The Diversity of Visualisation:

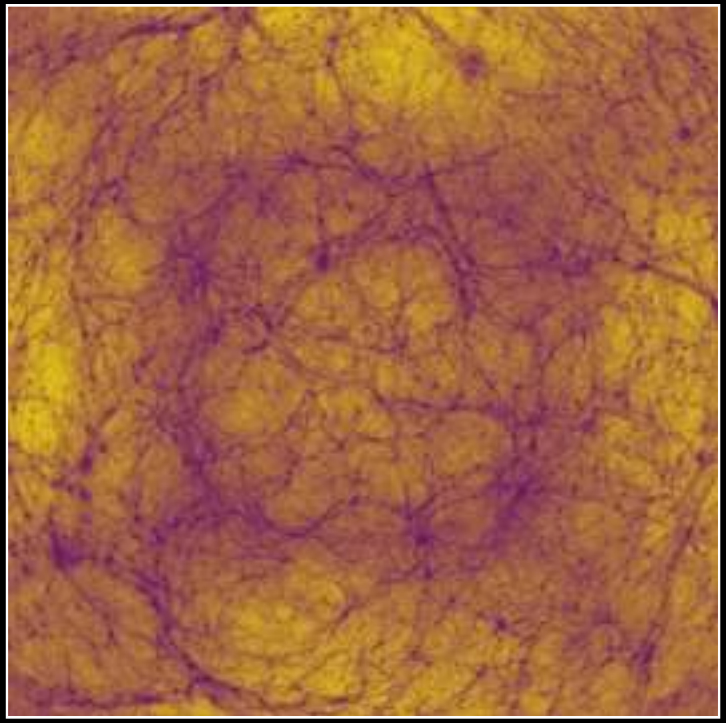
Selected visualisation projects from 2011

Paul Bourke

iVEC @UWA

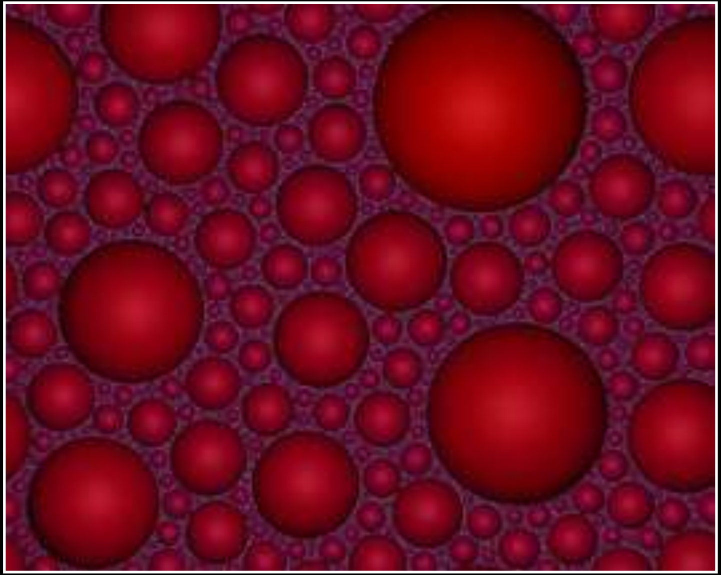
Contents

- Cosmology visualisations, in collaboration with
  Dr Alan Duffy (ICRAR, UWA) and
  Dr Rob Crain (Leinden Observatory, the Netherlands).



- 360 degree video capture for immersive environments.
  In collaboration with Sarah Kenderdine,
  City University, Hong Kong.

- Space filling random packings.
  In collaboration with John Shier.



Visualisation of cosmological simulations

- Present three examples: “COSMOS”, “GIMIC”, and “KINETIC”.

- Characteristics:
  - Large numbers of points, minimum 200 million, maximum 1 billion (COSMOS).
  - Generally three types of particles: Dark Matter, Stars, Gas.
  - Relative numbers of each type of particle may vary over time.
  - Each point has a region of influence, smoothing kernel.
  - Typically have multiple parameters per particle.
    Interest here in position, velocity (for time interpolation), mass, smoothing radius.

- Requirements / goals:
  - Explore pipelines appropriate for these types of data.
  - High impact images and animations.
  - High resolution fisheye images for digital planetarium projection.
    Targeting typically 3K square for an inhouse fulldome production and
    up to 8K square for high end planetariums.
  - Support for multiple projection types: orthographic, perspective, fisheye, spherical.
  - Produce all images as 16bit PNG to give enough dynamic range for postproduction effects.

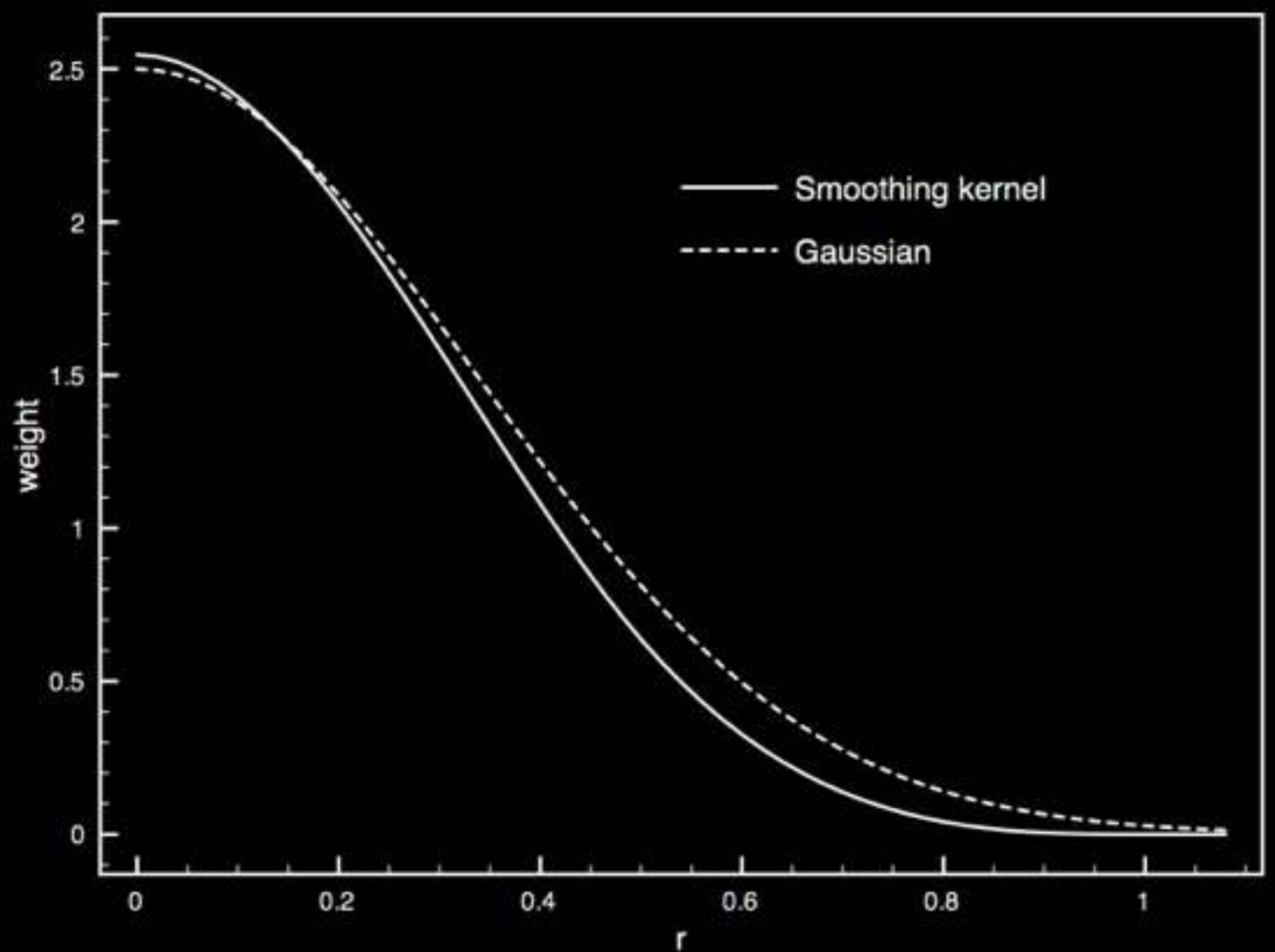
Gadget

Code base is Gadget (actually our private Gadget3 version). It's a C based code for cosmological N-body/SPH simulations on massively parallel computers with distributed memory. It uses an explicit communication model that is implemented with the standard MPI communication interface. It computes gravitational forces with a hierarchical tree algorithm (optionally in combination with a particle-mesh scheme for long-range gravitational forces) and represents fluids by means of smoothed particle hydrodynamics (SPH). It is both highly optimised and stable, and readily portability to supercomputers using standard libraries.

Alan Duffy

Smoothing kernel

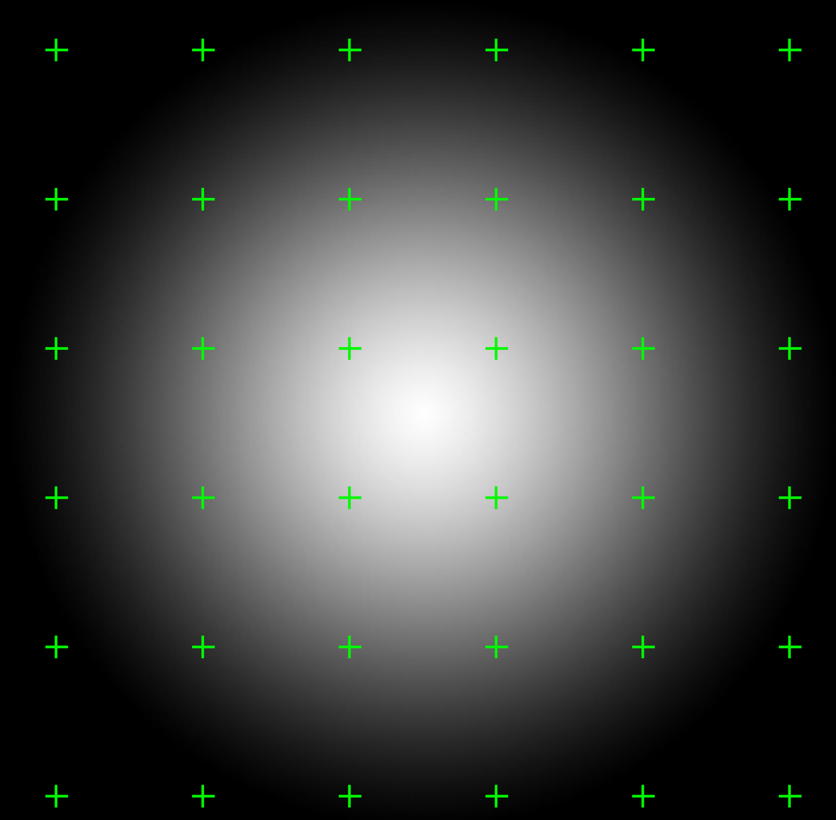
- If it were just “points” it would be much simpler.
- Region of influence are 3D functions of radius, similar to a “point spread function” in optics. Note this is used within the simulation software so not an arbitrary choice for the visualisations.
- For particles without a smoothing kernel (eg: stars) a Gaussian is used which allows the same pipeline to be employed. Use a single standard deviation, mass determines the amplitude.



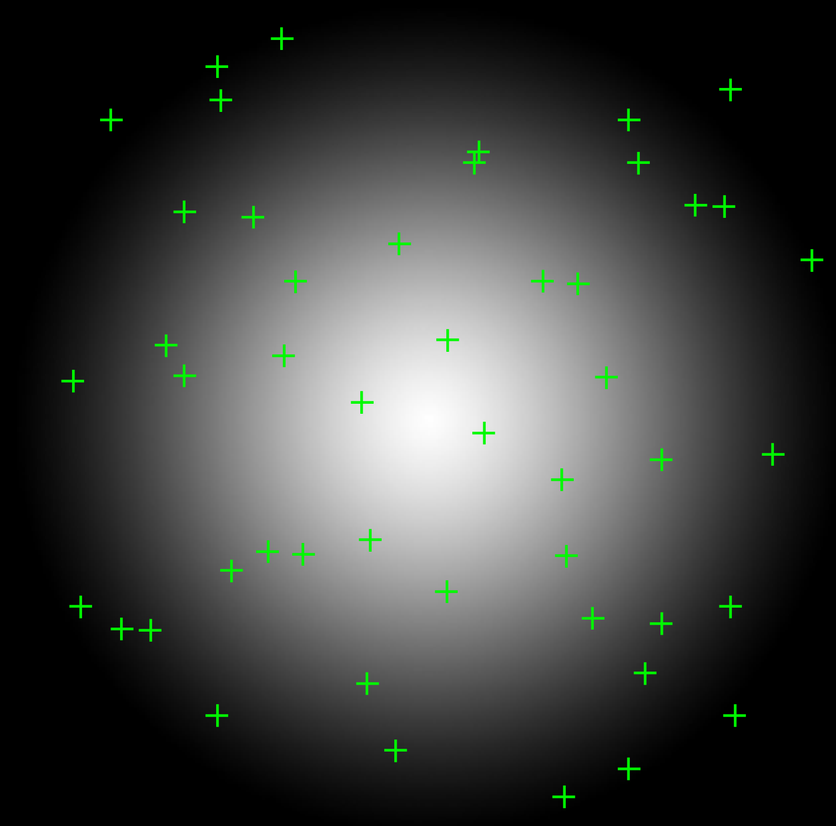
$$W(r) = \begin{cases} \dfrac{8\left(1 - 6\left(1 - \frac{r}{s}\right)^2 + 6\left(1 - \frac{r}{s}\right)^3\right)}{\pi s^3} & r < \frac{s}{2} \\[2ex] \dfrac{16\left(1 - \frac{r}{s}\right)^3}{\pi s^3} & \frac{s}{2} \le r < s \\[2ex] 0 & r \ge s \end{cases}$$

Smoothing kernel

- Easier to deal with when sampling into a volume.
  Decided not to do this here due to resolution constraints.
- Smoothing kernel radii are not necessarily “local”, sampling into a volume an be expensive.
- Implemented smoothing kernel by sampling (regular or stochastic) in 3D. Points are then projected onto plane, cylinder, or spherical surface. The image is then a histogram the projected points contribute their kernel weighted mass to.
- Only works because of the very large number of points, smooth histogram image forms quickly.
- Advantage of being able to form image with speed/quality trade-off.



Uniform sampling

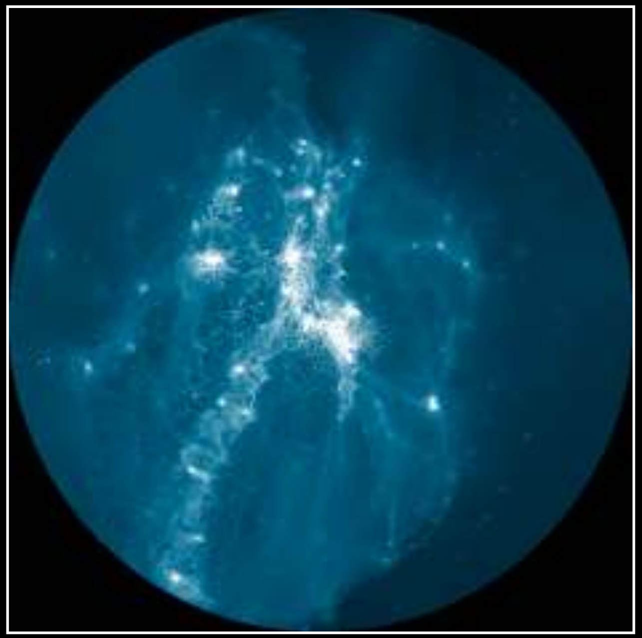


Random sampling

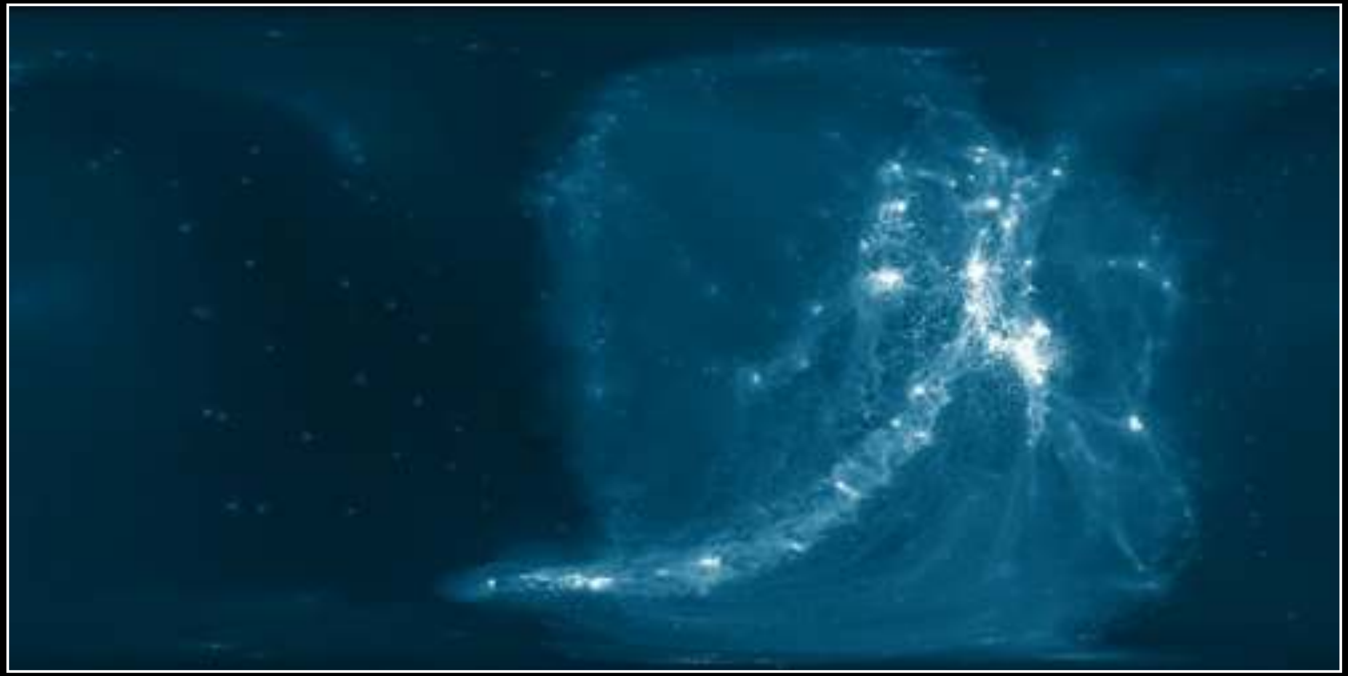
Projections

Orthographic

Perspective



Fisheye



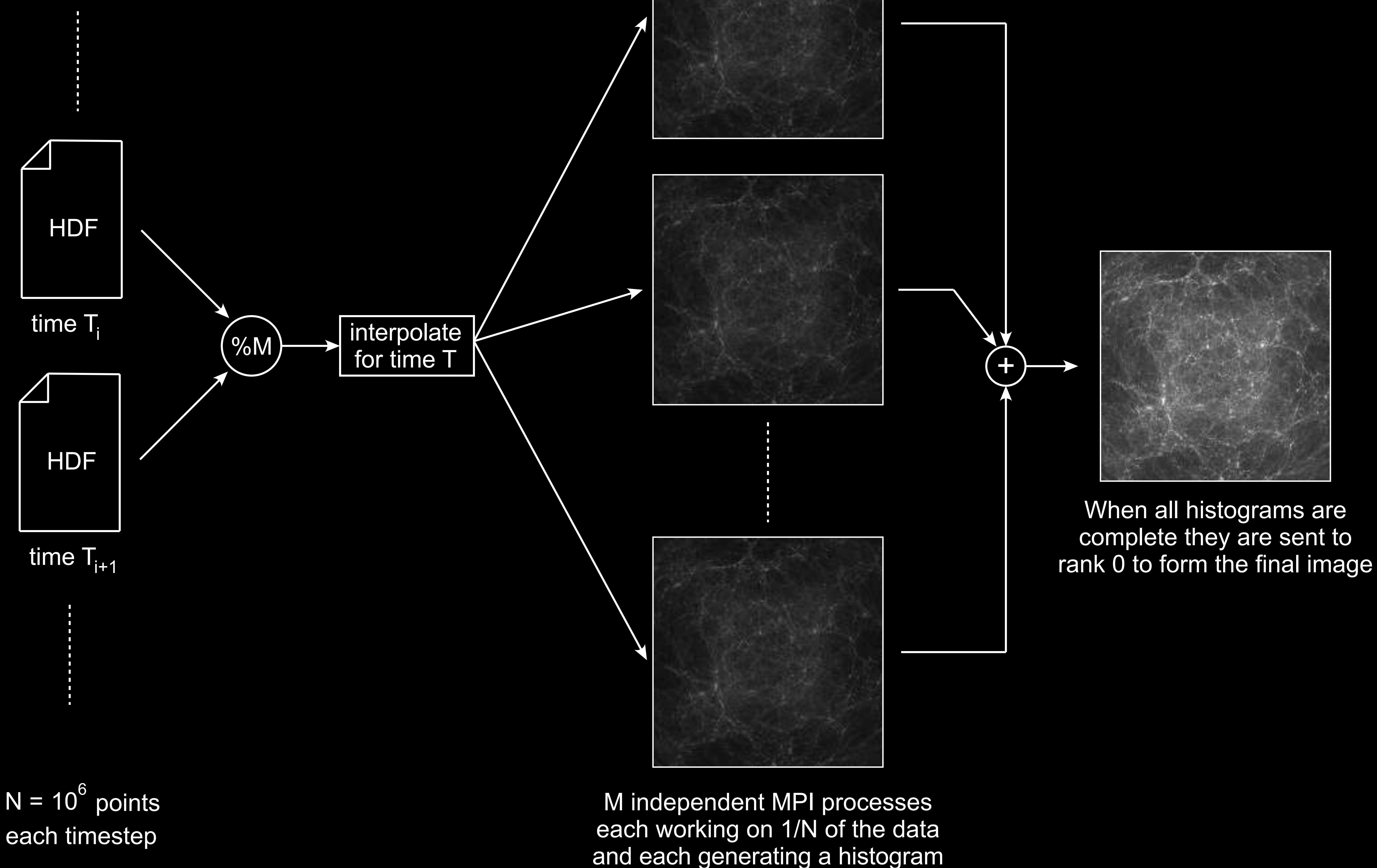
Spherical

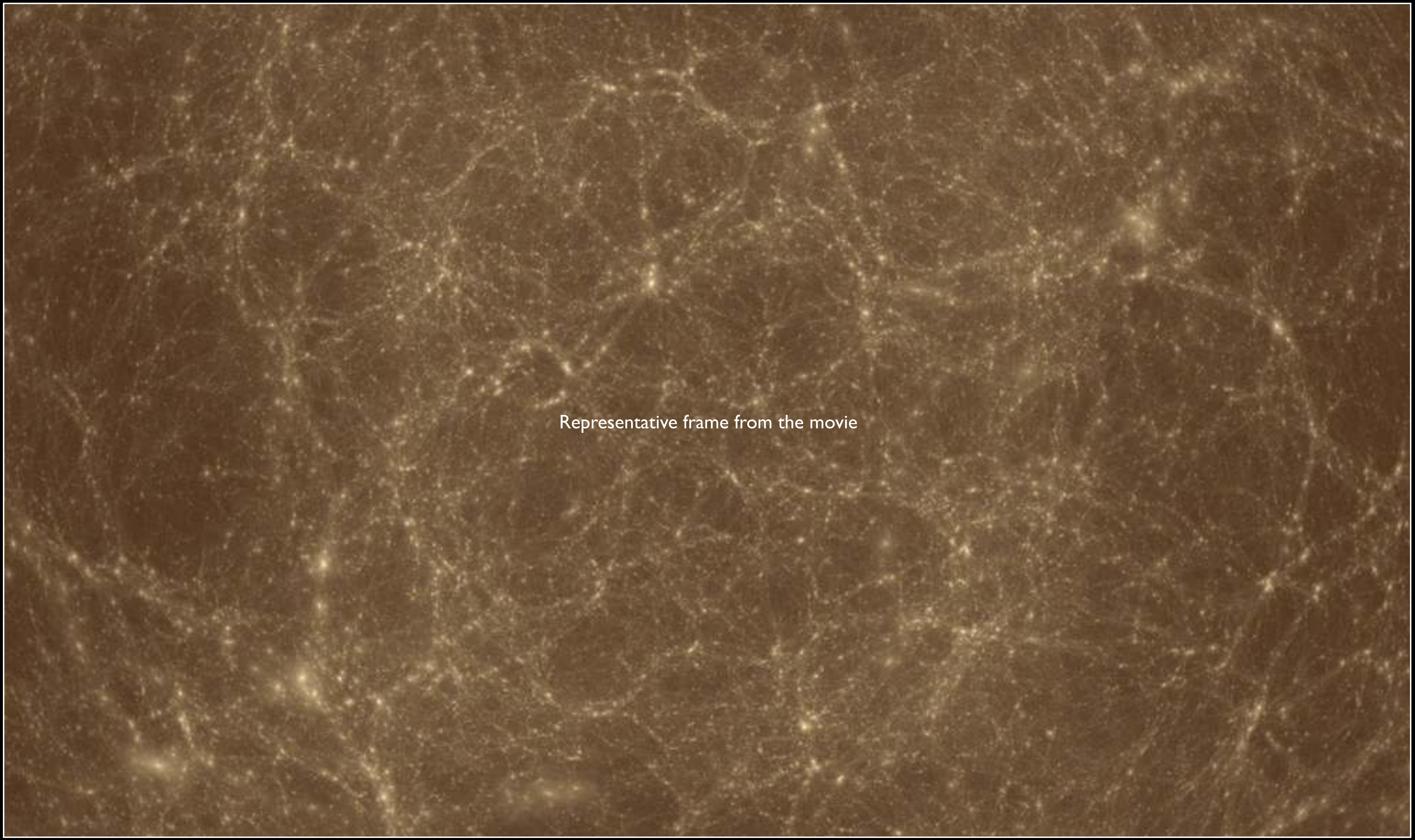
Cylindrical

COSMOS

- Simulation within a cubic region (periodic bounds) of the Universe just after the Big Bang.
- 600 million light years on each side of the cube.
- Shows dark matter collapsing over 14 billion years of cosmic time, forming filaments and collapsing haloes of the Cosmic Web.
- Note there is no smoothing kernel here, the images look smooth and continuous due to the 1 billion+ particles per time step.
- Even at 3Kx3K, if the whole dataset is in shot then on average there are over 100 points per pixel (if they were distributed uniformly).
- The final image is essentially a histogram formed on the projection plane.
- Original simulation computed on vayu (NCI).
  Used 1024 cores, 2.8TB RAM, took 19 hours (~20,000 CPU hours)
  Rendering performed on epic (iVEC).

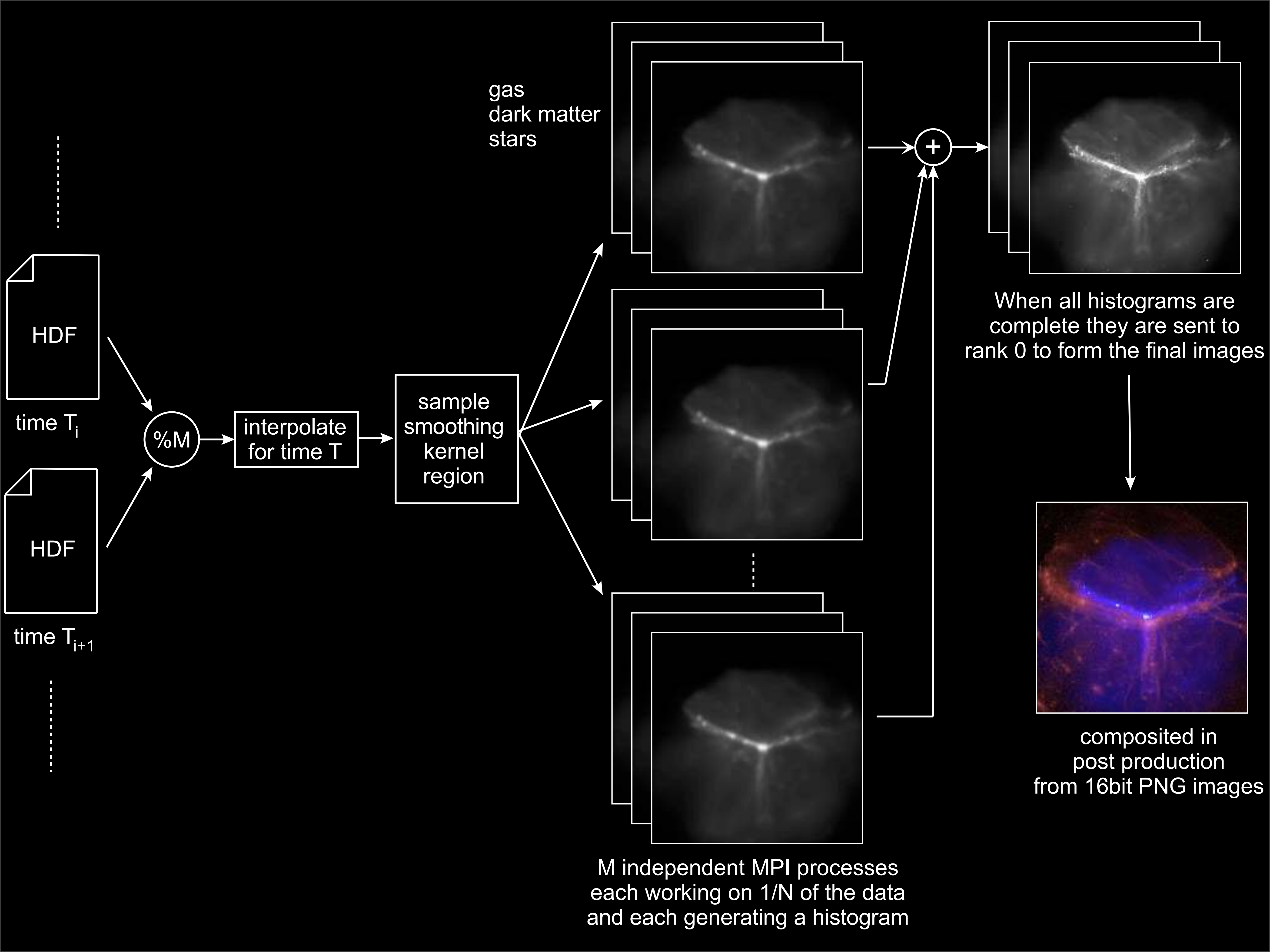
Rendering pipeline



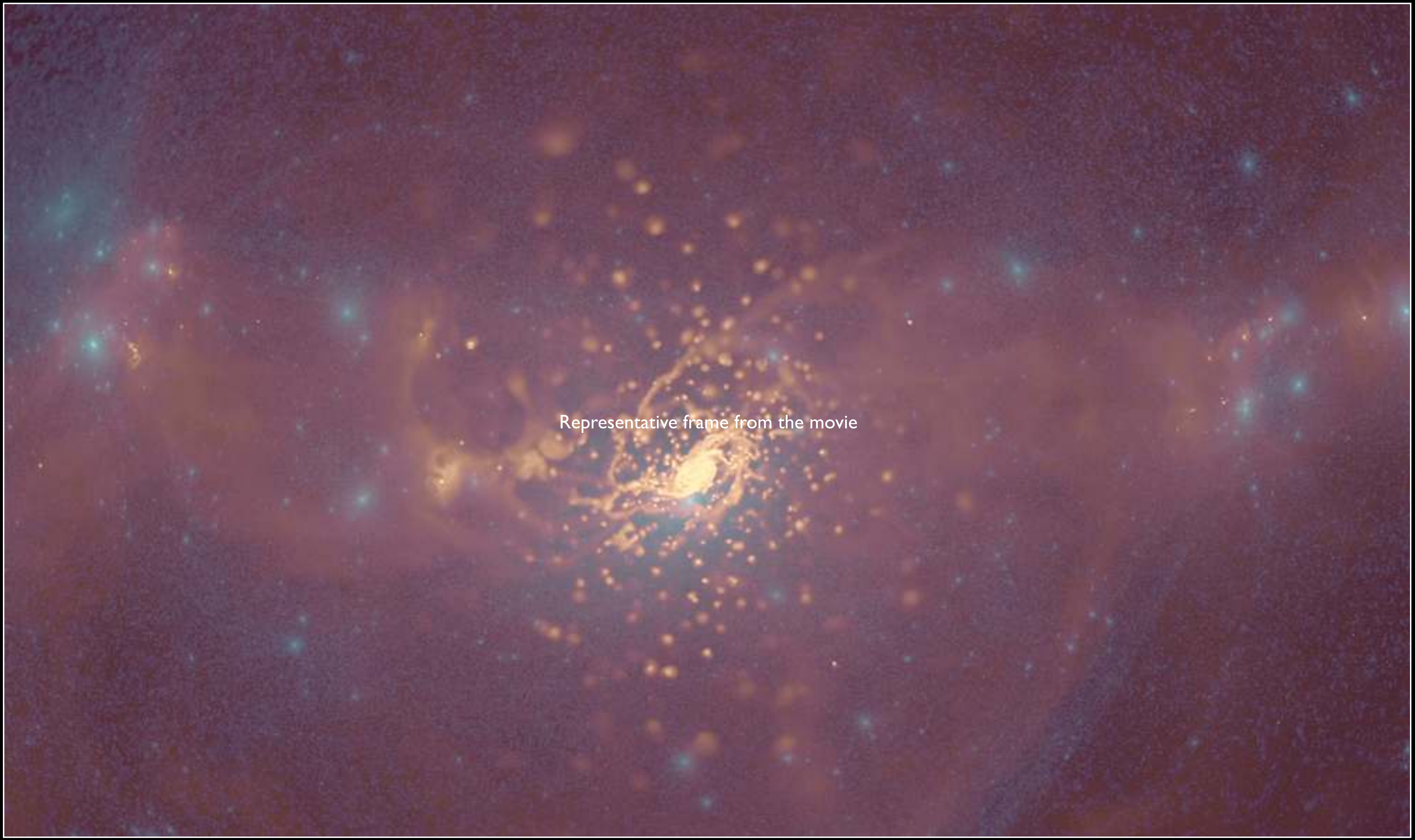

Representative frame from the movie

GIMIC

- Simulates the formation of a Dwarf Galaxy, similar to the Large Magellenic Cloud.
- The formation of these galaxies is a violent dynamic process.
- Dark Matters forms in filaments along which gas flows into the central disk where star formation occurs.
- Computed on cosma (Durham University).
  Used 32 CPUs, 92 hours (~3,000 CPU hours).
  Rendering performed on epic (iVEC).

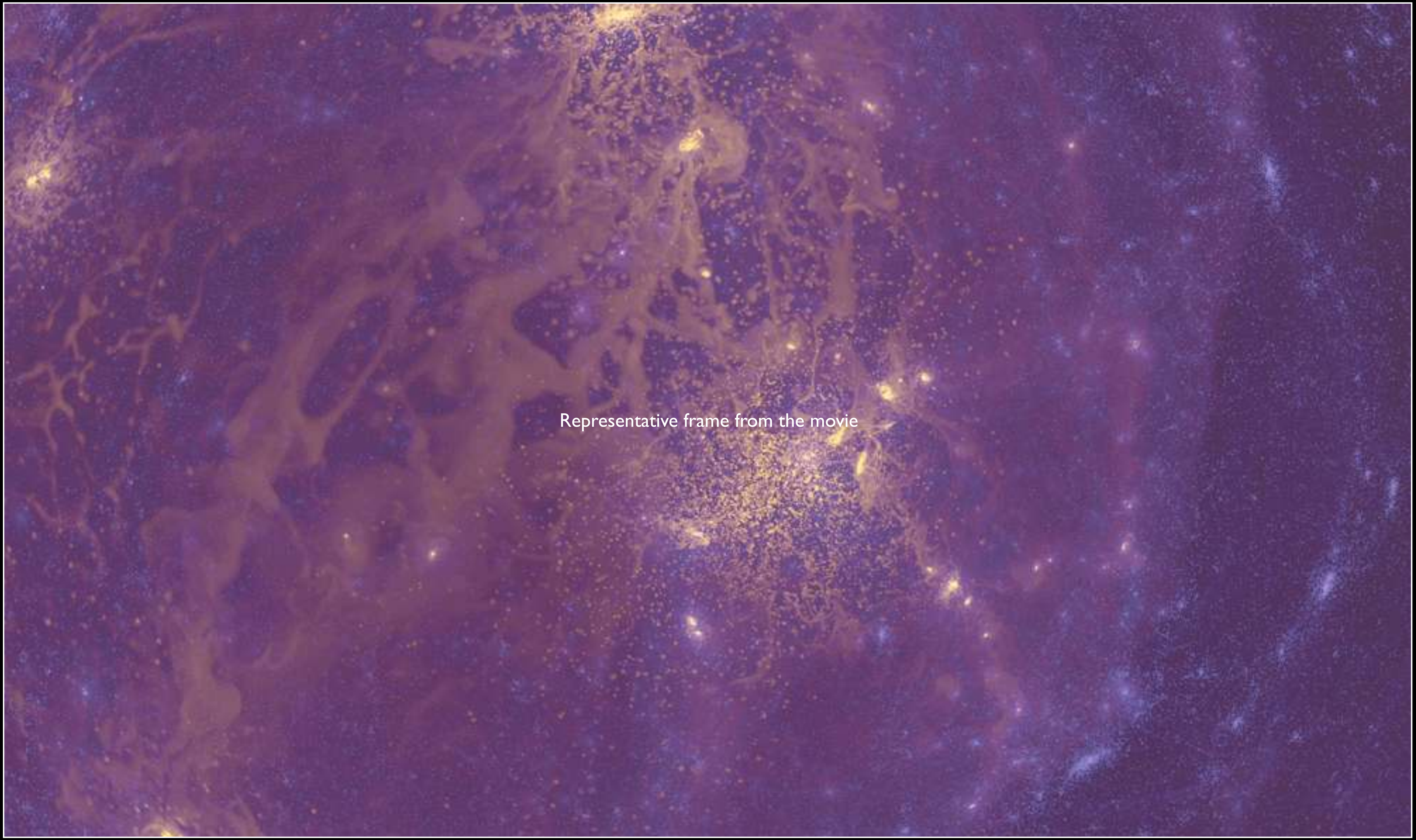


gas
dark matter
stars
HDF
time T_i
HDF
time T_{i+1}
%M
interpolate for time T
sample smoothing kernel region
+
When all histograms are complete they are sent to rank 0 to form the final images
composited in post production from 16bit PNG images
M independent MPI processes each working on 1/N of the data and each generating a histogram


Representative frame from the movie

KINETIC

- Simulation of the formation of a Spiral Galaxy similar to our own Milky Way but about half the current age.
- The Gas follows the Dark Matter along the filaments.
- Each of the small satellite galaxies are about the same mass as the GIMIC Galaxy.
- Computed on epic machine (iVEC).
  Used 1024 cores, 2.05TB RAM, took 470 hours (~500,000 CPU hours).
  Rendering performed on epic (iVEC).


Representative frame from the movie

360 degree video capture for immersive environments.

- How to capture sufficiently high resolution video for immersive environments?
- In particular for high definition dome and cylindrical displays.
- Place the user in a navigable real world recording (perhaps augmented) rather than purely virtual world.
- Applications in experiential environments, virtual heritage, and training/simulation.

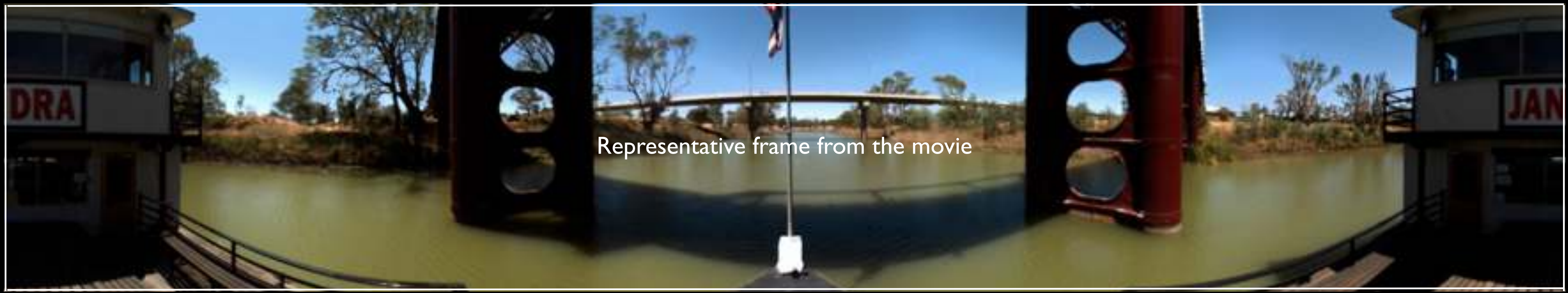
Running Room

Place Hampi

Single Sensor

- Single camera solutions are limited by the sensor resolution.
- Generally only support cylindrical panoramas, small vertical FOV.
- Typical maximum horizontal resolution of ~2000 pixels.

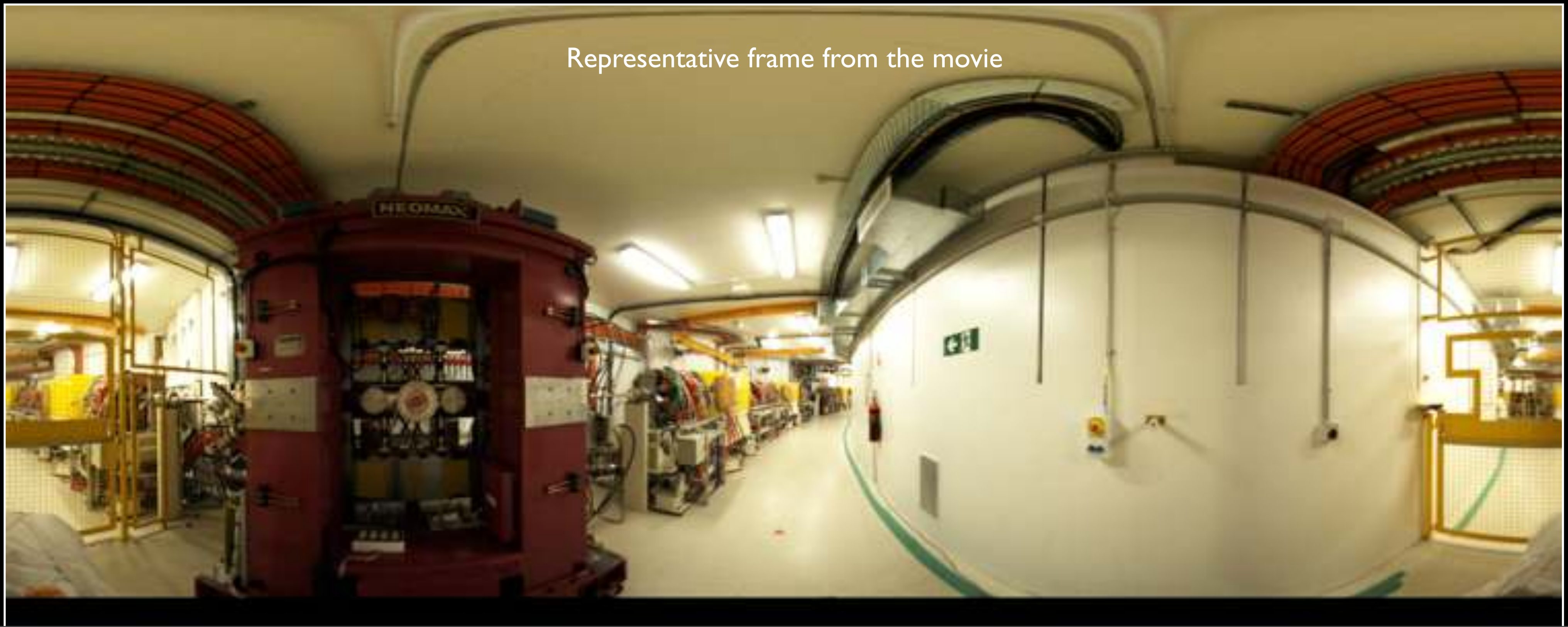
Simulation only, Courtesy iCinems



Representative frame from the movie

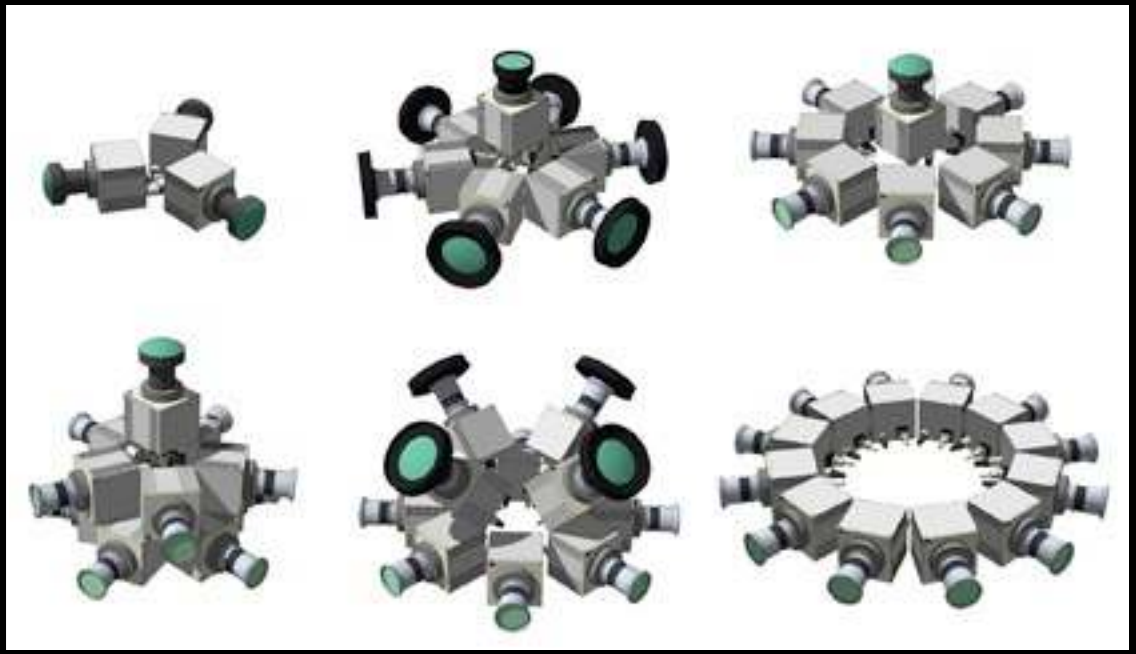
Multiple sensors

- Still very difficult to get final spherical panoramas above 4000 pixels horizontally.
- Also suffer from parallax error due to all cameras not having the same nodal point. Consequence is perfect stitching across all depths is impossible.


Representative frame from the movie

“Big end of town” solutions

- Generally large fixed cameras.
- Some solve the parallax problem by employing sets of mirrors.
  eg: “Making of Versailles 360”



SphereCam UNSW

Current options available from iVEC@UWA

- LadyBug3: 360x150 degree spherical video

Representative frame from the movie

~5400x2300 pixels, 15fps

- SLR and fisheye lens: 180x180 degree fisheye video

Representative frame from the movie

~1000x1000 pixels, 30fps

Current exploration

- Easy to capture high resolution still panoramas at 360x180 degrees.

~8000x4000 pixels
(3 shots with Canon 5D
and 8-15mm Canon fisheye lens)

- Inset video windows of the interesting action.

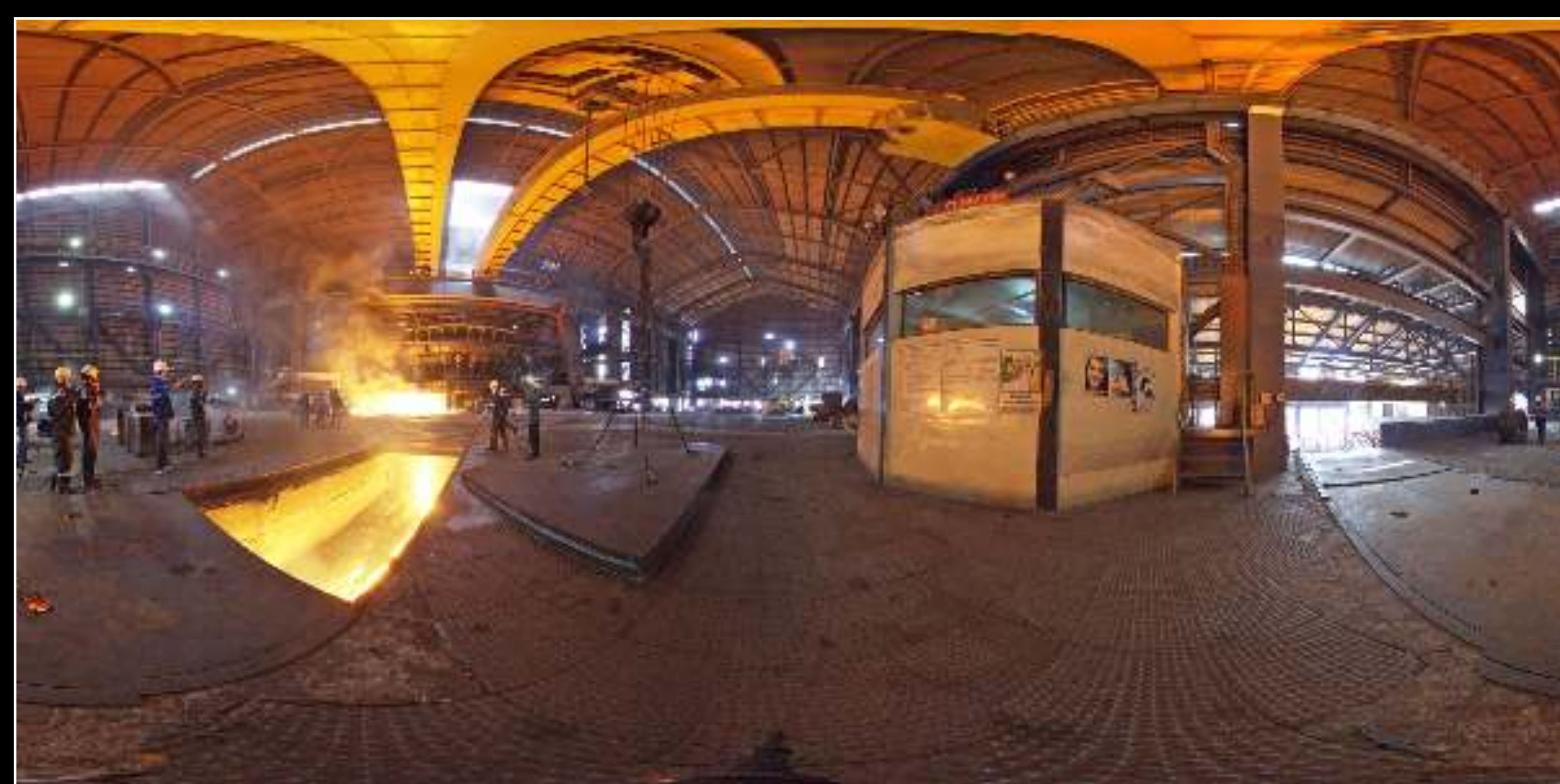
~3000x3000 pixels
(Red camera and
Sigma fisheye lens)

Process

Determine fisheye parameters: center and radius.
Key is measuring lens radial properties.

Transform fisheye movie frames into spherical coordinates.

Blending mask.



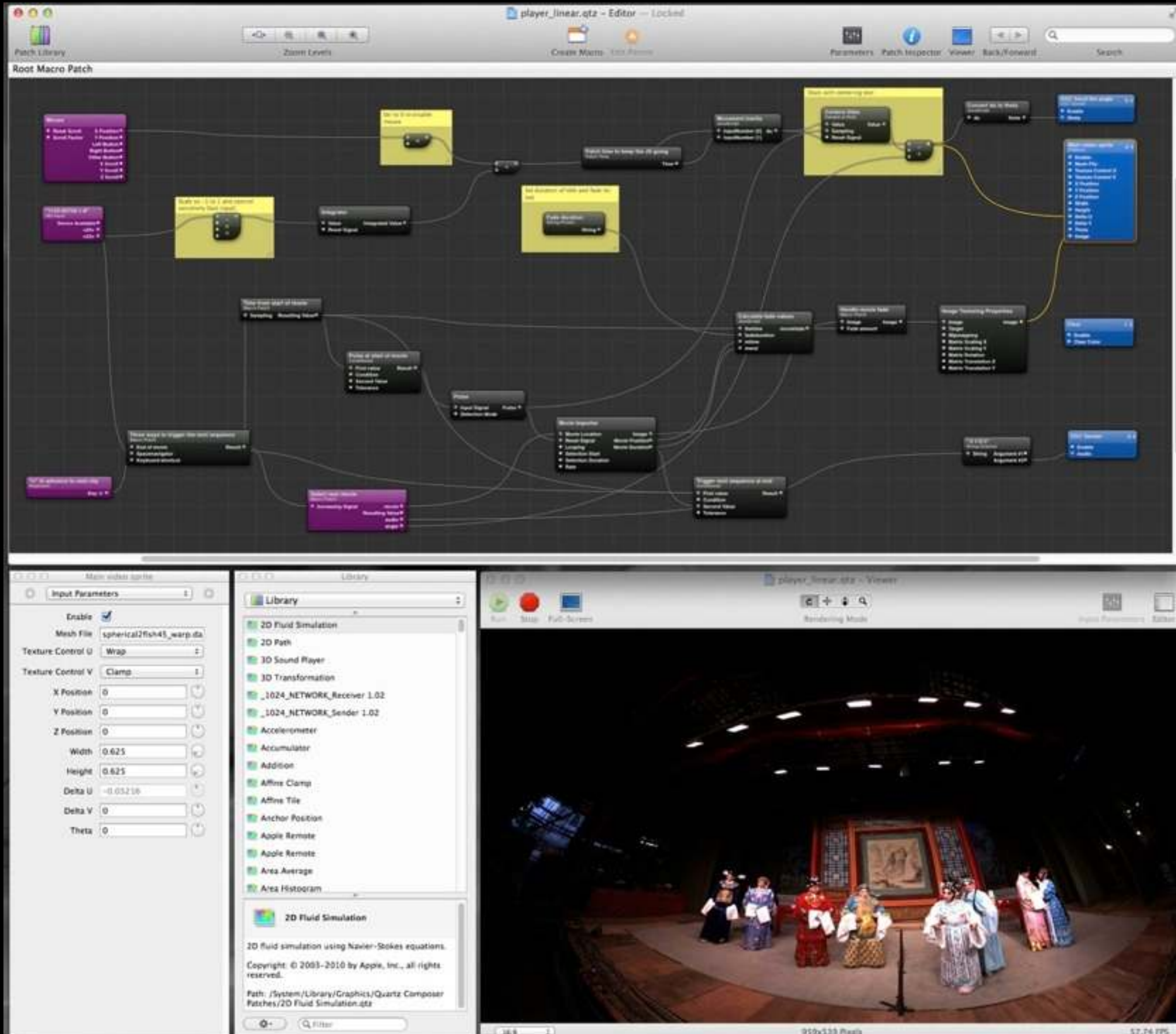
Final high resolution spherical panorama movie.

Result

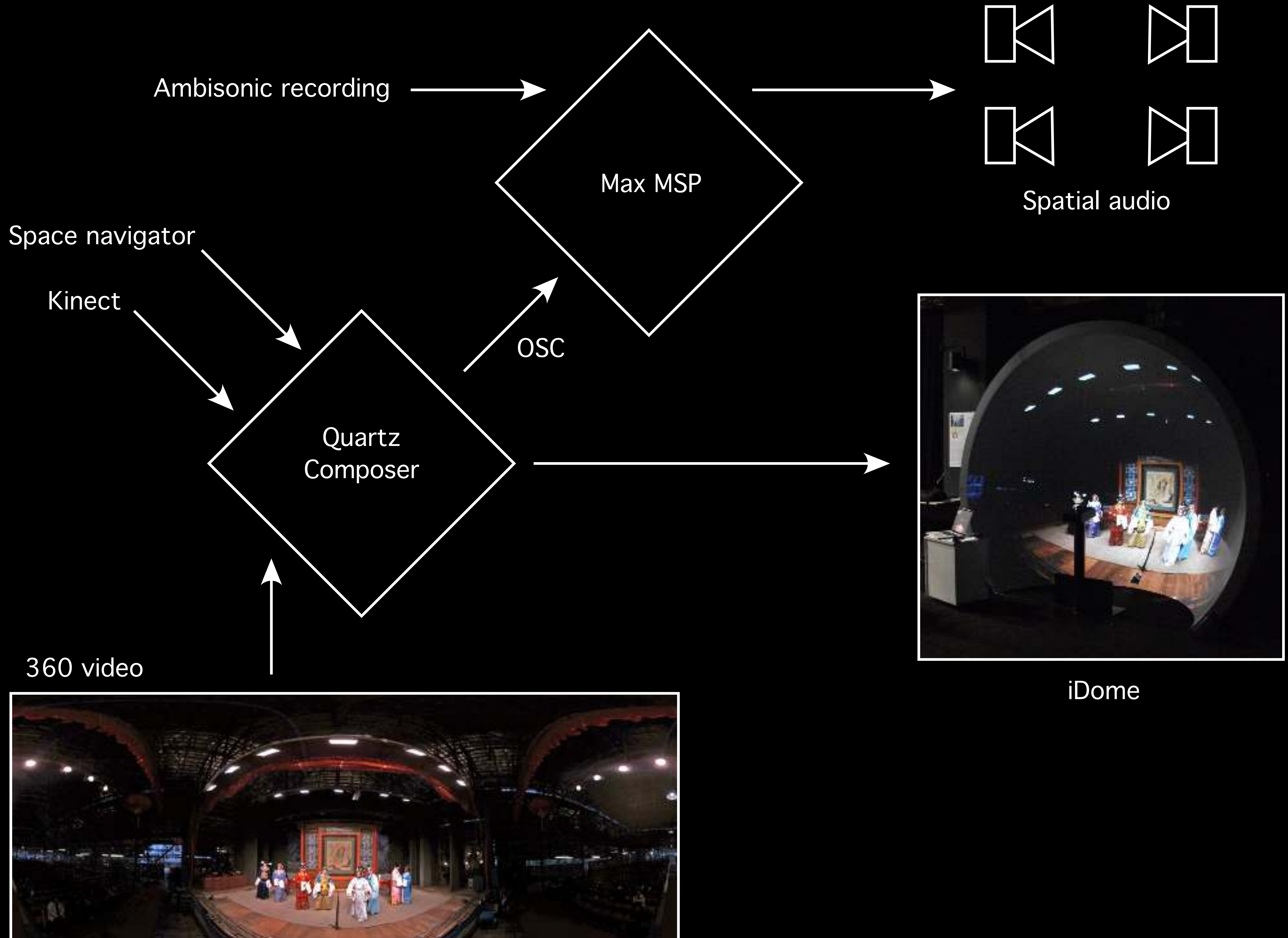
iJiao

- Showcase of cultural heritage of China.
- Venue: Main Gallery, Hong Kong Central Gallery.
- 360 degree video of various Taiping Qingjiao, also known as the Jiao festival.
- "The festivals, held throughout Hong Kong, appease the ghosts and give thanks to the deities for their protection. They take place every year or every five, eight, or ten years, depending on local customs. The religious rituals involved are meant to purge a community and prepare it for a new beginning."
  [Sarah Kenderdine]


Representative frame from the movie



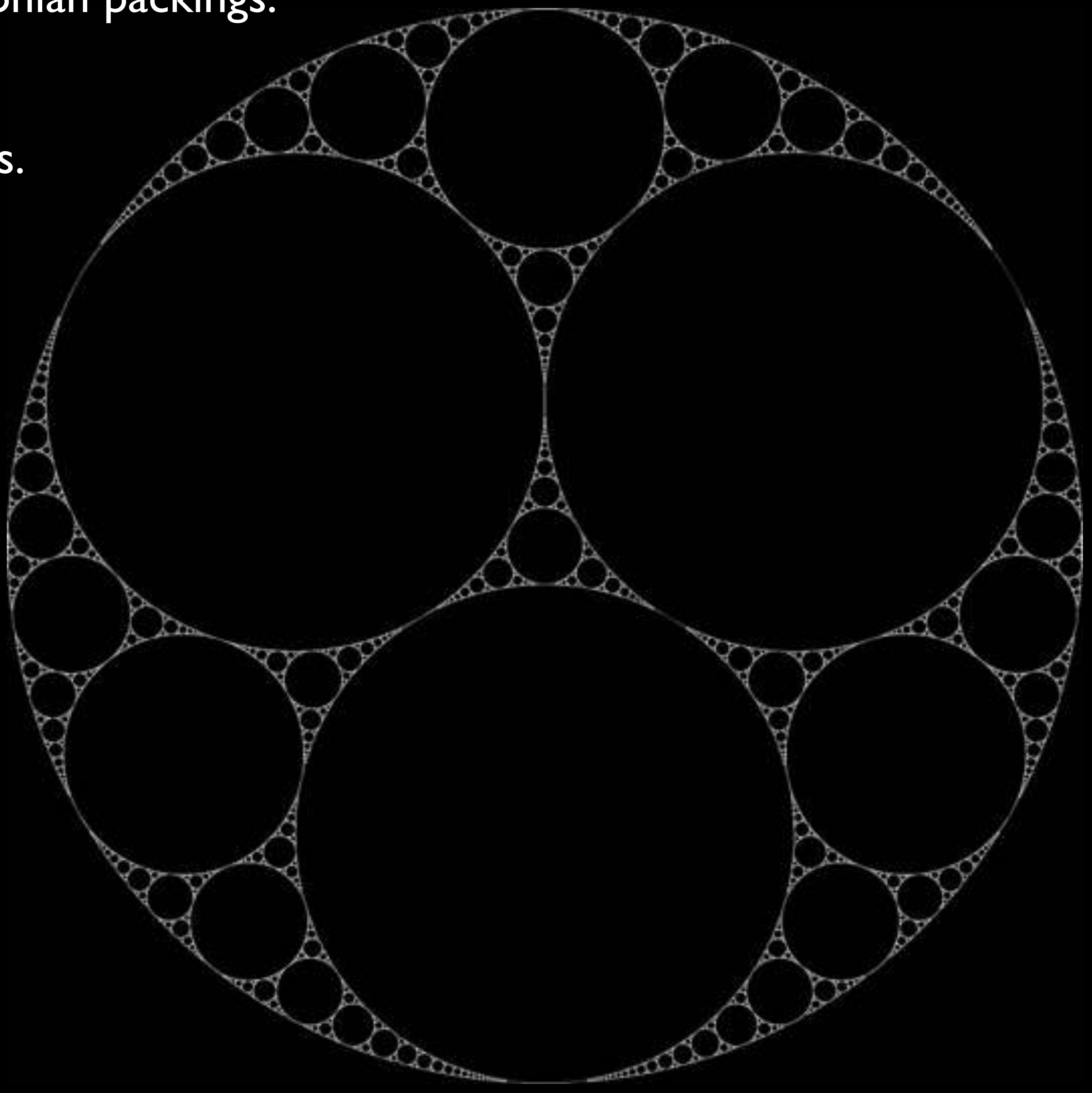
player_linear.qtz – Editor
Patch Library
Zoom Levels
Create Macro
Parameters
Patch Inspector
Viewer
Back/Forward
Search
Root Macro Patch
Main video sprite
Input Parameters
Enable
Mesh File
spherical2fish45_warp.da
Texture Control U
Wrap
Texture Control V
Clamp
X Position
0
Y Position
0
Z Position
0
Width
0.625
Height
0.625
Delta U
Delta V
0
Theta
0
Library
Library
2D Fluid Simulation
2D Path
3D Sound Player
3D Transformation
_1024_NETWORK_Receiver 1.02
_1024_NETWORK_Sender 1.02
Accelerometer
Accumulator
Addition
Affine Clamp
Affine Tile
Anchor Position
Apple Remote
Apple Remote
Area Average
Area Histogram
2D Fluid Simulation
2D fluid simulation using Navier-Stokes equations.
Copyright © 2003–2010 by Apple, Inc., all rights reserved.
Path: /System/Library/Graphics/Quartz Composer Patches/2D Fluid Simulation.qtz
Filter
player_linear.qtz – Viewer
Run
Stop
Full-Screen
Rendering Mode
Input Parameters
Editor



Ambisonic recording
Max MSP
Spatial audio
Space navigator
Kinect
OSC
Quartz Composer
360 video
iDome

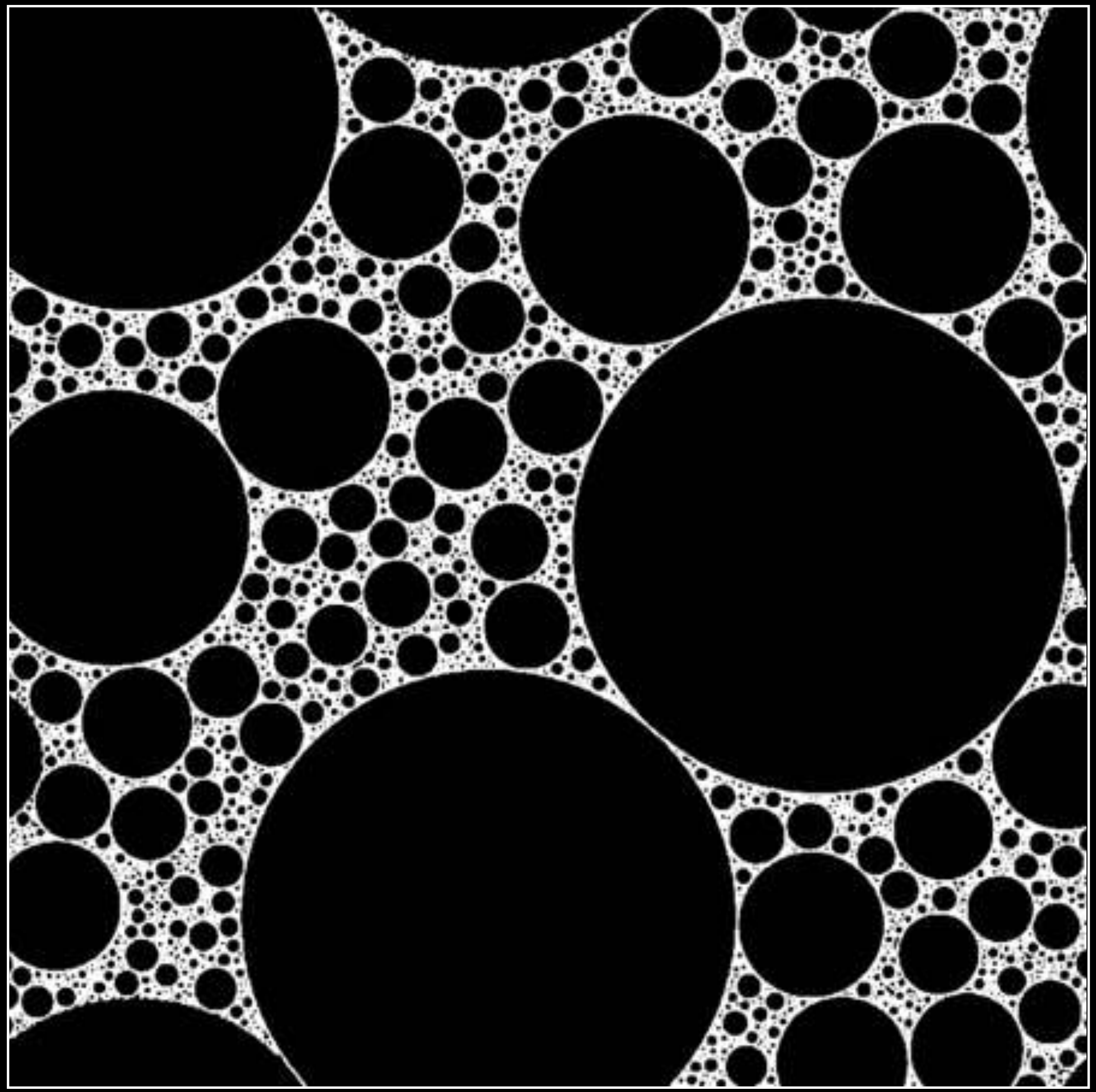
Space filling random packings

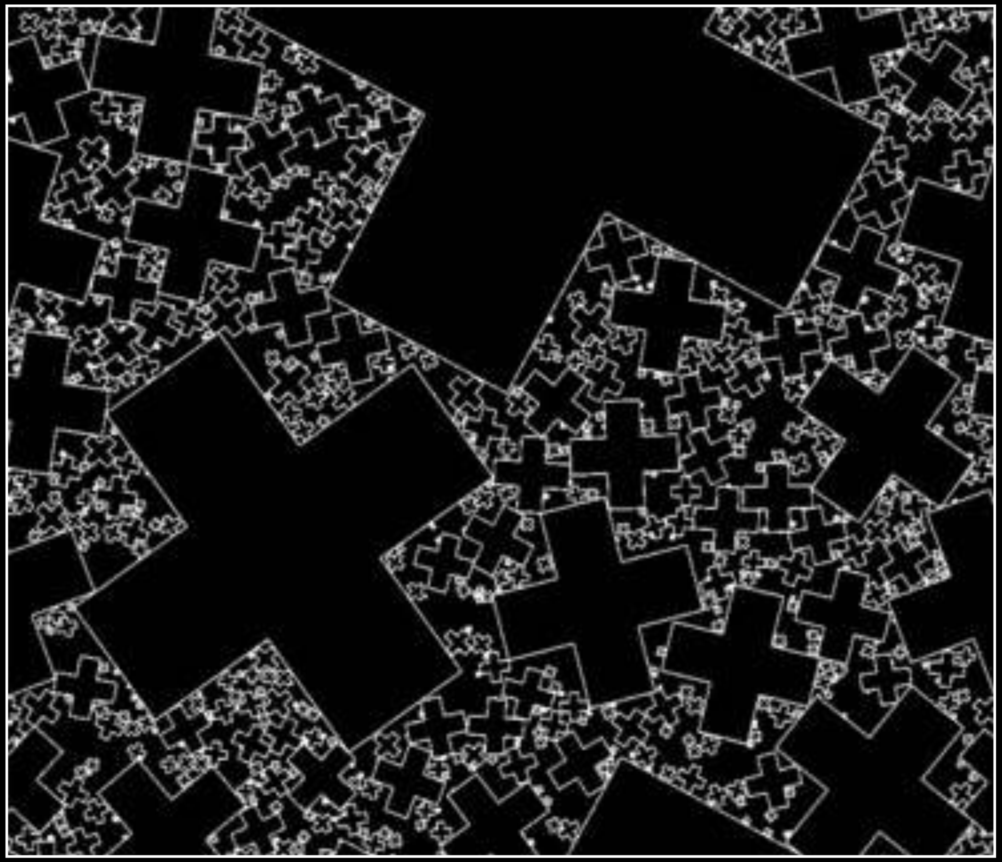
- Packing has been studied for some time with applications in material sciences, geology, and applications to a number of other physics process.
- Most study has been targeted at Apollonian packings.
- Apollonian packing has each object touching one or more of its neighbours.
- For example the pure Apollonian fractal is shown on the right.
- Circle that touches three neighbours is called Soddy circle.



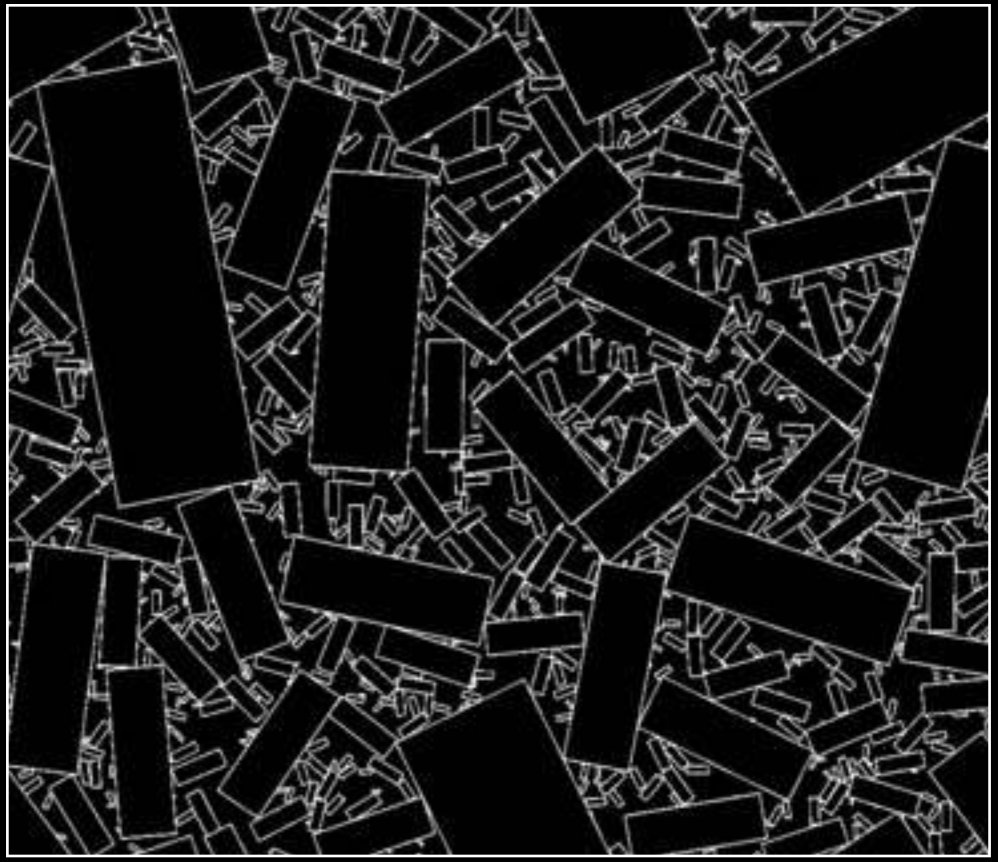
Random Apollonian packing

- Algorithm
  - Choose a random position not already occupied
  - Grow the object until it touches (kisses) another
  - Repeat



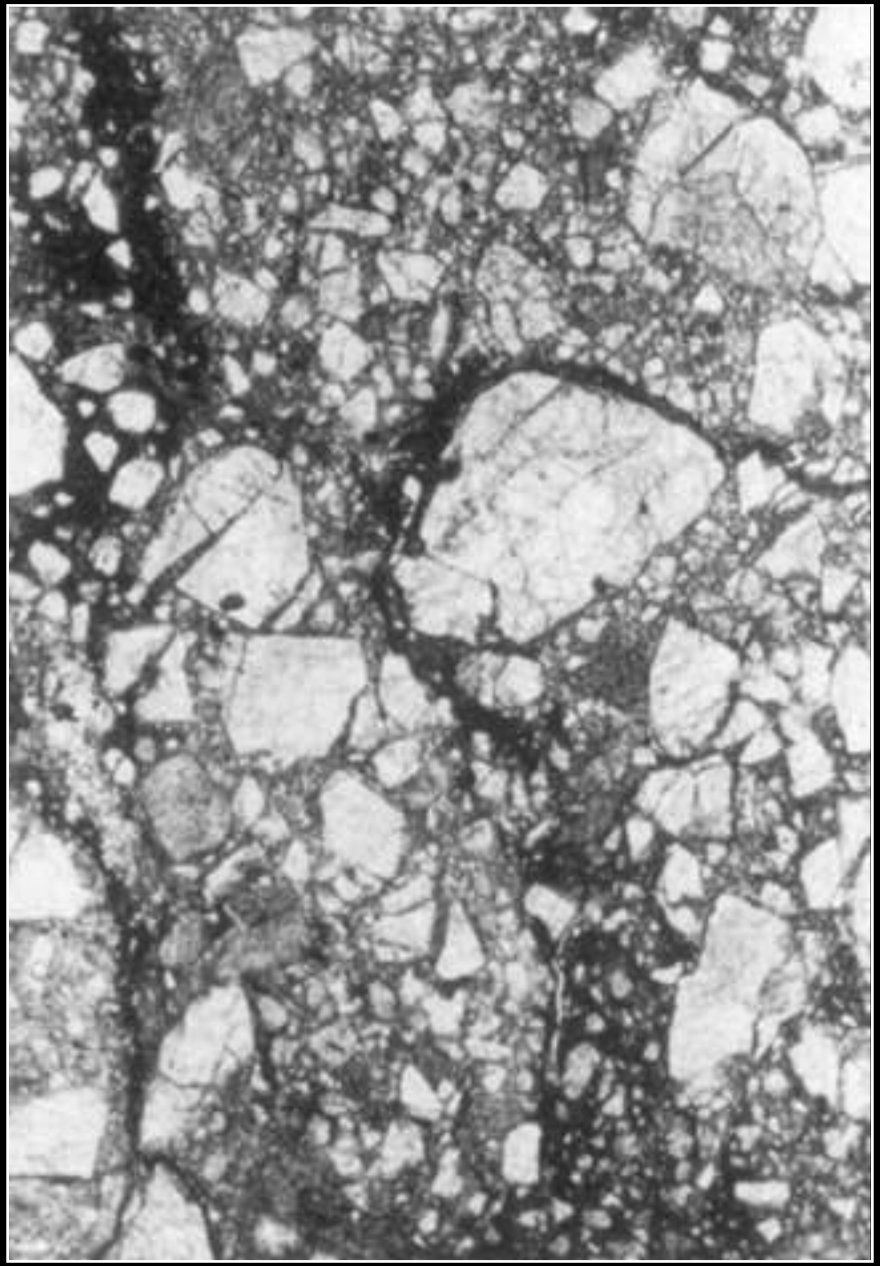


P.S. Dodds, J.S. Weitz. Physical Review E67, 016117 (2003)



Non-Apollonian packings

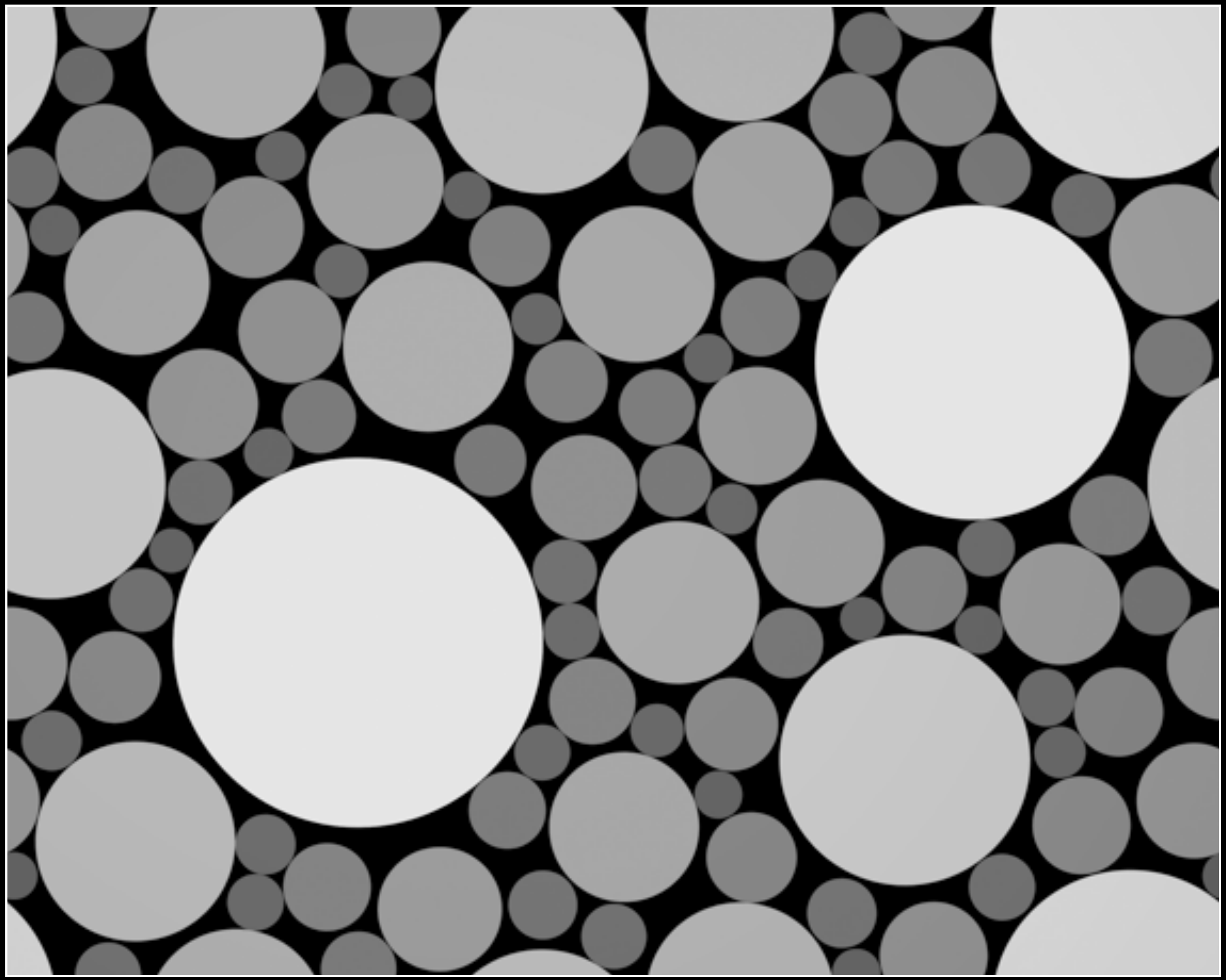
- There are other packings that are not Apollonian.
- Many deposit processes and most soap bubbles.



T.G. Blenkinsop, PAGEOPH, Vol 136, No 1, 1991

Random space filling packing

- Consider here random packings, that is, objects of fixed size are placed randomly into the empty space.
- Question: How might the size of the object decrease monotonically on each iteration in order for the packing to be space filling?



Area reduction function

- If A_0 is the initial area and g(i) is the reduction function then the area A_n occupied at the n'th iteration is give as:

$$A_n = A_0, A_0\, g(1), A_0\, g(2), \ldots. A_0\, g(n) = A_0 \sum_{i=0}^{n-1} g(i)$$

- The function g(i) used here is:

$$g(i) = \frac{1}{i^{c}}$$

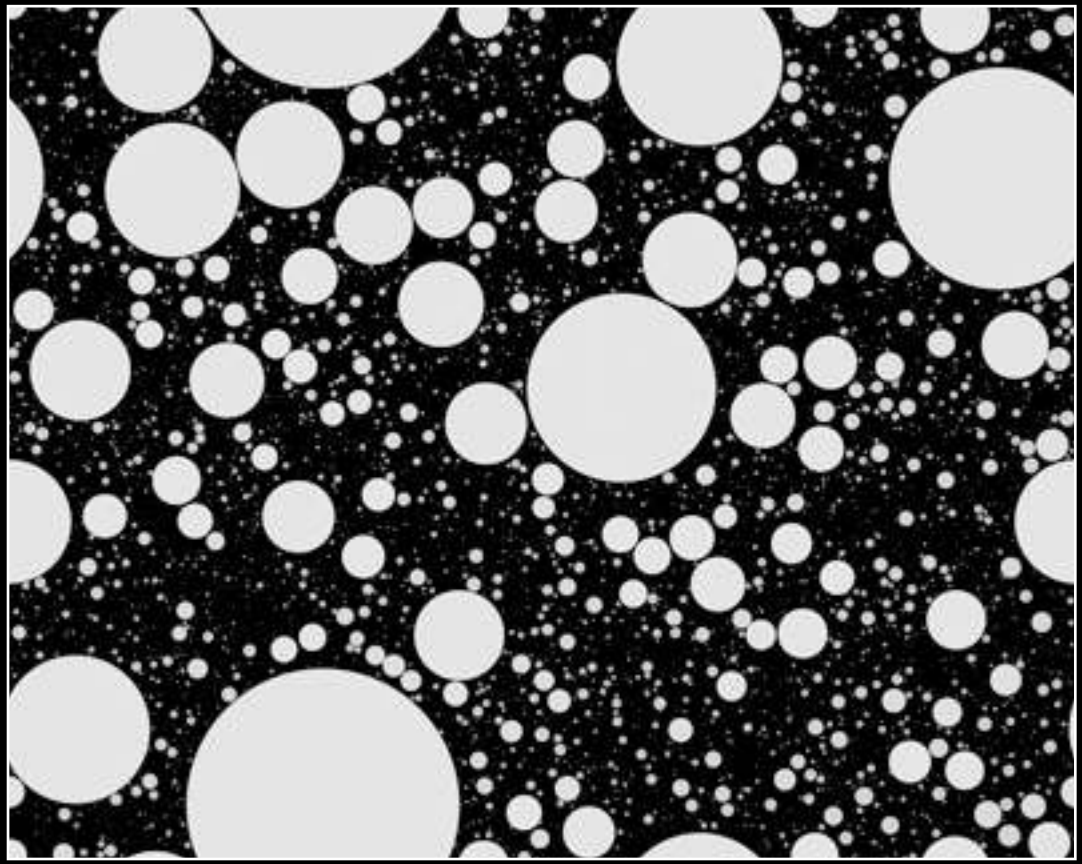
- Is this the only possible monotonically decreasing solution?

- At infinity the area is A_{total}, this is the Reimann Zeta function, it converges for c > 1.

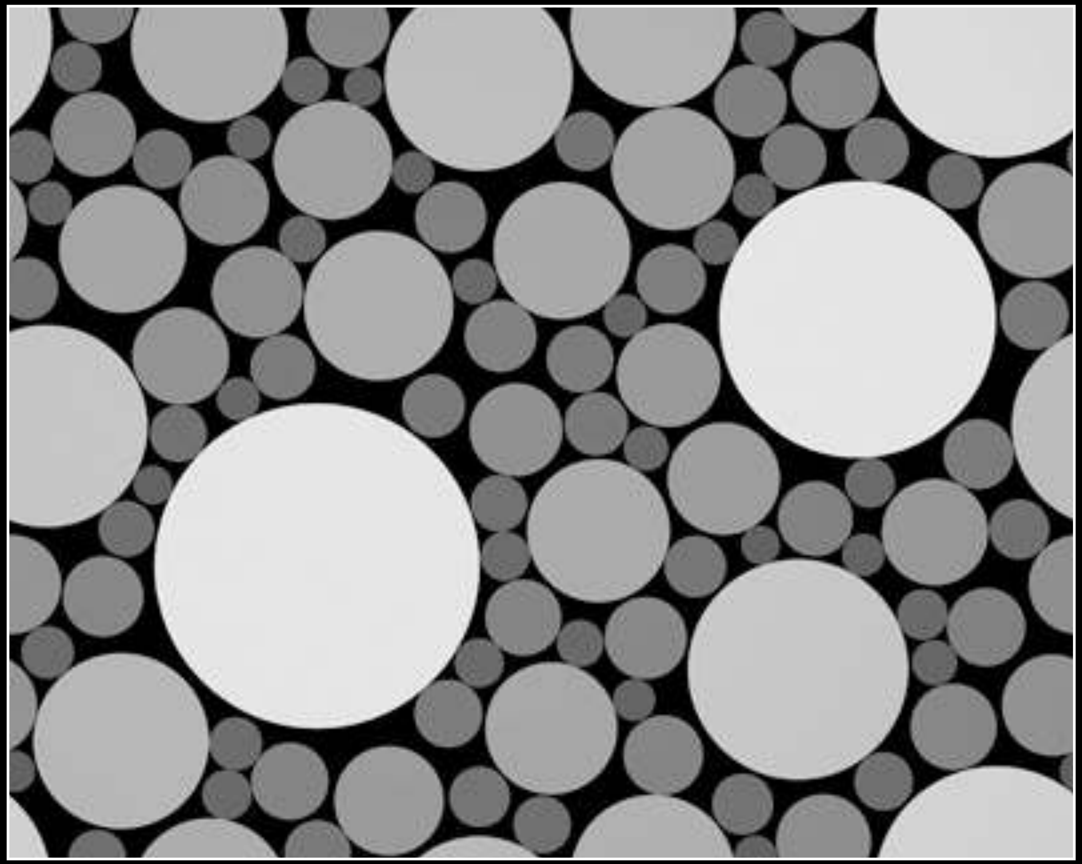
$$A_{total} = A_0 \sum_{i=0}^{\infty} i^{-c}$$

Notes

- The algorithm involves choosing a value of "c", this in turn determines A_0.
- Or one can choose A_0 which dictates "c".
- Pseudo code
  - select a random position
  - can the current object fit without overlap with existing objects?
  - if it can then
    - add the object
    - reduce the area for the next object by g(i)
  - repeat



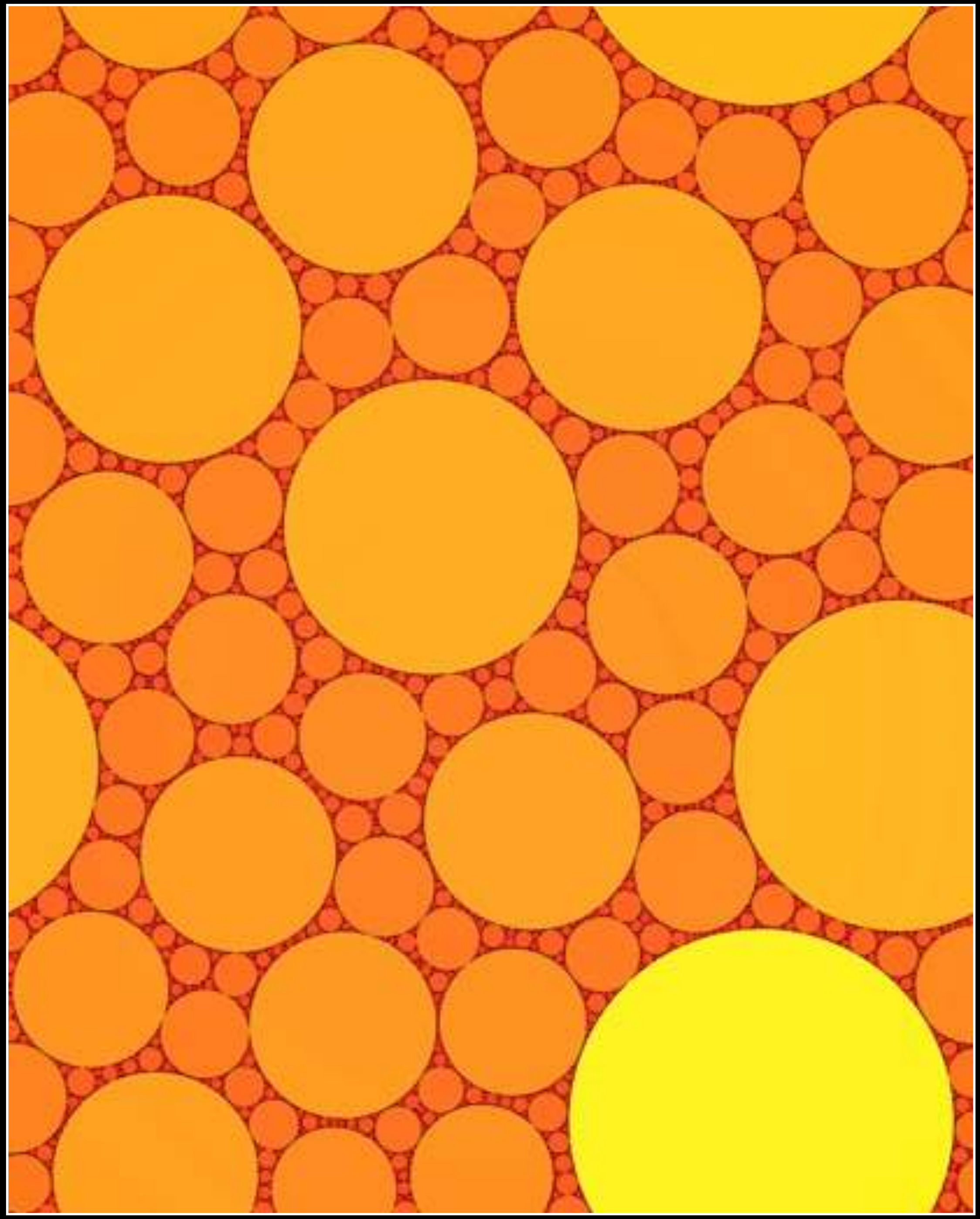
g(i) decreases too fast - not space filling

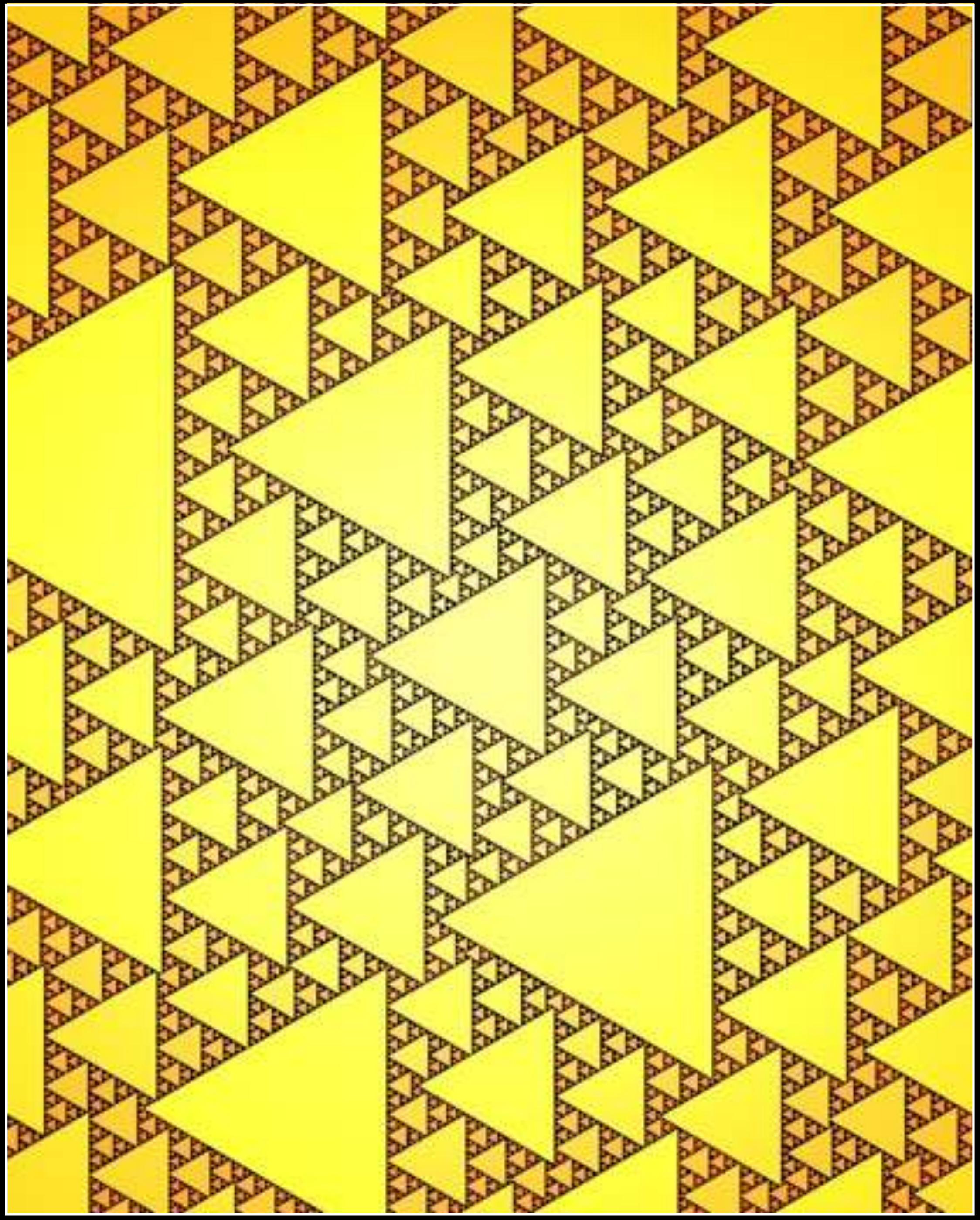


g(i) decreases too slow - run out of space

Examples

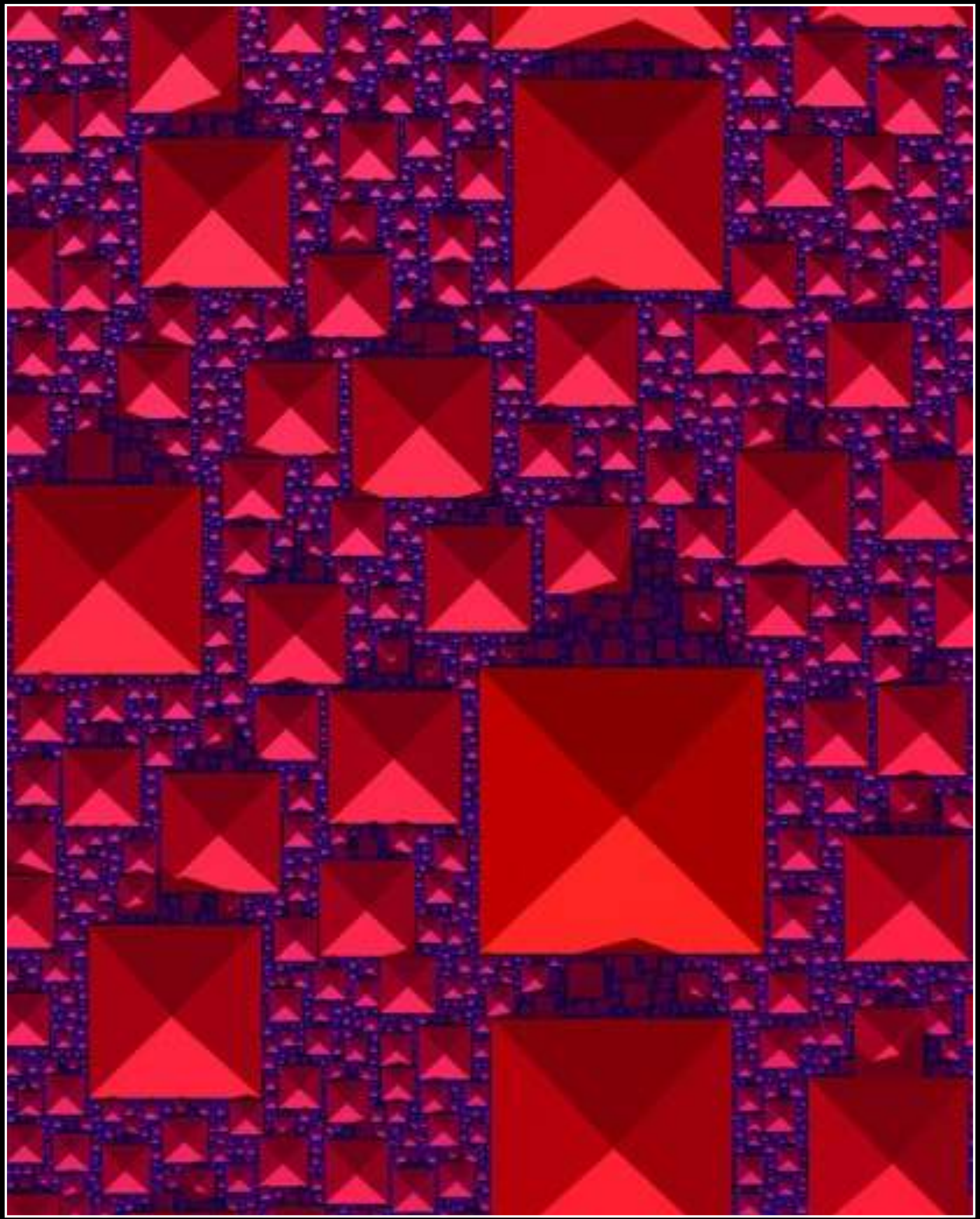
- Can use periodic bounds or simple containment.
- The shape doesn't affect the algorithm, g(i) determines the area of the added shape.

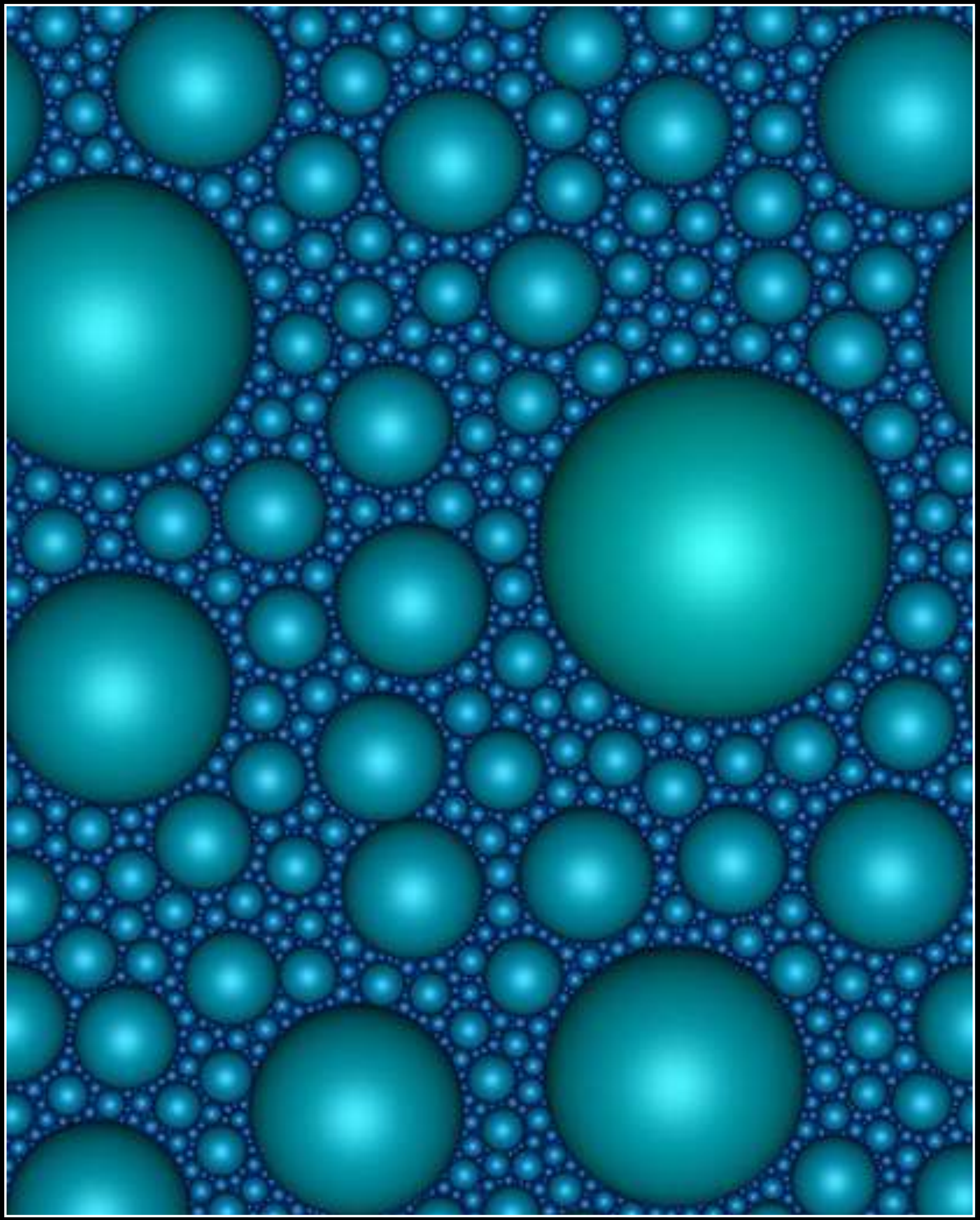




Examples

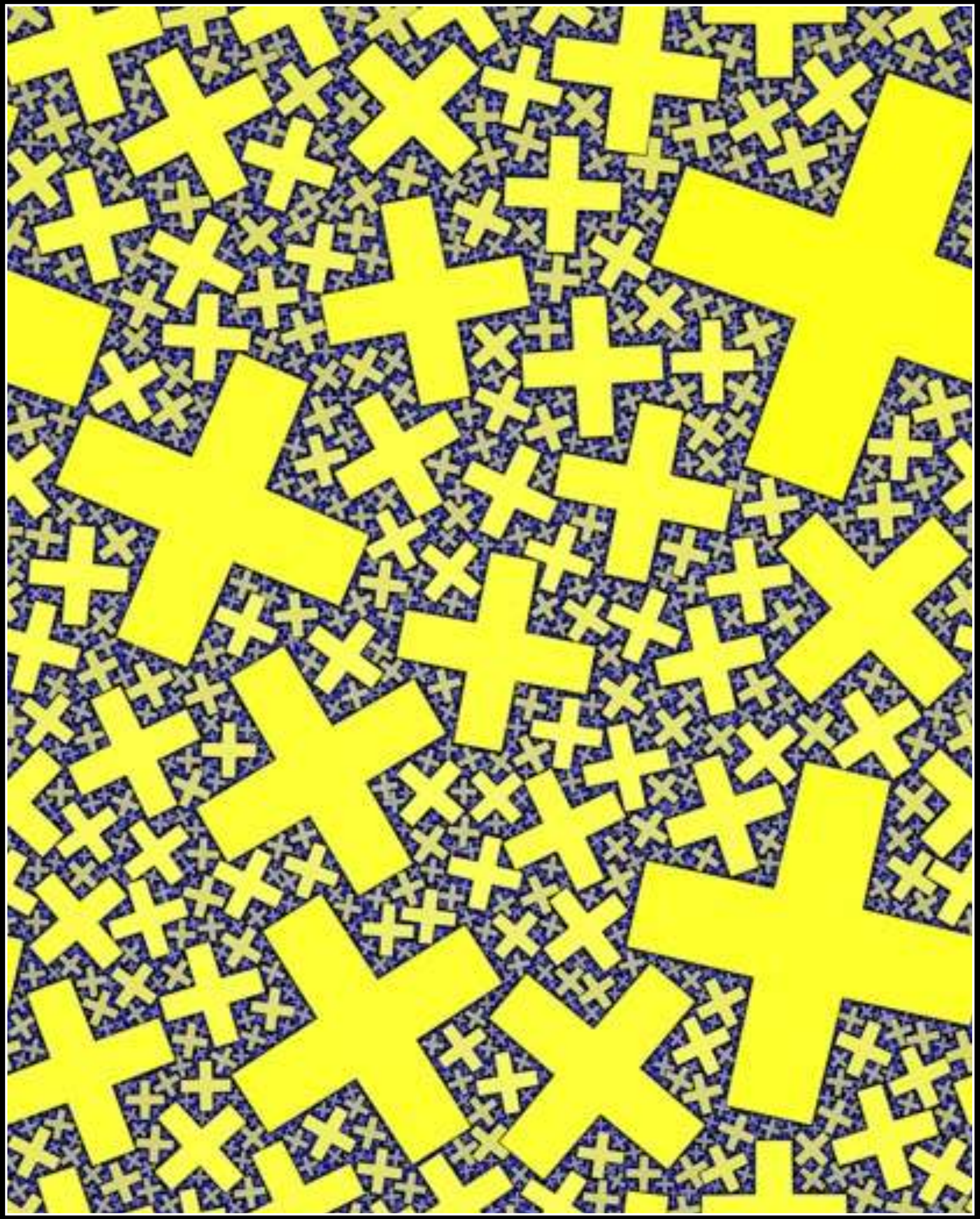
- Pseudo 3D examples.
- Only consider the intersection of the object with the planar surface.



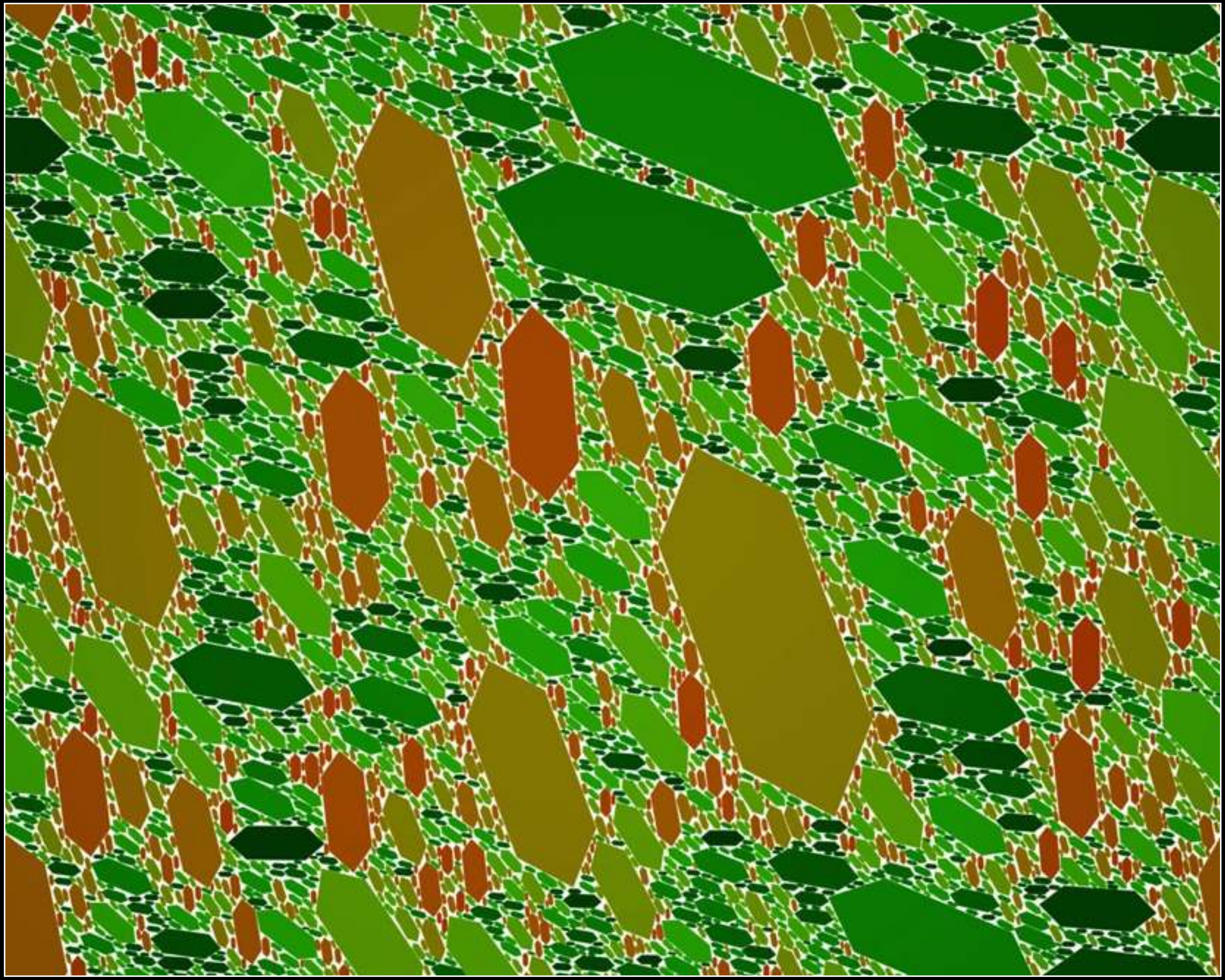


Examples

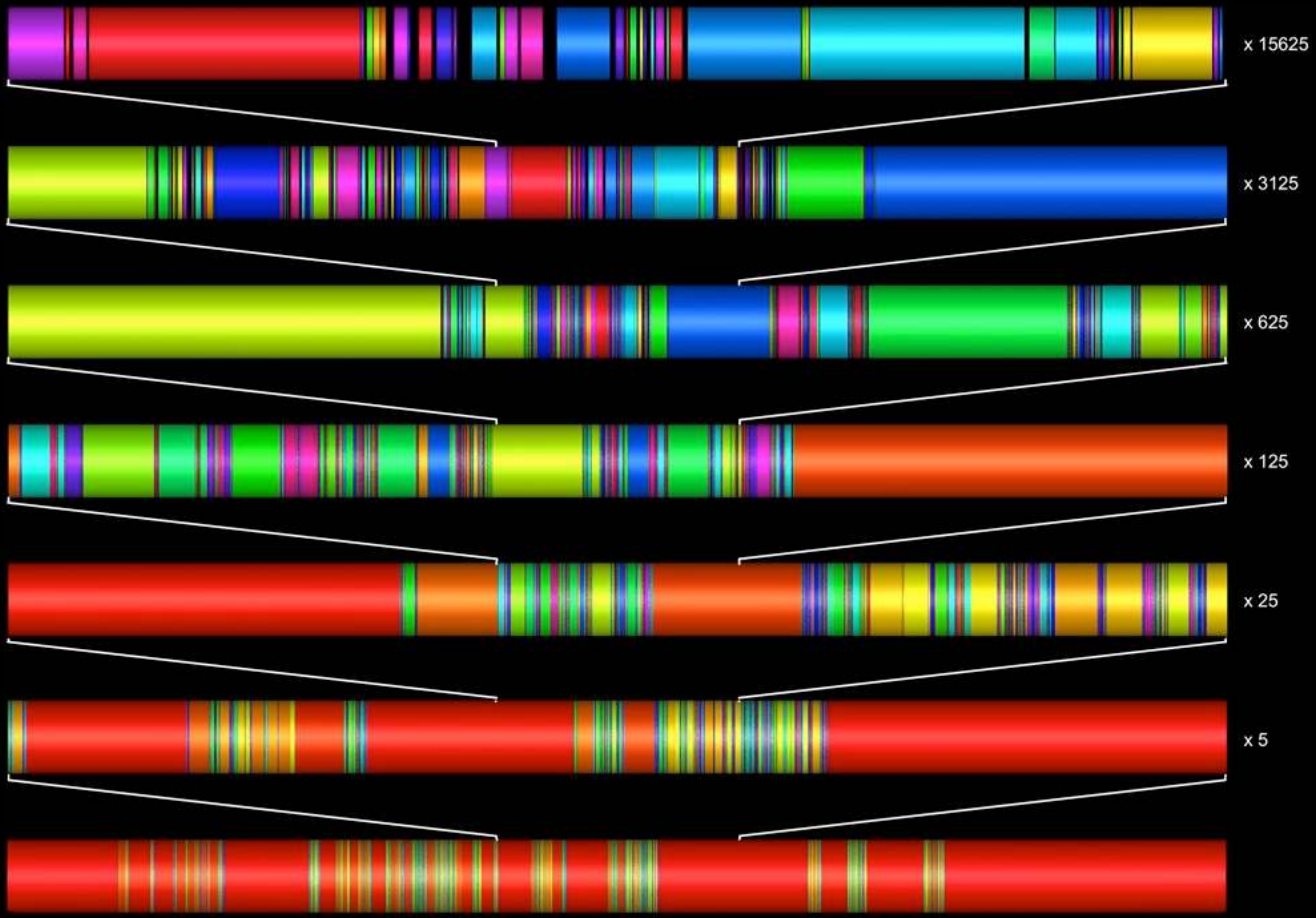
- No shapes have (yet) been found that don't space fill.
- The algorithm can run very slowly for highly concave shapes.



Example: aligned crystal domains

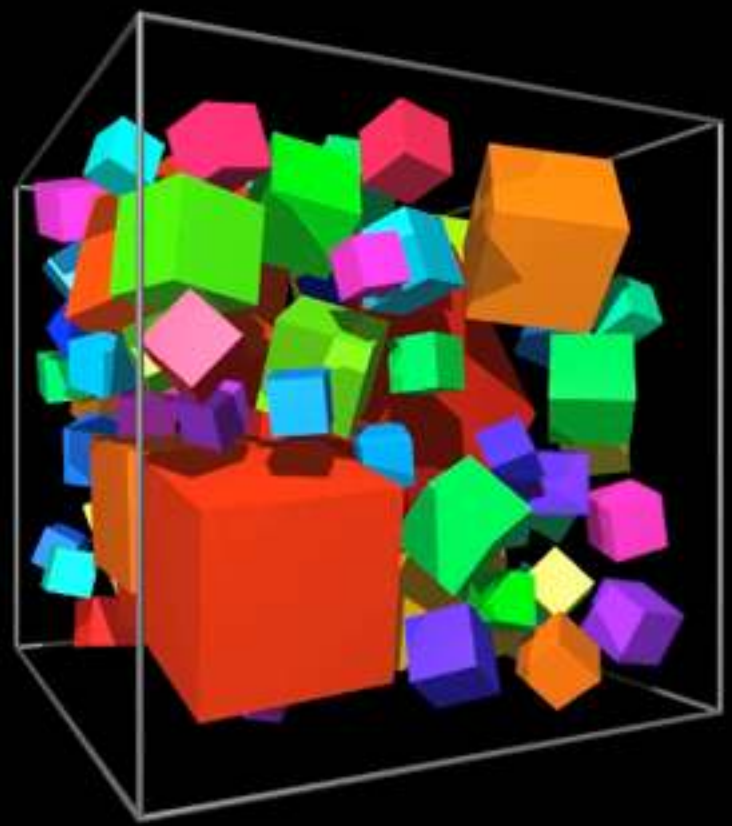


1 Dimension

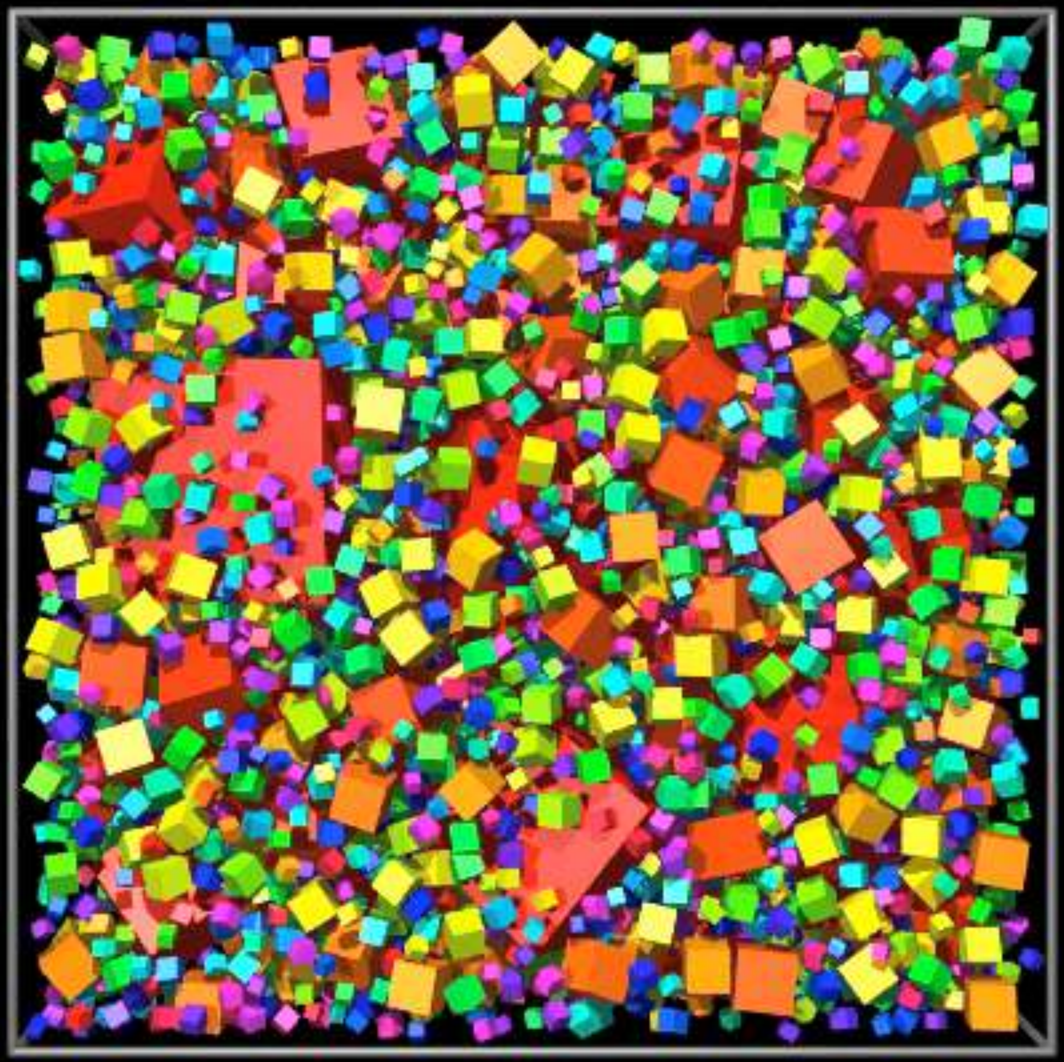


3 Dimensions

100 cubes

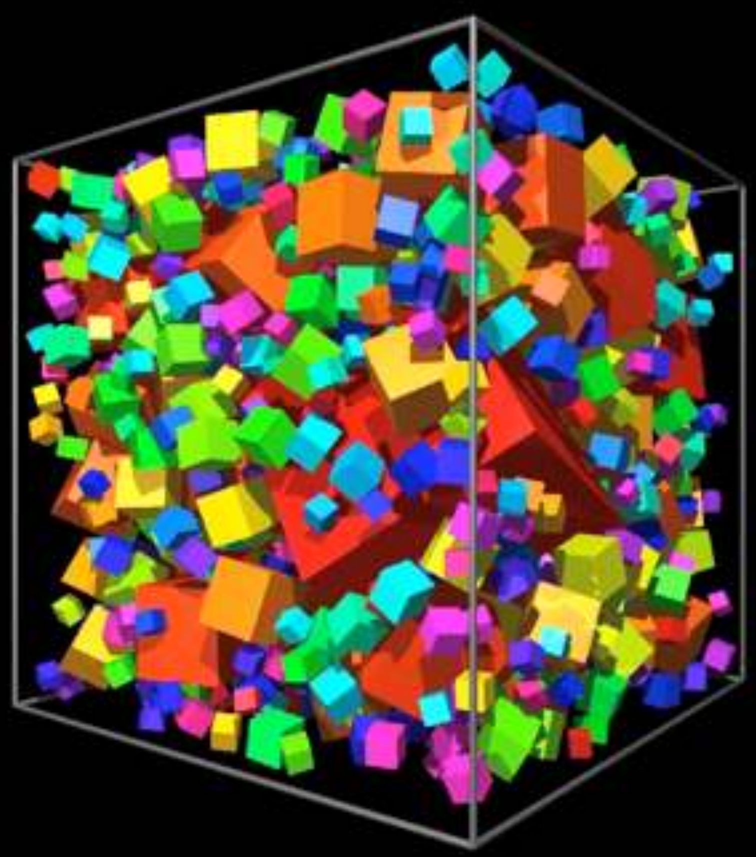


Representative frames from the movies

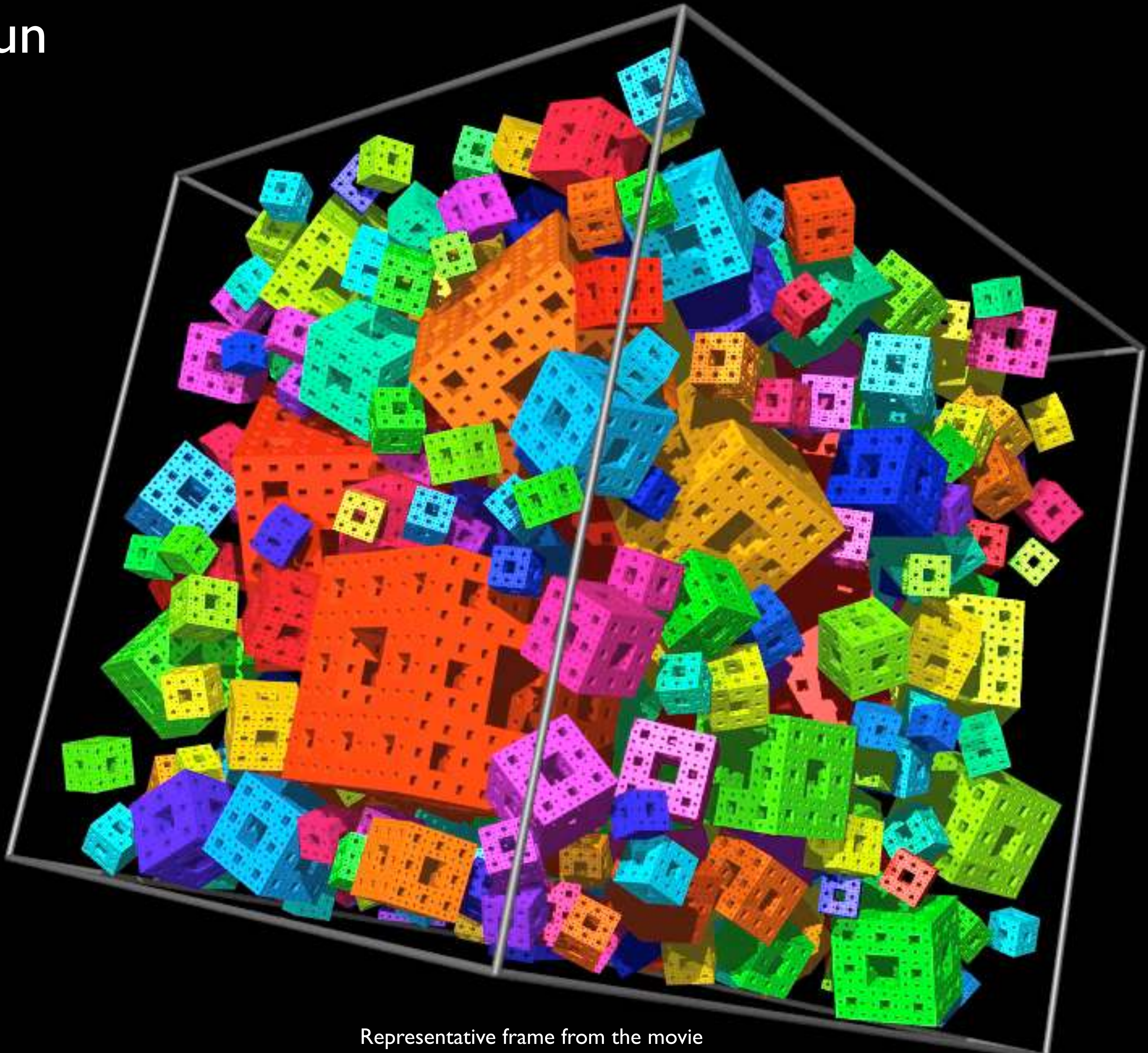


10,000 cubes

1000 cubes



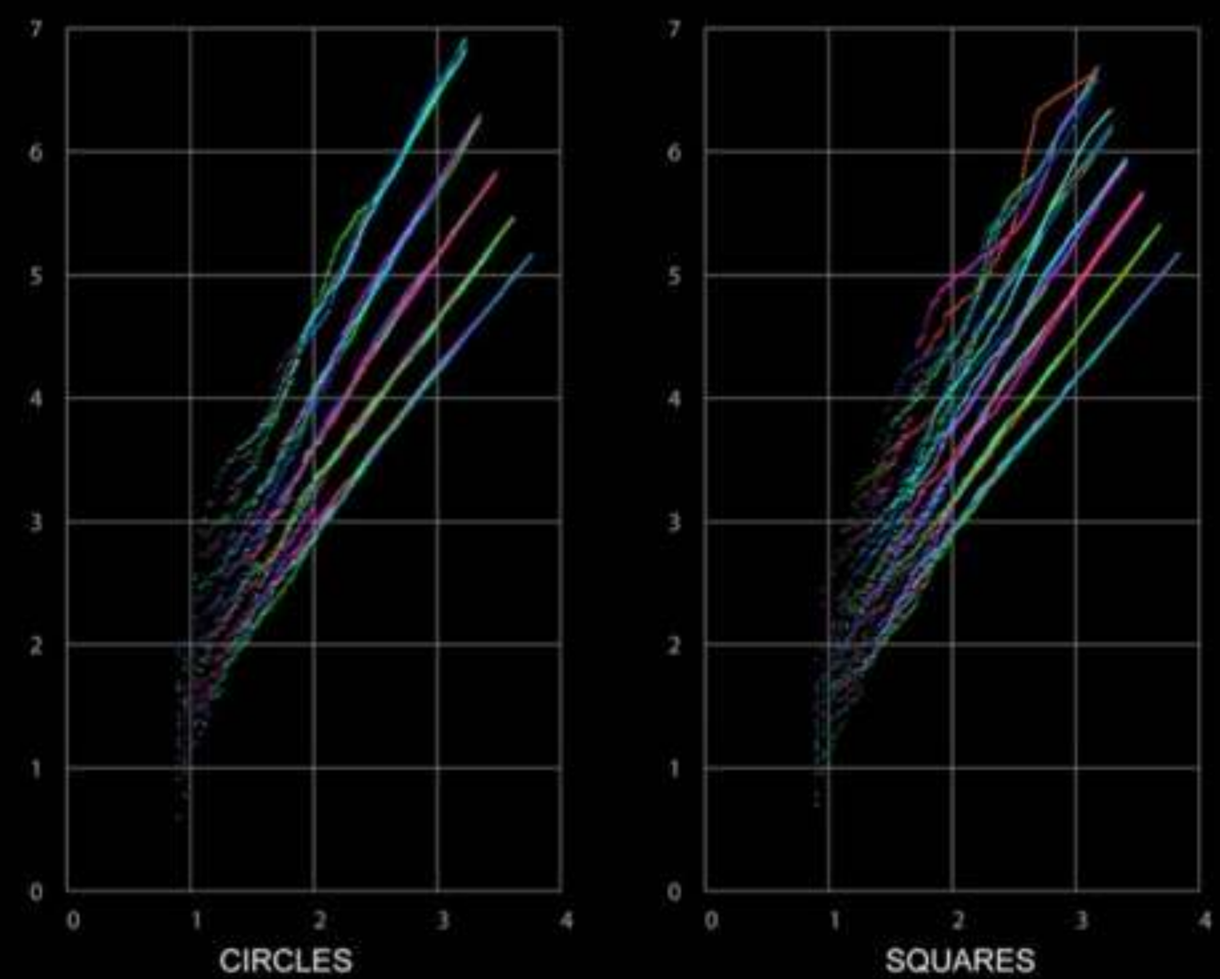
Just for fun

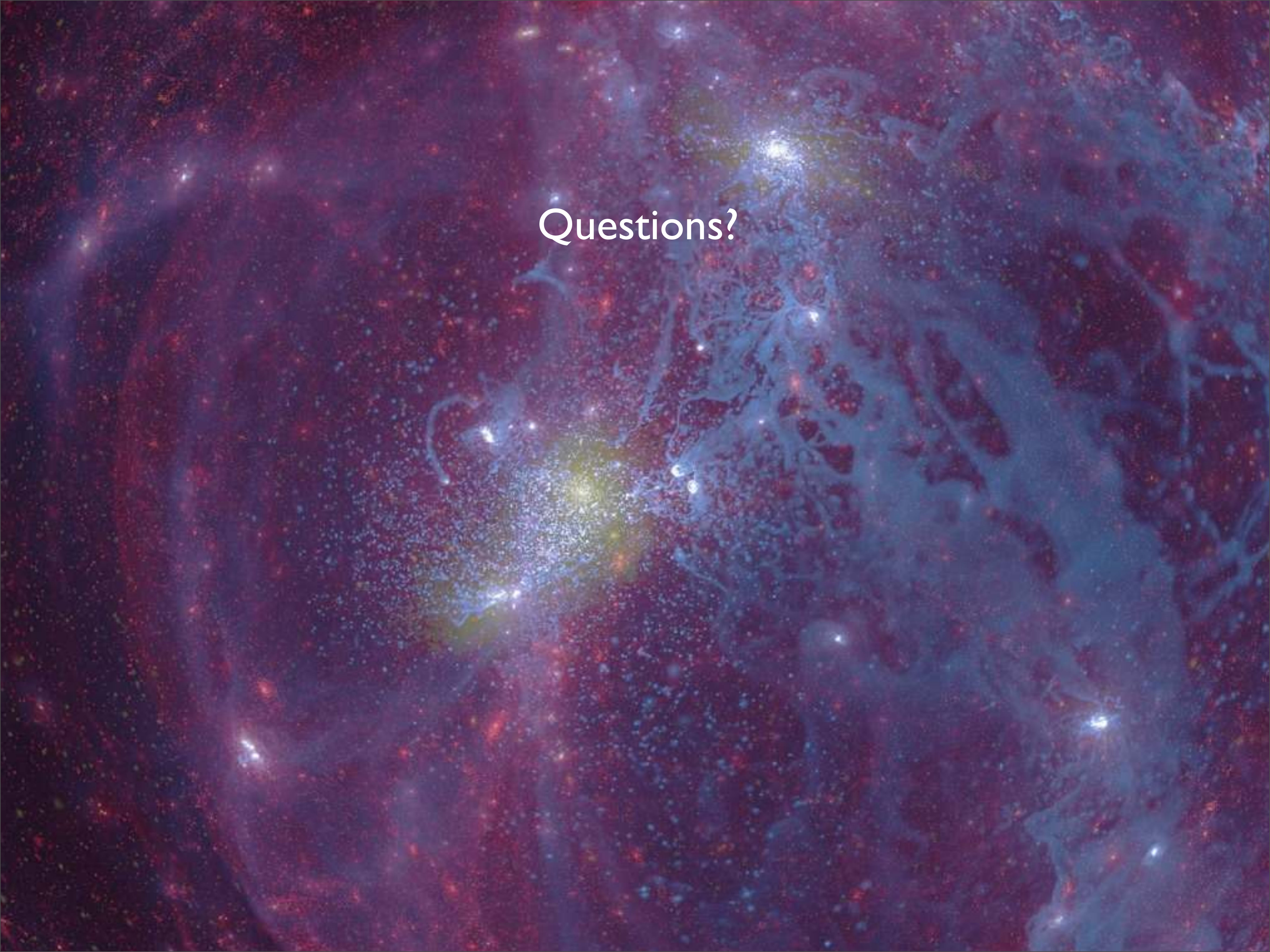


Representative frame from the movie

Current and future work

- Calculating fractal dimension (FD) for in each Euclidean dimension.
- Theoreticallly FD derived as D = d / c
  where "d" is the Euclidean dimension.
- Compare with reported dimensions from various physical processes.
- Compare area reduction functions from physical processes.
- Is the FD independent of shape?
  - We believe so.




Questions?