

# The Menger Sponge - A first exploration into 3D generated form

Richard Kennedy  
Technical University of Eindhoven  
Department of Industrial Design  
Building 32, Laplace  
5612 AZ , Eindhoven, The Netherlands  
E-mail: r.w.r.kennedy@student.tue.nl

## Abstract

In this article I describe programming a three-dimensional fractal structure as a first venture into formulaically generated forms, and the physical realisations of those structures with stereolithographic 3D printing. A Menger Sponge is generated and the implementation of the algorithm is discussed from the perspective of a non-experienced programmer. Converting the virtual model into a physical model is achieved by writing a custom stl file from within the program. The stl file was then printed on a high-quality stereolithographic printer. One model was sliced to illustrate the interior detail. The characteristics of the Menger Sponge are briefly discussed and some comparison of this technique with hand-crafting is made, a typical element of a TU/e Industrial Design student's Design and Research Process.

## Acknowledgements

Thanks are extended to Loe Feijs & Jun Hu of the Designed Intelligence Group, TU/e Industrial Design (ID) and especially Chet Bangaru, Jasper Sterk, Robin van Helmond and Jan Rouvroye of the D.Search Lab, TU/e ID, without who's expertise and patient help an artefact would have never materialised.

## Introduction

Inspired by Donald Schön's 1983 text: 'The Reflective Practitioner' [1], students of the ID department at the TU/e are encouraged to take a 'thinking-with-the-hands' approach to physically realising and validating design concepts. On the department's course: 'The Golden Ratio' an alternate mathematically descriptive approach is presented. Physical artefacts are produced by various CNC tools, the data to those being supplied by bespoke computer code written by the student. Participating in this course I selected a Menger Sponge as a means of exploration and to contrast the typical hand-crafted approach. Especially motivating was the self-referential quality of fractals which is also observable in nature's own 'design work'.

## The Menger Sponge

First described by the Austrian-American mathematician Karl Menger in 1926 while researching the concept of topological dimension [2][3]. A three dimensional generalisation of the Sierpinski carpet. The sponge is generated with a simple algorithm, most conveniently listed on Wikipedia [4]:-

- 1 Begin with a cube.
- 2 Sub-divide the cube into 27 equal smaller cubes.
- 3 Remove the smaller cube in the middle of each face, and remove the smaller cube in the centre of the larger cube, leaving 20 smaller cubes.
- 4 Repeat steps 2 and 3 for each of the remaining smaller cubes.

Each iteration of these steps gives a new 'level' of Menger Sponge. The number of cubes grows in a  $20^n$  arrangement where n is the number of levels. Thus at level 4 there are 160,000 cubes and at level 6

there are 64,000,000. Even at only level 4 (using a brute force approach) there are 1,280,000 vertices, 960,000 faces and 1,920,000 triangles!

The Menger Sponge has some remarkable qualities. It has infinite surface area and an infinitesimal volume. The fractal dimension of the Menger Sponge is 2.72683302786084 because each cube is subdivided into 27 with 7 removed leaving 20 new ones, one third of the dimension of the original. Hence:-

$$\frac{\log(20)}{\log(3)}$$

## Implementation

An encoding of this algorithm was made using the C programming language within Microsoft's MSDEV 6.0 Integrated Development Environment (IDE). Principally this choice was due to some basic familiarity with C and the fact that it incorporates a debugger, an invaluable tool for the non-confident programmer! OpenGL was employed to display the cubes in wireframe form to visually inspect the progress of the work. Algorithmically the Menger Sponge lends itself readily to nested for loops. A triplet of for loops, one for each of the x, y and z directions was employed to 'walk' through the 27 smaller cubes made out of each 'big' cube starting point.

Many iterations were taken regarding the data structures used to represent the cubes. As work progressed the data became more abstracted until the notion of a cube was simply 8 points in space, each point represented by a triplet of floating point numbers (one for each axis). From these 8 points 12 triangles or facets could be derived and a normal for each face determined. For each of the 27 smaller cubes in each iteration the 216 points needed (8 x 27) share so many that in fact only 64 unique points are needed. The spatial points are therefore shared between many cubes.

Dividing any cube into the 27 smaller ones was achieved by observing that calculating the distance across the corners and then dividing that dimension into thirds could be combined with the triple-nested for loops. Thus each step walked x, y and z *amounts* of this dimension. The following vector calculus was employed to determine this dimension. If the rearmost, bottommost, rightmost corner of the cube is taken as a vector A, and the frontmost, topmost, leftmost corner as a vector B, then the one third division of this diagonal can be calculated with the following:

$$V = x * (B - A) / |B - A| + A$$

Where V is the vector we are looking for, A is the starting point and B the end point. x is the amount we want to move from the one, to the other (one third, two thirds etc). This technique is called Lerp, which is short for Linear Interpolation by Distance. Informally you take two points in space and move a certain distance along this connecting line, finding a new, third point in space.

Increased levels of course required a considerably longer time to calculate and needed a much larger amount of memory. At level 4 the IDE had to be coerced into obtaining more than its normal fair share of computer memory from the heap. In addition to memory and processing limits physical modelling has its own restrictions such as the minimum size the CNC machines can produce. The minimum dots per inch (DPI) level of detail in the case of 3D printers.

## Moving to the physical

At this stage a program had been written to essentially generate a huge number of points in space and display those graphically on screen via OpenGL in wireframe triangle form. In order to grasp a physical Menger Sponge this data had to be meaningfully supplied to the department's Objet stereolithographic 3D printer. Using very old-fashioned C file writing code, the data was meaningfully written into the standard currency of the 3D printer: the stl file. This consists of an enormous list of formatted facets, each one

consisting of a single triangle - defined by three vertices and a normal. The normal being a vector indicating what constitutes the outward face of the facet. A function was developed to parse through the large set of cubes and write those to a text file in the correct format. The stl file was first opened in a 3rd party program called MeshLab (V1.2.2), which is very useful for examining one's 3D model. MeshLab reported 282,201 vertices and 1,824,240 faces for a level 4 model. Both a level 3 and level 4 model was generated. Staff members of the D.Search Lab used special software to slice the level 4 STL model to reveal the interior detail and these models were duly submitted to the department's Object 3D printer.

## Discussion

This project was executed in only ten days with an emphasis on learning about the descriptive, mathematic approach to realising physical artefacts. No reference to anyone else's implementations of the Menger Sponge were made as this exercise was primarily a learning exercise. Unanticipated problems were the most interesting learning moments. The solution was therefore by no means optimal. With an initial fair self-evaluation of a mediocre level of programming skills this endeavour alone improved my programming ability and confidence considerably. Understanding the relationship between algorithms and data structures became especially apparent.

Experiencing these issues for one class of fractals would doubtless be useful when exploring others. For instance the four-step Wikipedia procedure fails to mention an important issue: when making a cube into 20 sub-cubes (27 minus the 7 you discard) you also actually need to remove the bigger cube you began with. This is not a trivial issue if you have used a static list to contain all of your cubes. This in itself a workaround solution to the large memory footprint required.

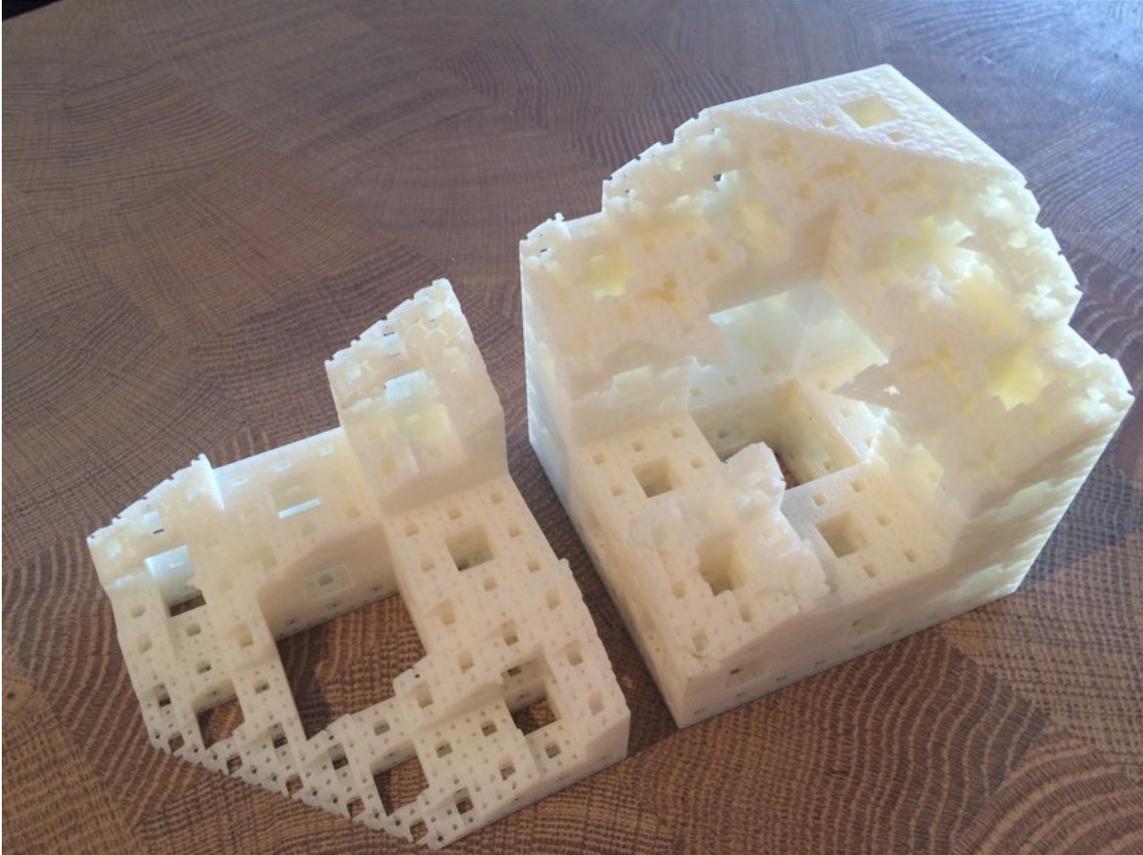
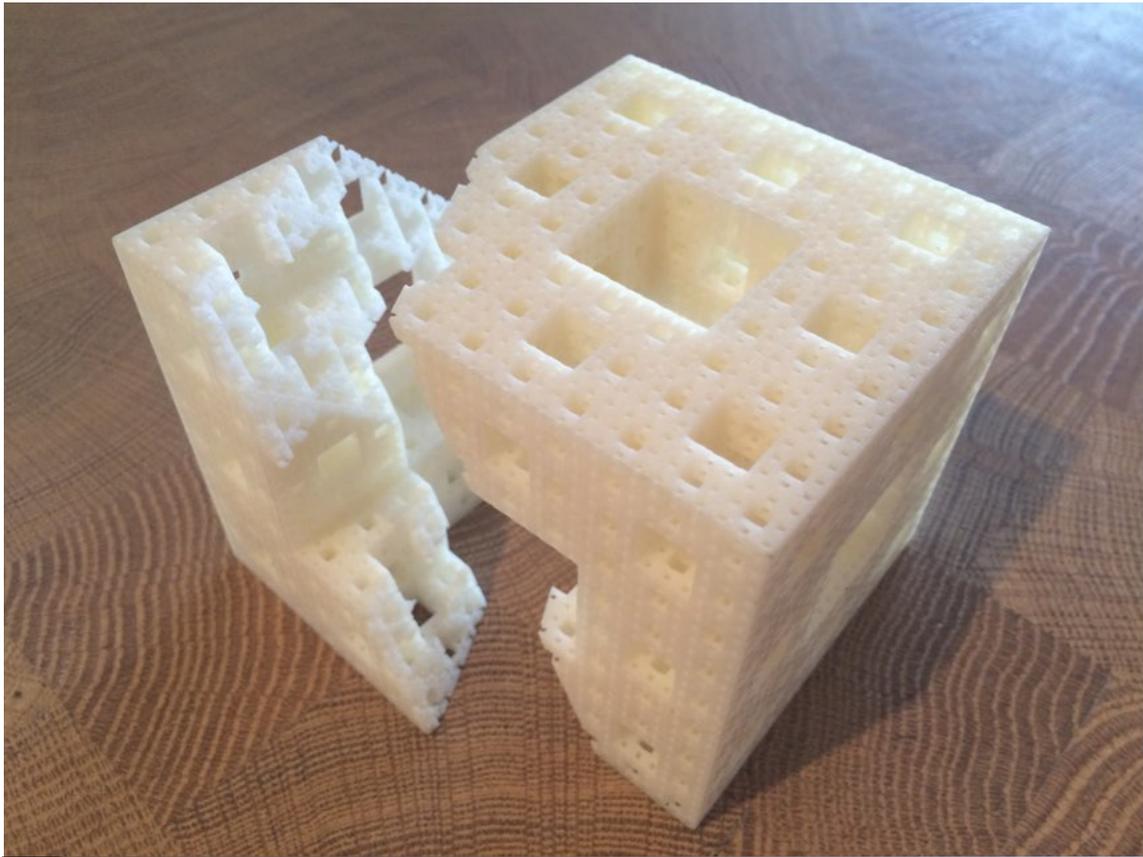
On a higher level of reflection, one learns to let the computer do as much of the work as possible. Throughout the project the numerous rewrites resulted in less and less actual code, making the code continually clearer. A general movement towards more abstracted data structures was noted and a more harmonious blending between algorithms and data structures occurred. The use of OpenGL allowed the data to be visualised very quickly and there was an awful lot of data, generated by contrastingly small blocks of code.

It is marvellous to physically experience an artefact that has only existed 'virtually' for the majority of its creative phase. Especially the level 4 Menger Sponge with the slice has received a lot of attention as the internal detail - which is normally concealed behind a facade of regularity - is revealed. Additionally the resin is slightly translucent so the detail is illuminated differently according to the density which varies considerably from area to area.

I see no conflict between a descriptive, mathematical approach and a hand crafted approach. Each has its own advantages and limitations. In fact it might be interesting to pursue a combination of the two. Programming and CNC manufacture can also arguably be considered craftsmanship in its own right, with as much scope for expressivity. The work was rewarding and interesting. Thus enthused, further exploration into fractals will follow.

## References

- [1] Schön, Donald A. (1983). *The reflective practitioner: how professionals think in action*. New York: Basic Books. ISBN 046506874X. OCLC 8709452.
- [2] Menger, Karl (1926), "Allgemeine Räume und Cartesische Räume. I.", *Communications to the Amsterdam Academy of Sciences*. English translation reprinted in Edgar, Gerald A., ed. (2004), *Classics on fractals, Studies in Nonlinearity*, Westview Press. Advanced Book Program, Boulder, CO, ISBN 978-0-8133-4153-8, MR 2049443
- [3] Menger, Karl (1928), *Dimensionstheorie*, B.G Teubner Publishers
- [4] Wikipedia, *Menger Sponge*, [https://en.wikipedia.org/wiki/Menger\\_sponge](https://en.wikipedia.org/wiki/Menger_sponge). Last accessed 22nd April 2016.



**Images 1 & 2:** Level 4 Menger Sponge with slice printed on the Object stereolithographic 3D printer.