Computational Design

Paul Bourke

paul.bourke@gmail.com

iVEC @UWA

Outline

- Introduction to iVEC
- (Science) Visualisation
- Serial vs parallel computing
- Euclidean geometry. Equation vs algorithm
  - Supershape, parametric equations
  - Space filling algorithm
  - dForm surfaces, form determined by simulation
  - Constructive Solid Geometry (CSG)
- Fractal geometry, modelling natural forms
  - Diffusion limited aggregation
  - Landscape and planet modelling
  - L-Systems
  - 3D replacement rules
  - Tiling and texture synthesis
- Newish Technologies
  - Rapid prototyping, tactile visualisation
  - 3D reconstruction, creating 3D textured models from photographs

Introduction to iVEC

- A joint venture between the 5 research institutions in Western Australia.
- The 4 public Universities in WA and CSIRO.
- Provides advanced computing to researchers in the state.
  - Supercomputing
  - Storage
  - Visualisation
  - Facilitates collaboration between partners
  - Comprises of staff with wide range of expertise
- Partners subscribe to be members of iVEC => no end user charges.
- Three teams
  - Supercomputing technologies
  - Data storage and management
  - Visualisation
- I am the director of the iVEC facility here at the University.

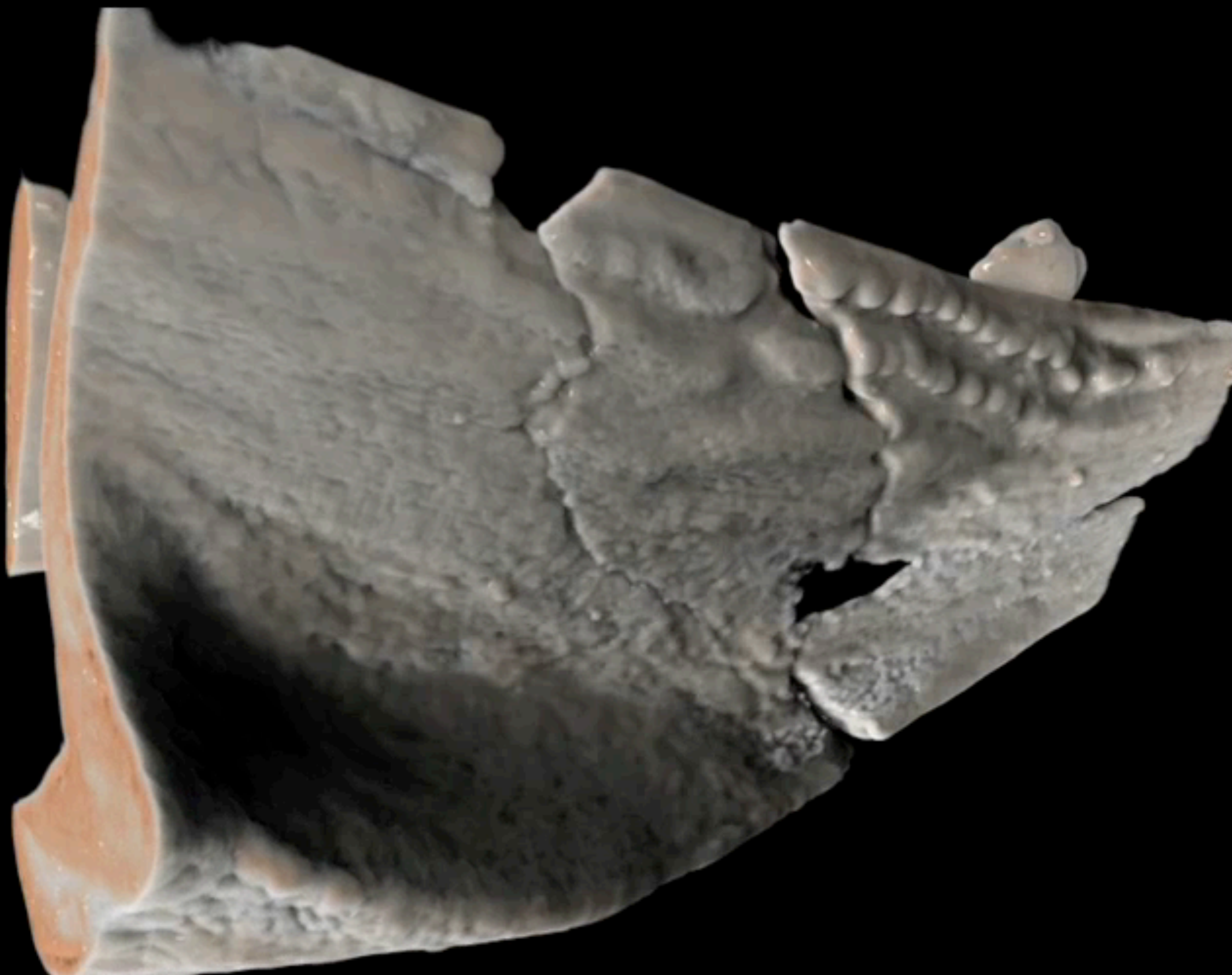
Visualisation

- Term used very broadly ... can mean different things to different people.
- My definition: Visualisation is the process of applying advanced computing techniques to data in order to provide insight into the underlying structures, relationships and processes.
- “Turning data into images and animations to assist researchers”.
- Wide range of sources of data: observational, simulation, theoretical.
- Techniques that find application across a wide range of disciplines.
- Often employs novel capture methodologies, display technologies and user interfaces.
- Frequently requires high performance computing and sophisticated algorithms.
- Outcomes
  - Revealing something new within datasets.
  - Finding errors within datasets.
  - Communicating to peers.
  - Communicating to the general public.
- Technologies
  Display hardware: stereo3D, immersion, fidelity.
  Algorithms: graphics, computer science, mathematics.
  Techniques: volume visualisation, stereographics, haptics.

Visualisation laboratory @ UWA

- By definition visualisation often targets/leverages the characteristics of the human visual system
  - depth perception
  - peripheral vision
  - visual fidelity

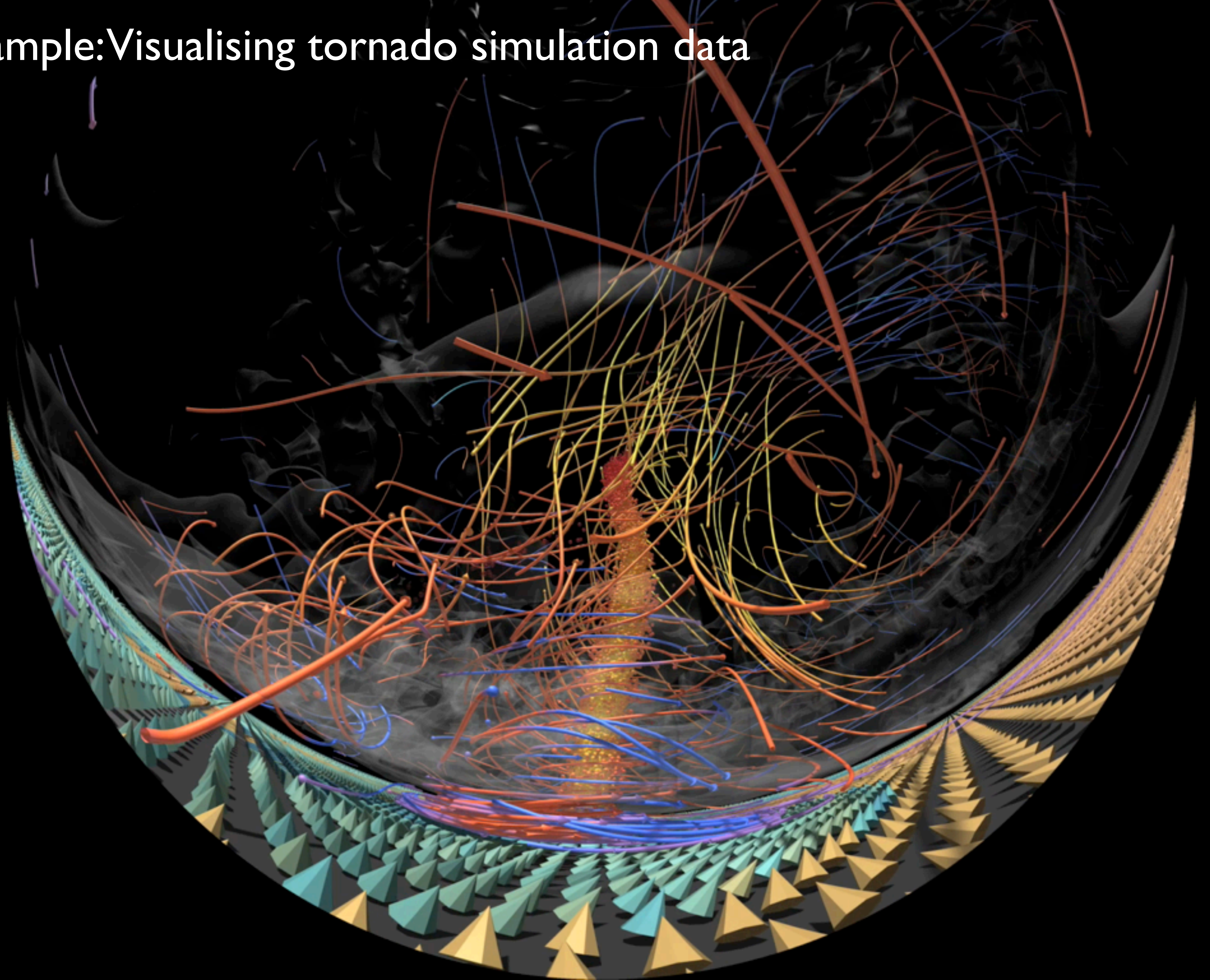
Example: Jaw of placoderm fish - CT scan



Rows of raised cusps are visible along the jaw bone making up the back of the placoderm pharynx.

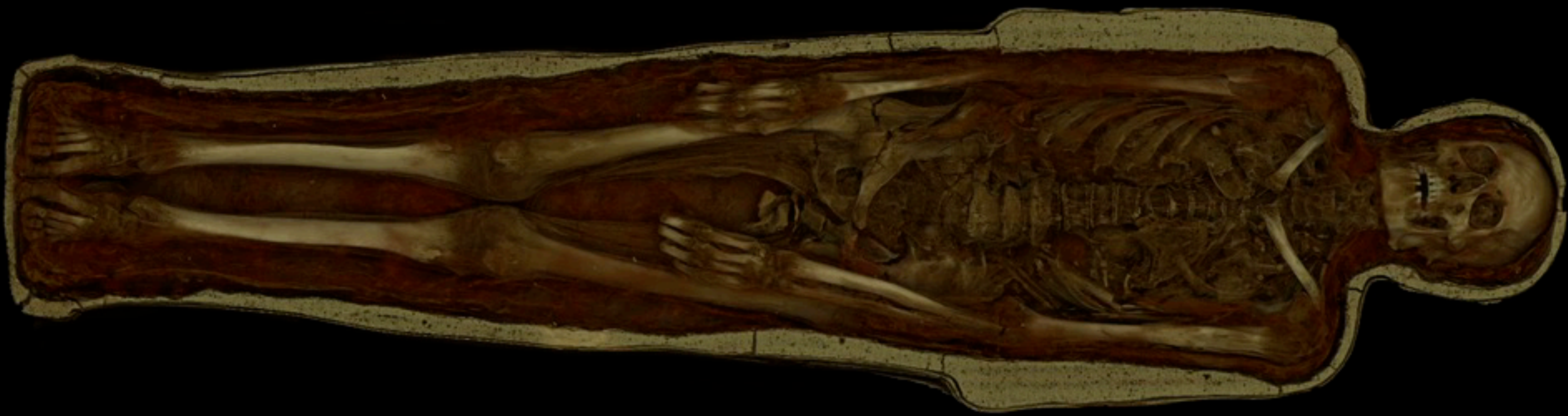
Example: Rabbits liver - Cancer research

Example: Visualising tornado simulation data



Example: Molecular motors

Example: Pausiris - MONA Museum exhibition

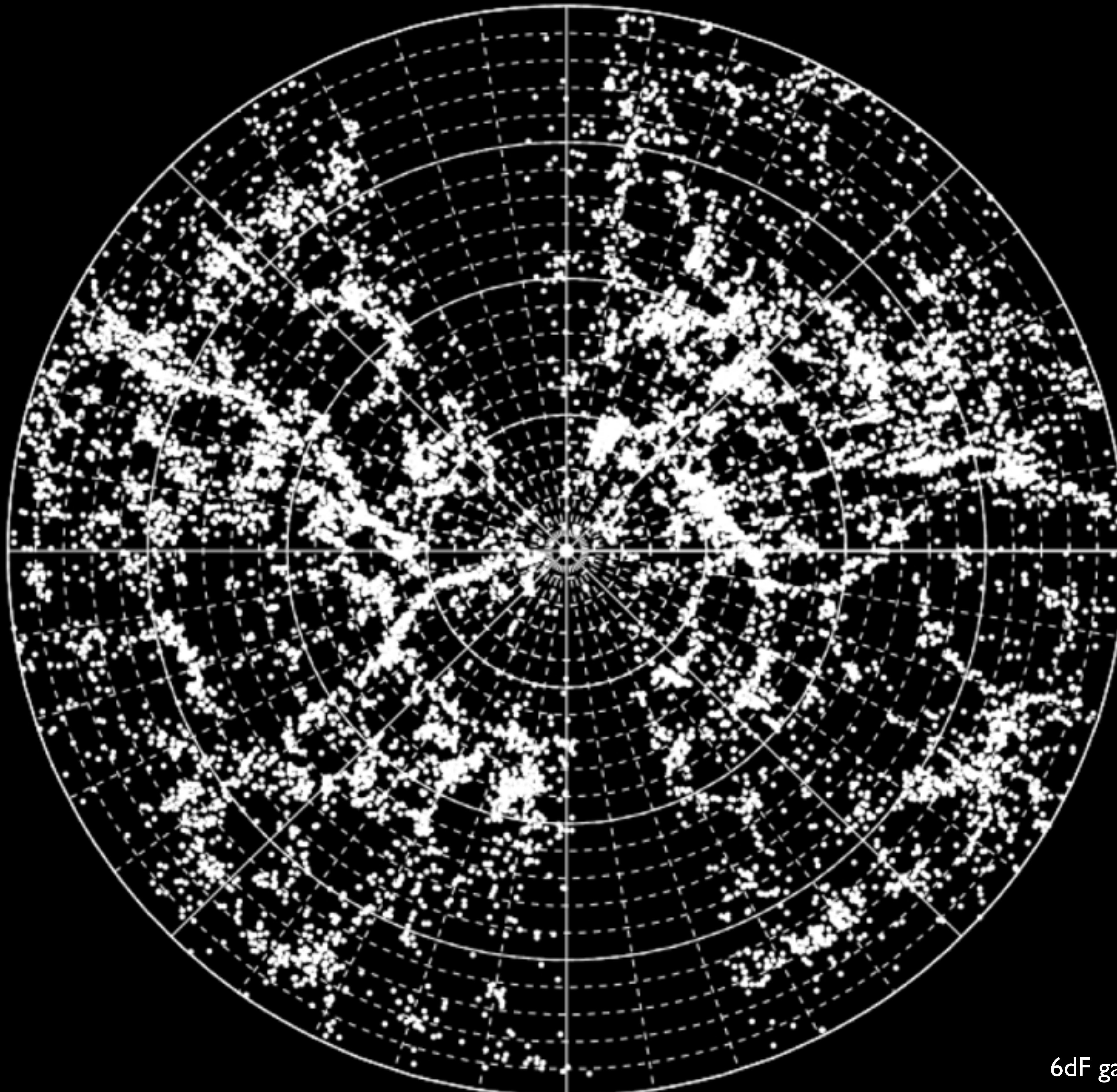


Example: Astrophysics

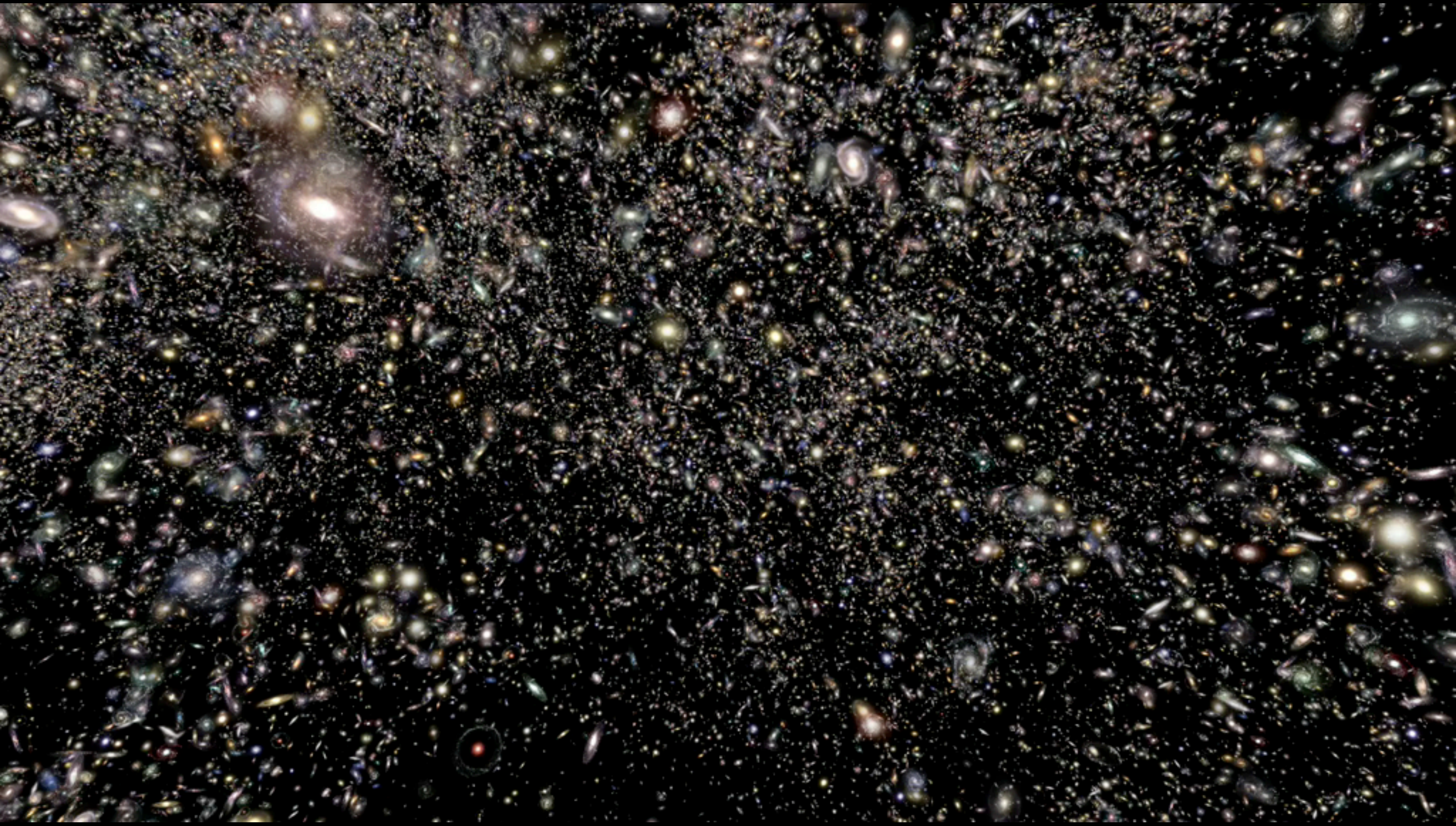
- Example of visualisation of simulation data in astrophysics.
- The formation of the large scale structure of the Universe.
- An exercise in computational design of reality.

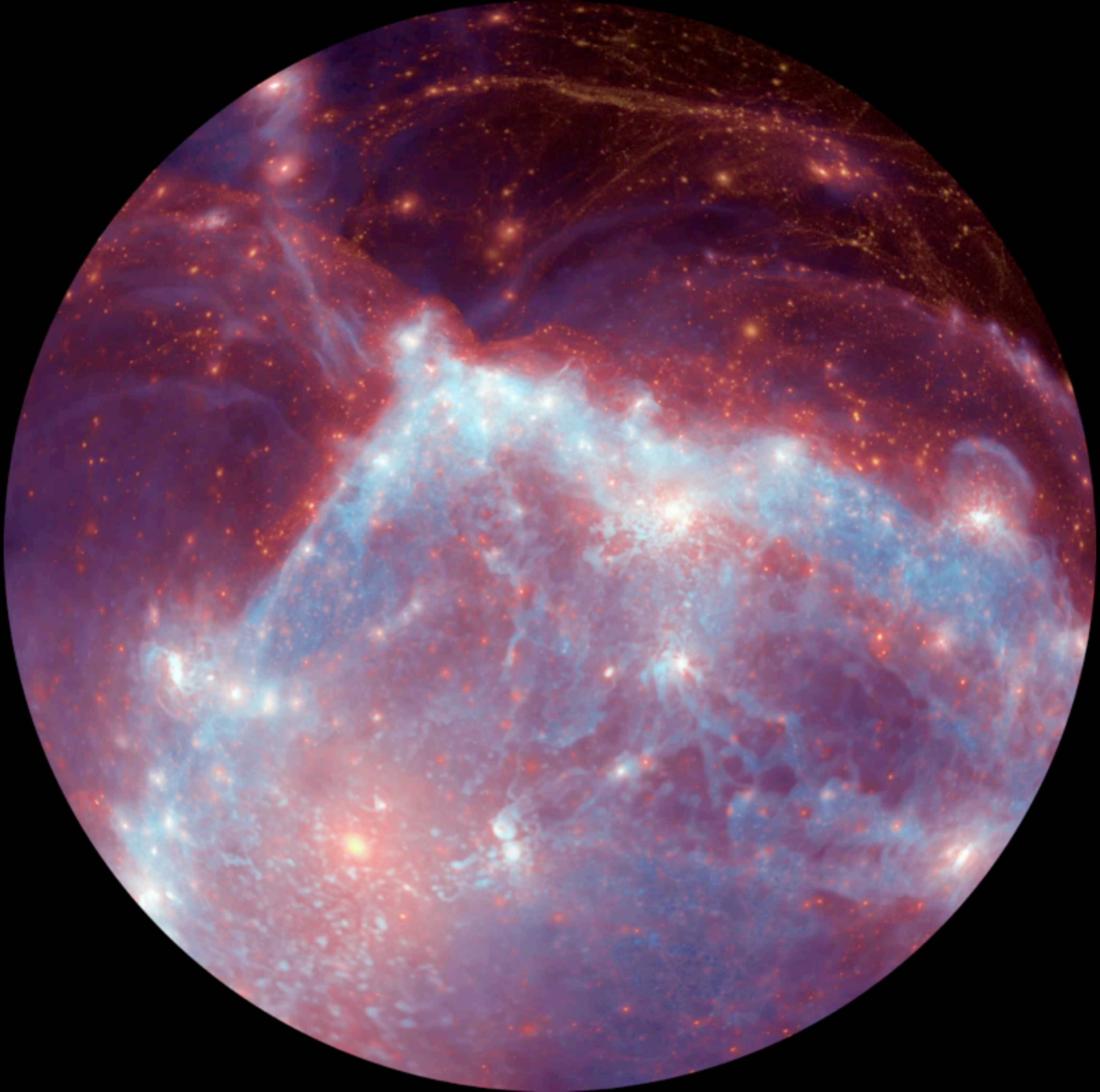
- Proposed laws of Physics go into a simulation of the formation of the universe from the big bang.

- Original simulation computed on vayu (NCI).
  Used 1024 cores, 2.8TB RAM, took 19 hours (~20,000 CPU hours)
  Visualisation performed on epic (iVEC).



6dF galaxy survey



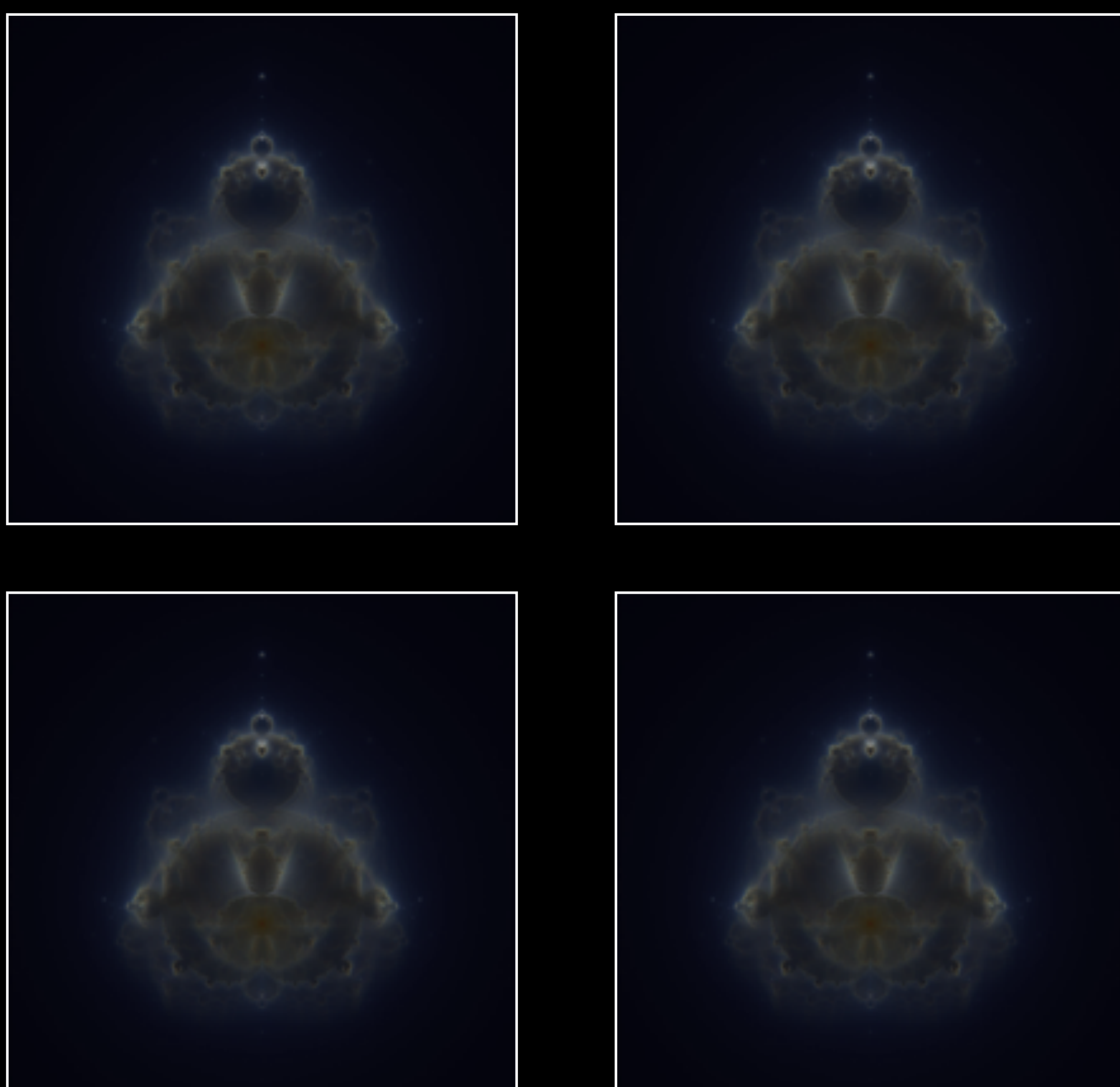


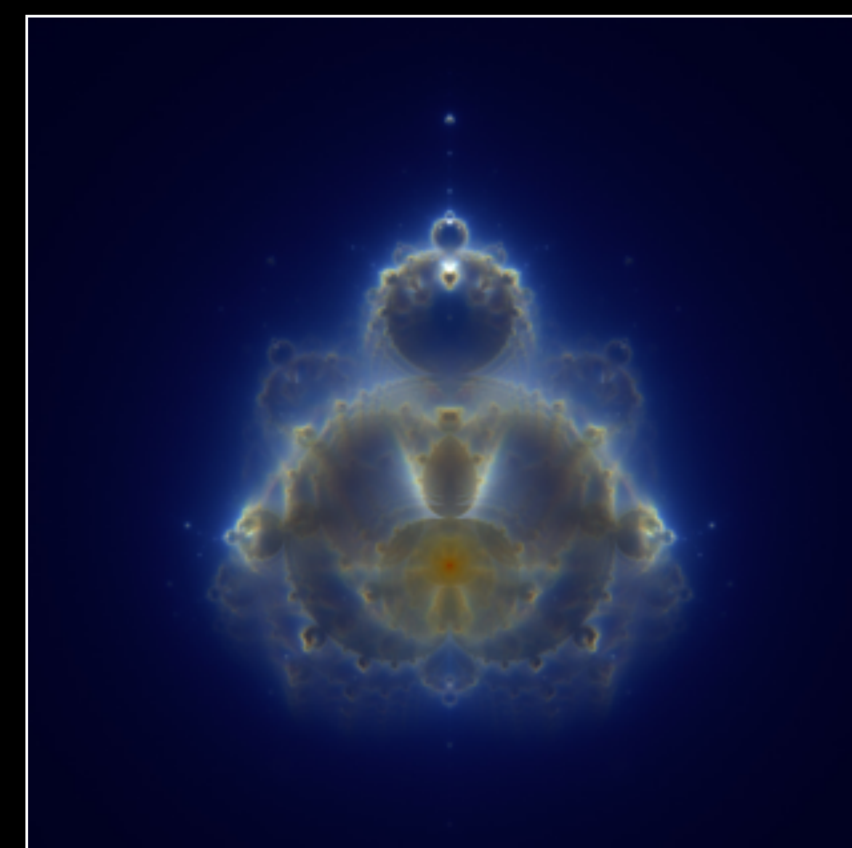
Serial vs parallel computing

- For some time CPUs have not been getting faster, or at least only modest yearly gains.
- Requirements in many areas require orders of magnitude improvement in performance.
- The only way compute performance will increase significantly is through parallel processing.
- Challenging since much of software base is written for serial implementation.
- How often does software perform faster when you add more cores?
  How often when more CPUs are added?
  How often when a cluster is used?
- "Supercomputers" generally have the same processors as other computers, but lots of them.
- Require fast interconnection between the processors.
- We are largely concerned with "scaling", how does the software/algorithm performance increase with CPUs?
- If we add twice the computing resource does the performance increase by 2?

Data parallel

- Sometimes the data can be split and processed in pieces and the results combined at the end.
- Simple example is the computation of the average.
  One can compute the average of a number of segments of the data and then compute the average of those sub averages.

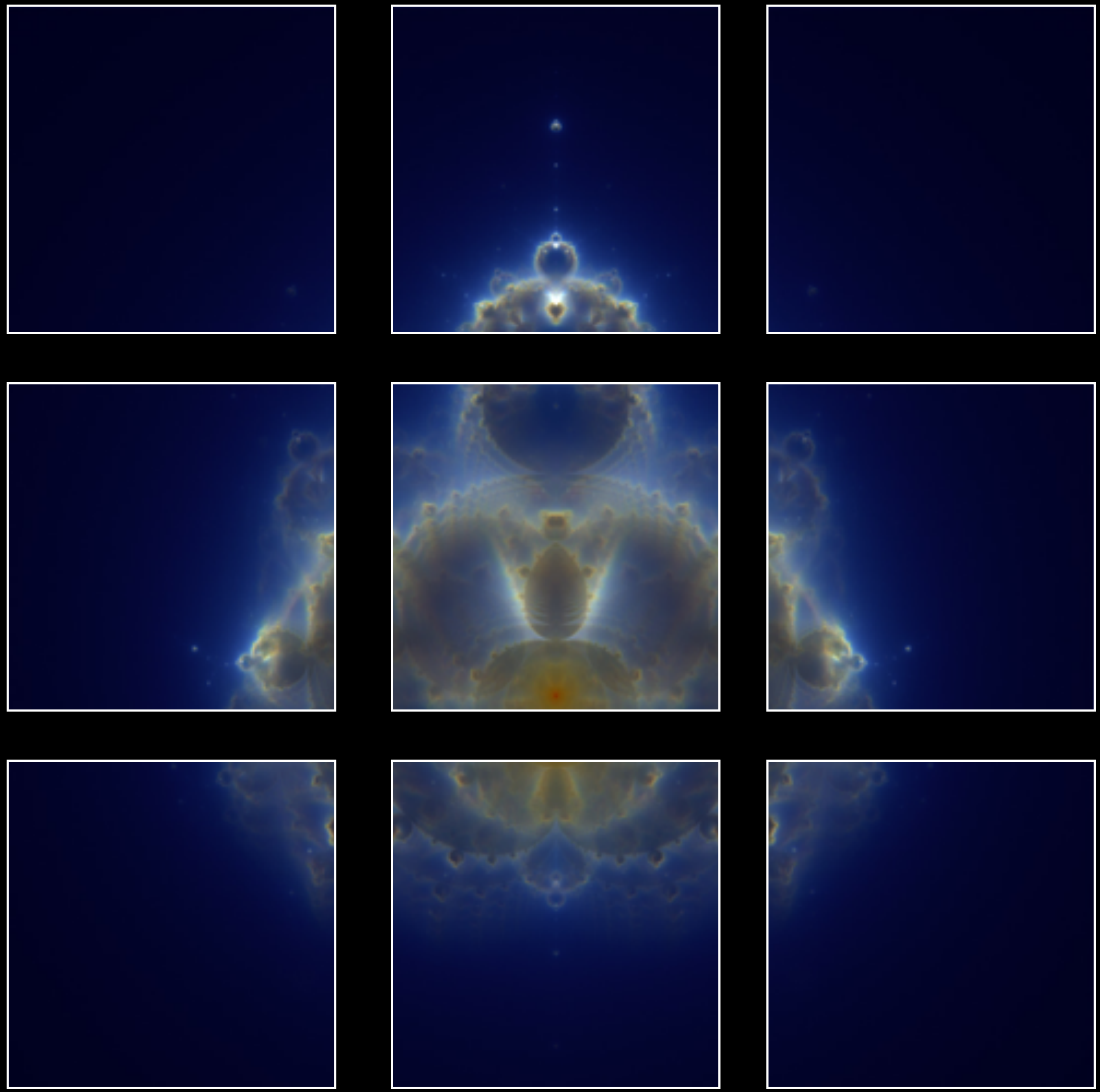




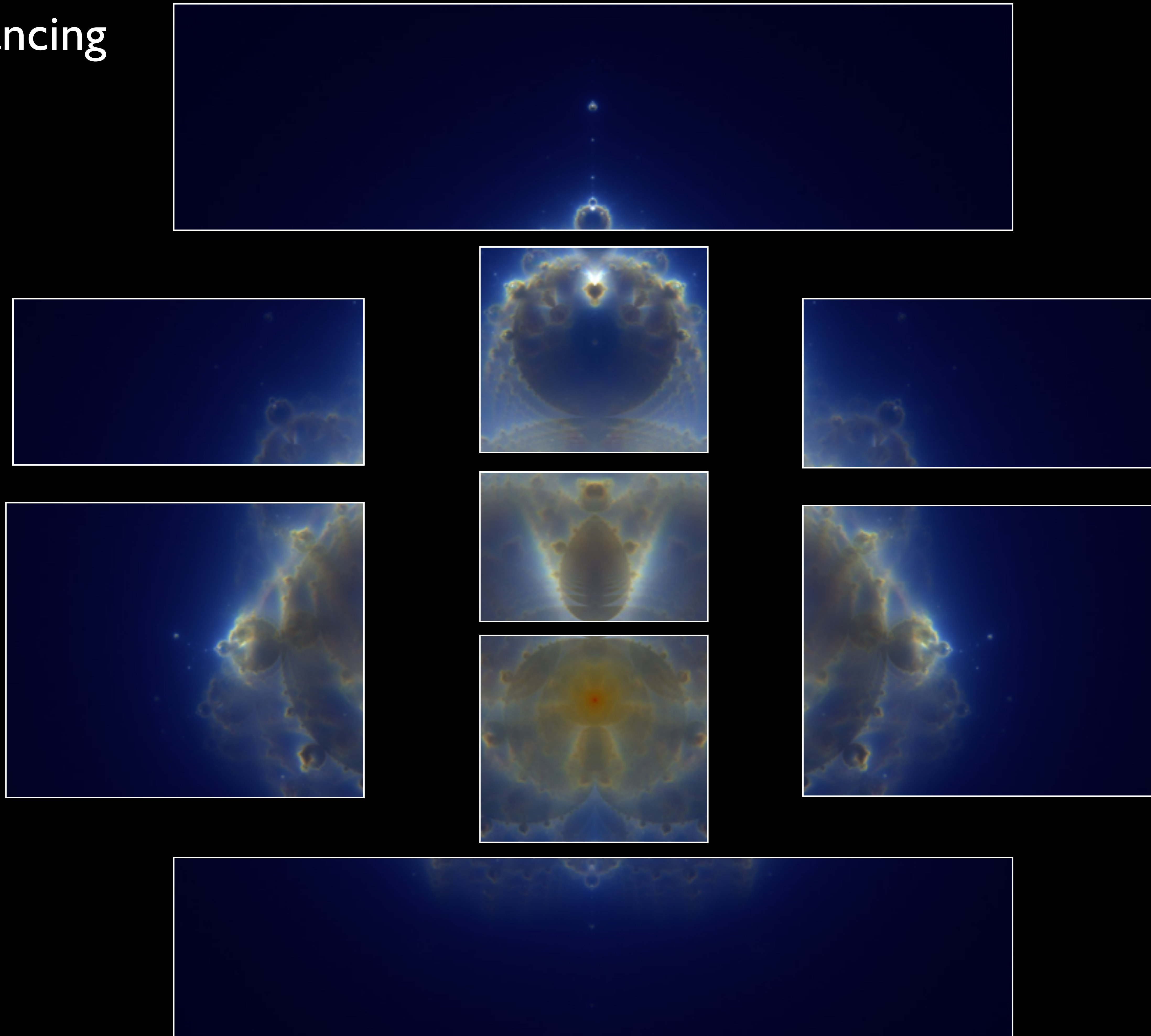
http://paulbourke.net/fractals/buddhabrot/

Result space parallel

- For some problems one can compute the result in sections, combining the sections at the end.
- The example here is called image space parallel, compute parts of an image on each processor and combine them at the end.
- A key to most parallel computing is how to "load balance".
- No point splitting a process up into N chunks for N processors if at the end one is waiting for a single chunk to complete.

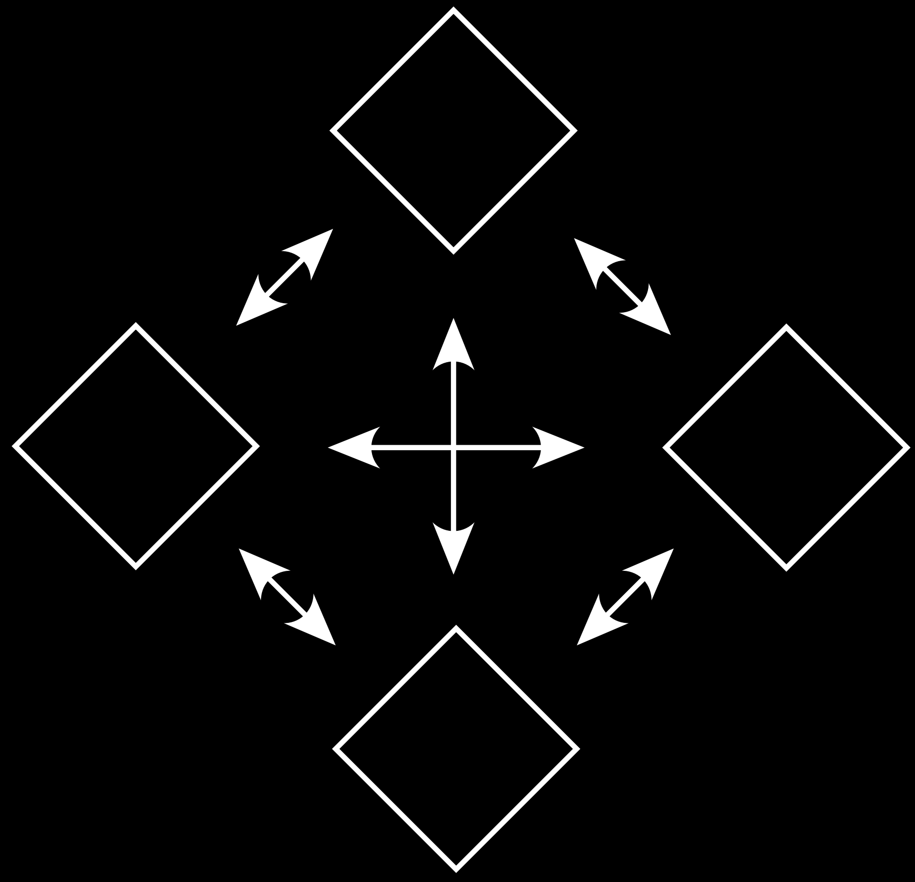


Load balancing



Parallel processing

- Previous two are example of what we call trivially parallel.
- Not suited to some algorithms, for example where the whole state needs to be updated with a knowledge of the whole system. Data or result cannot be so easily partitioned.
- In these cases the interconnection between separate processors becomes important as each processor need to communicate with the others to maintain the same state.

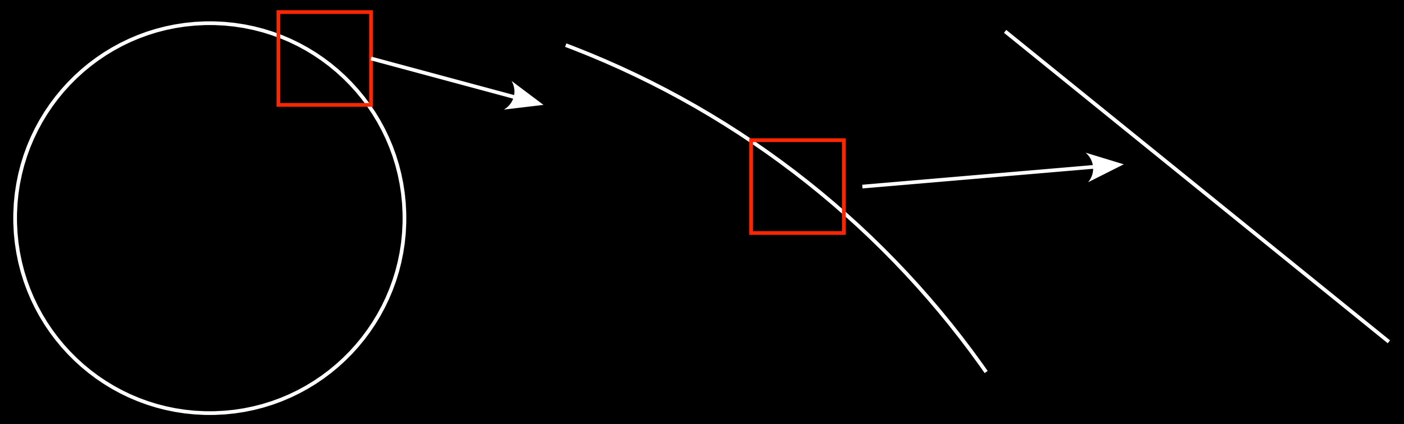


Euclidean geometry

- Library of Euclidean forms
  http://paulbourke.net/geometry/

- Characteristic of Euclidean geometry, as you examine it in more detail it becomes more boring.

- eg: As you zoom into a circle it looks more and more like a line. See later in comparison to fractals.



- Select a number of my projects that illustrate computation and creation of 3D form.

- Parametric equations: equations control geometry

- Algorithmic generation: geometry created by a process

- Simulation: set up conditions and let software find a stable or minimal solution

- Constructive solid geometry (CSG)

Supershapes: Parametric equation

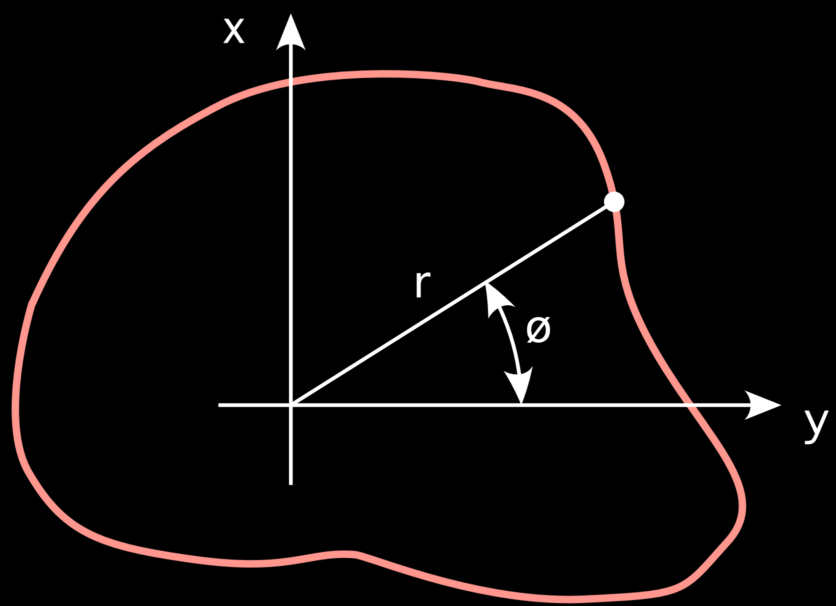
- A simple example of a parametric model.
  http://paulbourke.net/geometry/supershape/

- Proposed as a model for generating plant structures and other organic shapes by Johan Gielis.

- Polar coordinates:
  Vary "ø" from 0 to 360 degrees, the formula gives the distance "r" at that angle.

- Parameters are "m", "n1", "n2", "n3".

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



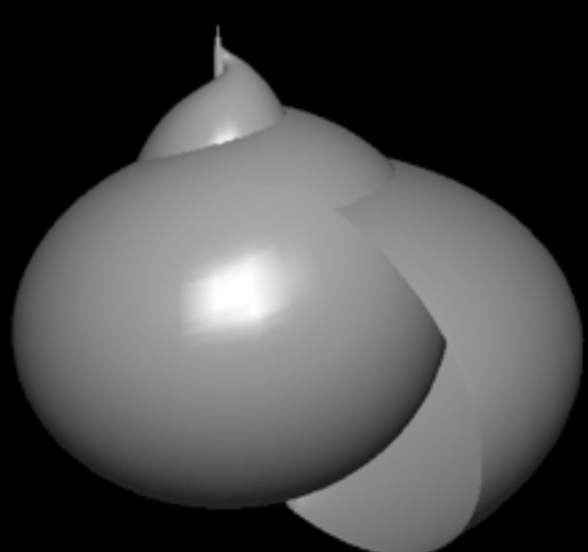
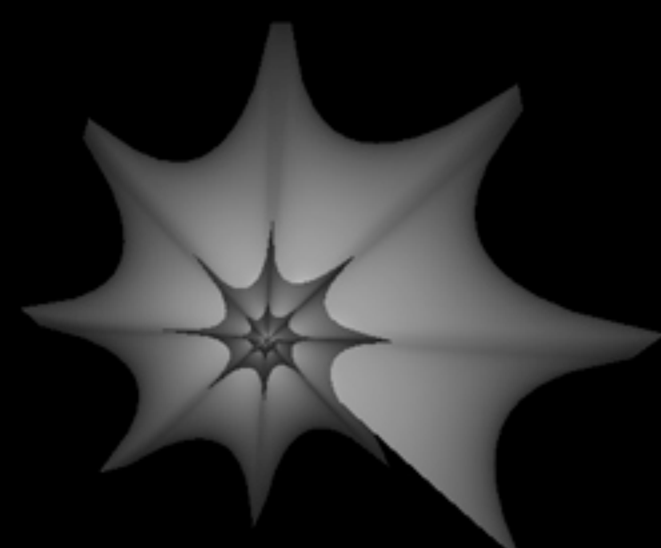
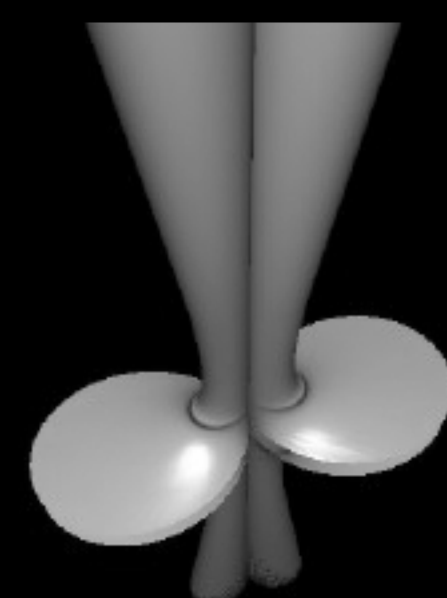
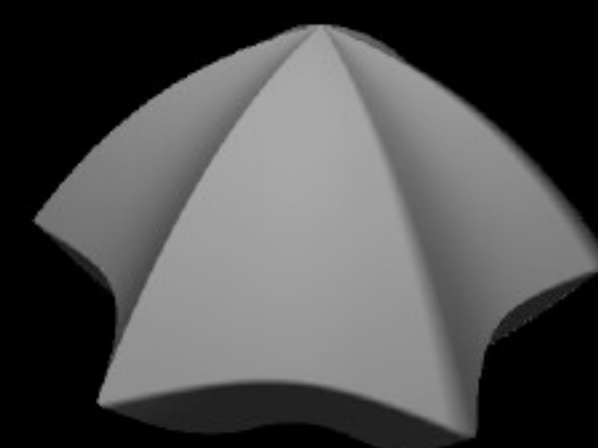
Extension to 3D

- Similar to 2D except polar coordinates in 3D consist of two angles (longitude and latitude) and radius.
- For any longitude and latitude the formula gives the radius at that position.
- If the radius was constant for all angles it would be a sphere.

$$r(\phi) = \left[\left| \frac{1}{a} \cos(m \phi / 4) \right|^{n2} + \left| \frac{1}{b} \sin(m \phi / 4) \right|^{n3} \right]^{-1/n1}$$

$$x = r1(\theta) \cos(\theta)\, r2(\varphi) \cos(\varphi)$$
$$y = r1(\theta) \sin(\theta)\, r2(\varphi) \cos(\varphi)$$
$$z = r2(\varphi) \sin(\varphi)$$

$-\pi/2 \leq \varphi \leq \pi/2$ eg: latitude
$-\pi \leq \theta \leq \pi$ eg: longitude

Questions

- When one has a parametric equation for creating geometry, questions are:

  Does varying the parameters lead to predictable outcomes?
  Can one choose parameters in such a way to create a predefined object?
  Do the parameters have intuitive meaning?

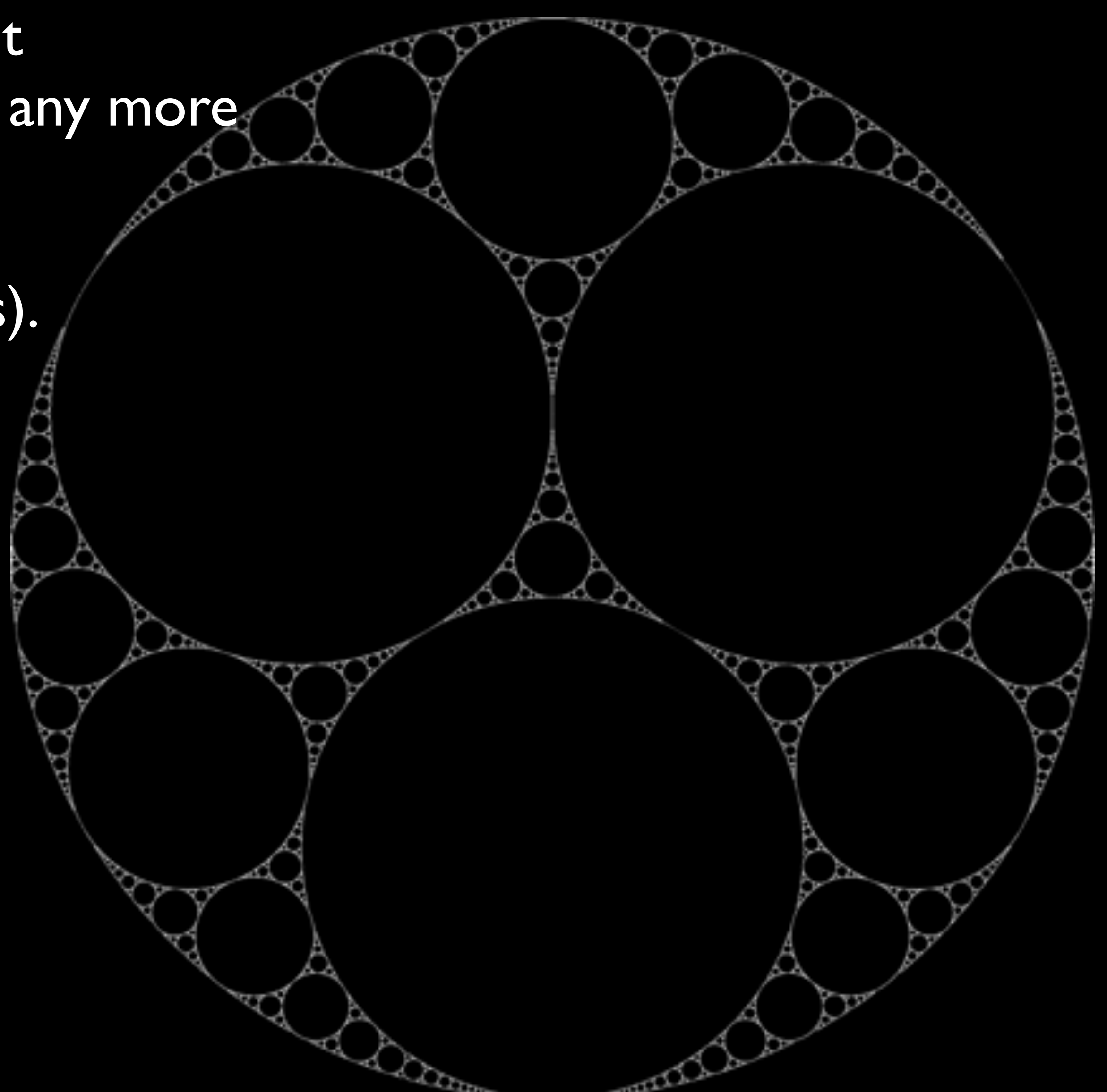
- Elegant when a few number of parameters can define a wide range of shapes.
  See fractals later.

- Not so elegant when there is not a predicable behaviour of the parameters.
  Become an exercise in trial and error.

- Quote: John von Neumann
  "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk."

Example: Space filling - algorithmic

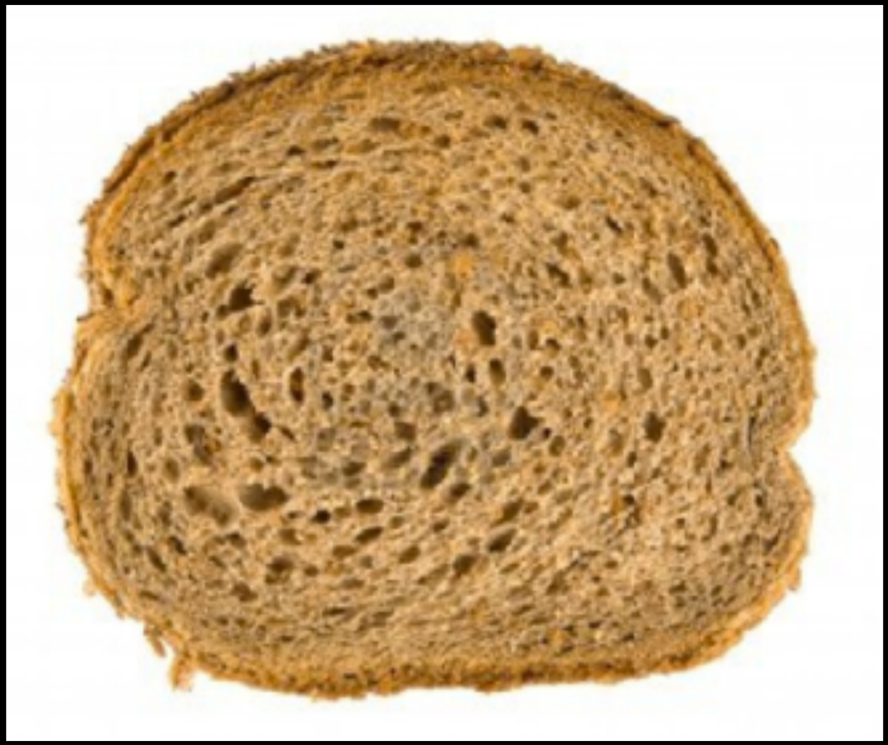
- Many natural structures tend to fill space.
- Space filling arises in nature due to competition for some resource.
- A previous model for creating these space filling has been so called Apollonian packings.
- Apollonian algorithm (or variations)
  - find some empty space
  - seed that location with a new object
  - grow the object until it touches another object
  - optionally shift the object until it cannot grow any more
  - place the object and repeat the process
- Key characteristic is that the objects touch (kiss).
- Not a fixed equation but a process.



Non-Apollonian packings


Bubbles in a plate

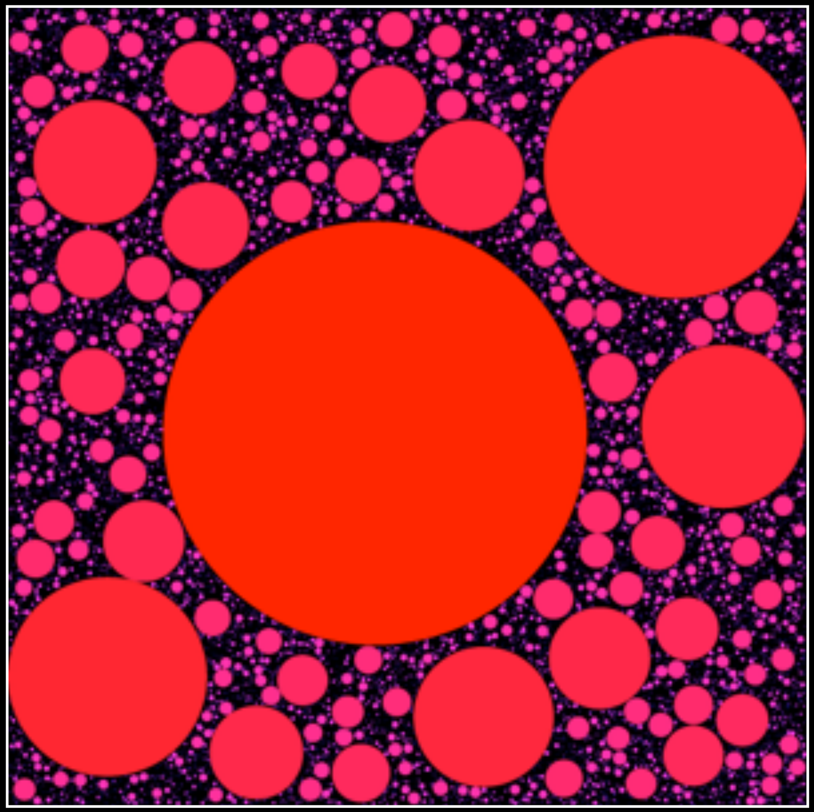

Rocky river shore


Bread air pockets

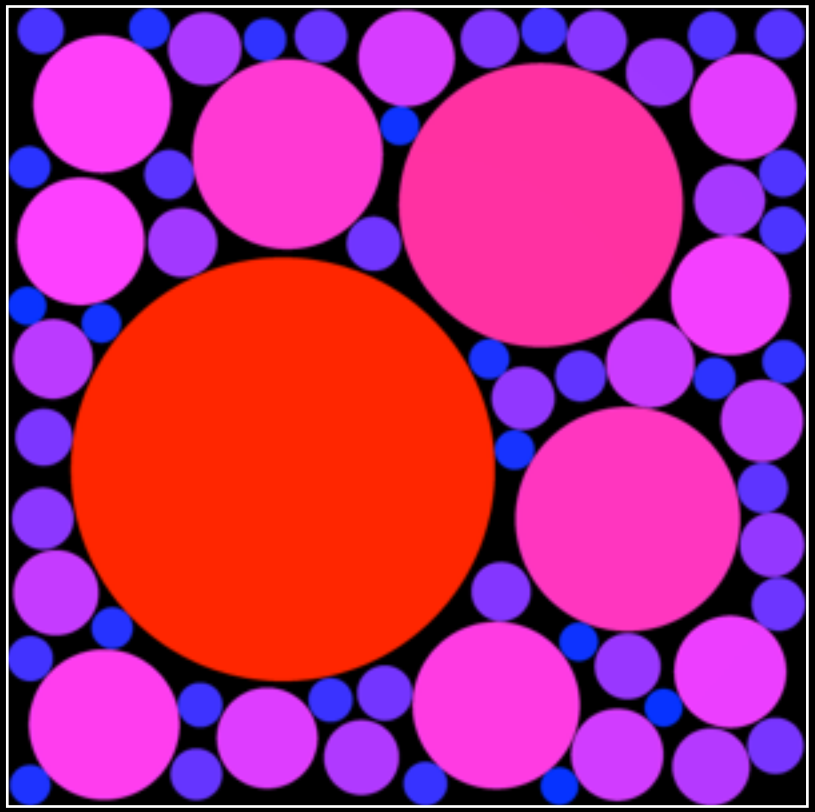

Lily pond

New model

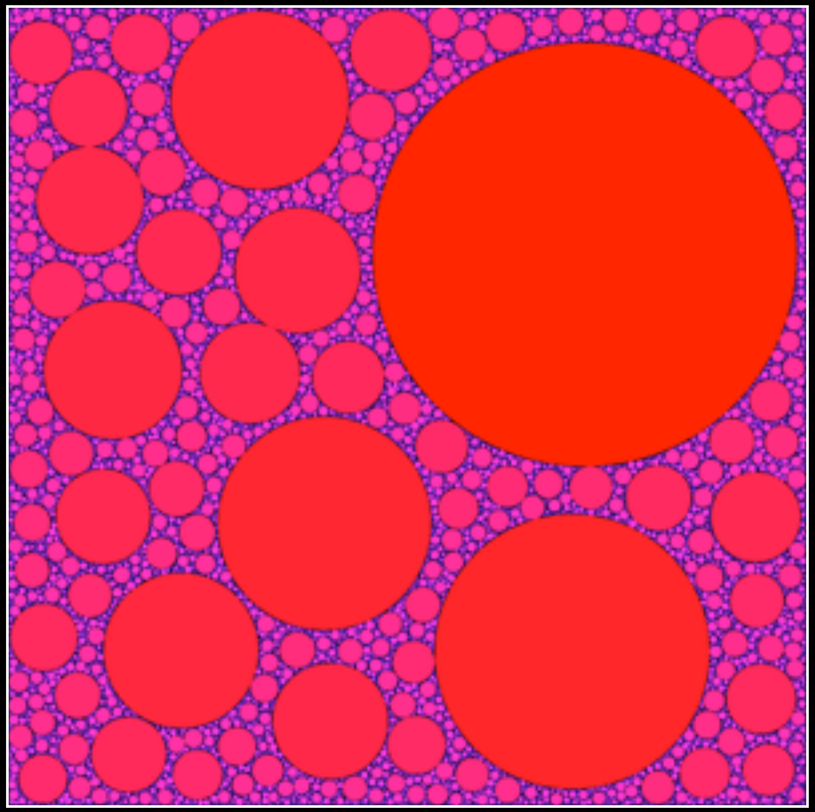
- Algorithm:
  - randomly find a place to locate the next object of a predetermined size without intersecting existing objects
  - add that to the database of existing objects
  - reduce the size of the object
  - repeat until some object count is reached or space is filled to some pre-specified level
- The trick ("magic") is how to reduce the size of the object such that the process never stops and it fills space (without gaps).



If size is reduced to fast space is not filled

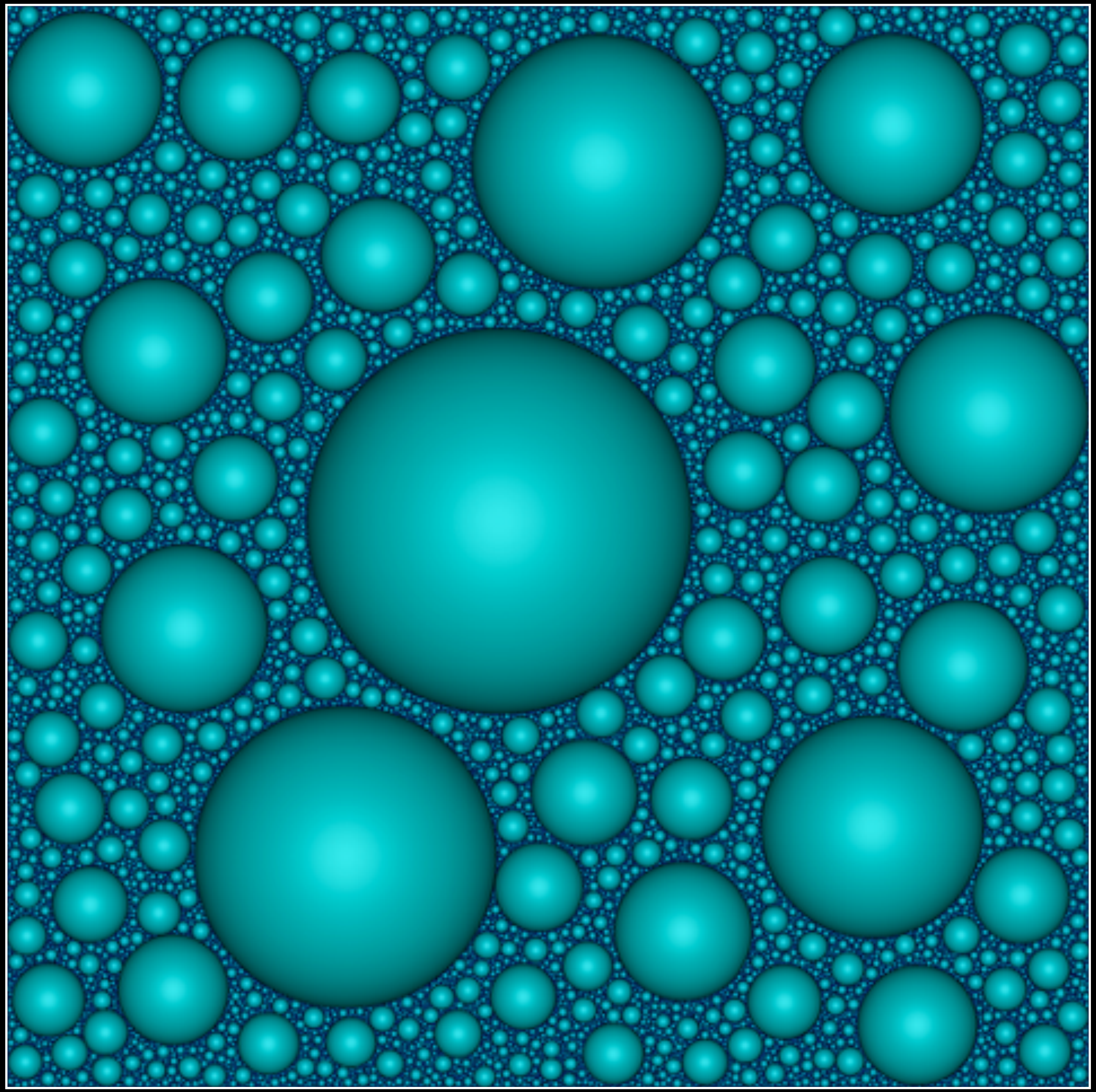


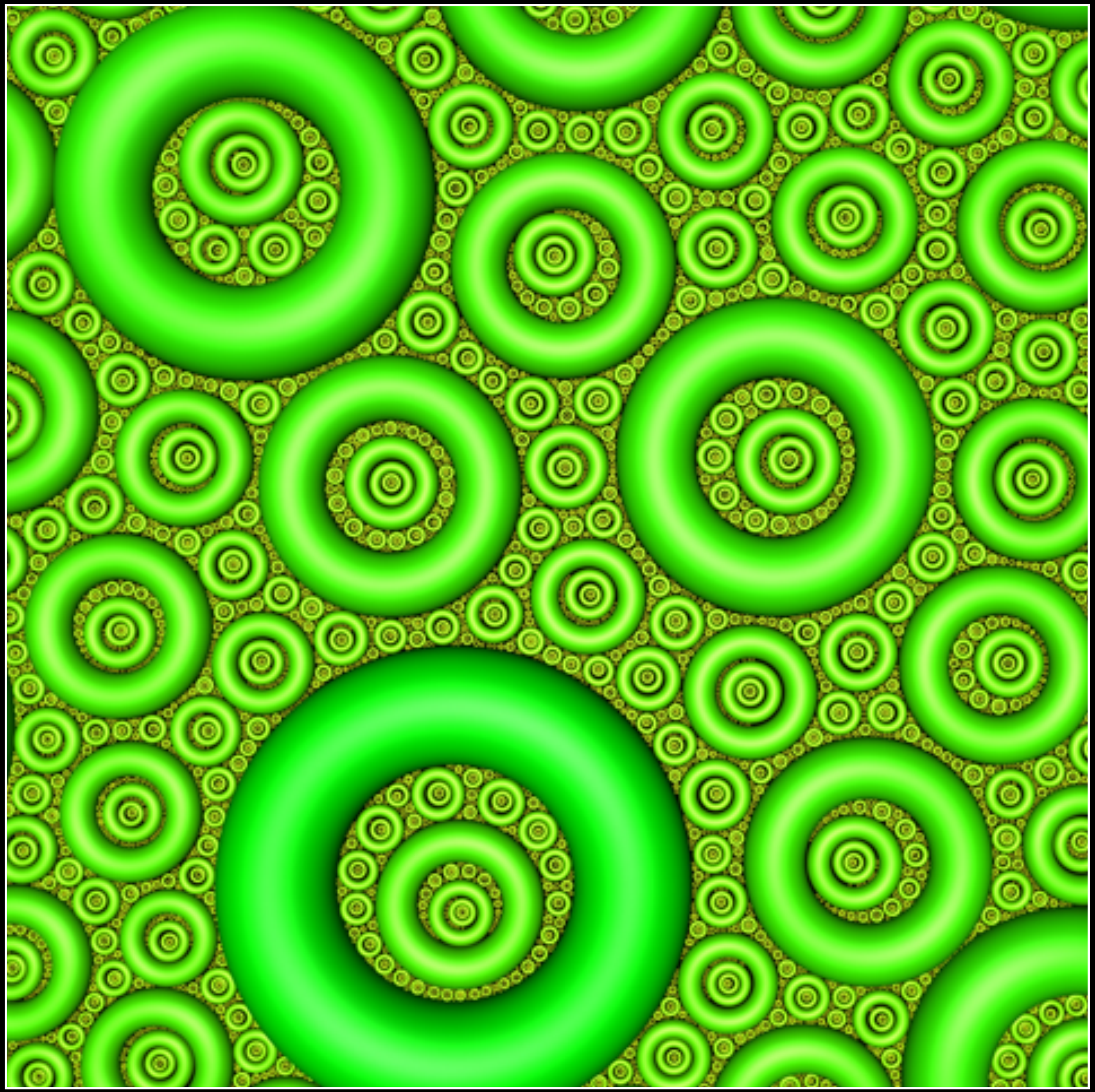
If the size is not reduced fast enough there will be no space for the next object of the intended size



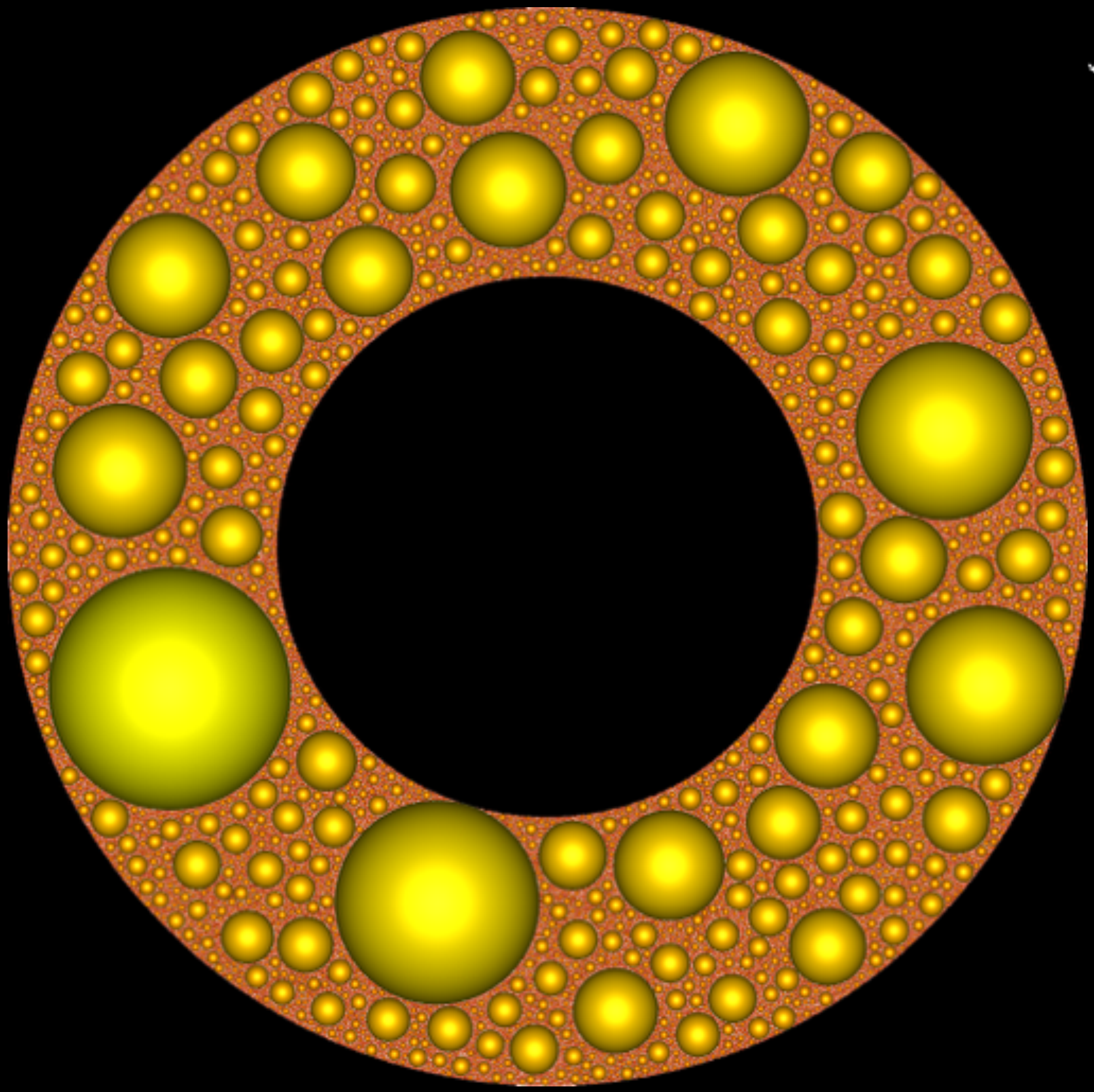
If reduced exactly right then space is filled.

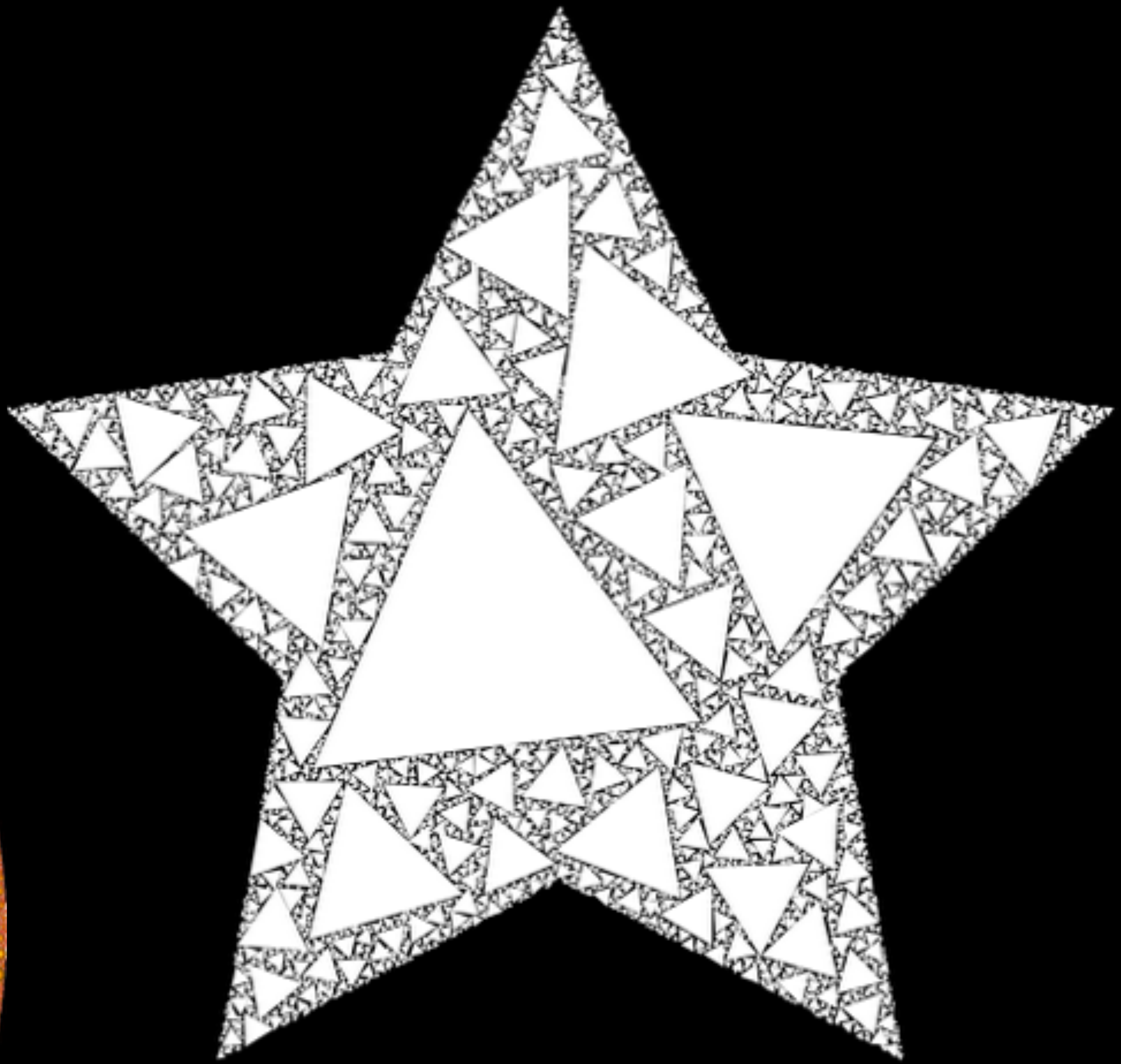
2 dimensions



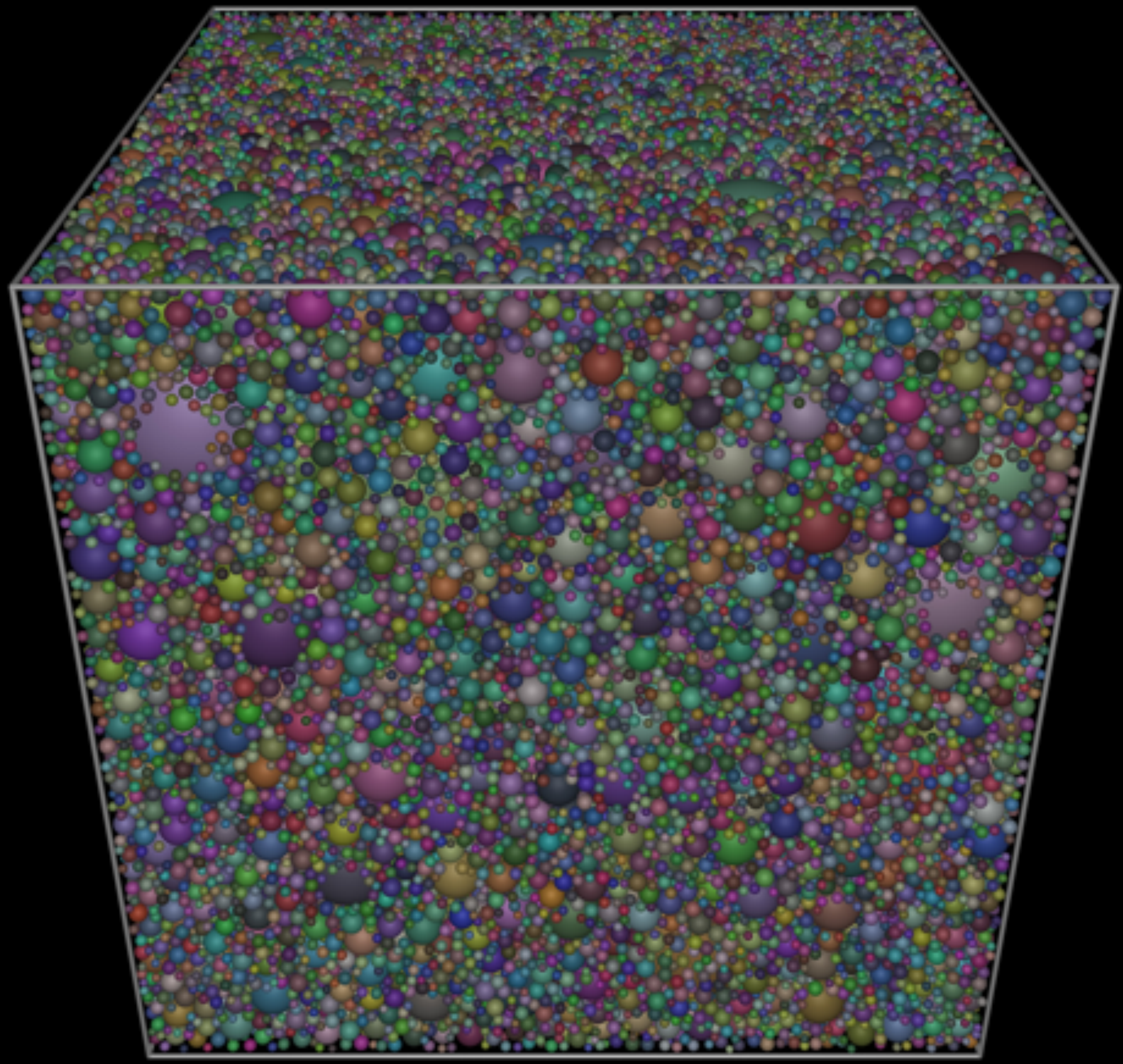


Fill any shape with any shape





3 dimensions



100,000 spheres

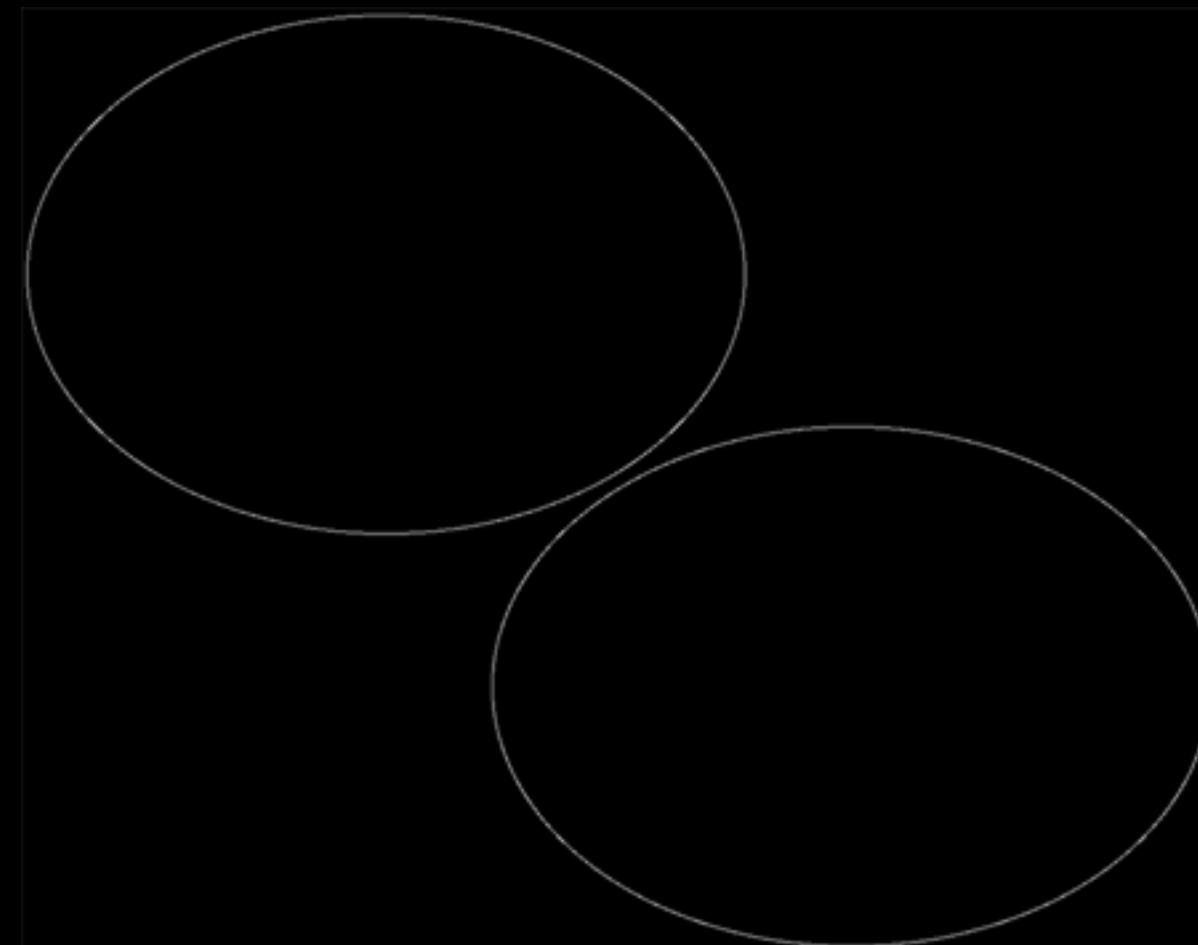
10,000 torii

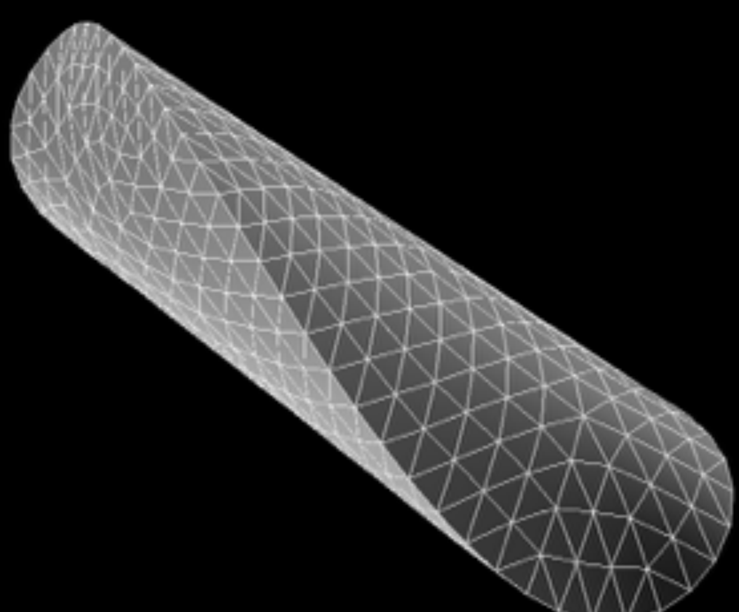
dForm surfaces: Simulation - find surface of least stress

- Credited to Tony Wills.
- http://paulbourke.net/geometry/dform/
- Example of a 3D shape that while simple to understand involves complex physical simulations to form digitally.

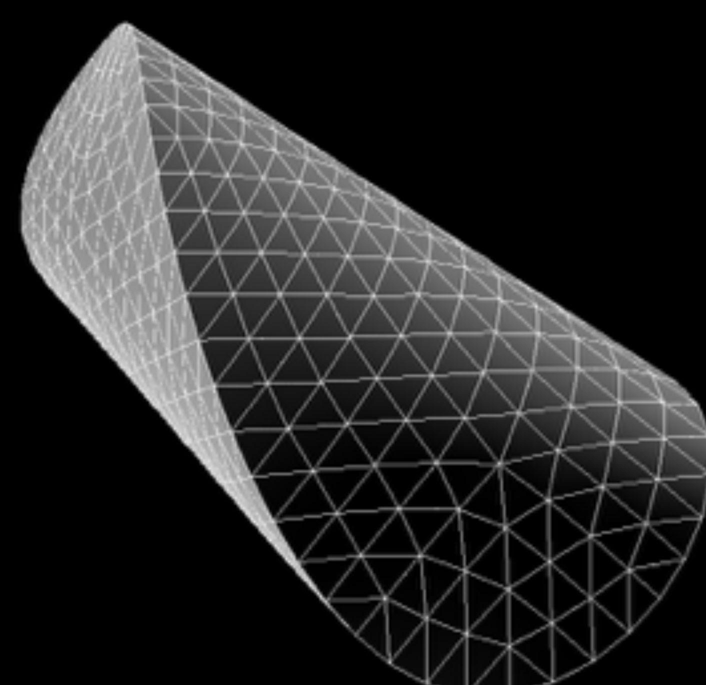
Concept

- Take two outlines of identical length.
- Non-elastic material.
- Stitch them together, possibly starting at different positions. What 3D shape do they form?
- Simplest case is 2 ellipses, 2 circles is always boring.

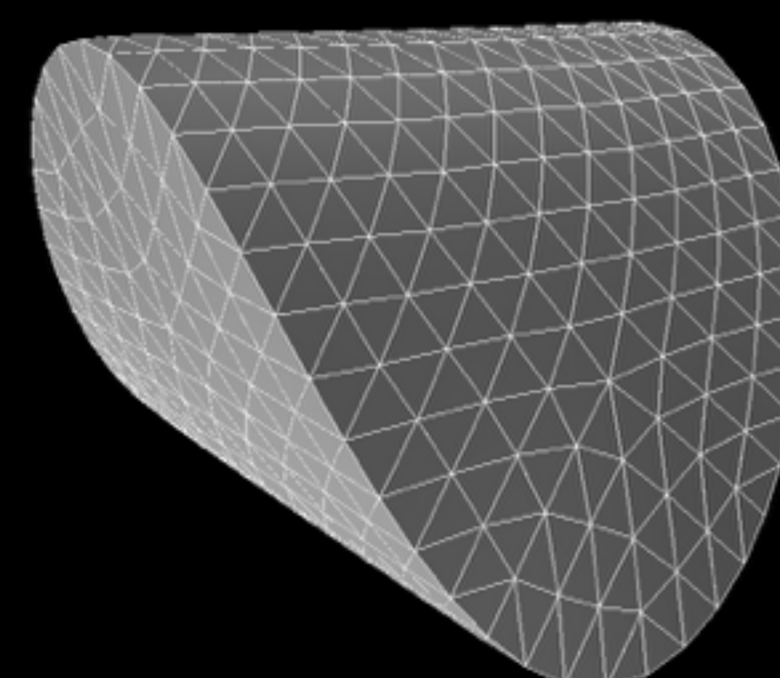




22.5 degrees

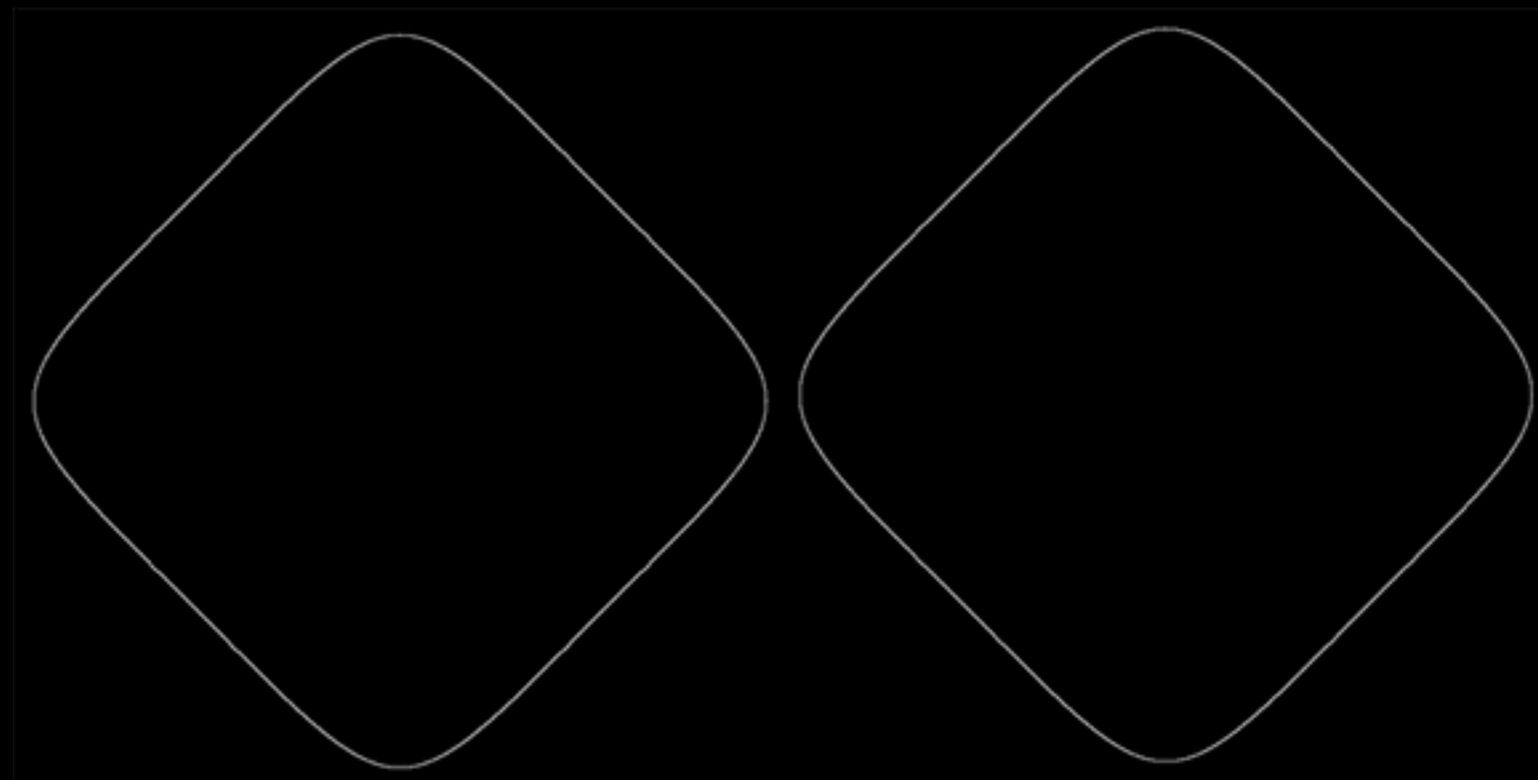


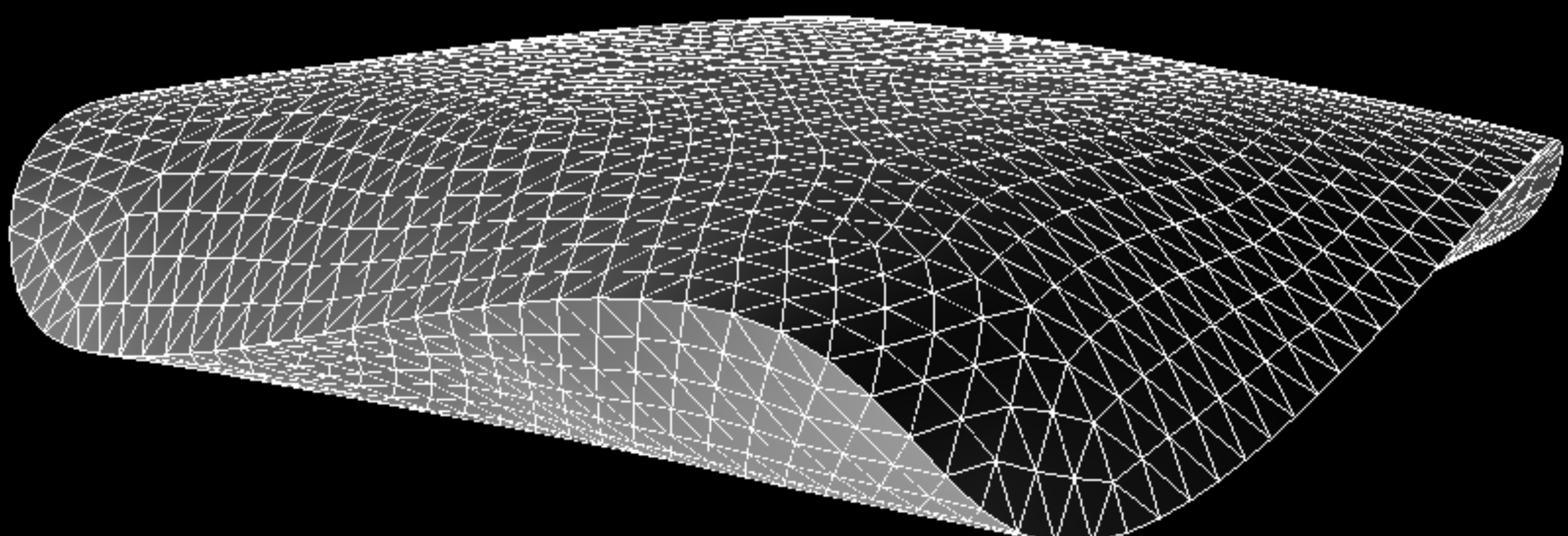
45 degrees



90 degrees

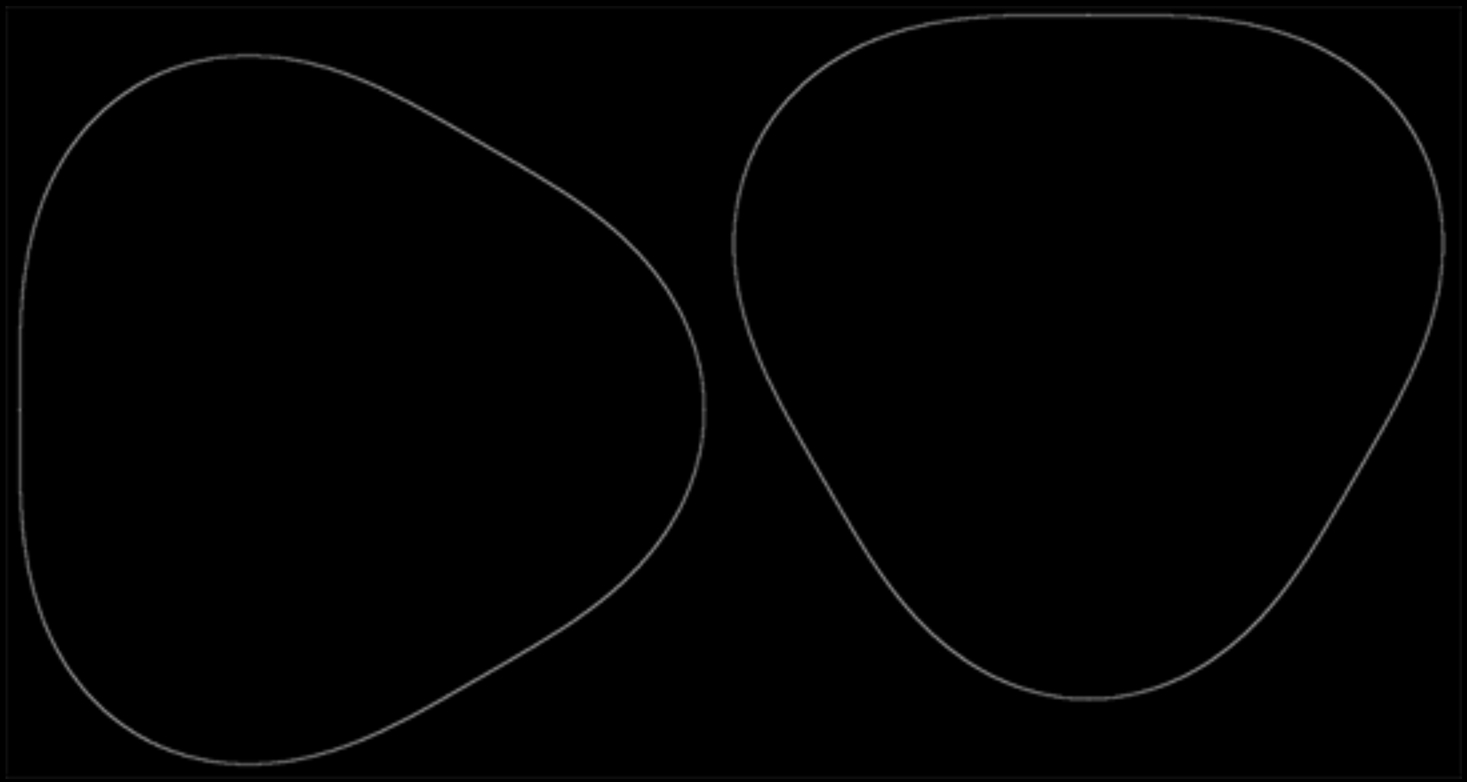
Example: Rounded squares

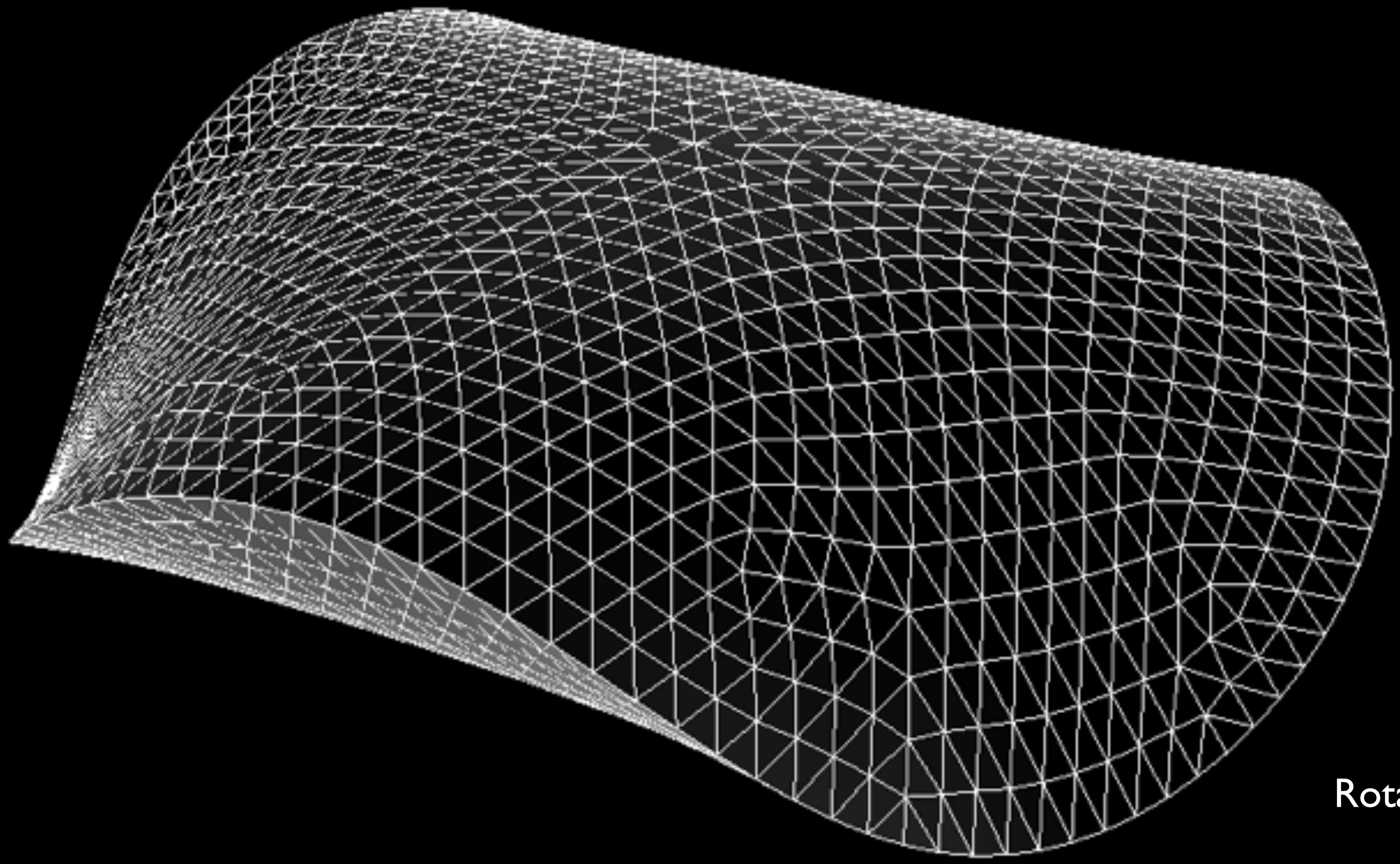




Rotated 30 degrees

Example: Rounded triangles





Rotated 60 degrees

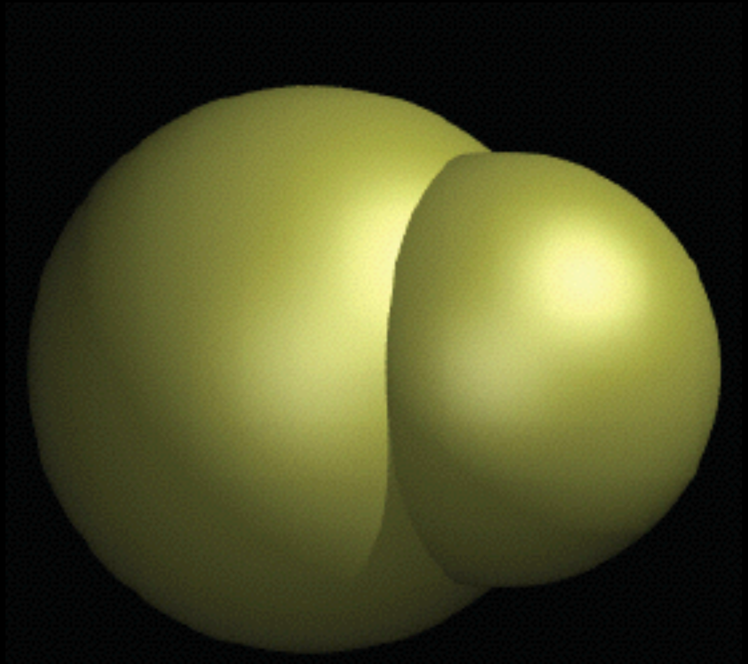
Algorithm

- Algorithm involves creating a mesh representation of the two surfaces.
- Virtually stitching them together = joining together the boundary vertices of the mesh.
- Assuming an elastic surface ... this will result in elastic stress/stretching of the surfaces.
- Every possible configuration of these meshes has some metric, total stress say.
- Apply physics to minimise that stress by deforming the surfaces.

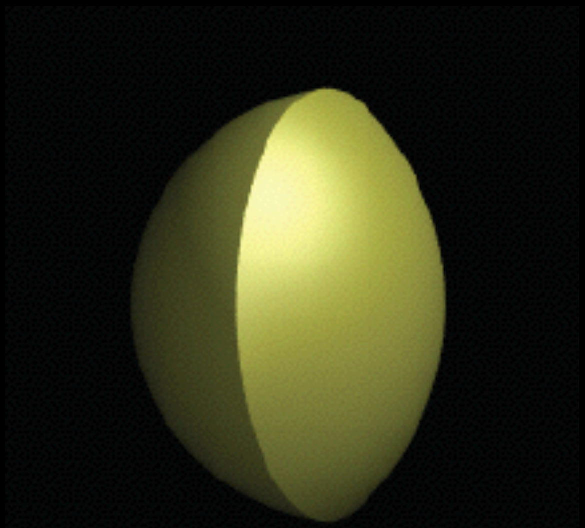
- Computationally impossible to scan the entire solution space.
- The algorithm attempts to move the system towards the configuration of minimum stress.
- Question, how does one know the algorithm will find the global minimum rather than just a local minimum?

Constructive solid geometry (CSG)

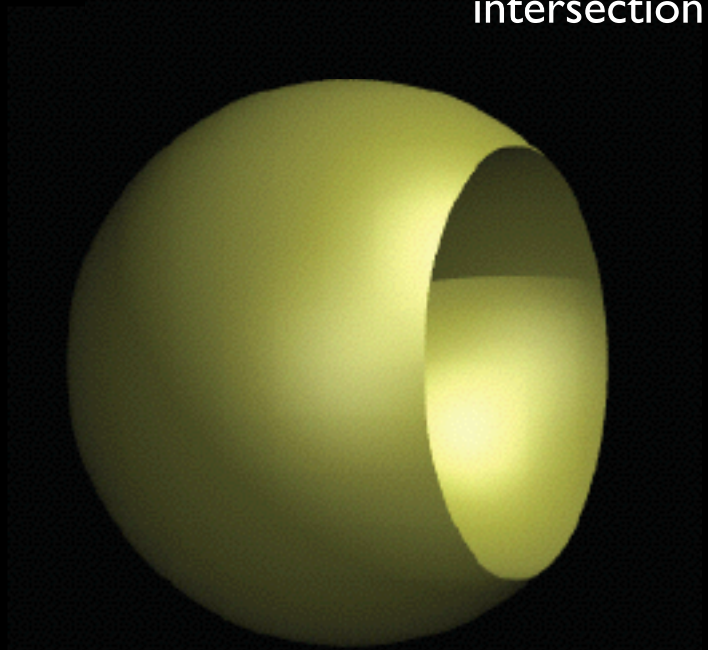
- A well established branch of computer based modelling.
- Everything has physical thickness and is solid. Has a clear notion of the inside and the outside.
- As distinct from more traditional modelling with infinitely thin lines or planes.
- Solid objects are combined in pairs using logical operations
  union - add the two objects together
  intersection - result is the space contained inside both objects
  difference (subtraction) - subtract one object from the second
- In the extreme case there is only one geometric primitive, a "half space".
  A cube is built from 6 half spaces suitable rotated and with the intersection operator applied.
- A very useful construction for modelling objects defined by such logical operations.
  Very difficult to construct solutions otherwise, for example, as a mesh definition.



union



intersection

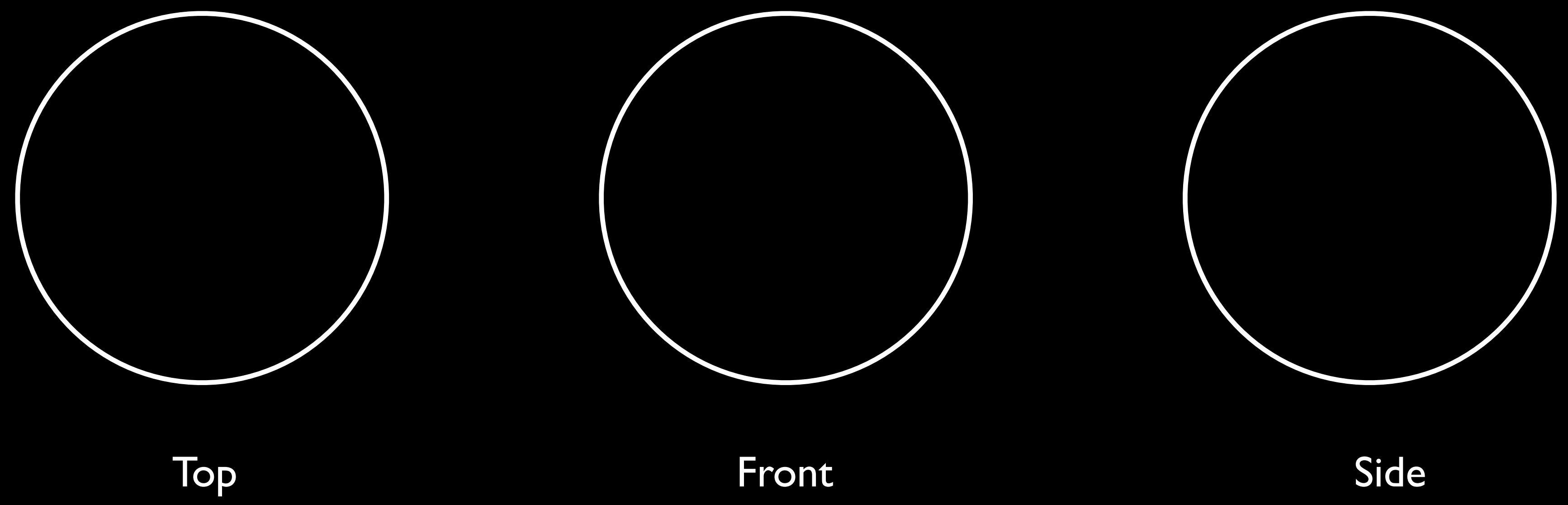


difference

Question

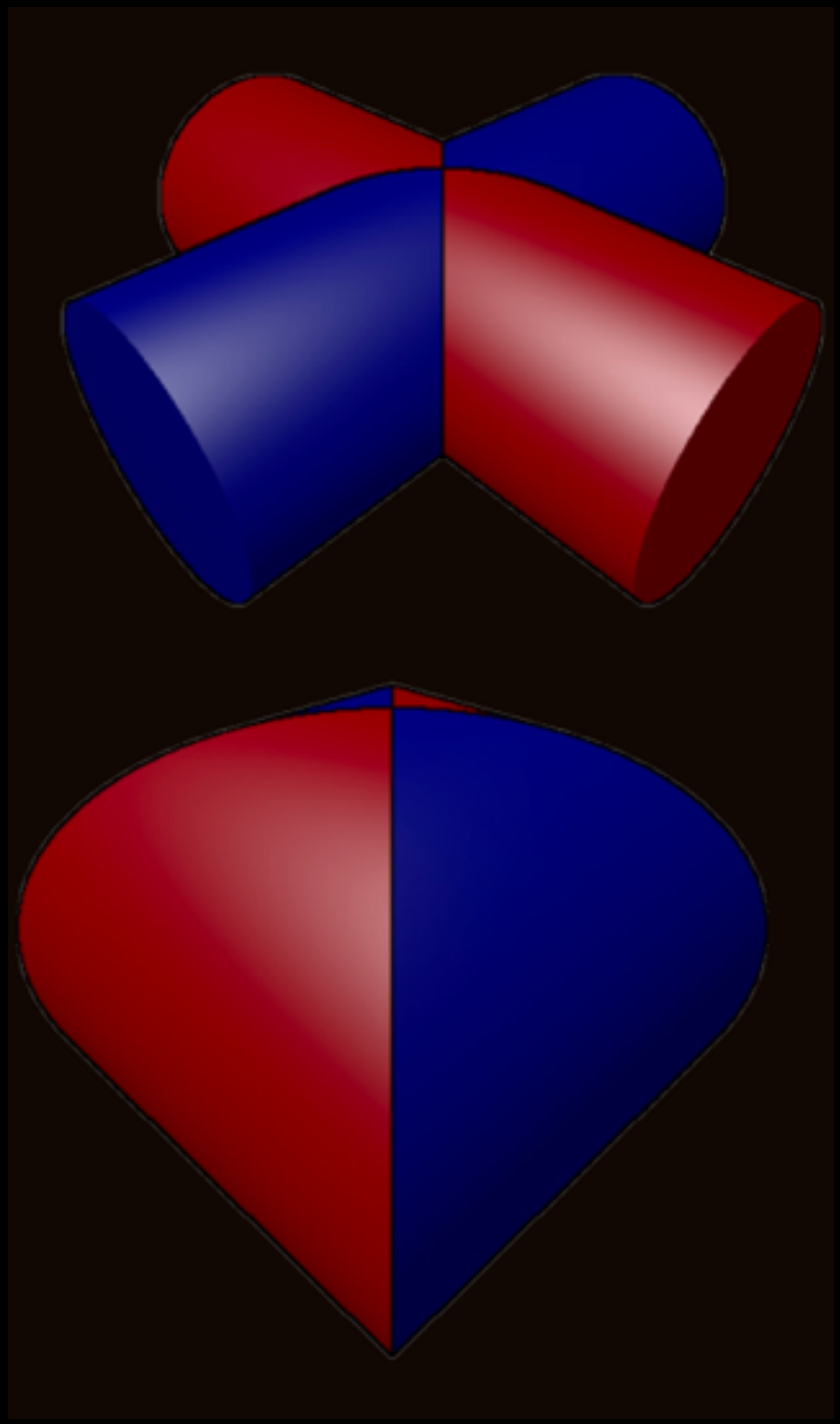
- What is the object that is a circle in plan, front, and side view?

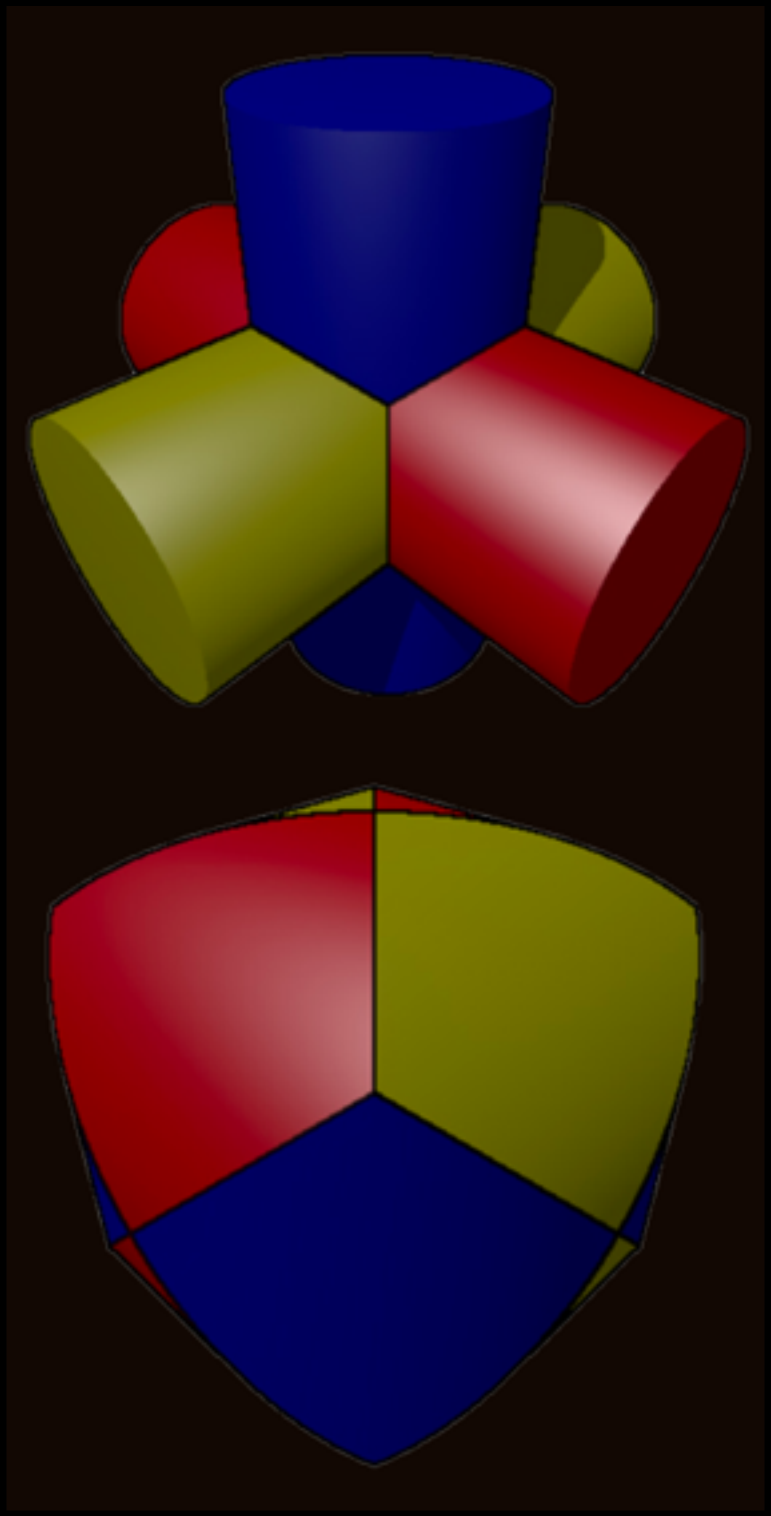
- Besides a sphere ...



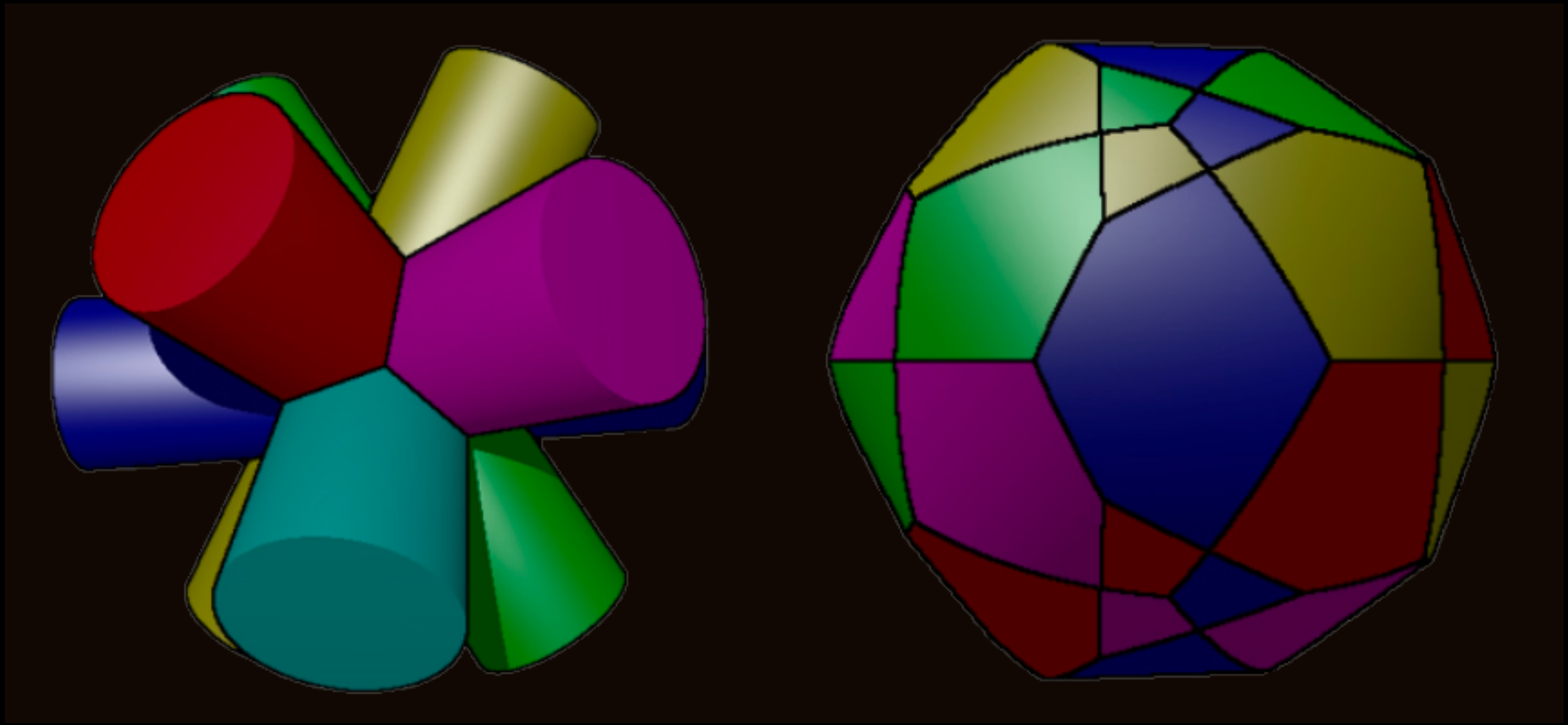
Answer

- What is the geometry resulting from the intersection of three cylinders?
- An interesting shape if you want to create one on a lathe, it rolls on three axes but has corners.



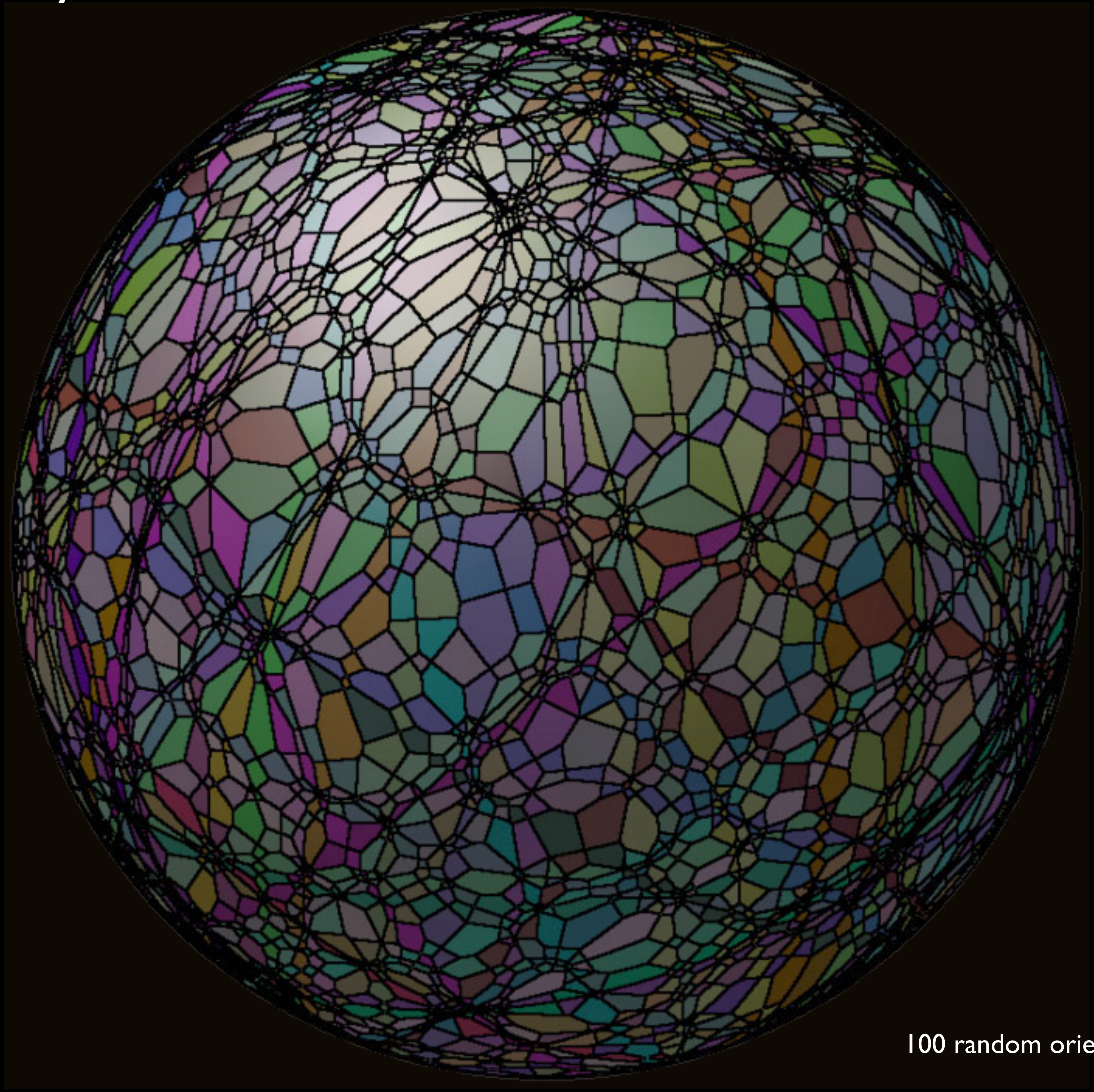


More cylinders



5 cylinders

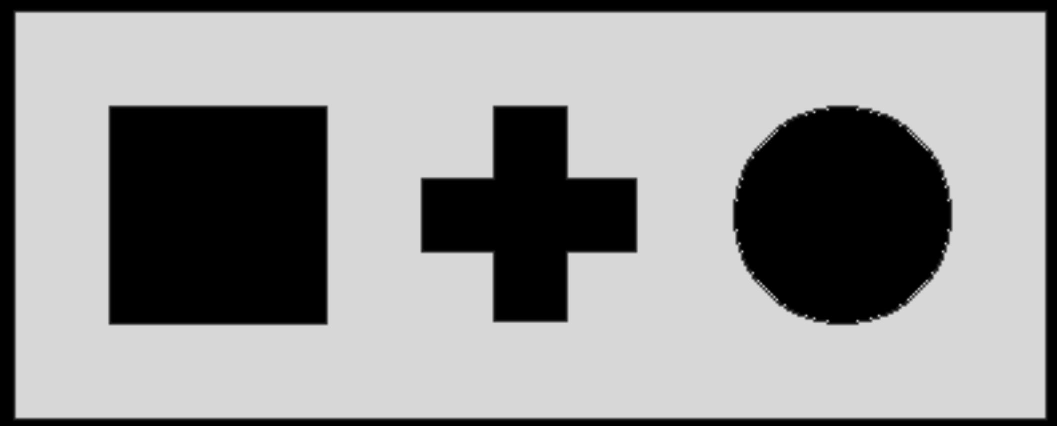
Even more cylinders

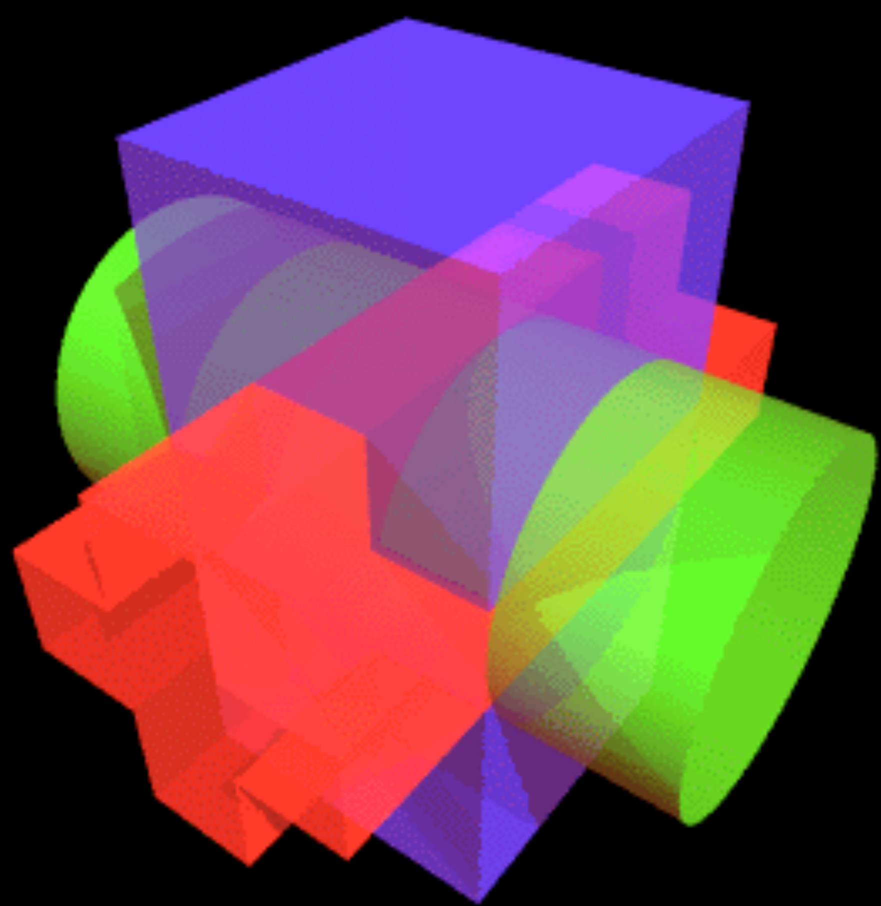


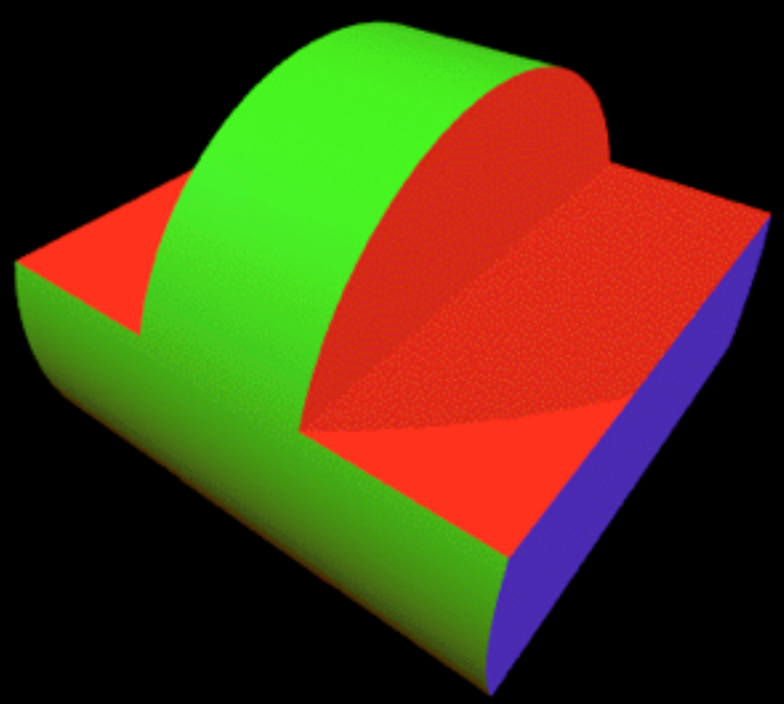
100 random orientated cylinders

Another example

- What single object can slide though all slots in the following?





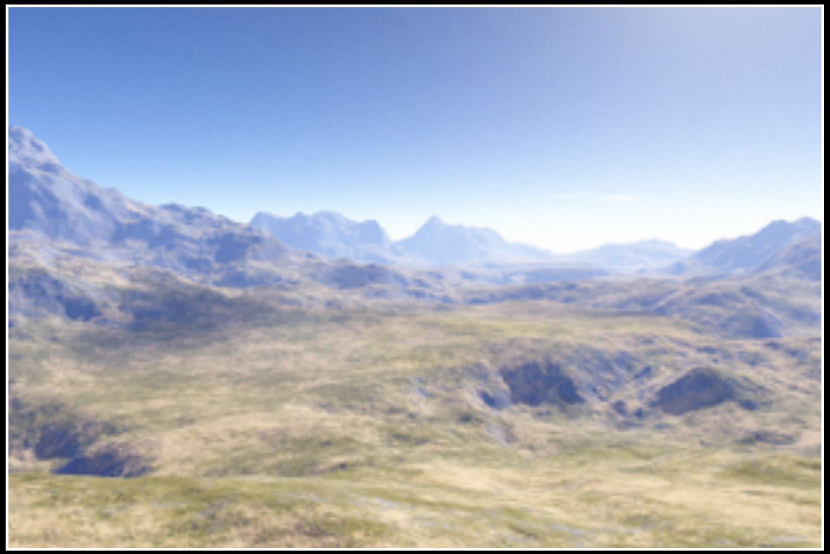


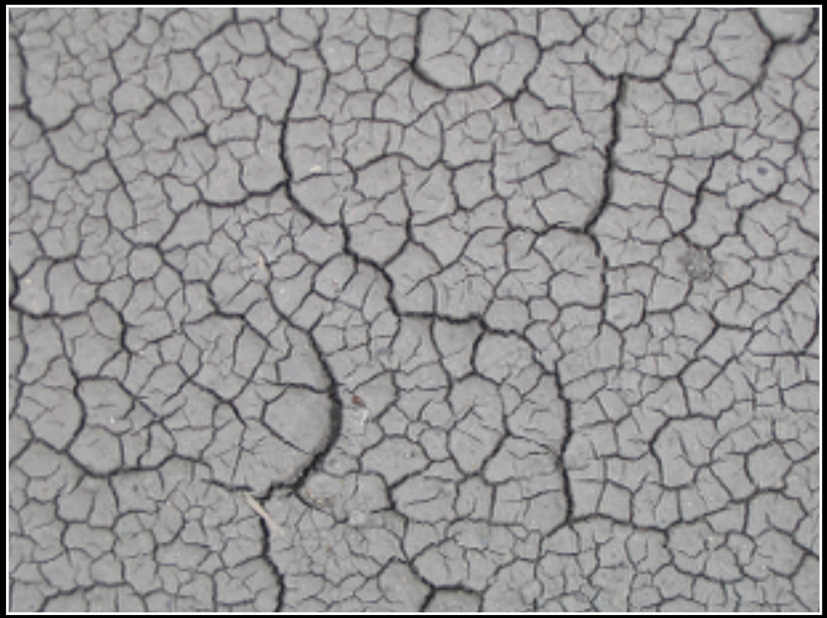
Fractal geometry

- Generation of form and/or texture.
  http://paulbourke.net/fractals/

- Random and fractal generators have been widely used to create game assets, objects for virtual environments, and in the animation/movie industry.

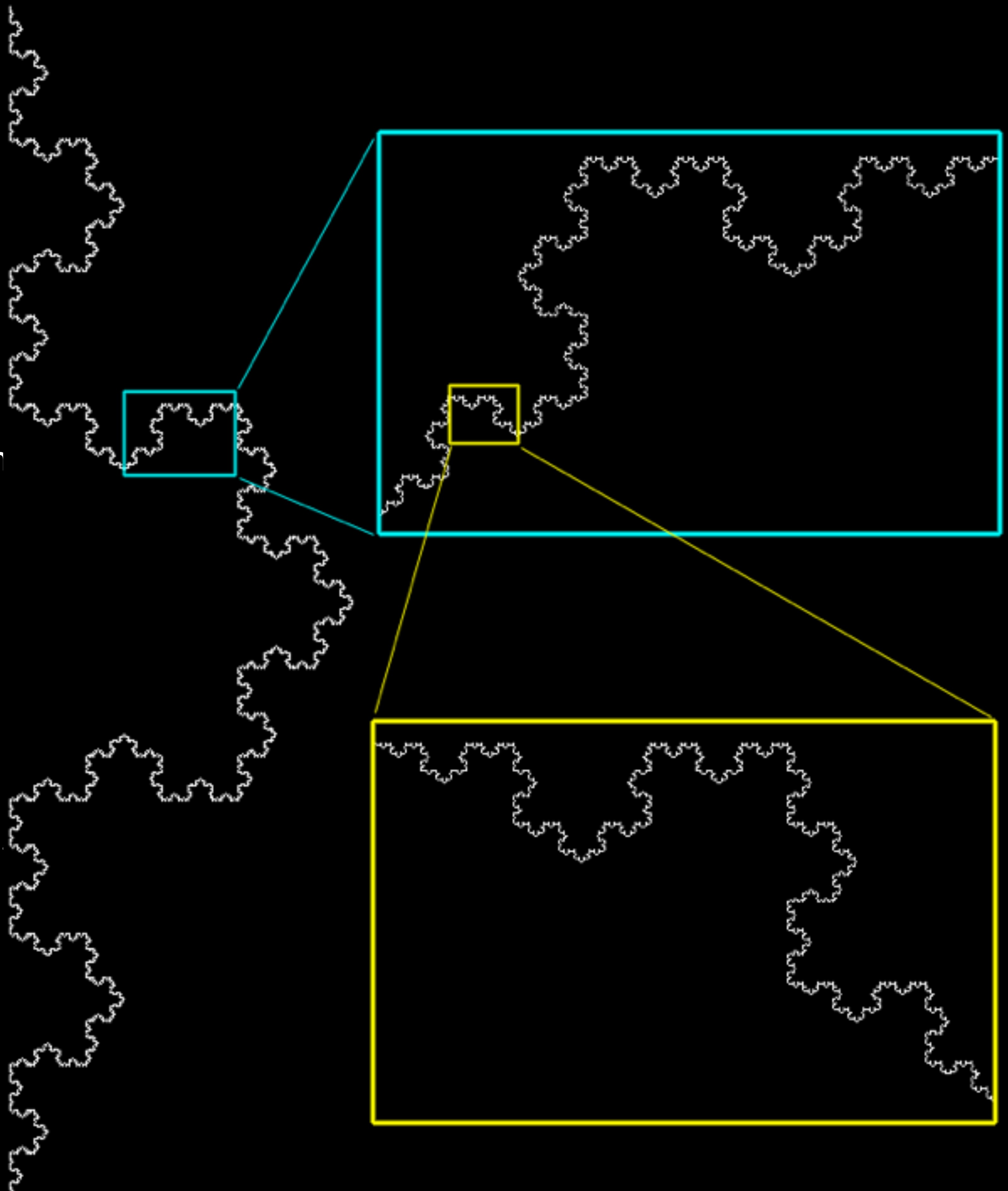
- Well know examples include
  - Perlin noise for clouds
  - landscape and terrain generation
  - fractal plant creation
  - ocean waves
  - texture generation





Fractal geometry

- Fractal geometry vs Euclidean geometry.
- Self similarity across scales.
- Euclidean geometry gets boring the more one zooms in.
- Koch snowflake is "self exact" across scales



Fractal dimension

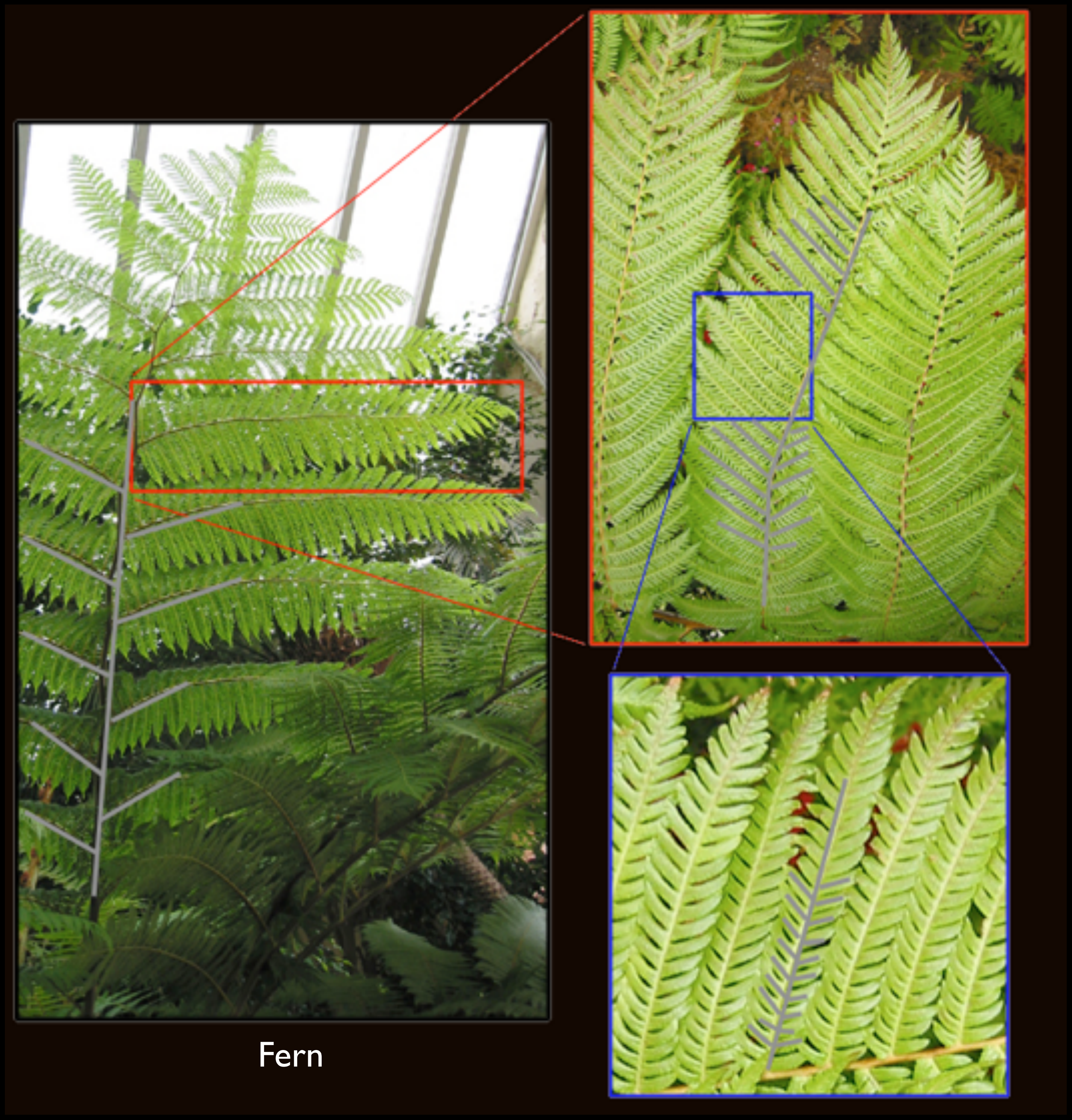
- Concept of dimension for Euclidean geometry is based upon the notion of scaling.
  If you scale a line by a factor of 2 its length increases by a factor of 2.
  If you scale a square by a factor of 2 its area increases by a factor of 4.
  If you scale a cube by a factor of 2 its volume increases by a factor of 8.

$$\text{quantity} = \text{scale factor}^{\text{dimension}}$$

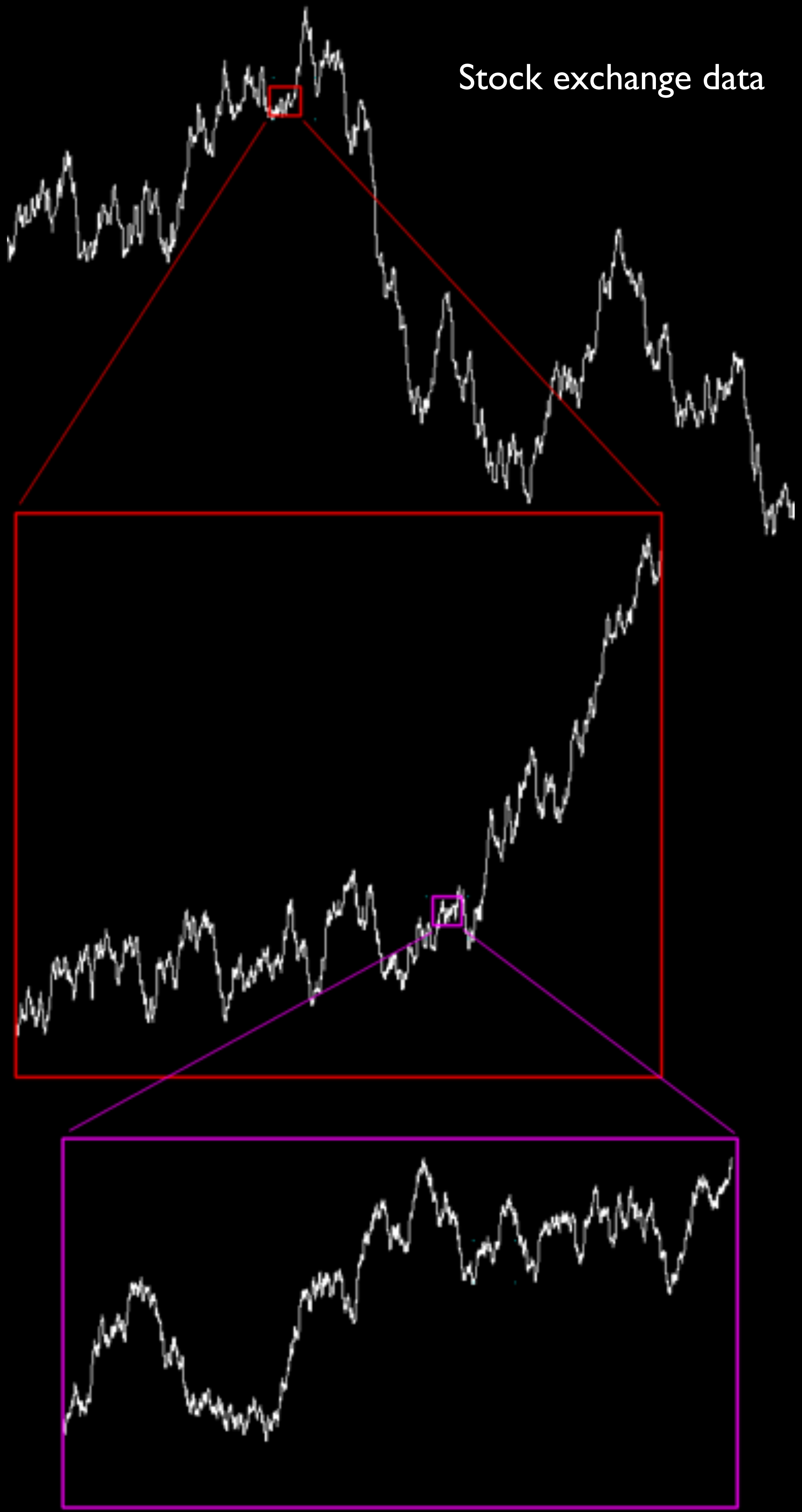
- There are geometric shapes where the dimension power above is not an integer (1,2,3) but rather some fractional amount.
- These are called fractals.
- There are idealised mathematical fractals, natural objects are generally approximations to this, but in the same way as a sphere in nature (eg: moon) is only an approximation to an Euclidean sphere.

Self similarity

- Real objects are generally approximations to mathematical fractals.
- Generally across a smaller range of scales.



Fern



Stock exchange data

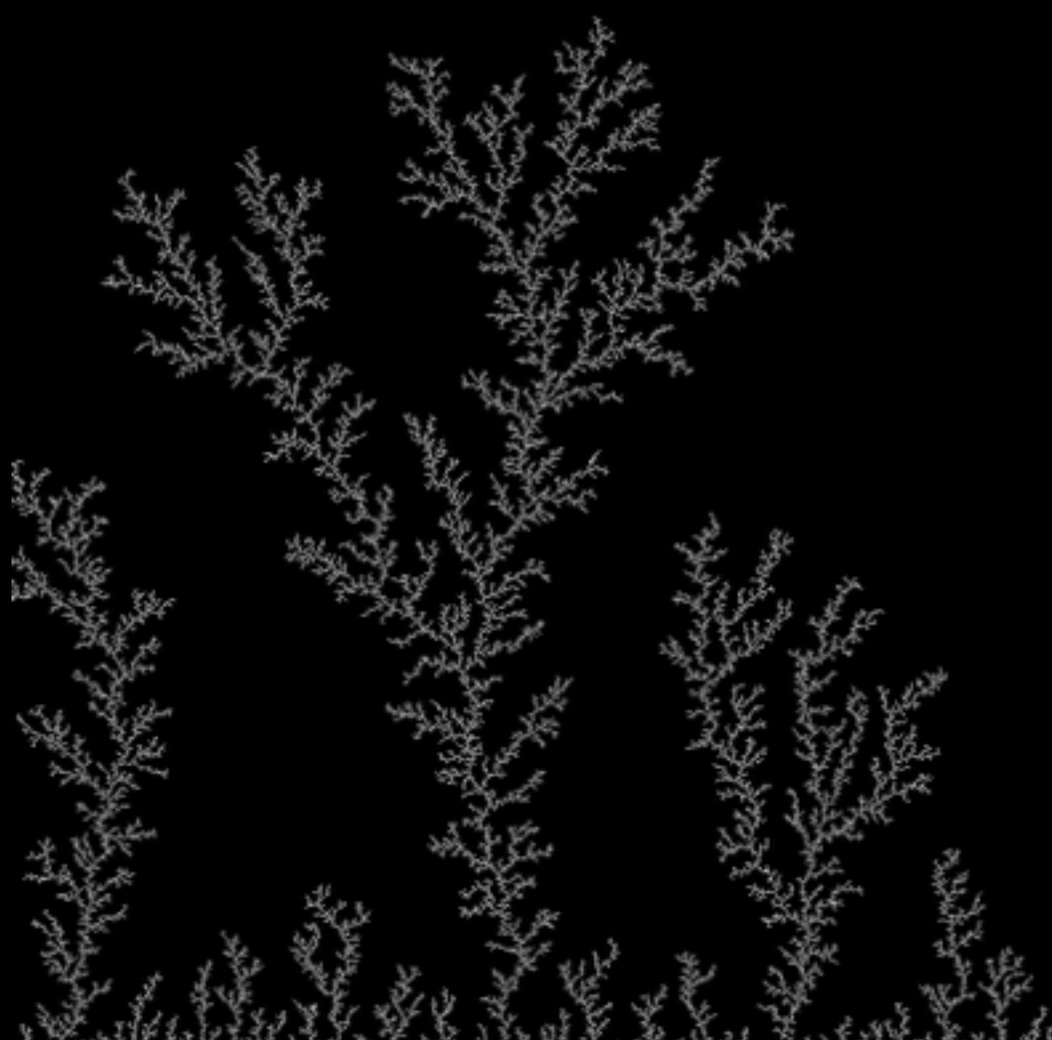
Self similarity

- Consequence of self similarity is the difficulty of judging the scale of an object without context.

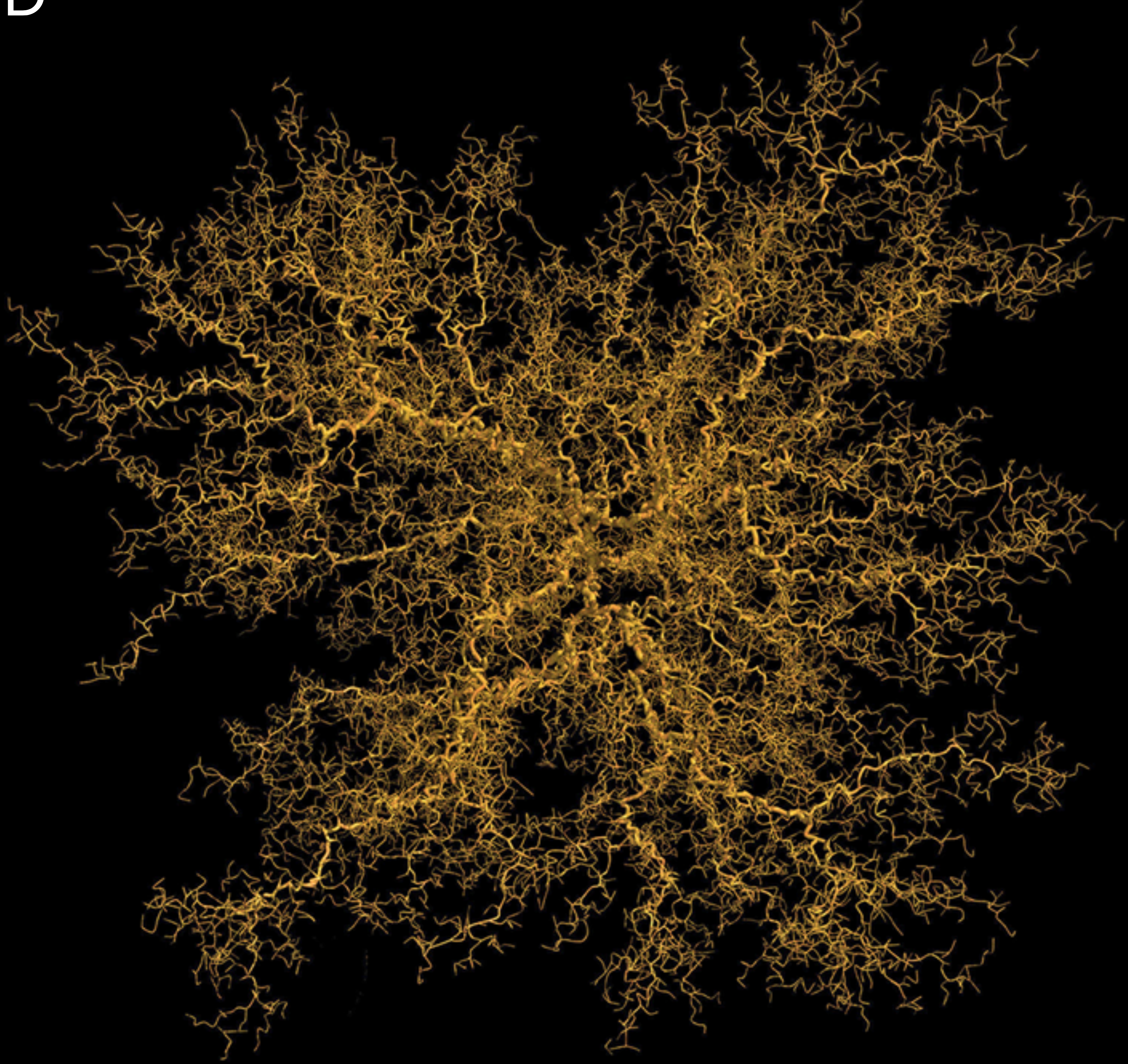
Rough stone? Rocky beach? Small cliff? Mountain side? Aerial view?

Diffusion limited aggregation (DLA)

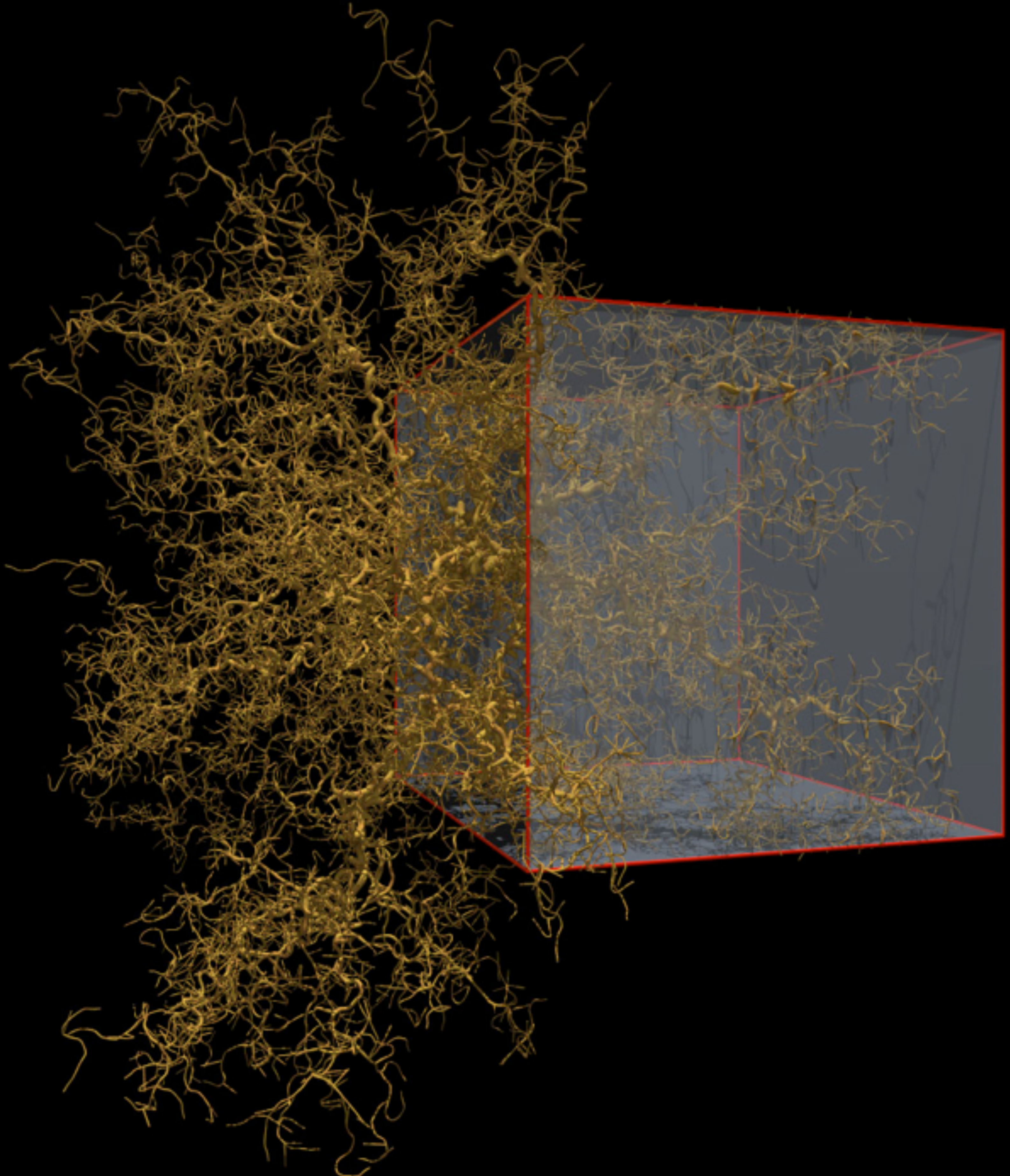
- Model for a number of processes in chemistry.
- Algorithm
  1. Particles enter the plane from the surroundings.
  2. Move around randomly.
  3. When they strike another particle they stop.
  4. Got to step 1.
- Drunk analogy.



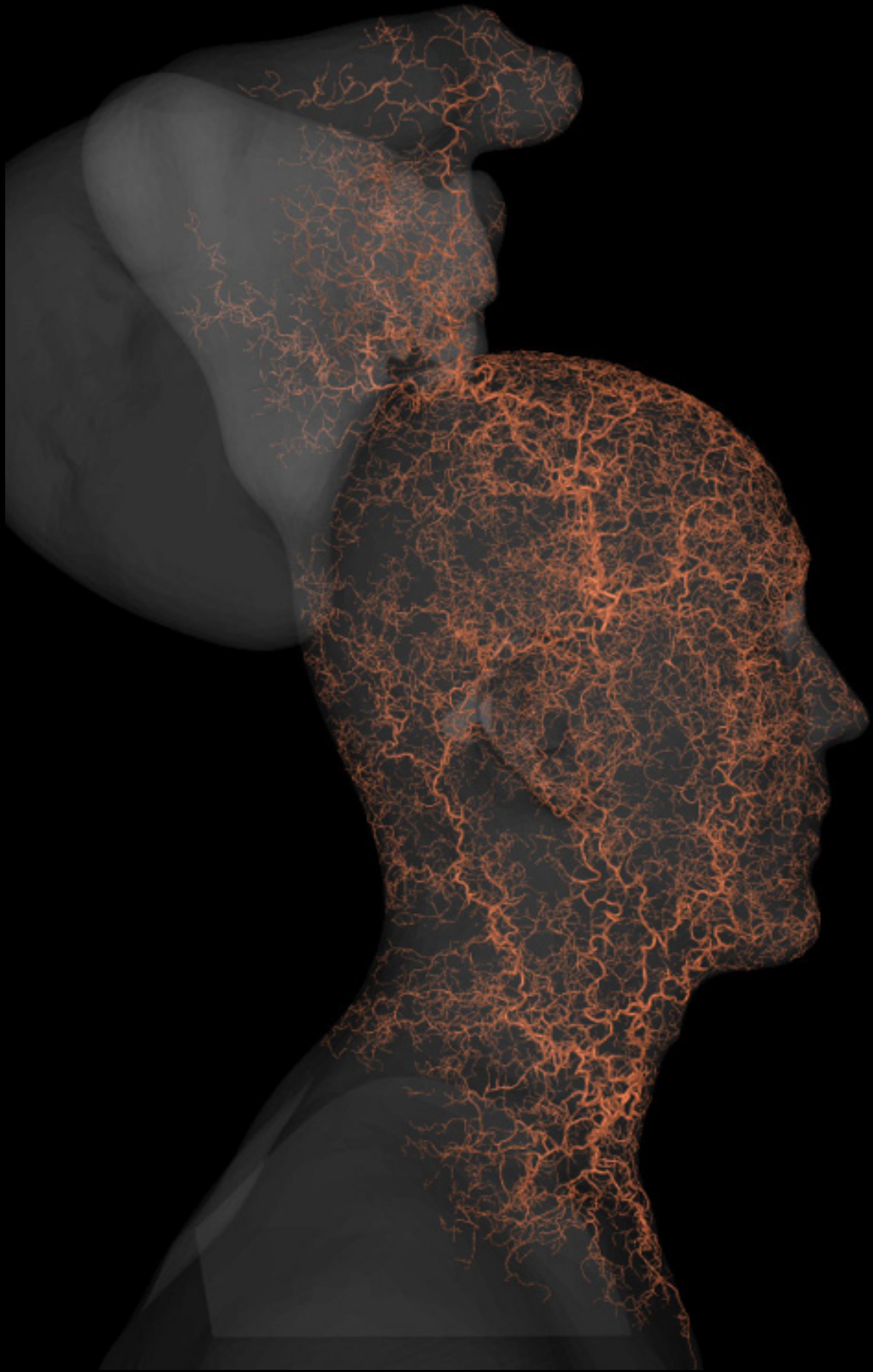
DLA in 3D



Constrained DLA



Constrained DLA



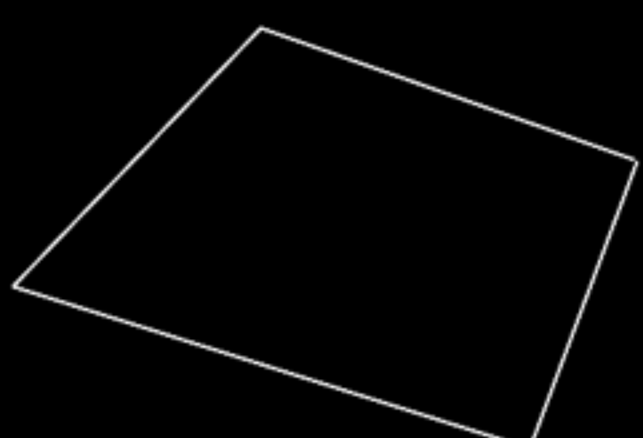
Model for river systems, Egypt

Landscape and planet modelling

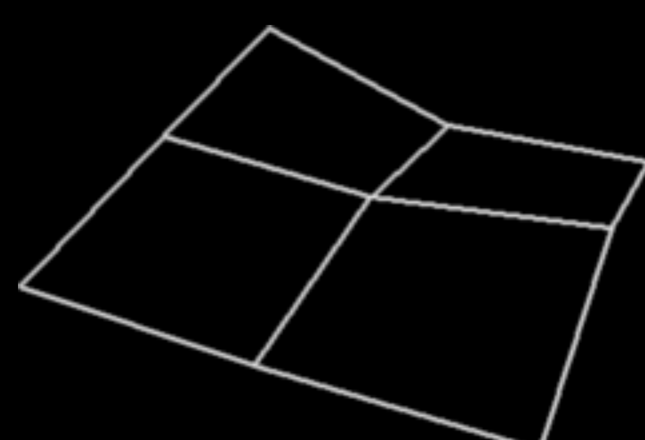
- Unlike Euclidean parametric modelling described earlier, the parameters for fractal process are generally very few yet result in infinitely detailed objects.

- Are generally algorithmic rather than close equations, need to be computed rather than just a given.

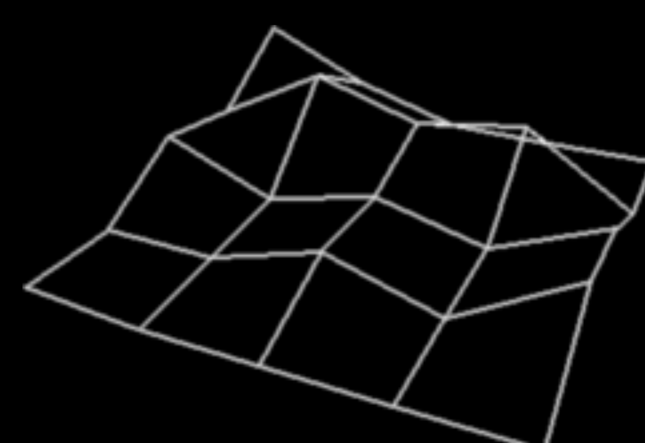
- Random displacement algorithms.
  1. Start with a single polygon
  2. Subdivide into each polygon into 4 pieces
  3. Random perturb each vertex
  4. Reduce the amount of random variation
  5. Go to 2



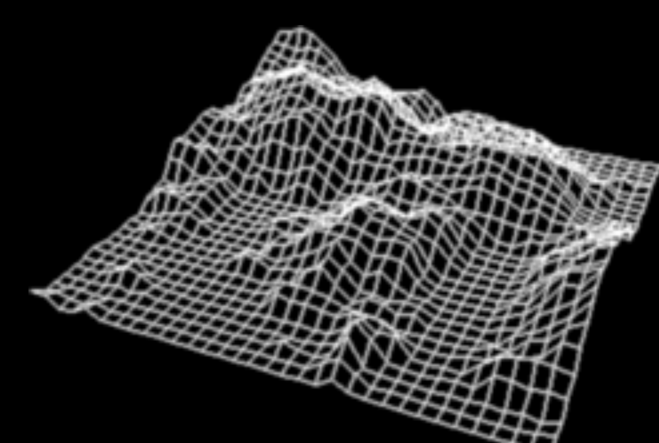
Initial polygon



Iteration 1

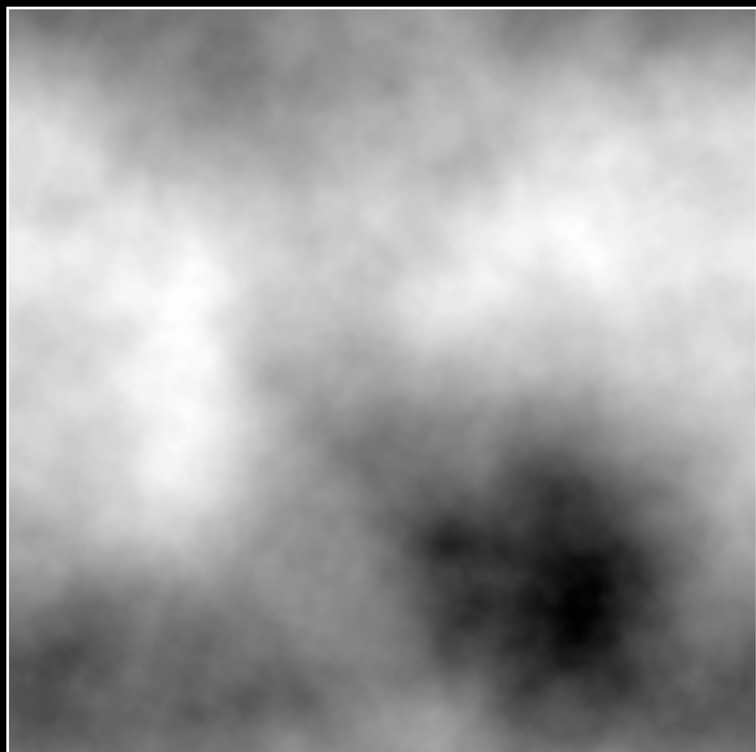


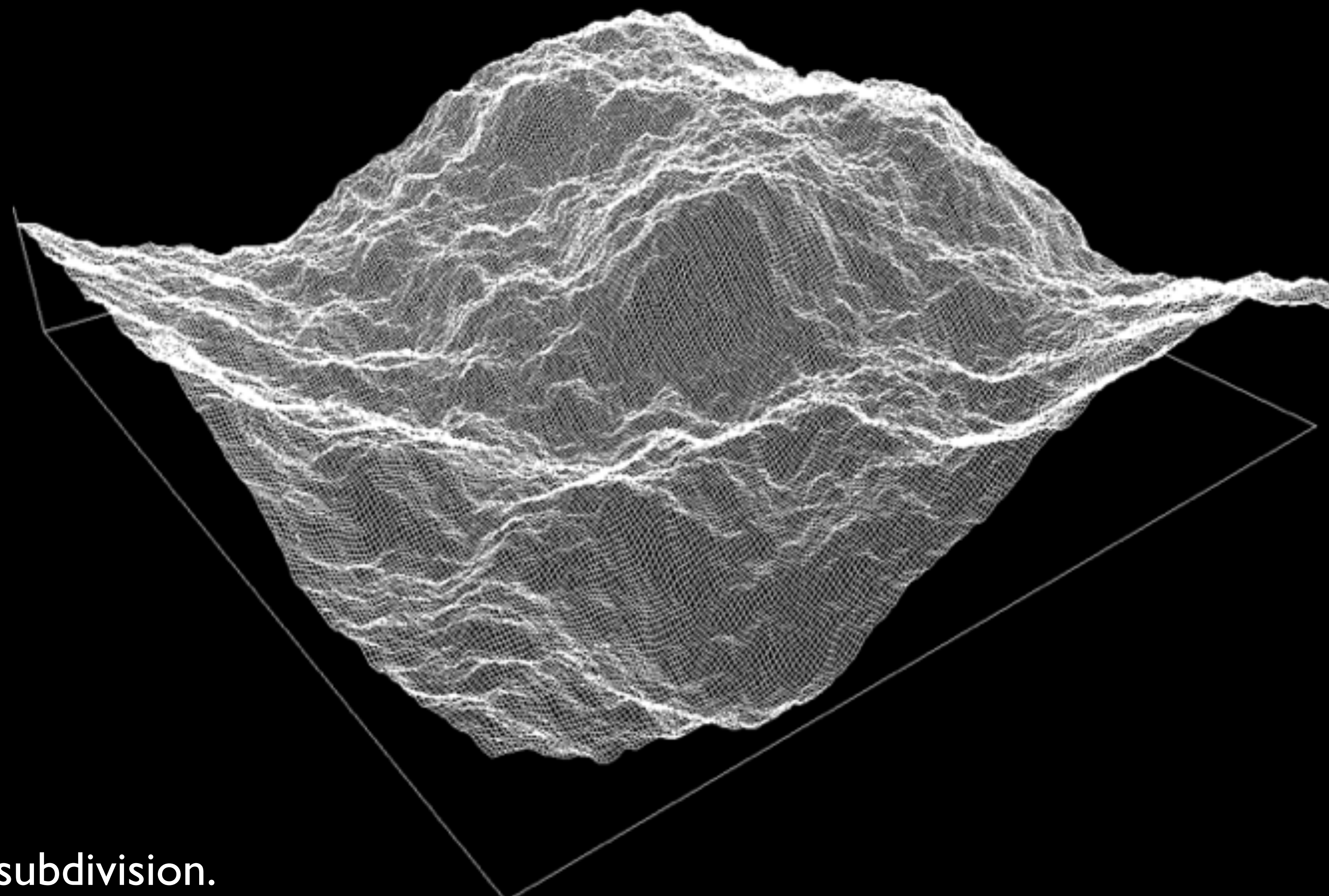
Iteration 2



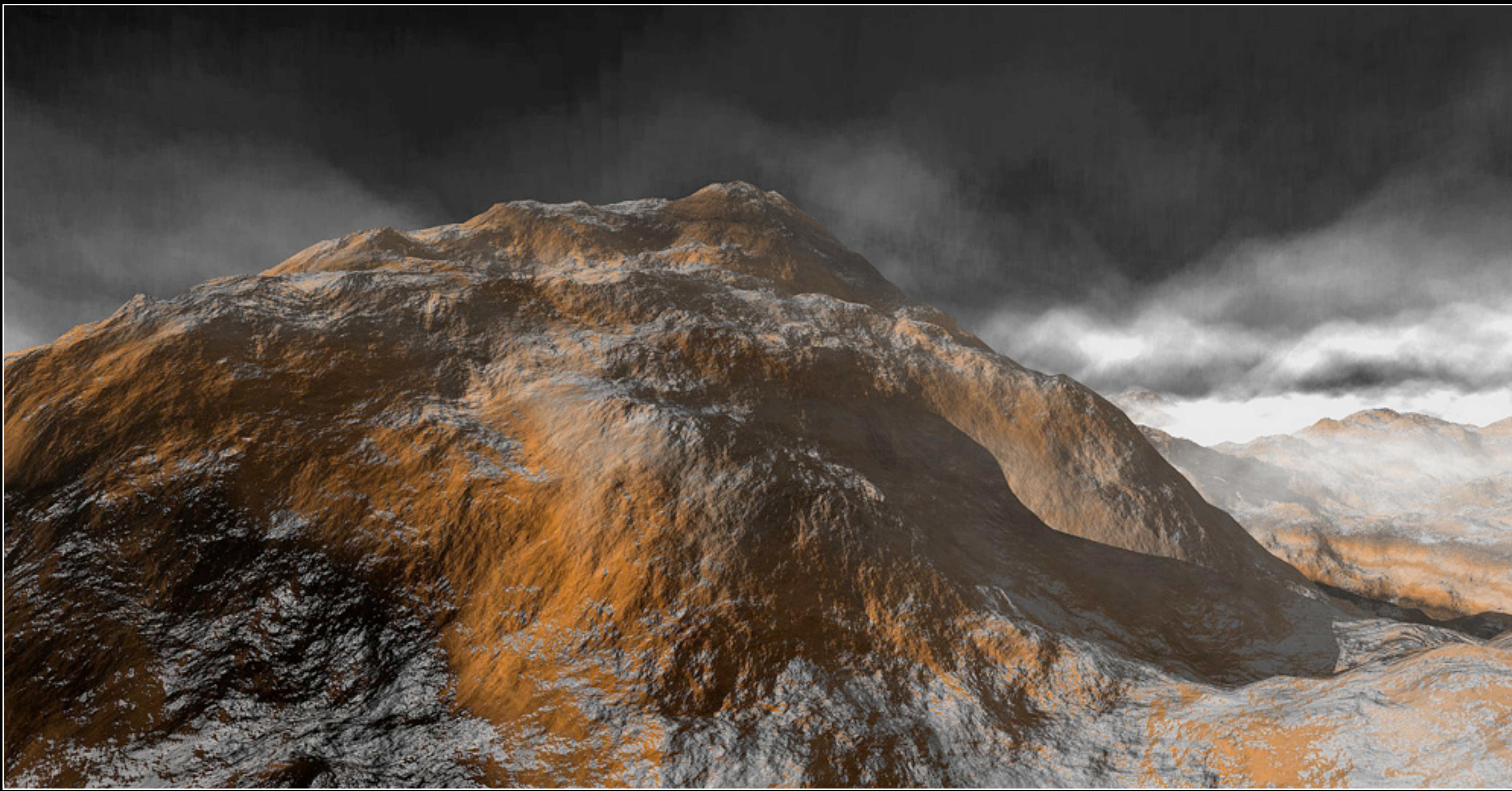
Iteration 5

Landscape modelling



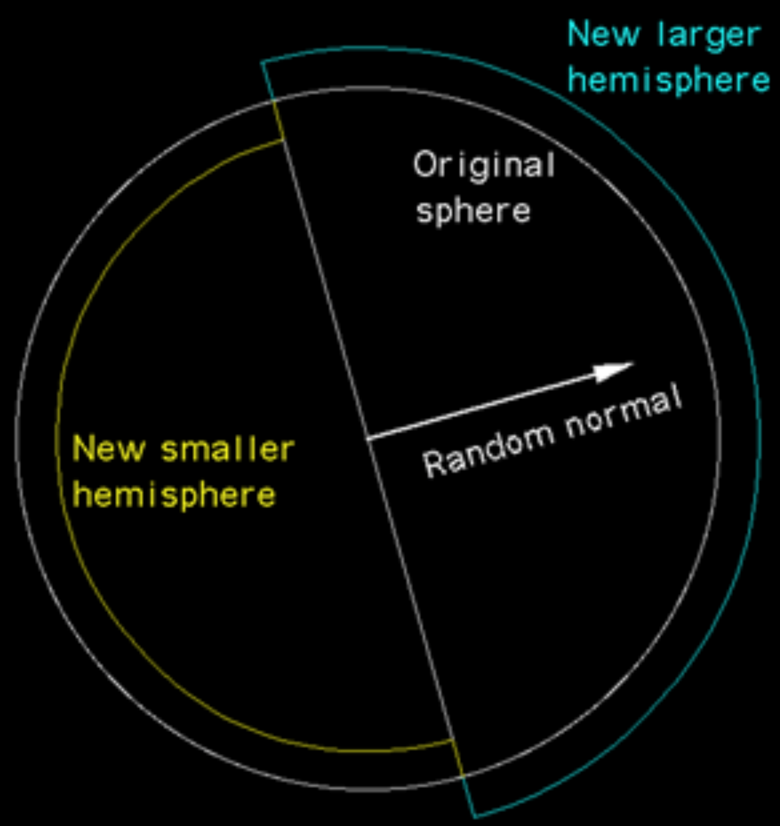


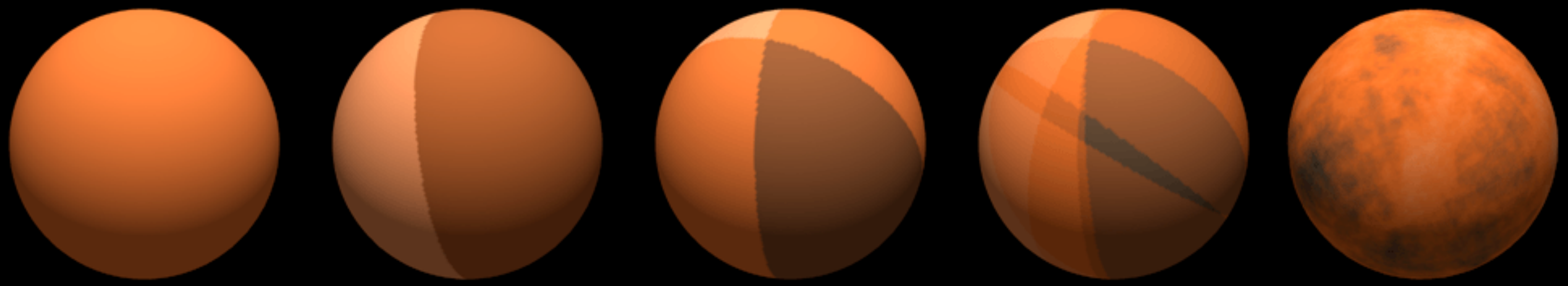
- Often called midpoint subdivision.
- Important characteristic is one can refine the detail to any arbitrary level, view from space or stand on the surface.



Planet modelling

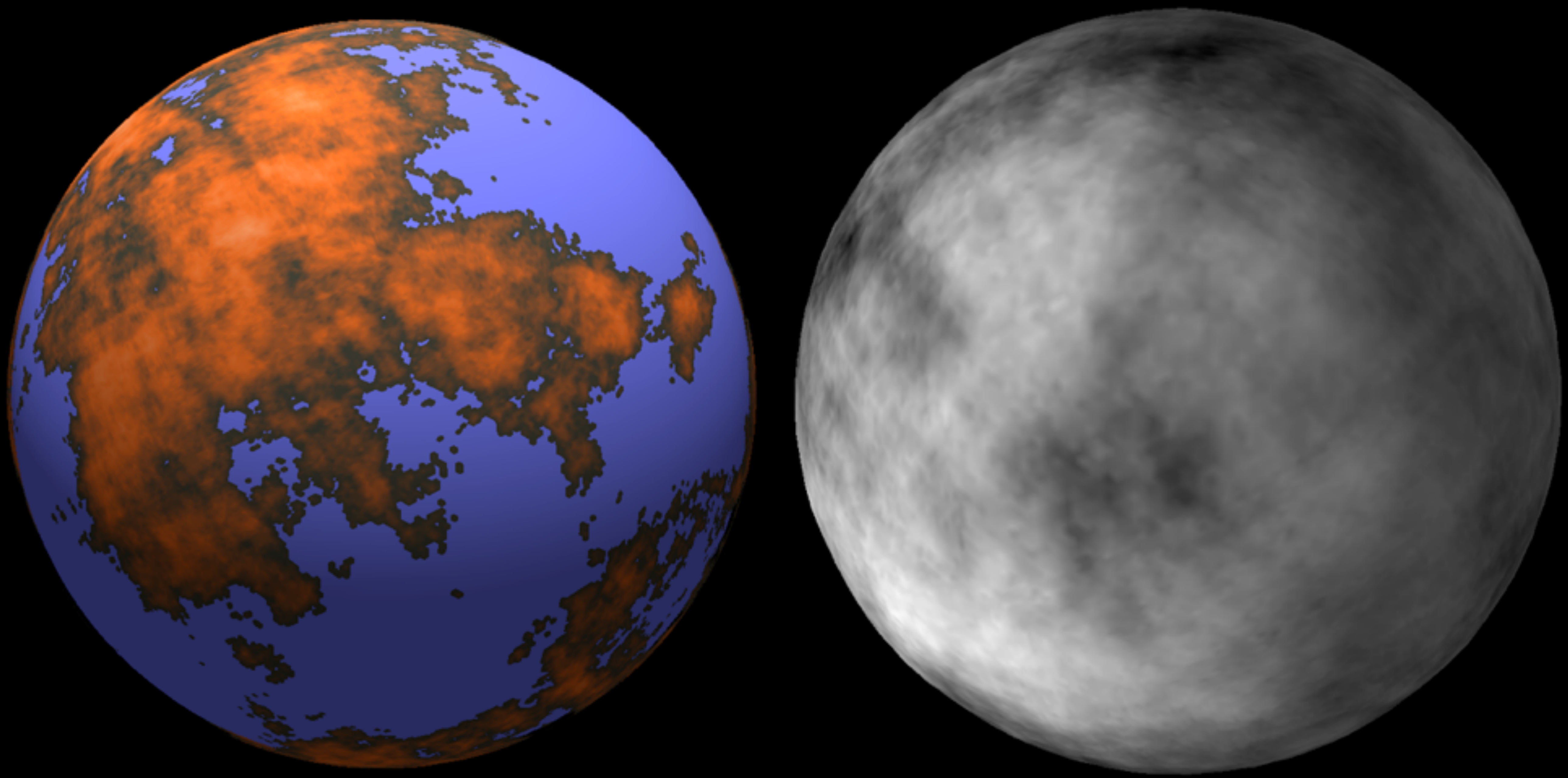
- Radial displacement algorithm.
- Similar idea except applied to a sphere rather than a plane.
- On each iteration split the sphere at a random angle and displace each half of the surface.





Initial sphere | Iteration 1 | Iteration 2 | Iteration 10 | Iteration 1000

Planet modelling, or planet cloud modelling



Lindenmayer Systems (L-Systems)

- Early seminal publication
  Lecture Notes in Biomathematics" by Przemyslaw Prusinkiewcz and James Hanan.

- Based upon so called "rewriting rules".

- A string of characters is rewritten based upon replacement rules.
  Some characters are given graphical meaning.

- Example of an initial string (axiom) might be: F+F+F+F
  A rewriting rule may be: F --> F+F-F-FF+F+F-F
  That is, on each iteration replace every instance of F with F+F-F-FF+F+F-F

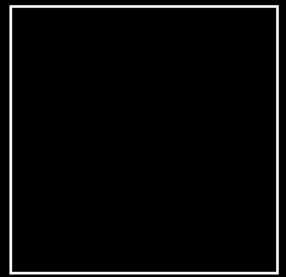
- So first iteration would be a new string:
  F+F-F-FF+F+F-F + F+F-F-FF+F+F-F + F+F-F-FF+F+F-F + F+F-F-FF+F+F-F

- Second iteration would result in the string: F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF + F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-FF+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F+ F+ F-F-FF+ F+ F-F-F+ F-F-FF+ F+ F-F

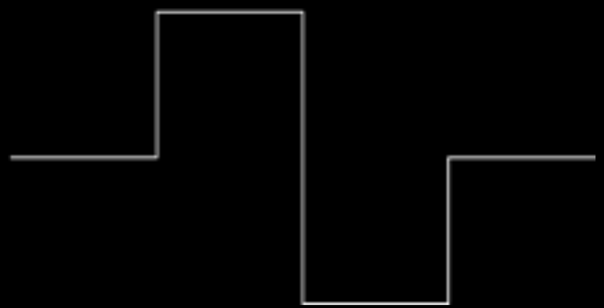
- And so on ...

L-Systems

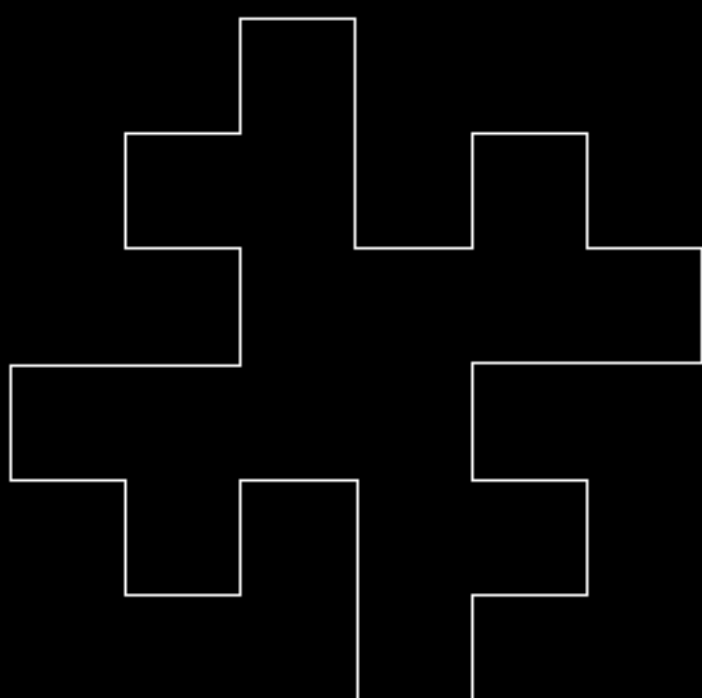
- If the characters are given meaning such that:
  "F" means move forward one unit.
  "+" means turn 90 degrees clockwise
  "-" means turn 90 degrees anticlockwise
  There is a standard library of character meanings.
- Then the axiom F+F+F+F is just a square
- Replacement rule F --> F+F-F-FF+F+F-F is
- Originally called "turtle graphics".



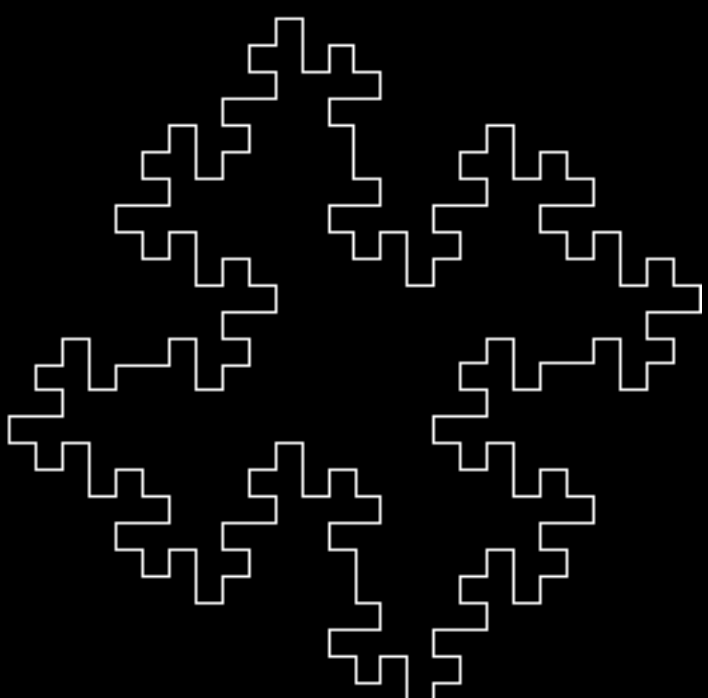
Axiom



Replacement rule



First iteration



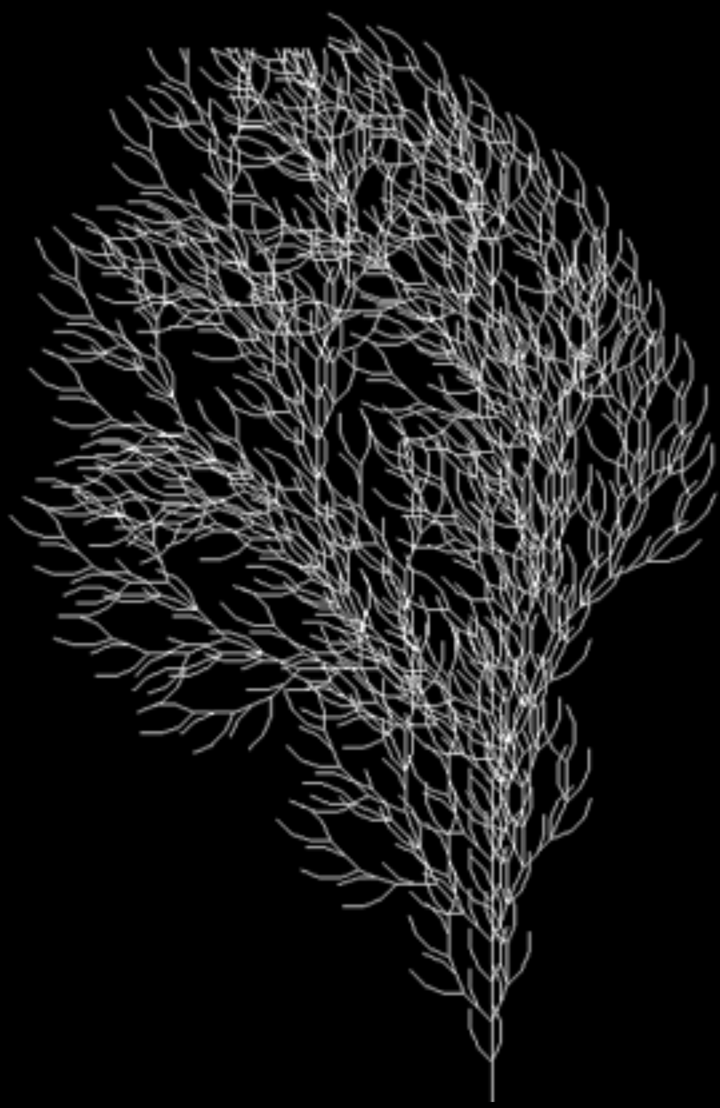
Second iteration

Examples: Tree structures

Axiom = X
F --> FF
X --> F-[[X]+X]+F[+FX]-X
angle = 22.5

Axiom = X
F -> FF
X -> F[+X]F[-X]+X
angle = 20

Axiom = F
F -> FF+[+F-F-F]-[-F+F+F]
angle = 22.5



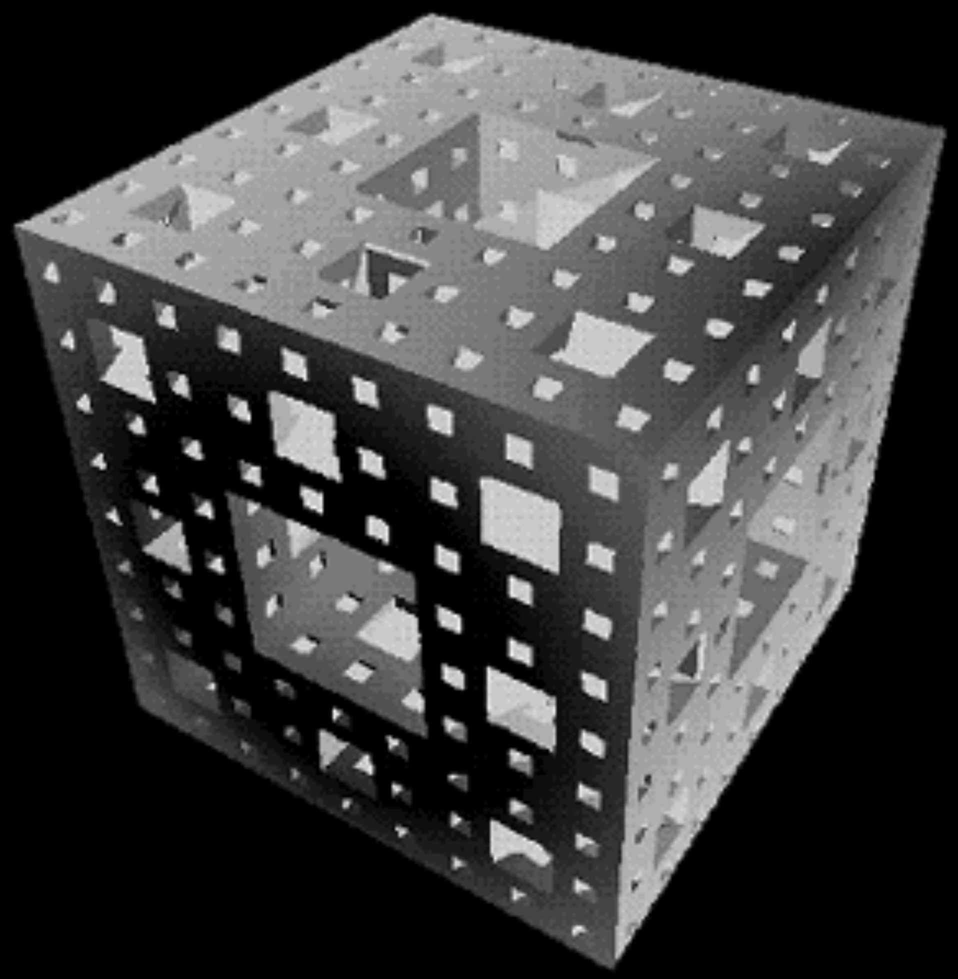
Examples in 3D

3D replacement rules

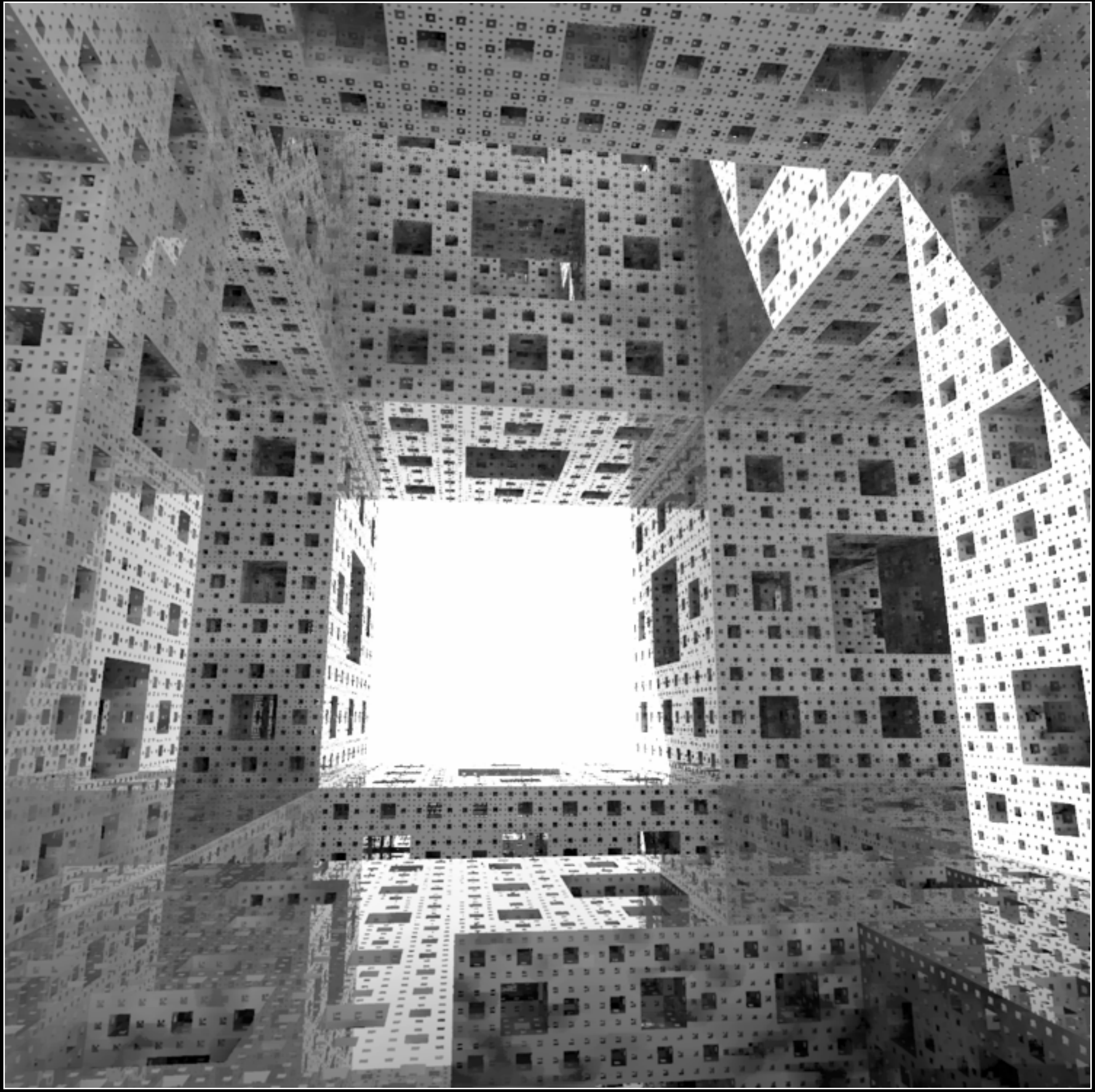
- L-System principle can be extended into 3D and rules simplified to iteratively replacing shapes with other shapes.
- Well known example is the Menger sponge.
- The replacement shape is the same as the original shape just scaled.

Axiom

Replacement rule



Second iteration



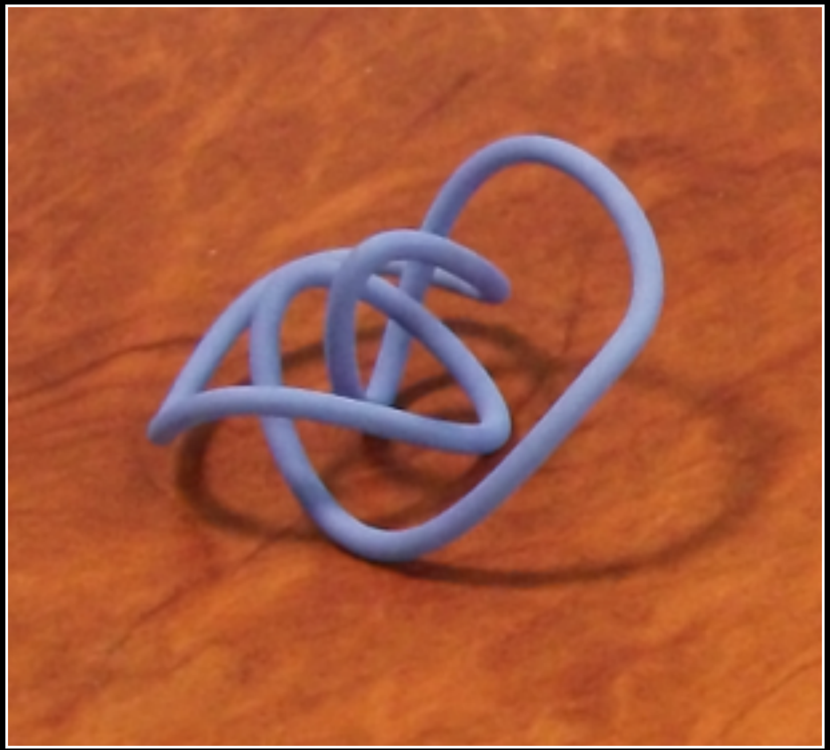
Technology #1: Rapid prototyping (RP)

- Has been available for 20+ years but the technology was expensive and imposed constraints on what could be built.
- Increasingly accessible process.
- A number of online bureau services.
- ZCorp revolutionised the process perhaps 10 years ago with their range of colour 3D printers.
- Now two main players/technologies
  - Zcorp: colour but limited structurally
  - ObJet: highest resolution but monochrome
- Still limitations to colour fidelity.

Constrained DLA

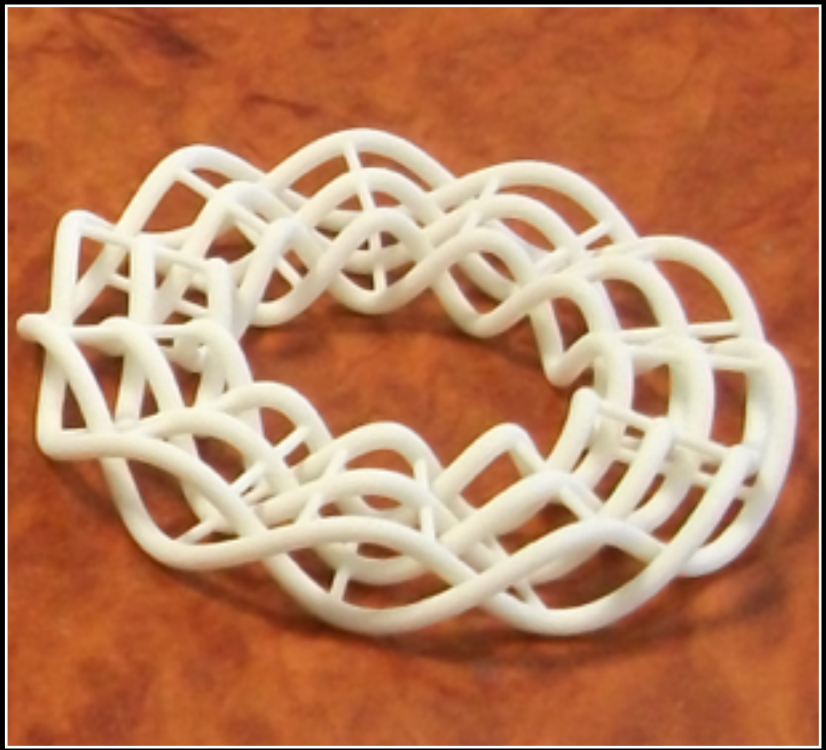
RP in visualisation

- Visualisation is the process of applying advanced computing techniques to data in order to provide insight into the underlying structures, relationships and processes.
- The range of printable objects has increased rapidly in the last 6 years and continues to expand.
- RP allows us to explore geometry in the same way as we explore objects in real life.



Visualisation in knot theory

Printing in metal



Visualisation in neuroscience

Examples: Chemistry

Series of peptides (Chemistry UWA)

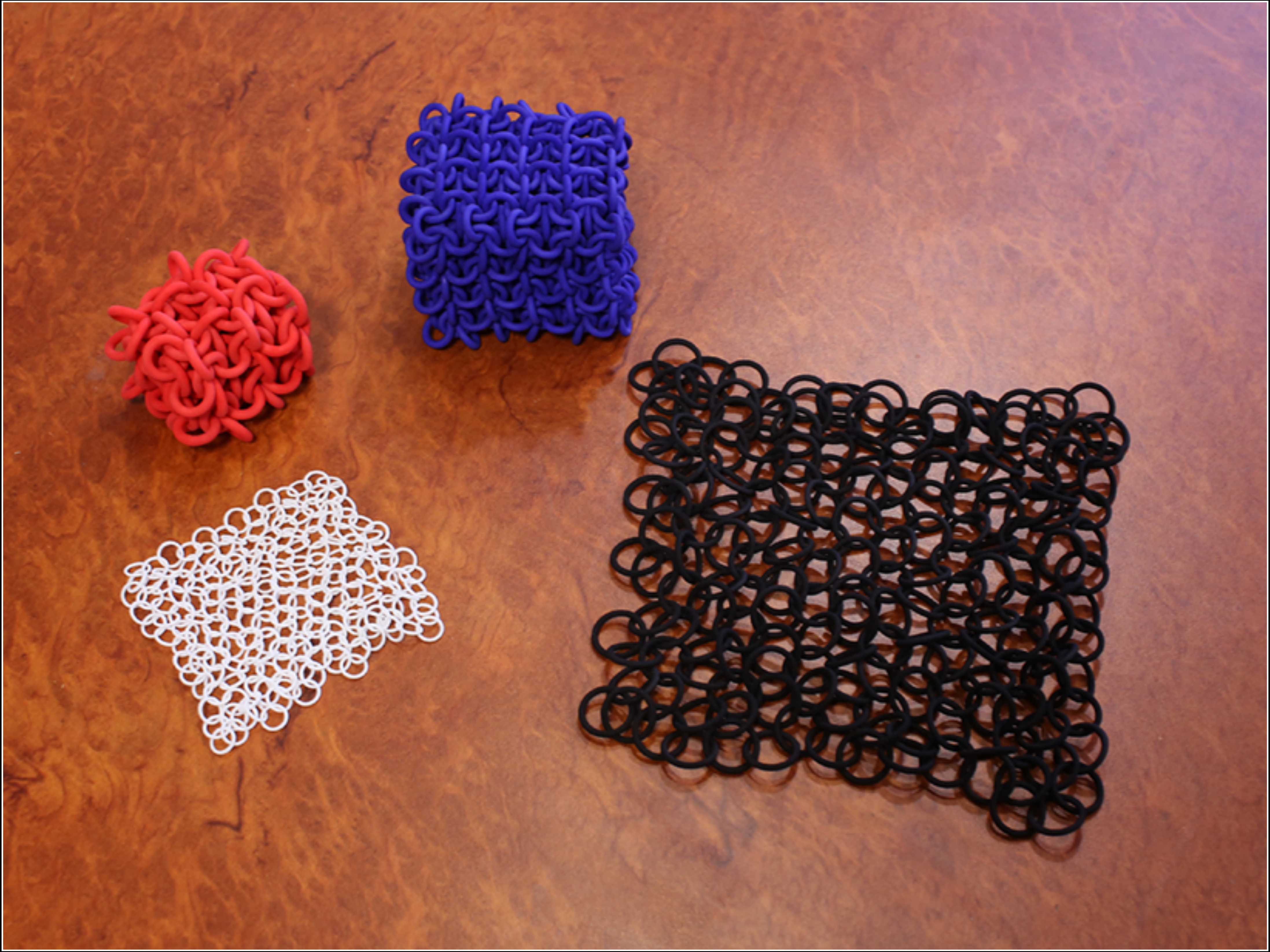
Examples: Geoscience

Fossils (Geology)

Examples: Mining (reconstructions, see later)

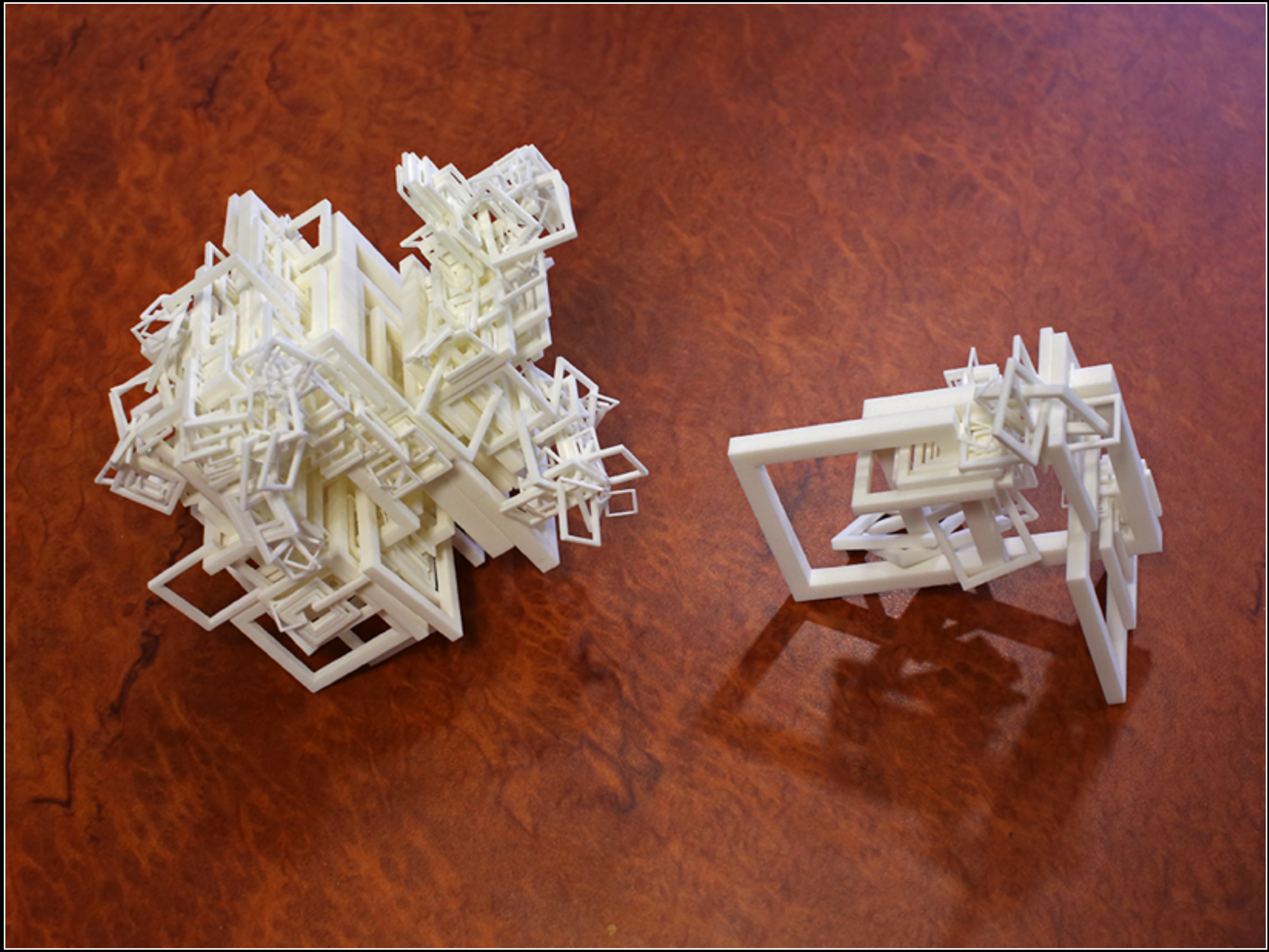
Mine pits (Geoscience)

Examples: Mathematics



2D and 3D chainmail

Examples: Mathematics, packing theory



Packing of square torii

Technology 2: 3D reconstruction from photographs

- Photogrammetry: The process of deriving some 3D quantity from a collection of photographs.

- Algorithms have developed significantly in recent years.

- A relatively low number (< 10) of photographs are required for 2.5D objects.

- Full 360 objects may require more than 30 photographs.

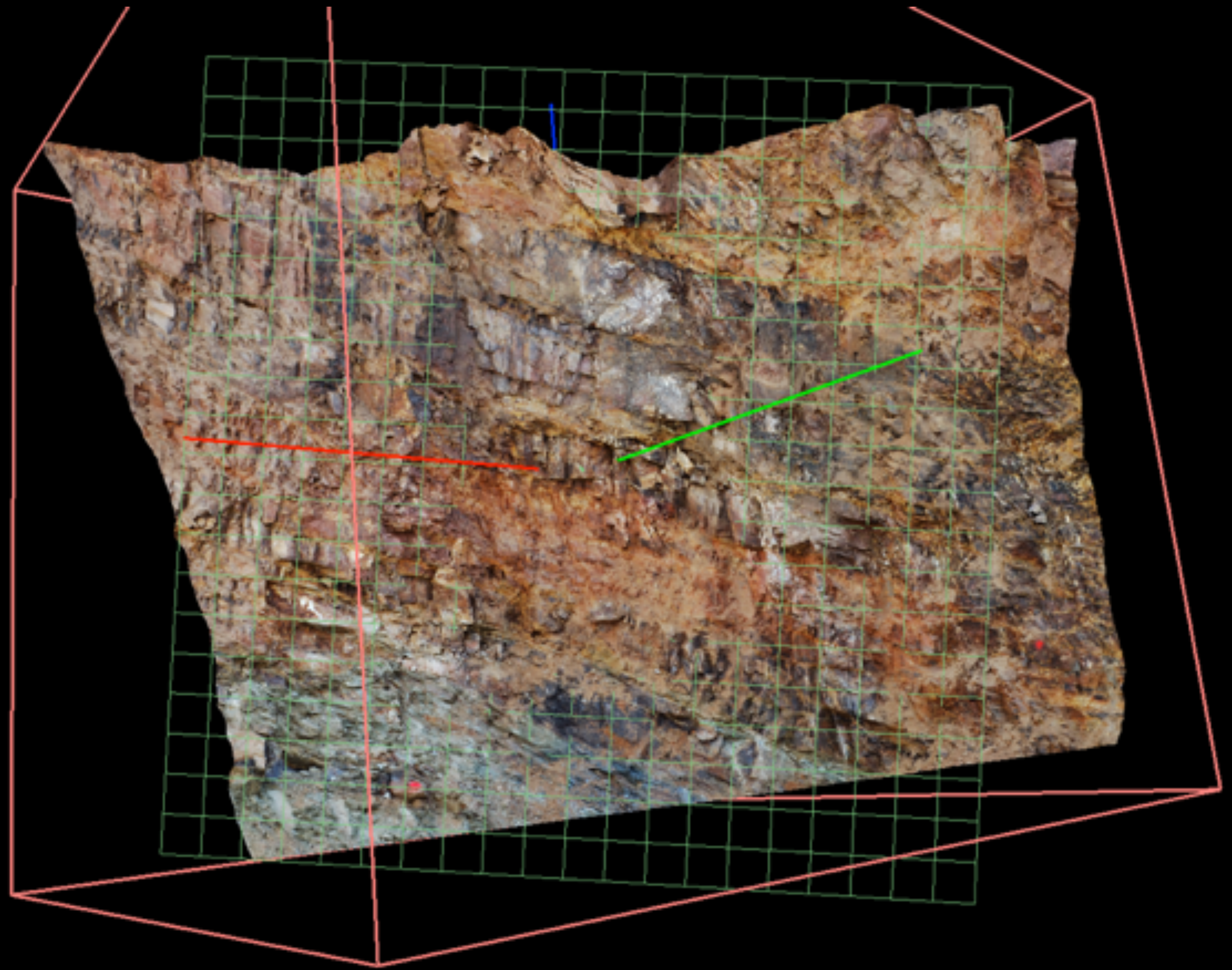
- Some skill in image capture to achieve best results.
  - requires fixed focal (prime lens), as such point and click cameras can be used.
  - cannot reconstruct what is not photographed (obviously).
  - no point taking multiple photographs from the same position.

- Online services exist for performing the reconstruction, eg: 123D Catch from AutoDesk and PhotoSynth by MicroSoft ... but are generally not leading edge.

- There are a number of open source implementations of "bundler" as well as affordable commercial solutions such as PhotoScan.

Motivations

- Creating 3D textured assets for virtual worlds and games.
  Avoid time consuming 3D modelling.
- Recording objects of historic or archaeological importance.
- Non-intrusive capture for medical and forensic applications.
- Capturing geological structures for analysis.



Other technologies

- In some areas it is starting to replace technologies such as laser scanning. LIDAR - light detection and ranging.
  - particularly so for capture of object in difficult locations
  - only requires a modest investment

- Another technology are so called depth cameras
  - Primesense (eg: Kinect)
  - Structured light techniques (eg: Artec Scanner)

- The above do have some advantages
  - LIDAR generally gives better accuracy
  - Structured light can cope with (limited) motion unlike photogrammetry and LIDAR.

- Future: Light field cameras (plenoptic camera).

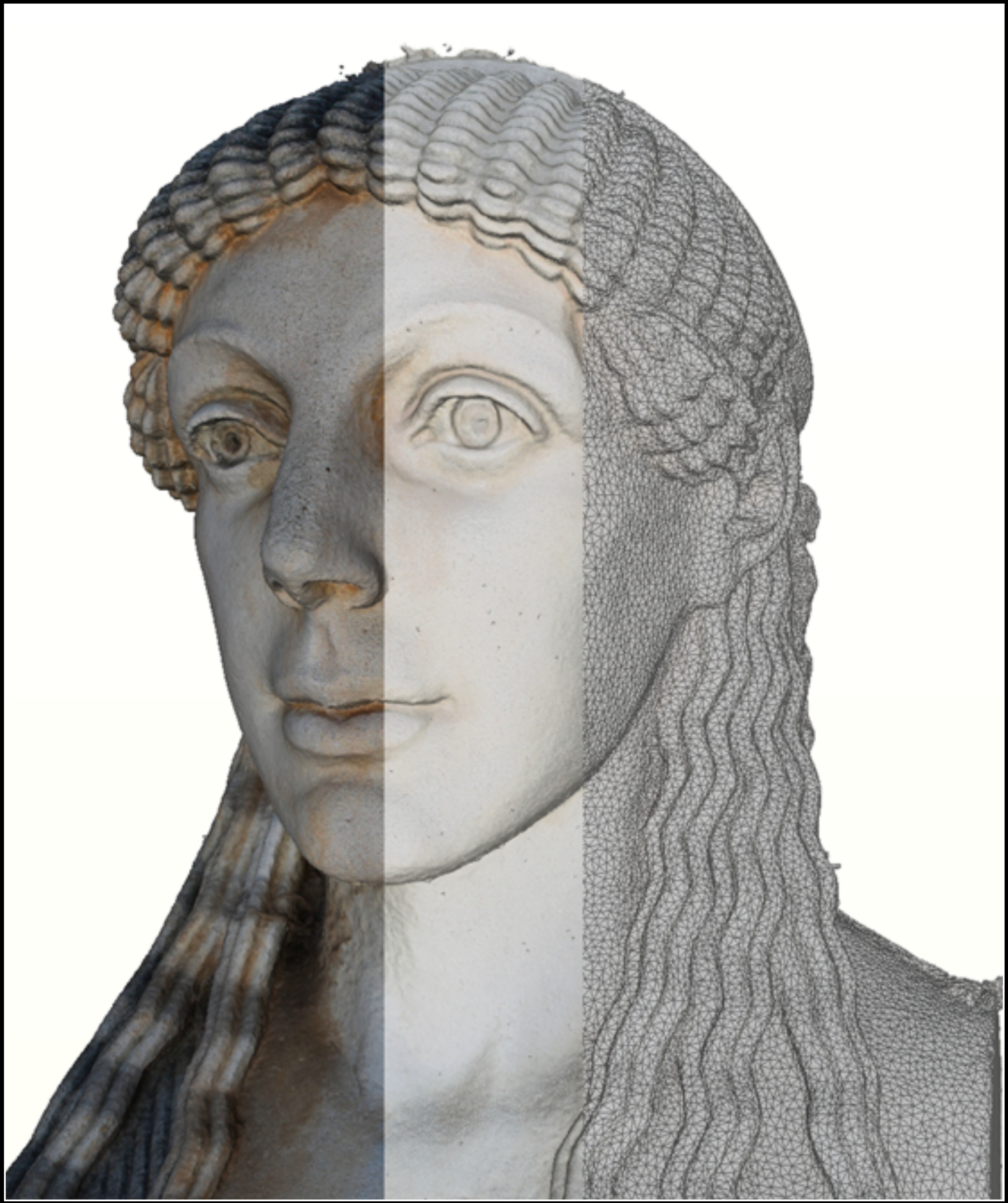
LIDAR

Structured light

Example: Indian temple engraving

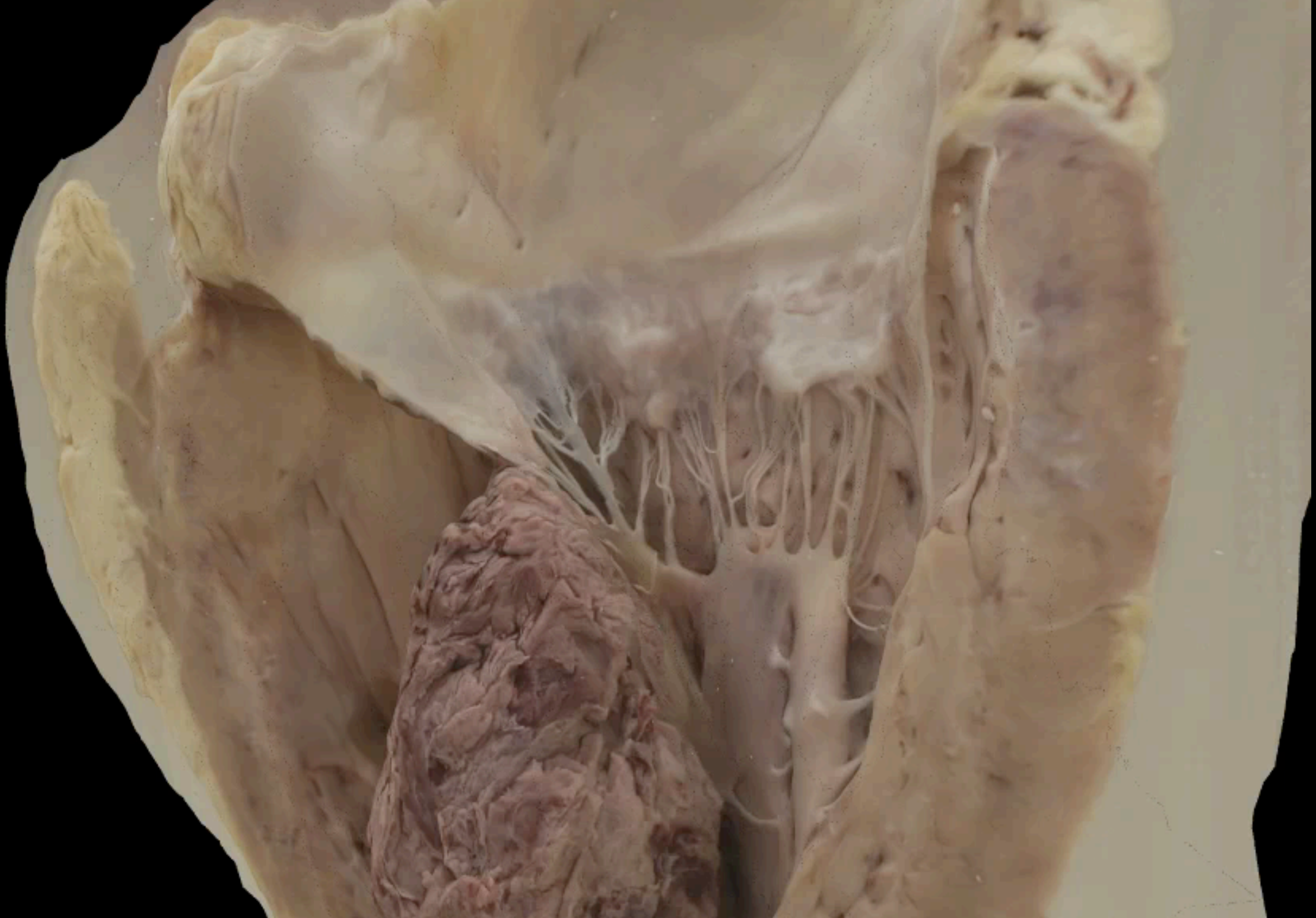
Example: Aphrodite (UWA)

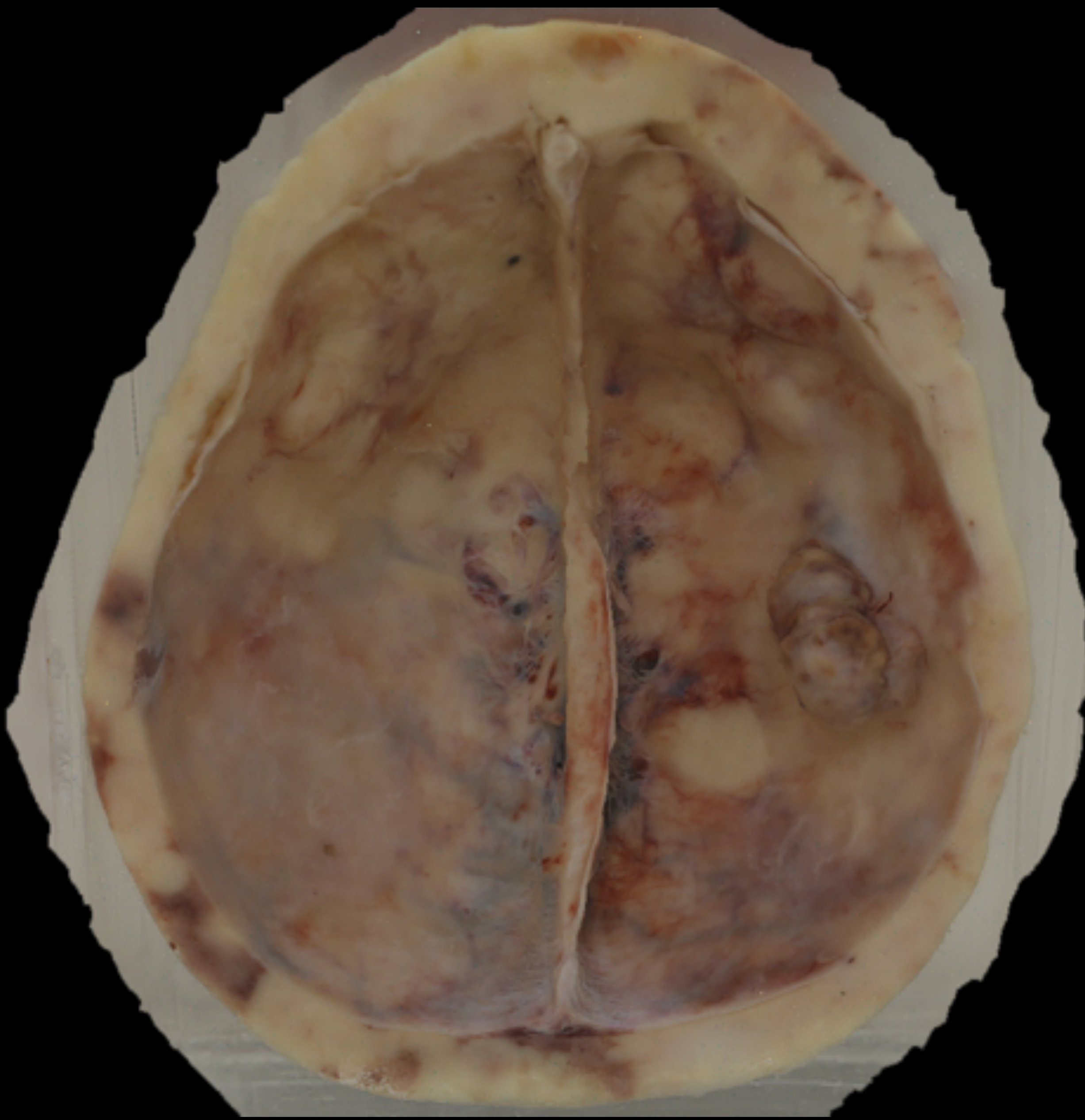
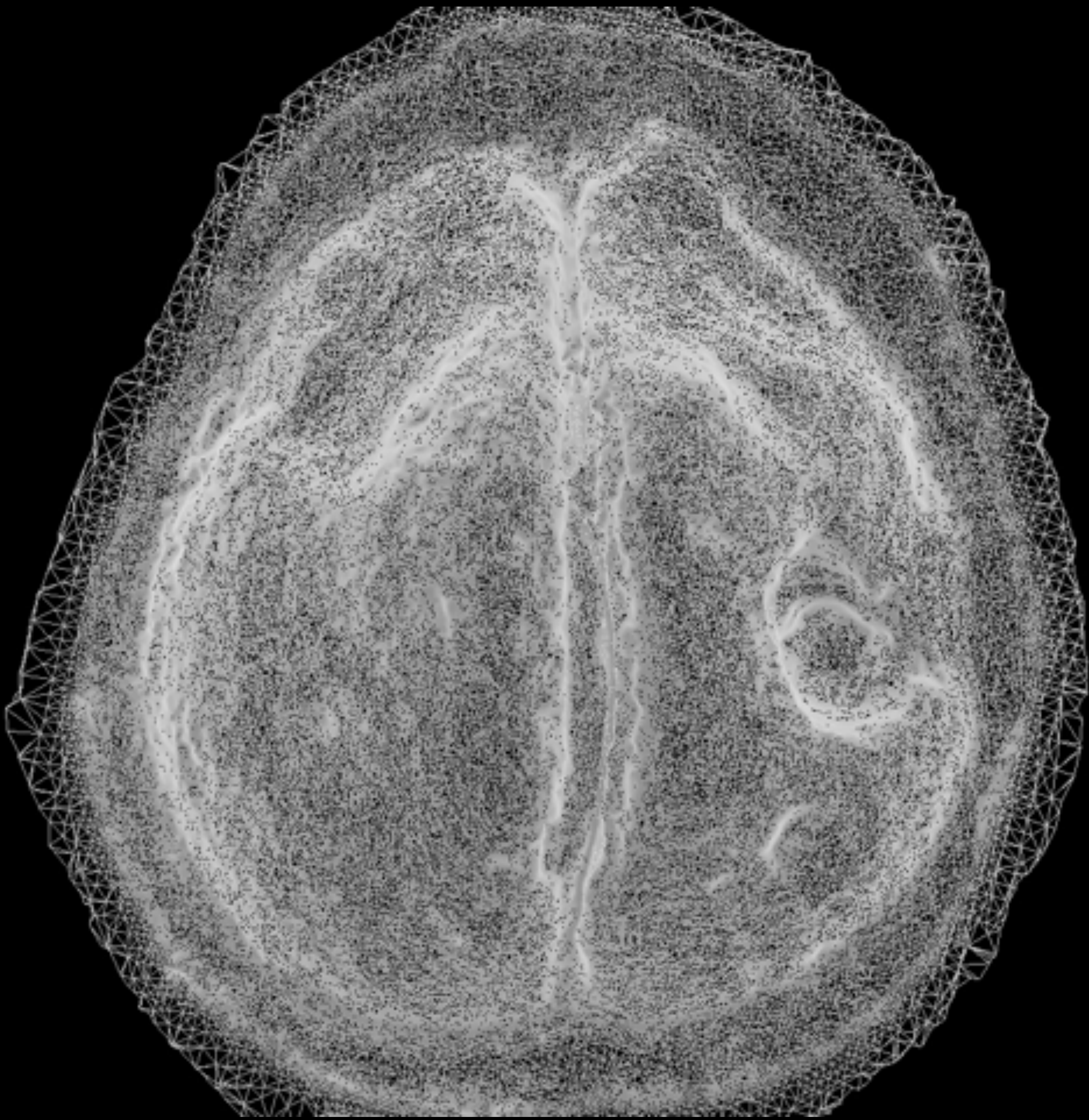
Example: Aphrodite (UWA)



Examples: Mining

Example: Archaeology

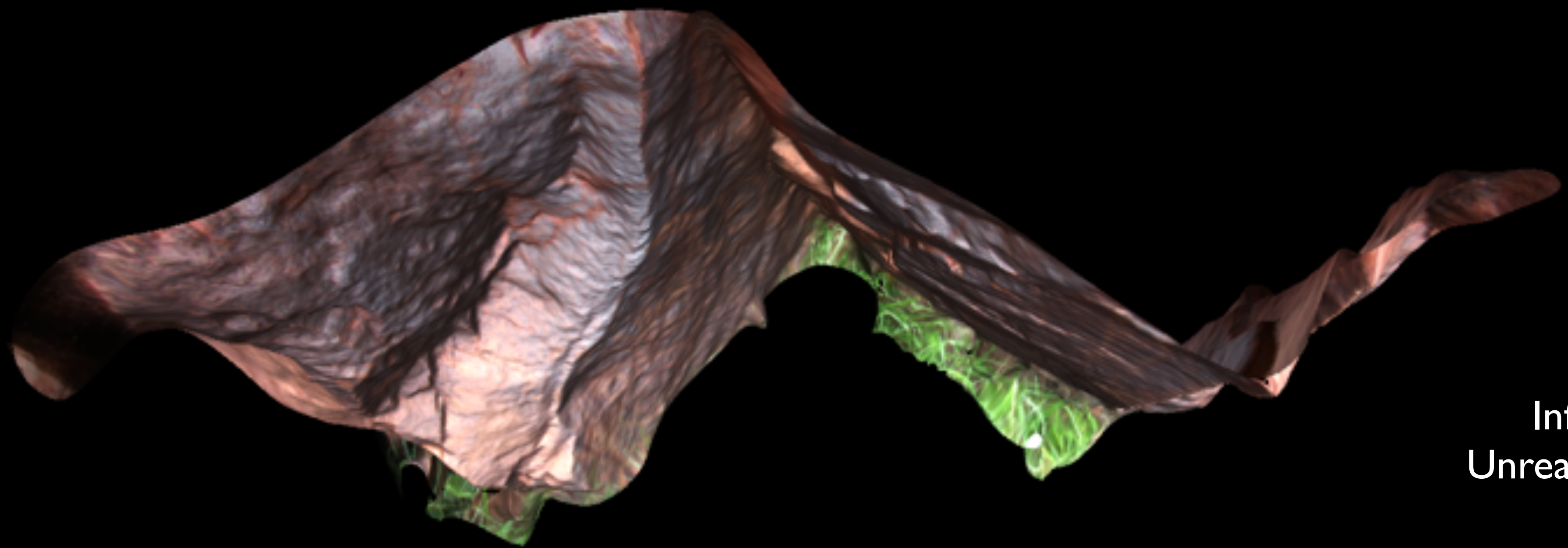


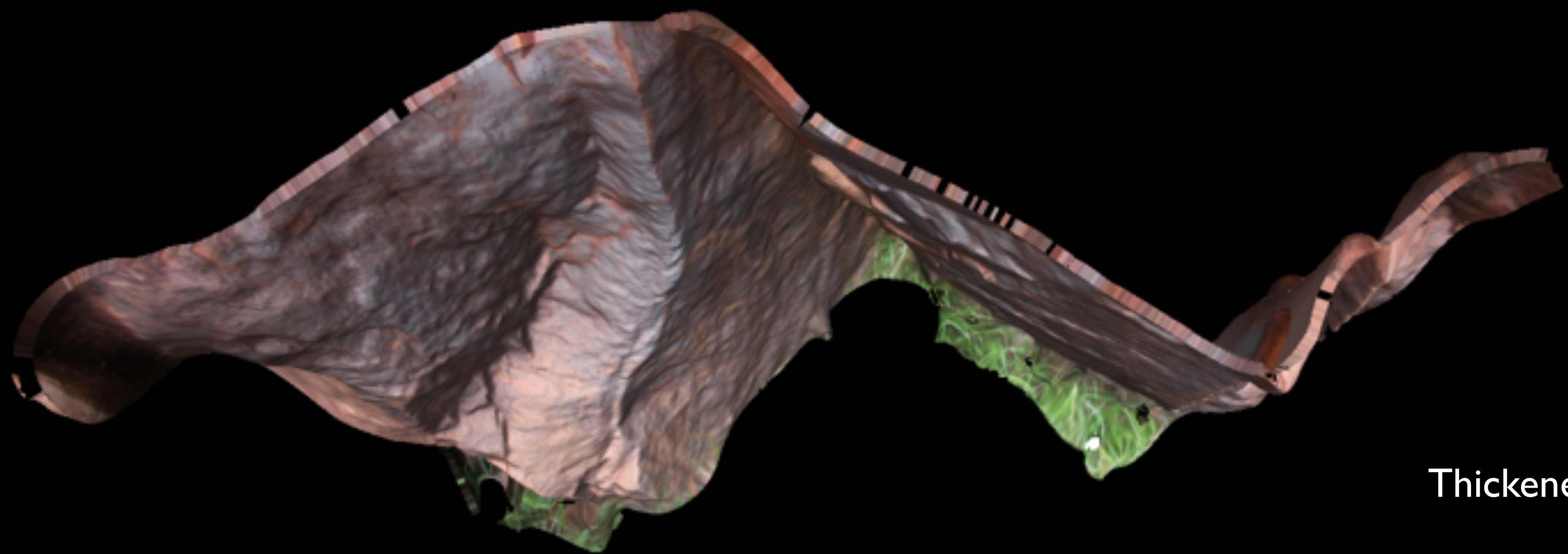
Rock shelter

Closing the loop

- Reconstruct 3D model from photograph - edit/transform - 3D print.

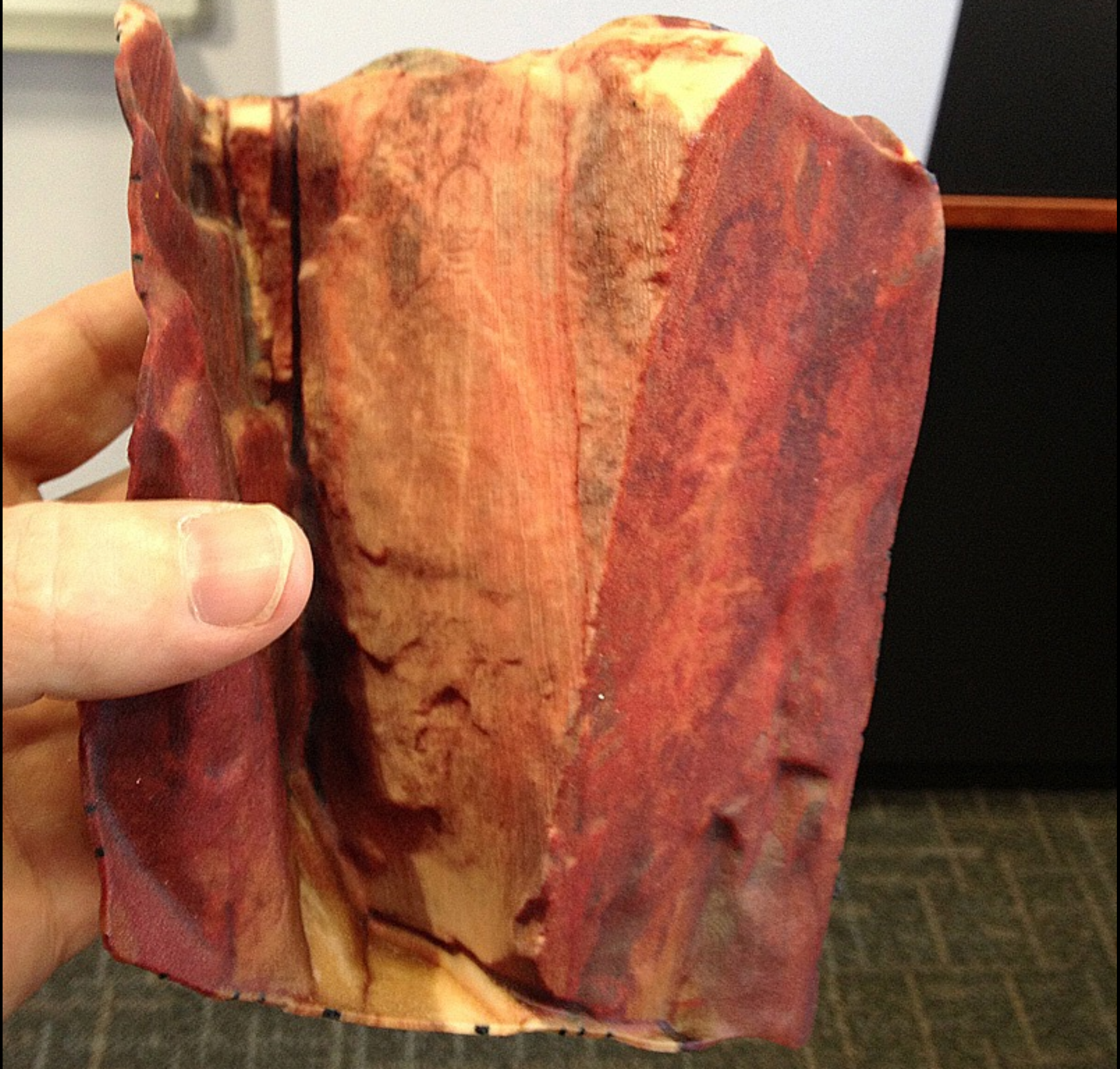


Infinitely thin surface
Unrealisable by a 3D printer



Thickened "realisable" surface


G07
Paul Bourke



Questions?

For all those who dismissed Erich von Däniken.

In case you doubted aliens visited West Australia