2D and 3D objects through computational methods. Some of the early studies done on geometrical modelling evolution in a 3D space are polygonal sequencing operators [1] by McGuire and Exploration of the lattice deformation [2] by Watabe and Okino. More studies were carried out by using different encoding such as the work by Sims [3] using directed graph encoding in morphology and behaviour evolution of virtual creatures in a 3D environment. Jacob and Hushlak [4] displayed the used of L-system encodings for their work in creating virtual sculptures and furniture designs. Bentley [5] explore into the evolutionary variable and fixed length direct encoding on solid objects such as tables, cars, boat and even a the layout of a hospital department.

The Superquadrics equation in representing geometric shapes was introduced by Barr. It has been used as quantitative models for diverse applications in computer environments [6,7]such as computer graphics as well as in computer visions [8]. Since then, Superquadrics has been extended in local and global deformations to be able to model natural and considerable precision of synthetic shapes.

By generalizing the Superellipses and Superquadric formula, Gielis was able to come up with another equation which is the Superformula equation to describe shapes by its internal symmetry and internal metrics [9]. The Superformula equation is then further used to represent shapes in various fields such as engineering [10] and it has been used together with EA to achieve a certain target shapes [11].

EAs are inspired by natural selection of the fittest and it has been used as an optimization technique to solve engineering, mathematical, computational and many more complex problems. EAs main genetic operators comprise population, parent, recombination, mutation, offspring, and survivor selection. It has four different classes, which are Genetic Algorithms, Evolutionary Programming (EP), Evolution Strategies (ES), and Genetic programming [14]. Each class utilizes different approaches in solving complex problem while maintaining the main genetic operators.

In this paper, we introduce the approach of using Superformula to create non-target based 3D shapes through EP. The focus of this study is to investigate the ideal range of parametric value to supply to the Superformula in order to automatically generate 3D shapes and objects through EP. The results from the investigations can be used to determine the suitable range of numbers to be used as parameters in Superformula. By finding these range of numbers, the search space for novel and unique 3D objects and shapes can be narrowed down.

The next section of this paper will discuss the background of the Superformula and how it is able to generate 3D shapes from generalizing from Superellipses and Superquadric equations. It will be followed by another section on the flow of Evolutionary Programming. The experimental setup section will be presented next and followed by the results section. The last section will be the conclusion and future work.

II. METHOD

A. *Superformula:*

The Superformula is simple geometric equation form from generalisation of a hyper-ellipse. It was found to be able to model forms of a large variety of plants and other living organisms [13]. The generalization of Superellipse equation is as follows:

$$r(\theta) = \frac{1}{\sqrt[n_1]{\left[\left(\left|\frac{1}{a}.\cos\left(\frac{m}{4}.\theta\right)\right|\right)^{n_2}\right]+\left[\left(\left|\frac{1}{b}.\sin\left(\frac{m}{4}.\theta\right)\right|\right)^{n_3}\right]}} \quad (1)$$

The distance in polar coordinates is denoted by r, for n_i and m$\in R^{+}$; a, b$\in R_0^{+}$; a > 0, b > 0 are responsible for the size of the supershapes with the usual value of equals to one. Symmetry number is control by m while the shape coefficients are control by n1, n2 and n3 with real valued parameters. From equation (1) forms the superecllipse 2D shapes henceby multiplying 2 superecllipse equations together it allows the extention towards 3D shapes:

$$x = r_1(\theta)\ .\ \cos\theta\,.\,r_2(\varphi)\,.\cos(\varphi) \quad (2)$$
$$y = r_1(\theta)\ .\ \sin(\theta).\ r_2(\varphi)\,.\cos(\varphi) \quad (3)$$
$$z = r_2(\varphi)\,.\sin(\varphi) \quad (4)$$

θ, denotes longitude with $-\pi \leq \theta \leq \pi$,
φ, denotes latitude with $-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}$

As such, more complex 3D shapes can be generated. Preen [10] has shown more complex shapes such as the Mobius strip, shell and even torus shapes can be generated with the Superformula.

B. *Evolutionary Programming:*

Evolutionary Programming (EP) serves as the EA method in this study. EP is one of the four major EA methods. It was first introduced by Fogel [15] to simulate learning processing aiming to generate artificial intelligence. Adaptive behavior is the key to EP and by using real-value parameters it can be integrated to the problem domain. The real-value parameters of Superformula are used as the representation in EP for this study. Below is the pseudocode for EP in this study:

a. Generate initial population
b. Test each individual solution in the population
c. Parent selection
d. Mutation process
e. Offspring generation
f. Repeat step 2 to 5 until reach termination criteria

C. *Evaluation Function:*

Evaluation Function serves as a representation of requirement for a solution to adapt to. It is the basis of selection to aid improvements of the individual solution. From the perspective of problem-solving, it is the representation to the task to be solved in evolutionary background [14]. Basically it serves as a quality measurement of the individual solution presented in the population pool. In this study, the evaluation function is design to calculate the value obtain from the 3D object as well as from the Superformula.

$$\frac{\left((200000-\hat{V})+\sigma_x+\sigma_y+\sigma_z\right)^{-\sqrt{\left(n_{2,2}-n_{2,3}\right)^2}}}{m_1+m_2+1} \quad (5)$$

In equation (5), it was intended to find the spread of point x, y, and z over the symmetry number of any given object. A penalty will be imposed to the score if the dimension of the objects were too big and out of the boundary set. The reason for the penalty imposed is to maintain a reasonable dimension size. The values for m_1 and m_2 are responsible to the symmetry of the 3D object, both the values of m_1 and m_2 are added together with a constant of 1. The constant is used to counter the division by zero error in case the addition between m_1 and m_2 results in zero. Another penalty are in pose by using the power to the difference of $n_{2,2}$and $n_{2,3}$. In Superformula, the value of $n_{2,2}$ and $n_{2,3}$ is to control the thickness of each of the layers generated and with this penalty, thin layers or structure to the 3D objects can be avoided and printed out successfully without deformation.

III. EXPERIMENTAL SETUP

The population size model used is μ+λ with both parameters set to a size of 1 and 100 respectively which means the population size model will include the parent plus 100 offspring. Each individual in the population pool will be evaluated using the fitness function in equation (5) and hence the fittest individuals will be selected to seed the next generations. There will be five different sets of number range {0-20}, {0- 40}, {0- 60}, {0- 120}, and {0-240} each range of random numbers will be run five times and the final objects of each runs were then observed. Although the range sets of numbers overlap each time the upper bound is increased, as our results will show, there appears to be some so-called “sweet spots” of number ranges and others that are much less ideal. Hence, running a single experiment that covers the entire range of overlapping number ranges would not be able to identify these “sweet spots”. The number of generations set for this experiment is 10. Object evolved are first save into Autocad file format (.dxf) and later convert into a STereoLithography (.stl) format. With .stl format the object are then brought into the UP! Print preview as shown in Fig 1.

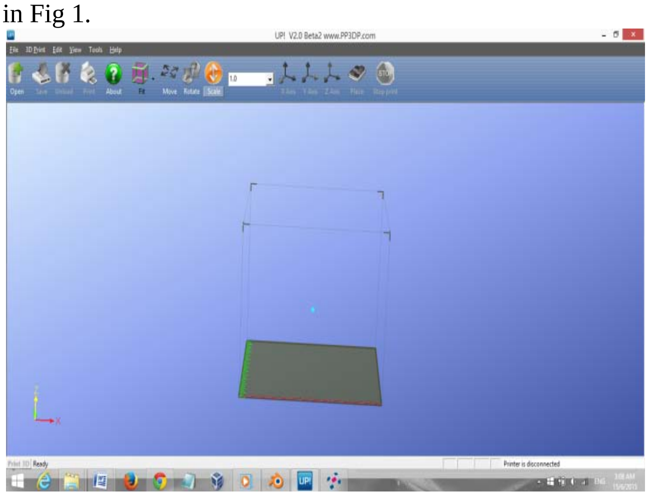


Figure 1. UP! 3D Printer interface

IV. EXPERIMENTAL RESULTS

Table I. Results Obtained from {0,20}

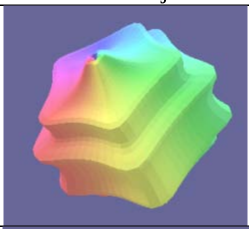
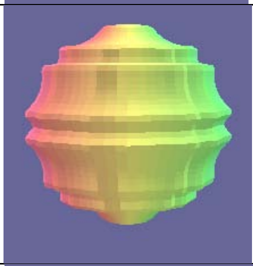
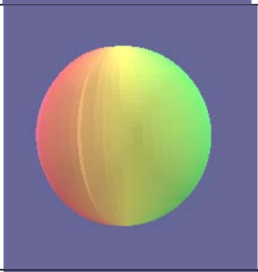
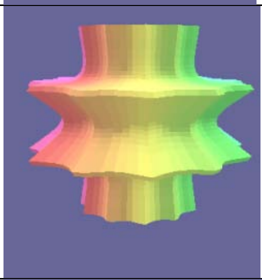
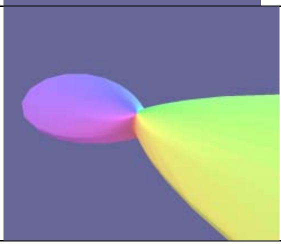
Run	Parameters	Evolved 3D object
1	m_1 =5 , m_2= 11 , $n_{1,1}$ =7.1, $n_{1,2}$ =8.3, $n_{1,3}$ =6.1, $n_{2,1}$ =12.8, $n_{2,2}$ =4.9, $n_{2,3}$=17.5	
2	m_1 =14 , m_2=17 , $n_{1,1}$ =10.6, $n_{1,2}$ =1.2, $n_{1,3}$ =15.2, $n_{2,1}$ =8.9, $n_{2,2}$ =15.7, $n_{2,3}$=1.6	
3	m_1 =5, m_2 =3, $n_{1,1}$ =4.2, $n_{1,2}$ =9.5, $n_{1,3}$ =0.2, $n_{2,1}$ =16.0, $n_{2,2}$ =1.5, $n_{2,3}$=12.0	
4	m_1 =11 , m_2=9 , $n_{1,1}$ =19.9, $n_{1,2}$ =11.5, $n_{1,3}$ =11.2, $n_{2,1}$ =12.0, $n_{2,2}$ =18.7, $n_{2,3}$=15.7	
5	m_1 = 2 , m_2=8 , $n_{1,1}$ =0.5, $n_{1,2}$ =6.7, $n_{1,3}$ =6.8, $n_{2,1}$ =18.3, $n_{2,2}$ =10.0, $n_{2,3}$=2.3	

Table II. Results Obtained from {0,40}

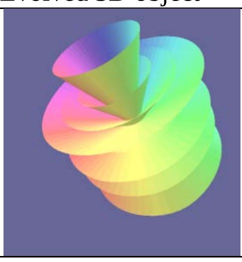
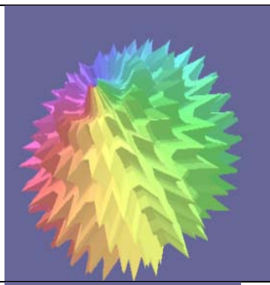
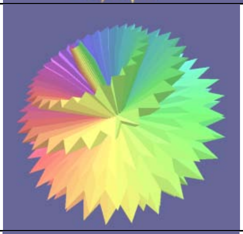
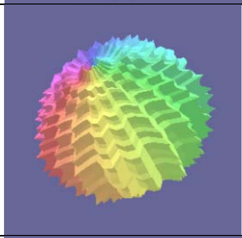
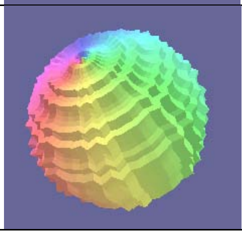
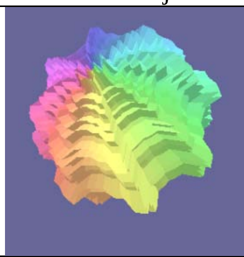
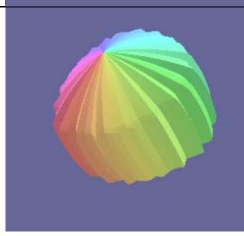
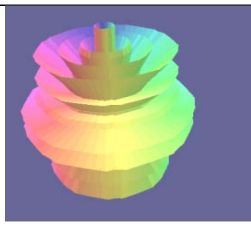
Run	Parameters	Evolved 3D object
1	m_1 =2 , m_2= 29 , $n_{1,1}$ =36.2, $n_{1,2}$ =8.1, $n_{1,3}$ =13.1, $n_{2,1}$=2.4, $n_{2,2}$=6.1, $n_{2,3}$=38.0	
2	m_1 =36 , m_2=19 , $n_{1,1}$ =22.8, $n_{1,2}$ =38.5, $n_{1,3}$ =38.2, $n_{2,1}$ =29.7, $n_{2,2}$ =15.8, $n_{2,3}$=29.2	
3	m_1 =30 , m_2=15 , $n_{1,1}$ =22.4, $n_{1,2}$ =11.3, $n_{1,3}$ =19.5, $n_{2,1}$ =4.5, $n_{2,2}$ =37.5, $n_{2,3}$=16.5	
4	m_1 =19 , m_2=33 , $n_{1,1}$ =15.2, $n_{1,2}$ =15.7, $n_{1,3}$ =11.3, $n_{2,1}$ =28.9, $n_{2,2}$ =27.1, $n_{2,3}$=5.0	
5	m_1 =27 , m_2=29 , $n_{1,1}$ =35.0, $n_{1,2}$ =7.3, $n_{1,3}$ =17.4, $n_{2,1}$ =37.5, $n_{2,2}$ =5.0, $n_{2,3}$=29.9	

Table III. Results Obtained from {0,60}

Run	Parameters	Evolved 3D object
1	m_1 =52, m_2=43 , $n_{1,1}$ =40.0, $n_{1,2}$ =57.7, $n_{1,3}$ =25.6, $n_{2,1}$ =21.6, $n_{2,2}$ =35.9, $n_{2,3}$=7.5	
2	m_1 =38, m_2= 6 , $n_{1,1}$ =53.2, $n_{1,2}$ =58.3, $n_{1,3}$ =15.3, $n_{2,1}$ =59.9, $n_{2,2}$ =15.7, $n_{2,3}$=14.9	
3	m_1 =18, m_2=28 , $n_{1,1}$ =38.1, $n_{1,2}$ =55.3, $n_{1,3}$ =1.2, $n_{2,1}$ =27.5, $n_{2,2}$ =50.0, $n_{2,3}$=50.9	

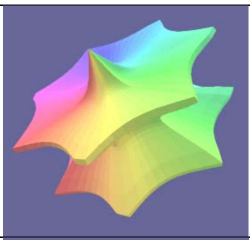
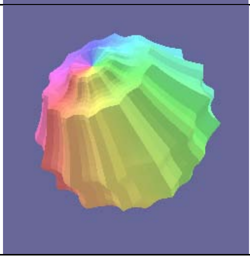
4	m_1 =7, m_2 =6 , $n_{1,1}$ =30.9, $n_{1,2}$ = 27.0, $n_{1,3}$ =24.5, $n_{2,1}$ =4.6, $n_{2,2}$ = 24.5, $n_{2,3}$=7.0	
5	m_1 = 47, m_2=11 , $n_{1,1}$= 35.4, $n_{1,2}$ = 17.2, $n_{1,3}$ = 11.8, $n_{2,1}$ =28.6, $n_{2,2}$ =22.0, $n_{2,3}$=5.1	

Table IV. Results Obtained from {0,120}

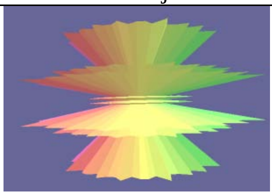
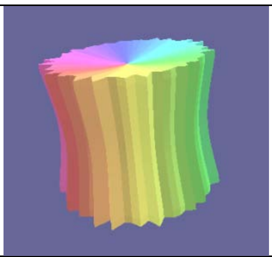
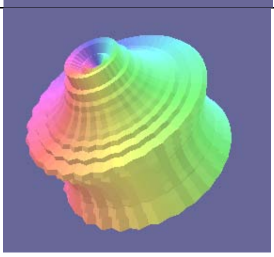
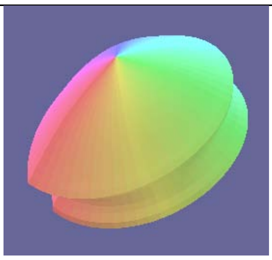
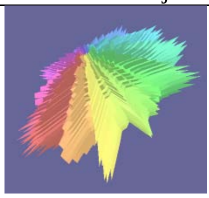
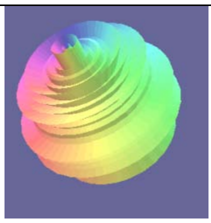
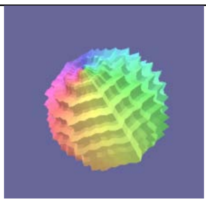
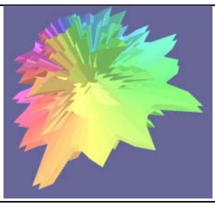
Run	Parameters	Evolved 3D object
1	m_1 =23 , m_2=44 , $n_{1,1}$ =79.8, $n_{1,2}$ =50.4, $n_{1,3}$ =83.2, $n_{2,1}$ =14.3, $n_{2,2}$ =57.2, $n_{2,3}$=70.5	
2	m_1 =25 , m_2=4 , $n_{1,1}$ =111.2, $n_{1,2}$ =98.2, $n_{1,3}$ =56.6, $n_{2,1}$ =67.1, $n_{2,2}$ =93.4, $n_{2,3}$=91.3	
3	m_1 =53 , m_2=120 , $n_{1,1}$ =24.2, $n_{1,2}$ =44.4, $n_{1,3}$ =21.5, $n_{2,1}$ =63.6, $n_{2,2}$ =0.3, $n_{2,3}$=90.2	
4	m_1 =61 , m_2=113 , $n_{1,1}$ =107.8, $n_{1,2}$ =26.1, $n_{1,3}$ =47.2, $n_{2,1}$ =90.5, $n_{2,2}$ =93.7, $n_{2,3}$=59.4	
5	m_1 =119 , m_2=6 , $n_{1,1}$ =33.2, $n_{1,2}$ =37.1, $n_{1,3}$ =83.4, $n_{2,1}$ =28.9, $n_{2,2}$ =96.5, $n_{2,3}$=6.2	

Table V. Results Obtained from {0,240}

Run	Parameters	Evolved 3D object
1	m_1 = 49 , m_2=182 , $n_{1,1}$ =22.6, $n_{1,2}$ =140.1, $n_{1,3}$ =24.0, $n_{2,1}$ =63.2, $n_{2,2}$ =76.0, $n_{2,3}$=53.6	
2	m_1 =177 , m_2 =176 , $n_{1,1}$ =106.7, $n_{1,2}$ =2.7, $n_{1,3}$ =16.5, $n_{2,1}$ =145.7, $n_{2,2}$ =220.3, $n_{2,3}$=130.0	
3	m_1 =110 , m_2 =189 , $n_{1,1}$ =34.7, $n_{1,2}$ =62.8, $n_{1,3}$ =63.8, $n_{2,1}$ =6.5, $n_{2,2}$ =36.4, $n_{2,3}$=69.2	
4	m_1 =189 , m_2 =217 , $n_{1,1}$ =109.8, $n_{1,2}$ =155.9, $n_{1,3}$ =27.7, $n_{2,1}$=176.0, $n_{2,2}$ =77.1, $n_{2,3}$=88.6	
5	m_1 =46 , m_2=67 , $n_{1,1}$=117.2, $n_{1,2}$ =214.1, $n_{1,3}$ =97.6, $n_{2,1}$ =40.2, $n_{2,2}$=105.6, $n_{2,3}$= 157.8	

From the results obtained, number range of {0- 20} , {0-60} and {0 -120} shows a diversity of 3D object evolved. While number range of {0-40} and {0-240} did not manage to evolve shapes that look different and unique but rather most of the time the objects evolved are in a spiky shapes. In Table 1 the parameters value for m_1 and m_2 maintain in a lower region and it is observed that with a lower value ofm_1 and m_2, shapes with less symmetrical points is evolved. The results from such a low symmetrical points produce two rather unique shapes in Table 1 object no.3 and no.5. But for object no.5, when it is brought into the 3D printing software the object could not be printed out. This might due to the reason that the object itself were rather too big and becomes incomplete due to constrain of the 3D environments. The fitness function used was supposed to impose a penalty to such a large object generated, but in this case it is observed that although the objects were large but the thickness of the entire object is rather thin hence the object only appear to be long instead of big in size. Object no.3 was printed out successfully and shown in Fig 2. In Table 5, the results from number range {0-240} shows more objects evolved into a spiky shape. From the parameter values shown in Table 5, when the valuefor m_1 and m_2 exceed the 120 mark, spiky shapes will appear and it was deduced that after the number range exceeded 120, the shapes will be in a low quality region. Some of the handpicked 3D objects were printed out and shown in Fig 3 to 6.

Figure 2: 3D object printed out from {0 – 20} run no.3

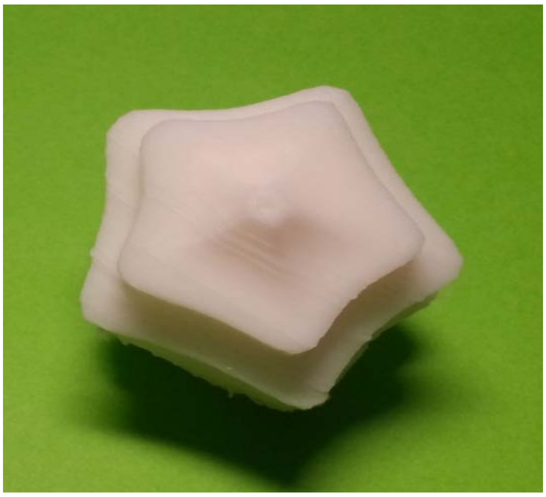


Figure 3: 3D object printed out from {0 – 20} run no.1

Figure 3: 3D object printed out from {0 – 40} run no.2

Figure 4 : 3D object printed out from {0 – 120} run no.3

Figure 5: 3D object printed out from {0 – 60} run no.2

Figure 6: 3D object printed out from {0 – 60} run no.4

From Table 3, the 3D objects shown were rather diversified, all 5 objects had different and unique symmetrical points. These observations can be used to conclude that the range number from 0 to 60 contains a high probability of generating unique and novel 3D objects.

V. CONCLUSION AND FUTURE WORK

From this study, the ideal range of numbers for the parameter value of the Superformula was investigated. Obtaining these values will be informative to serve as a reference point for future studies on the Superformula for automated 3D object and shape evolution. These findings will assist in decreasing the time needed and reduce the chance of devolving into a bad region of the parametric space.

Future work should be focused on finding the settings of the EA operators that works well for the Superformula. Other types of evolutionary algorithms could also be investigated for more diverse shape generations.

VI. ACKNOWLEDGMENT

This research is funded by the ERGS Research project ERGS0043-ICT-1/2013granted by the Ministry of Education, Malaysia.

VII. REFERENCES

[1] F. McGuire, "The origins of sculpture: Evolutionary 3D design". IEEE Computer Graphics Applications 13, 9–11 (1993)

[2] H. Watabe, and N. Okino, "A study on genetic shape design". In: Proceedings of the 5th International Conference on Genetic Algorithms. pp. 445–451. ICGA '93, Morgan Kaufmann Publishers Inc. (1993)

[3] K. Sims, "Evolving 3D morphology and behavior by competition. Artificial Life" 1, 353–372 (1994)

[4] C.J. Jacob, and G. Hushlak "Evolutionary and swarm design in science, art, and music". In: The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, pp. 145–166. Springer (2007)

[5] P. Bentley, "Generic Evolutionary Design of Solid Objects using a Genetic Algorithm". Ph.D. thesis, University of Huddersfield (1996)

[6] A.H. Barr, "Superquadrics and angle preserving transformations". IEEE Computer graphics applications 1: 11-23, (1981).

[7] A.H. Barr, "Global and local deformations of solid primitives". Computer Graphics, 18(3): 21-30, (1984).

[8] A. Jaklic, and F. Solina, "Superellipsoids gaining momentum". In: Proceedings of 7th Computer Vision Winter Workshop CVWW'02,, Bad Aussee (2002)

[9] J. Gielis, B. Beirinckx, and E. Bastiaens, " Superquadrics with rational and irrational symmetry". In: Proceedings of the eighth ACM symposium on Solid modeling and applications. pp. 262–265. SM '03, ACM (2003)

[10] R.J. Preen, L. Bull, "Evolution of Supershapes for the Generation of Three-Dimensional Designs". In Evolutionary Computing 20 Symposium: Proceedings of the 50th Anniversary Convention of the AISB 2014, Goldsmiths, University of London, pages 15-22.The Society for the Study of Artificial Intelligence and Simulation of Behaviour (2012).

[11] R.J. Preen, and L. Bull "Towards the evolution of vertical-axis wind turbines using supershapes", Proc. 50th Anniv. Conv. AISB Evol.Comput. 20 Symp., pp.15 -22 (2014)

[12] B. Barry, "3-D printing: The new industrial revolution". In: Business Horizons 55, 155—162(2012)

[13] J. Gielis, "A generic geometric transformation that unifies a wide range of natural and abstract shapes". American Journal of Botany 90, 333–338 (2003)

[14] A.E.Eiben, and J.E.Smith.Introduction to Evolutionary Computing. (2003).

[15] D.B. Fogel, and L.J. Fogel "An introduction to evolutionary programming". Artif.Evol., 1063: 21-33 (1996)

