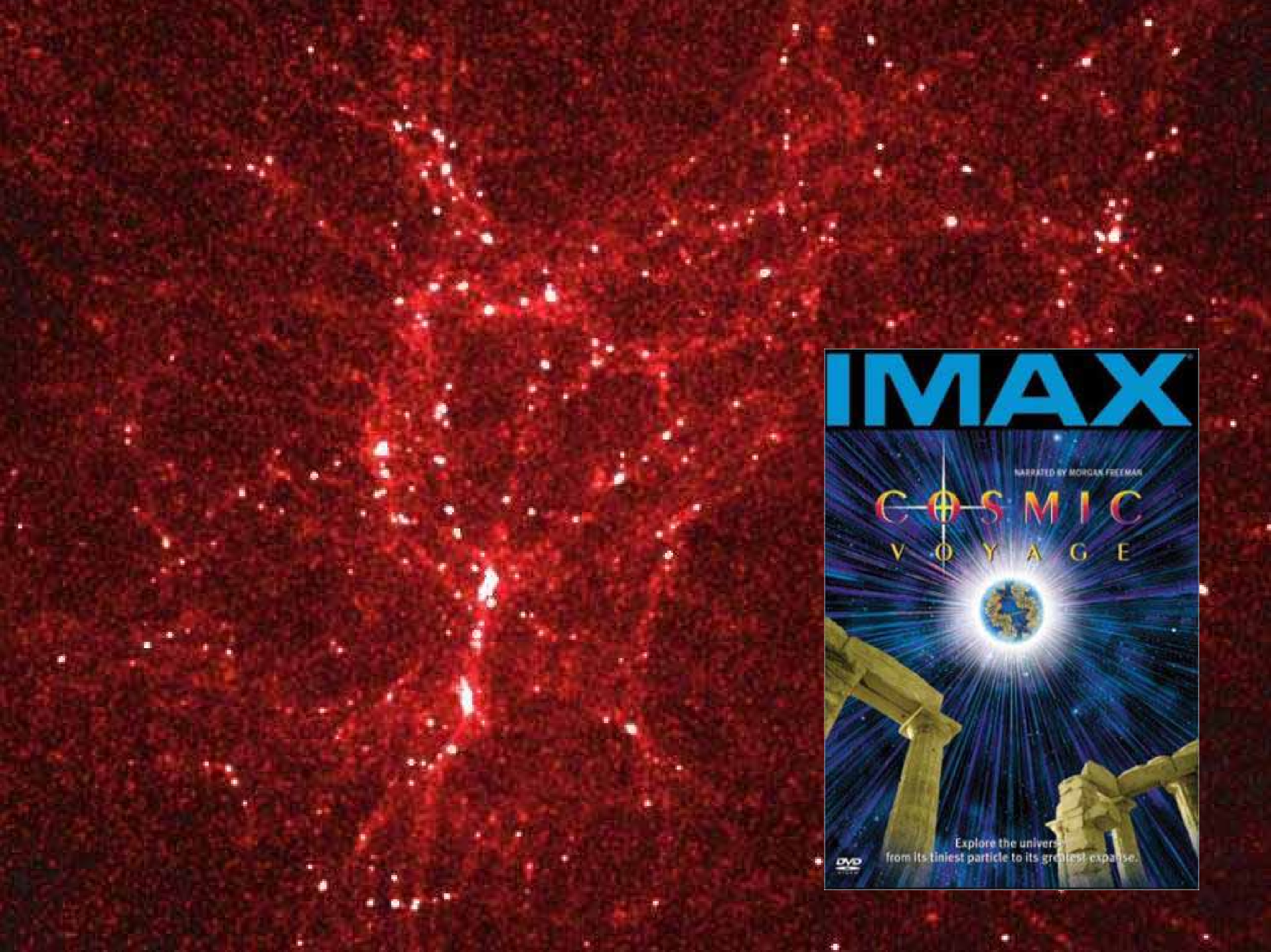
Accuracy & Aesthetics: Scientific Visualizations Using Hollywood Tools

Dr. Frank Summers

Space Telescope Science Institute

September 25, 2007


IMAX
NARRATED BY MORGAN FREEMAN
COSMIC
VOYAGE
Explore the universe
from its tiniest particle to its greatest expanse.
DVD

Research vs Outreach

- Pictures, plots, charts
- Illustrations to scientific argument
- Requires background knowledge to interpret
- Representational
- Content more important than form
- Knowledge discovery

- Images and movies
- Tells its own story
- Must play off of audience's knowledge
- More literal
- Visual message is the strongest
- Knowledge transfer

Commercial Software

Hubble: Galaxies Across Space and Time

Best Short Film 2004
Large Format
Cinema Association

a production of the Space Telescope Science Institute

http://hubblesite.org

hubble

galaxies across space and time

direction

Frank Summers

narration

Barbara Feldon

music

Jonn Serrie

production

Greg Bacon
Lynn Barranger
Ann Feild
Leigh Fletcher
Lisa Frattare
John Godfrey
Zoltan Levay
Bryan Preston
John Stoke
Frank Summers
Ed Weibe

support

Lucy Albert
Ian Griffin
Ray Villard

The Great Observatories Origins Deep Survey (GOODS) Team

Jim O'Leary, Maryland Science Center

digital and 15/70 post production

David Keighley Productions 70MM Inc.
An IMAX Company

Hubble Imax Movie

GOODS CDFS

627 M pixels

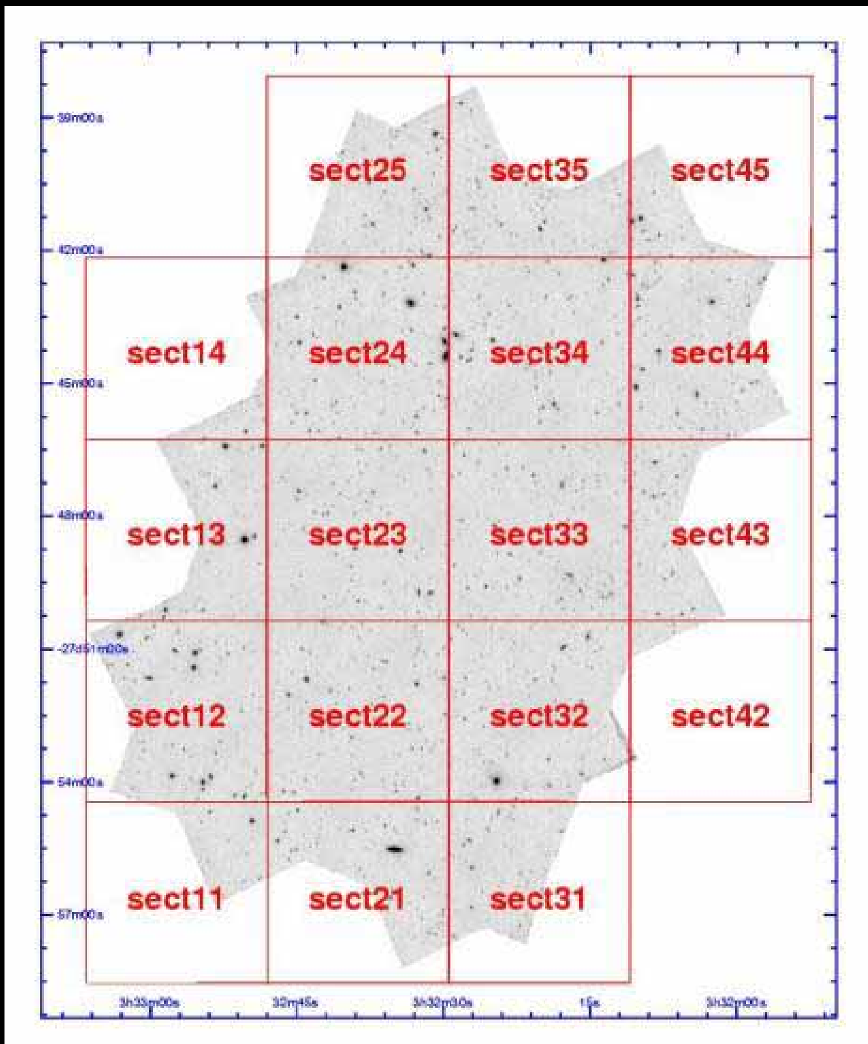
19,464 pixels

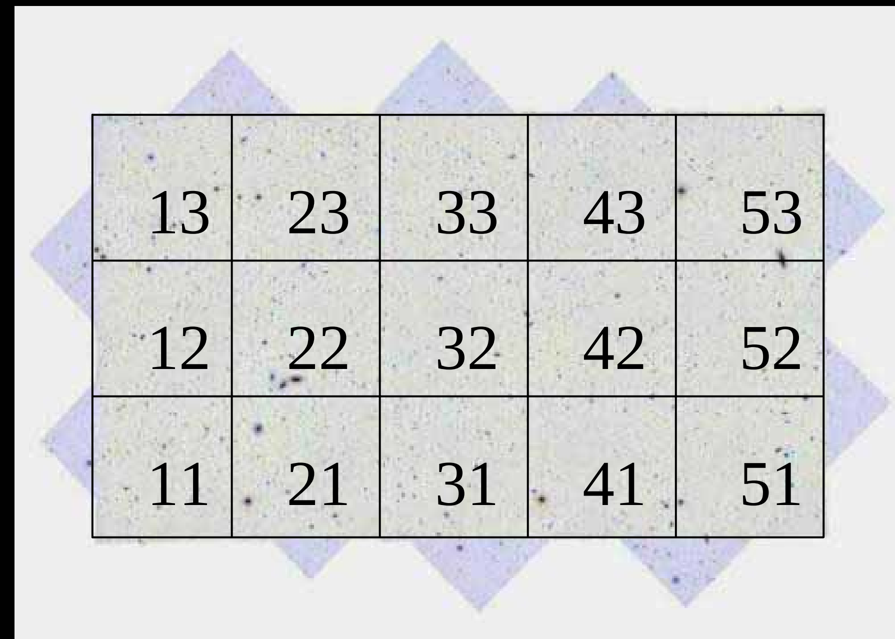
32,195 pixels

Resolutions

	Dimensions	Total Pixels
TV / VGA	640 x 480	300 k
XGA	1024 x 768	800 k
HDTV	1920 x 1080	2.1 M
WFPC2	1600 x 1600	2.6 M
Dome	< 3800 x 3800	< 14.4 M
ACS WFC	4096 x 4096	16.7 M
Viz Wall	5120 x 4096	21 M
IMAX	5616 x 4096	23 M
Sombrero	11,472 x 6429	74 M
GOODS	32,195 x 19,464	627 M

Sections to Stamps - CFITSIO

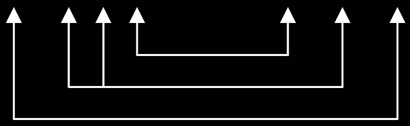




- Rotate & crop
- 5 x 3 stamps
  - 6,439 x 6,488 pixels each
  - 32,195 x 19,464 total
  - 627 M pixels

Combine - Photoshop

- 4 FITS files – 32 bits/channel
- 1 TIFF file – 8 bits/channel
- BVIZ to RGB



- Remove noise
- See faint sources
- Don’t saturate bright sources

Image Cleaning - Photoshop

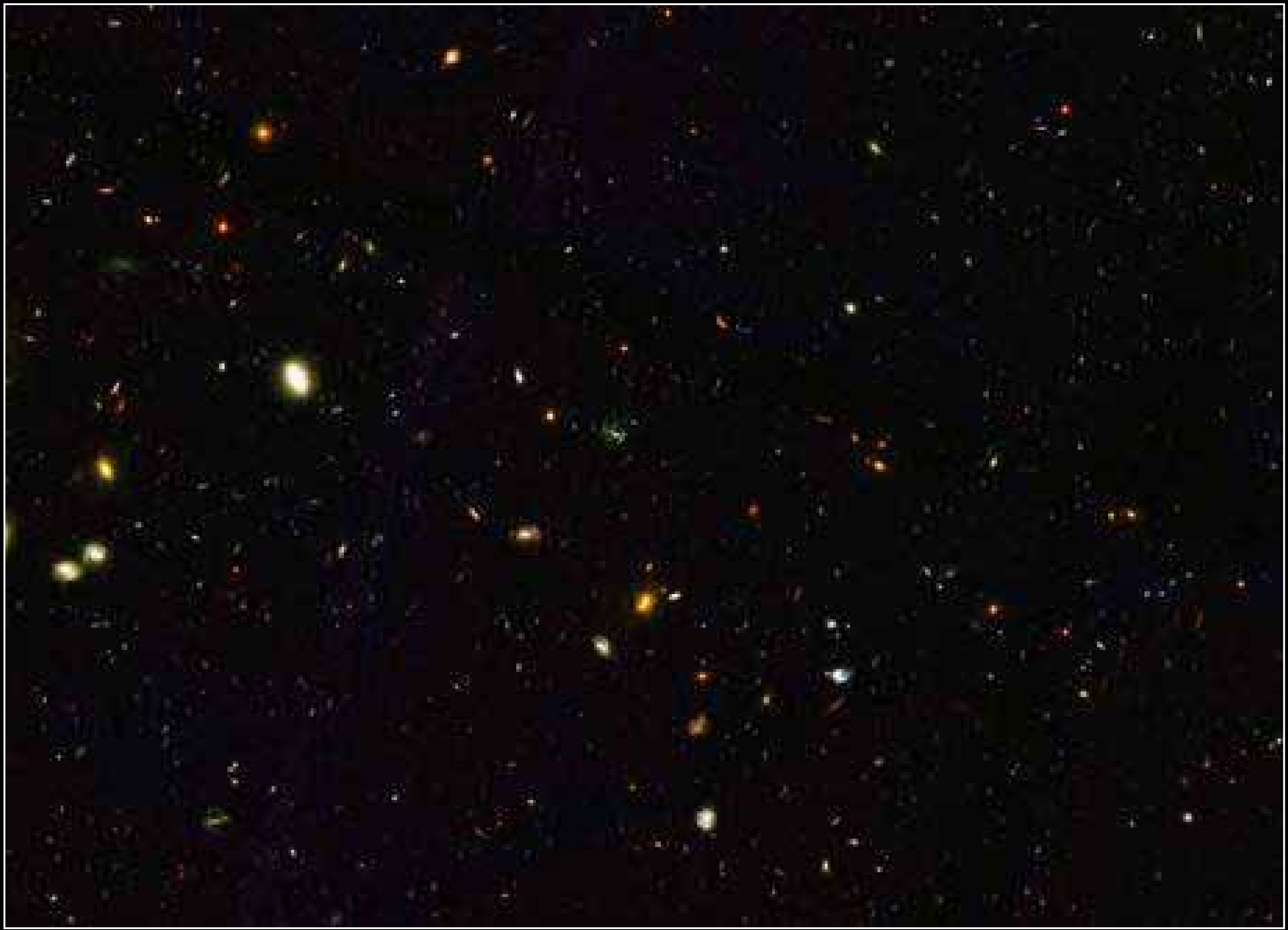


Image Cleaning - Photoshop

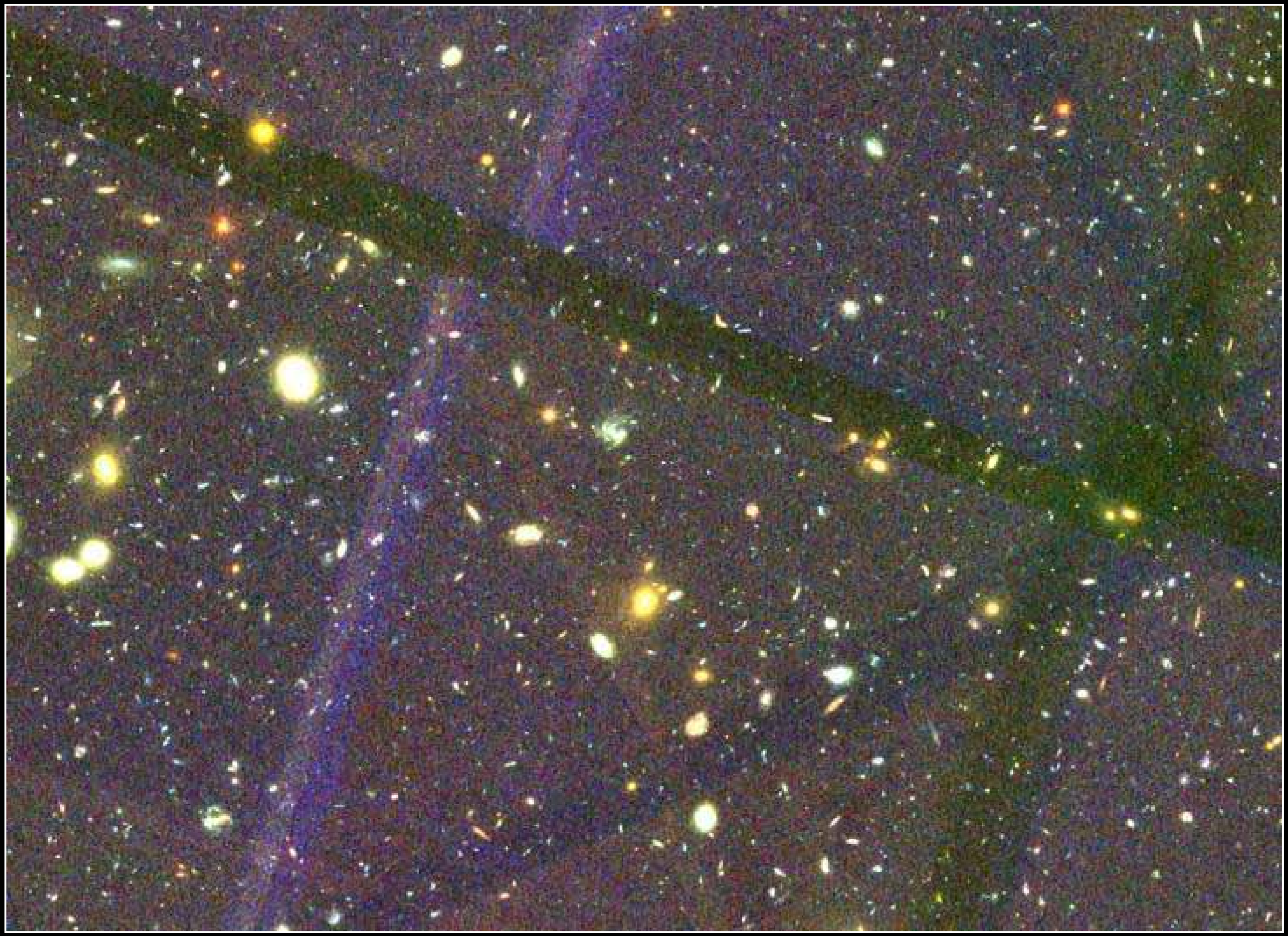
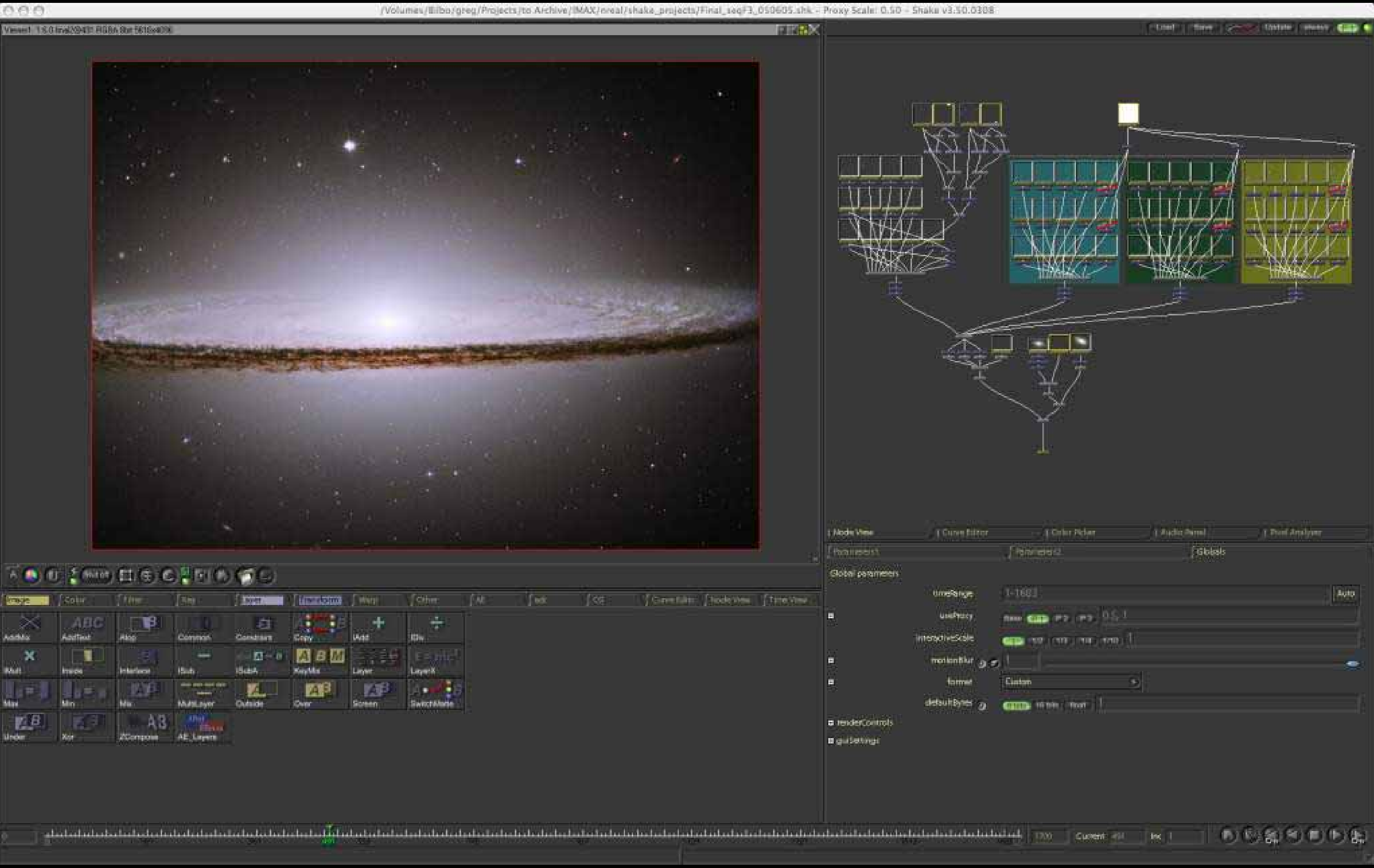
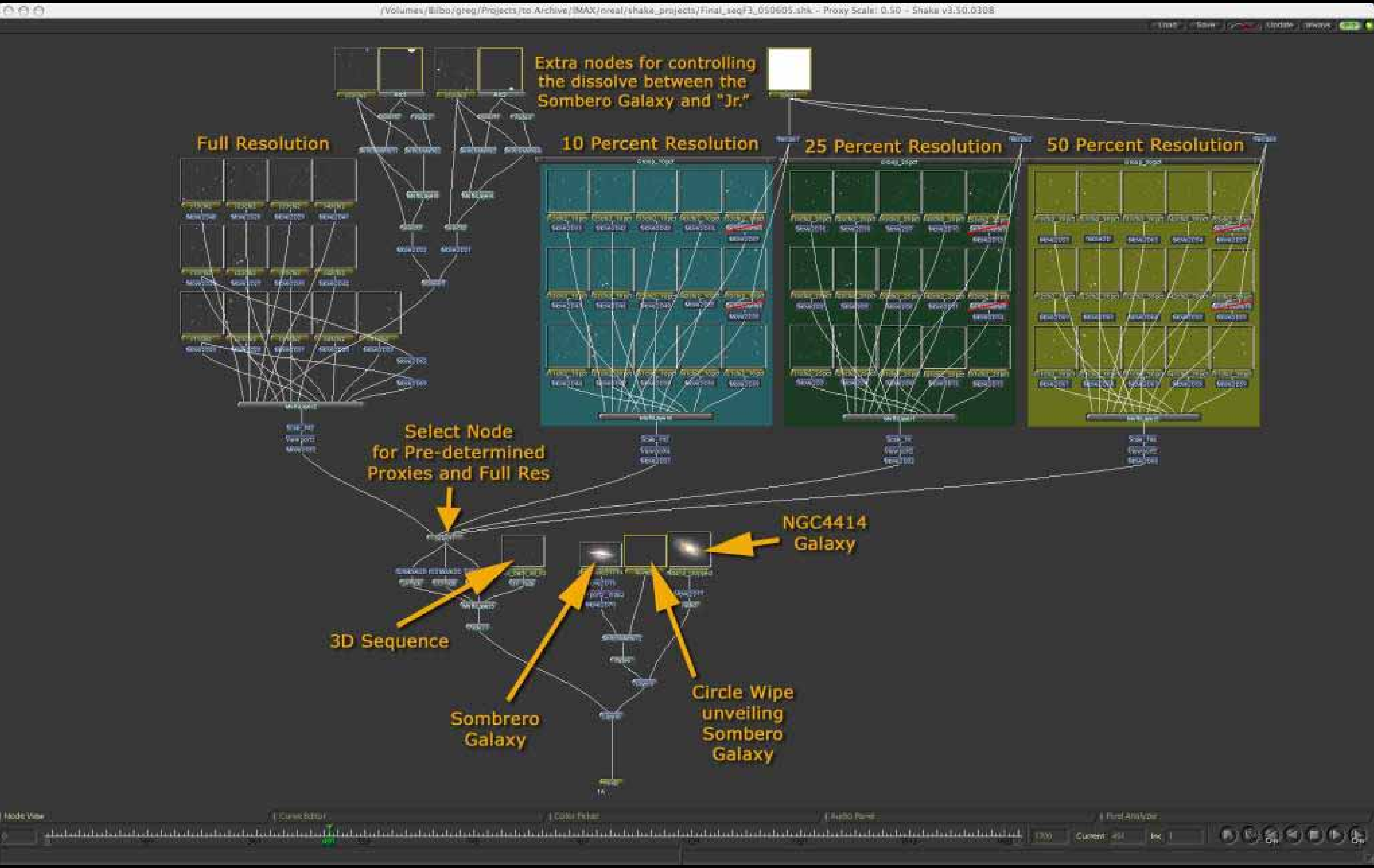


Image Cleaning - Photoshop

2D Cinematics - Shake

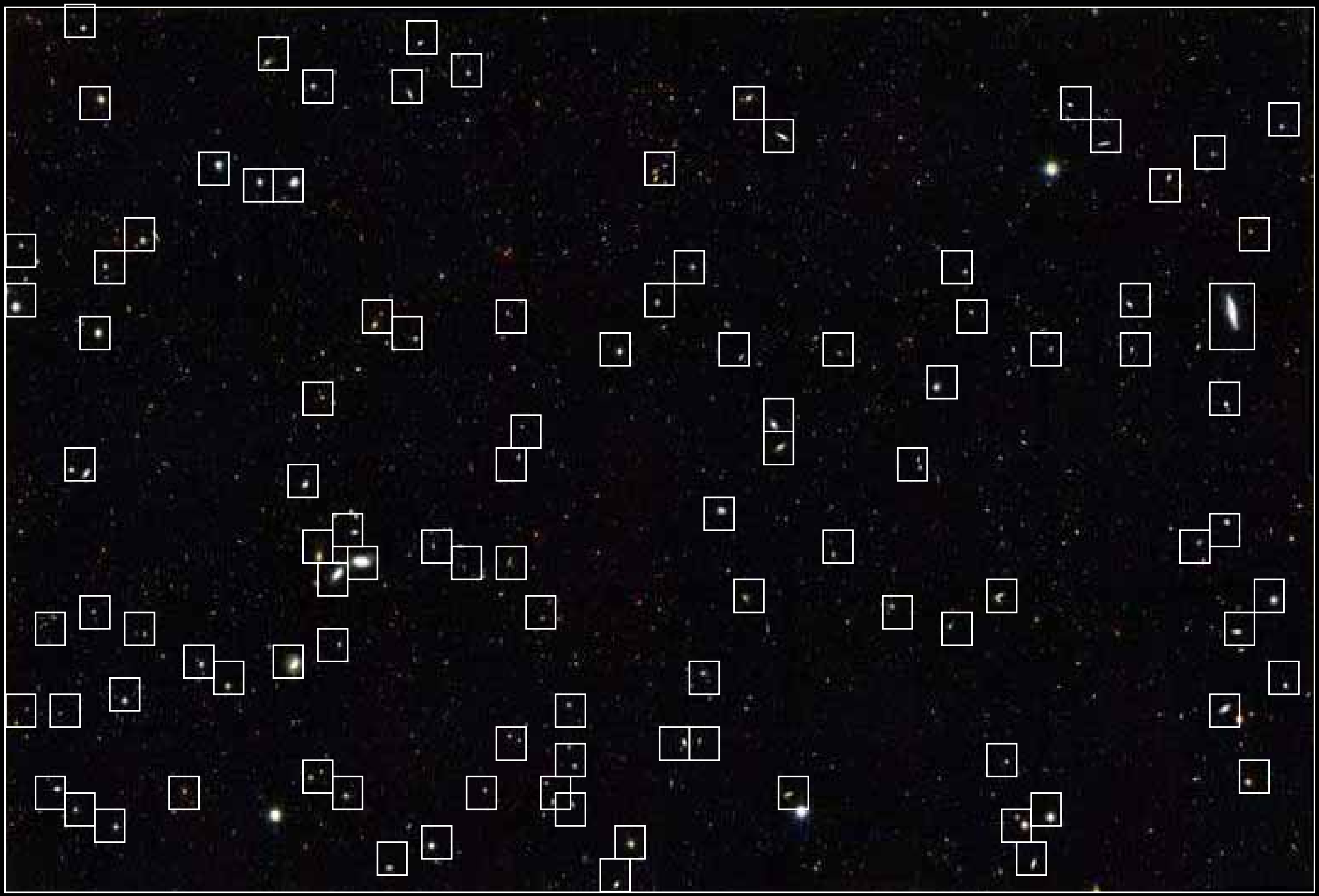


2D Cinematics - Shake

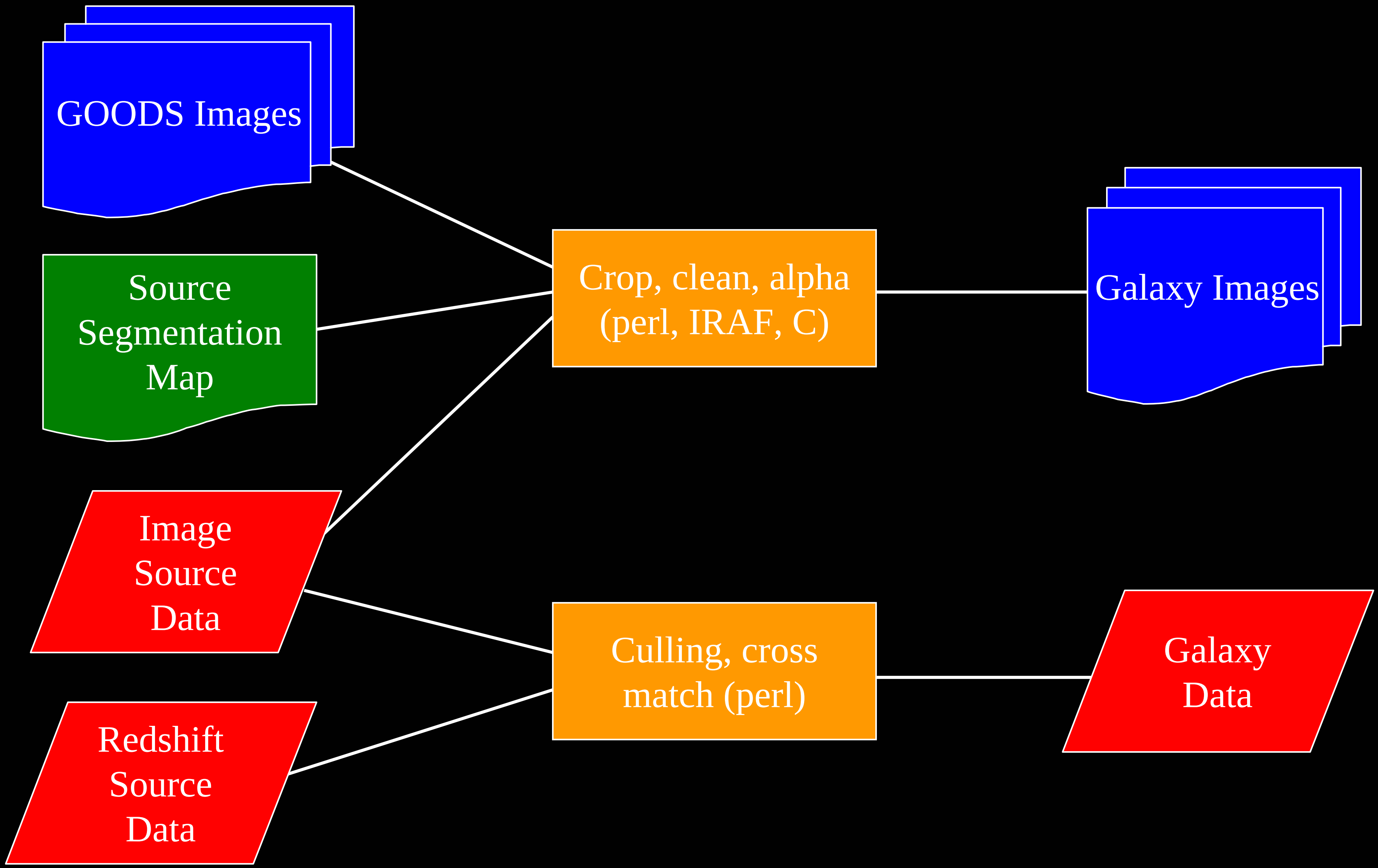


Galaxy Cut-outs & 3D Model

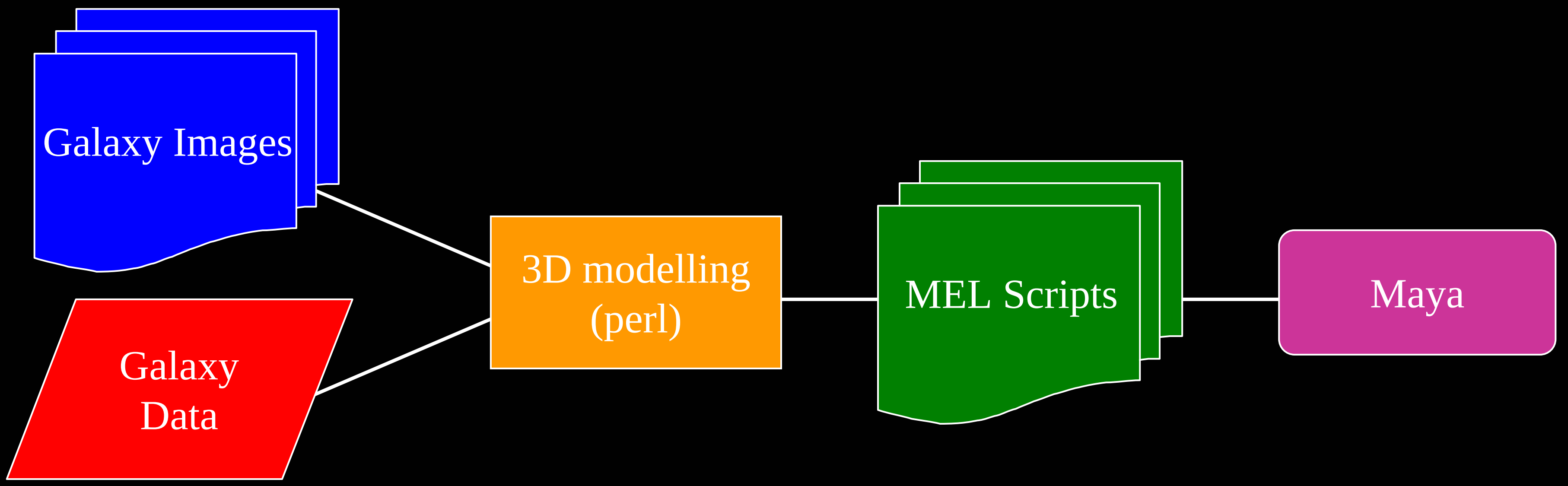
Galaxy Cut-outs & 3D Model



Data Pipeline



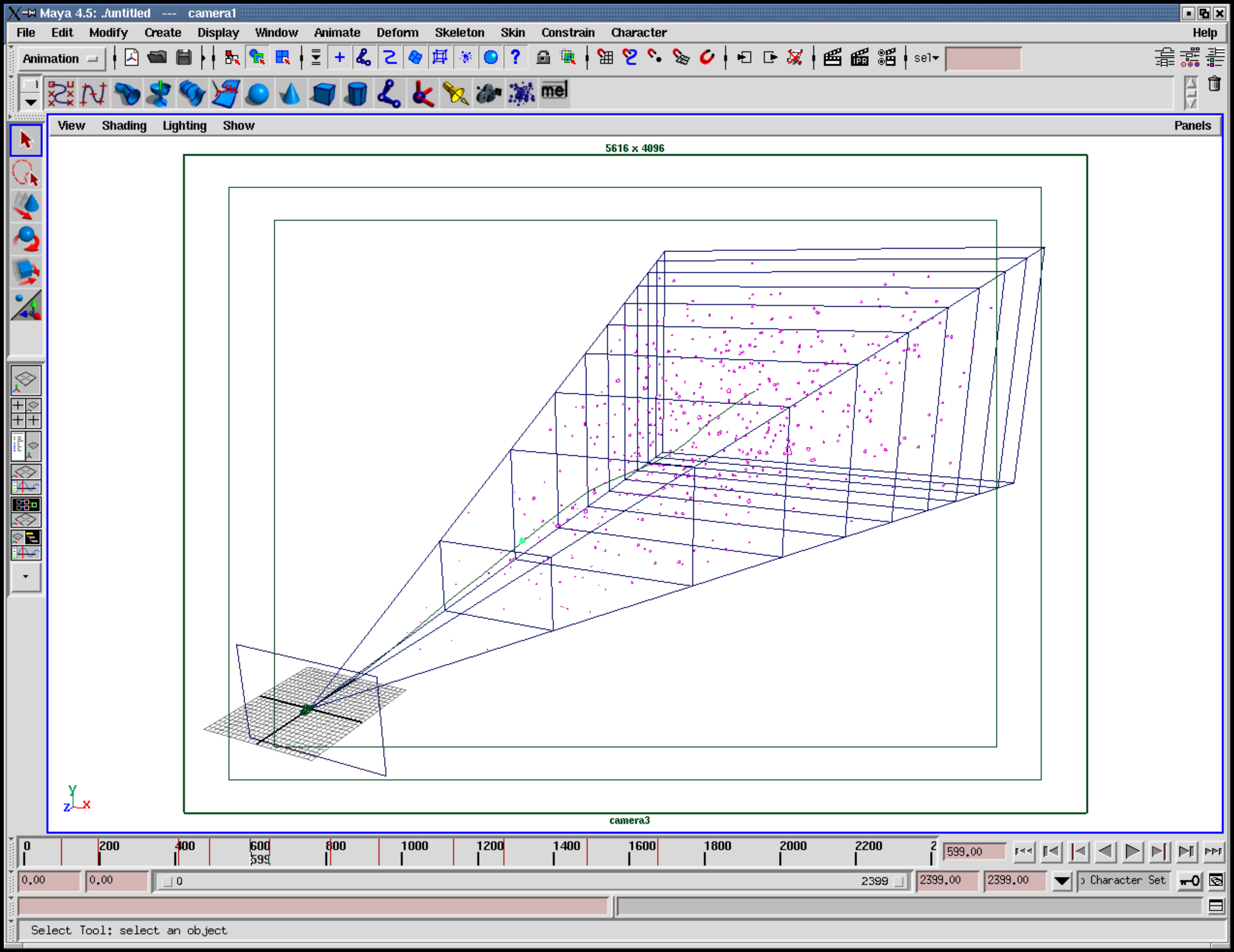
Data to Visualization

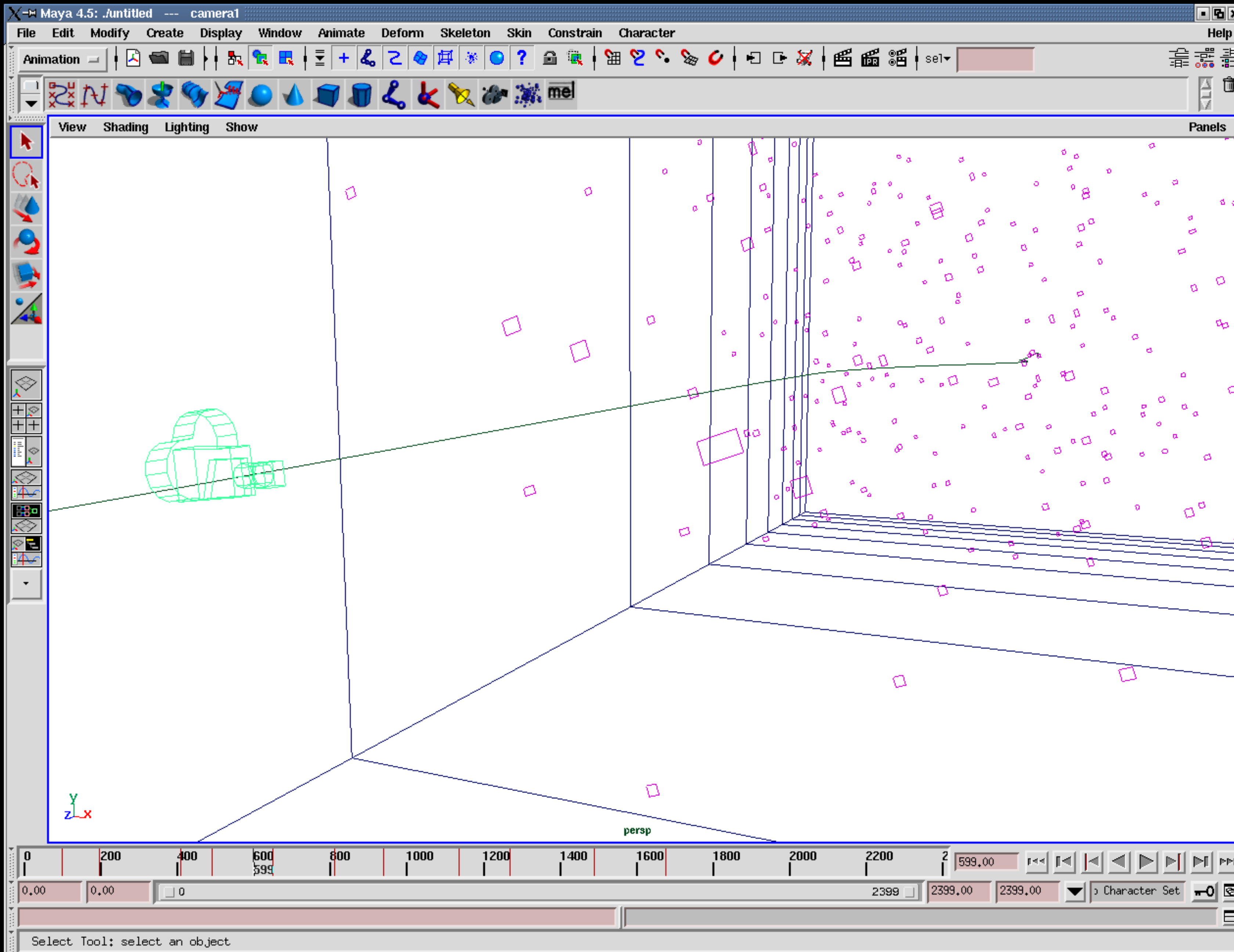


- Test in Maya
- Save as ASCII
- Edit
- Shortcuts

- Galaxy script
- Command script
- Camera script
- About a million lines of MEL

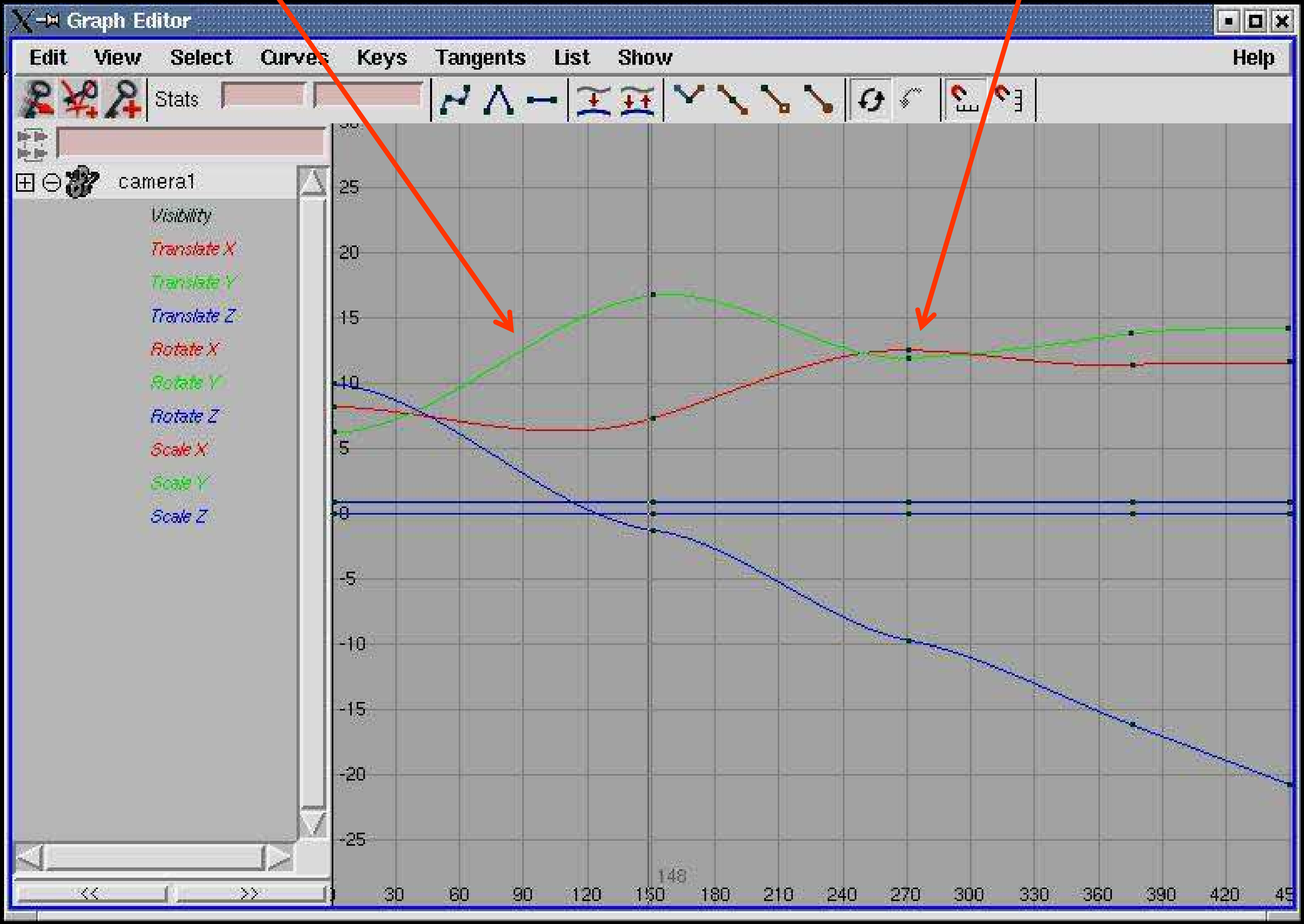
```
createNode transform -n "pPlane18471";
    setAttr …
createNode mesh -n "pPlaneShape18471" -p "pPlane18471";
createNode polyPlane -n "polyPlane18471";
createNode orientConstraint -n "pPlane18471_orientConstraint1" -p "pPlane18471";
createNode lambert -n "lambert18471";
createNode shadingEngine -n "lambert18471SG";
createNode materialInfo -n "materialInfo18471";
createNode file -n "file18471";
createNode place2dTexture -n "place2dTexture18471";
//
connectAttr "polyPlane18471.out" "pPlaneShape18471.i";
connectAttr "pPlane18471.ro" "pPlane18471_orientConstraint1.cro";
connectAttr "camera1.ro" "pPlane18471_orientConstraint1.tg[0].tro";
connectAttr "lambert18471SG.msg" "lightLinker1.lnk[18471].olnk";
connectAttr "file18471.oc" "lambert18471.ic";
connectAttr "pPlaneShape18471.iog" "lambert18471SG.dsm" -na;
```


Maya 4.5: ./untitled --- camera1
File
Edit
Modify
Create
Display
Window
Animate
Deform
Skeleton
Skin
Constrain
Character
Help
Animation
sel
mel
View
Shading
Lighting
Show
Panels
5616 x 4096
camera3
0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
599.00
0.00
0.00
0
2399
2399.00
2399.00
Character Set
Select Tool: select an object

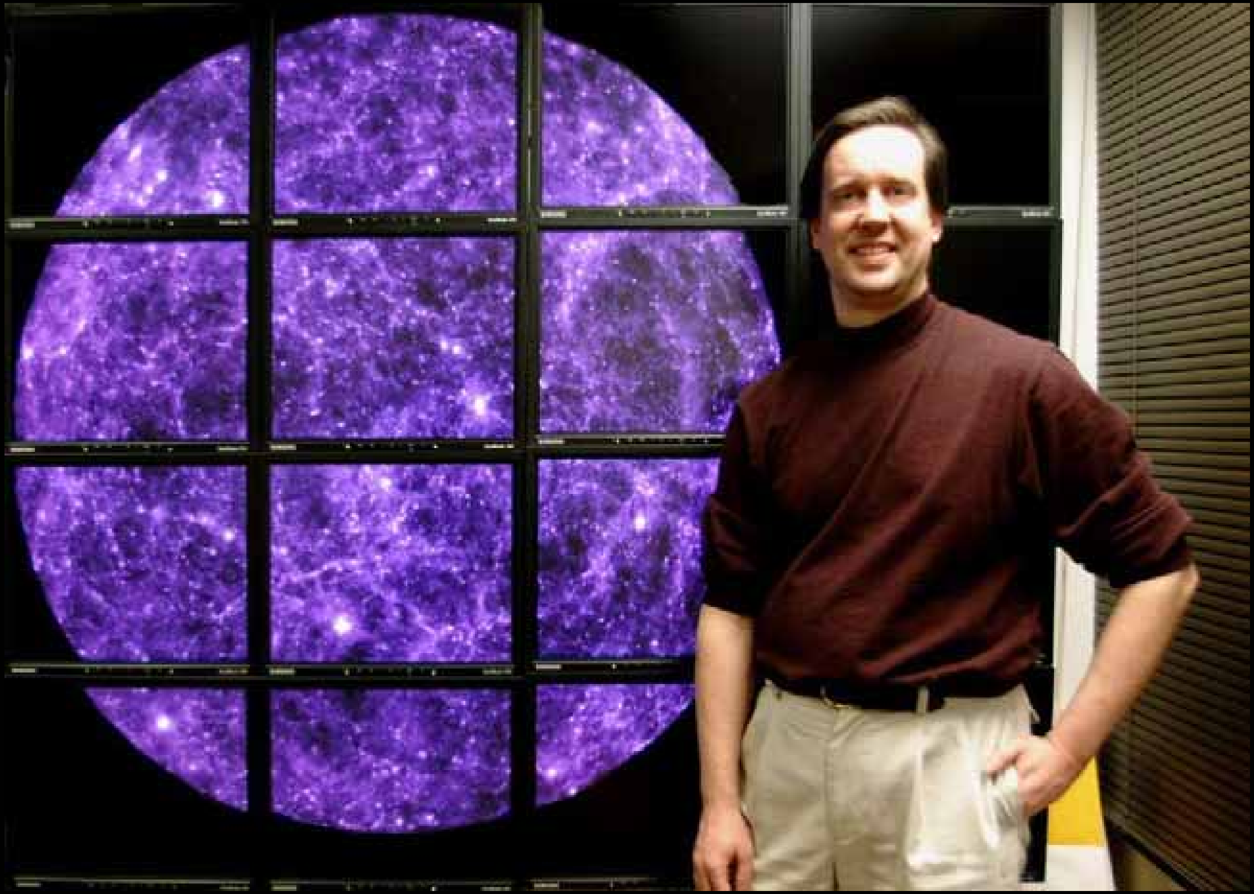

Maya 4.5: ./untitled --- camera1
File Edit Modify Create Display Window Animate Deform Skeleton Skin Constrain Character Help
Animation
sel
mel
View Shading Lighting Show
Panels
persp
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200
599
599,00
0,00
0,00
0
2399
2399,00
2399,00
Character Set
Select Tool: select an object

Camera tracks in x, y and z

Dots are keyframe positions



Visualization Wall



Free Software

Renderman Interface

- Pixar specification
- Renderers
  - Commercial: PRMan, Air, RenderDotC, 3Delight
  - Free: BMRT, Aqsis, Pixie
- APIs
  - C, Java, perl, python, Tcl
- RIB files

```
##RenderMan RIB-Structure 1.0
version 3.03
#
Projection "perspective"
Display "fisheye_splat.tiff" "tiff" "rgb"
ScreenWindow -1 1 -1 1
Format 480 480 1
Clipping 0.049 1000.0
#
WorldBegin
#
Surface "fjs_fisheyelens" "lens_angle" [180] "zdistance" [0.05] "scale" [0.05]
Polygon "P" [0.05 0.05 0.05 0.05 -0.05 0.05 -0.05 -0.05 0.05 -0.05 0.05 0.05]
#
Attribute "render" "integer visibility" [3]
#
Translate 0.95940 0.93531 1
Surface “fjs_splat" "splatcolor" [1 1 1] "radius" [1] "amplitude_g" [1] "sigma_g" [0.4]
"amplitude_e" [1.0] "sigma_e" [0.16] "percent_g" [1.0] "exposure" [1.0]
Disk 0 0.0110226 360
#
```

Shading

- Exact geometric modelling can get very complex
- Shading - use simple shapes and add complexity when drawing the surface
  - Texture - color, pattern
  - Bumps - small shape distortions
  - Light - reflection, transparency
- Programmability = Flexibility

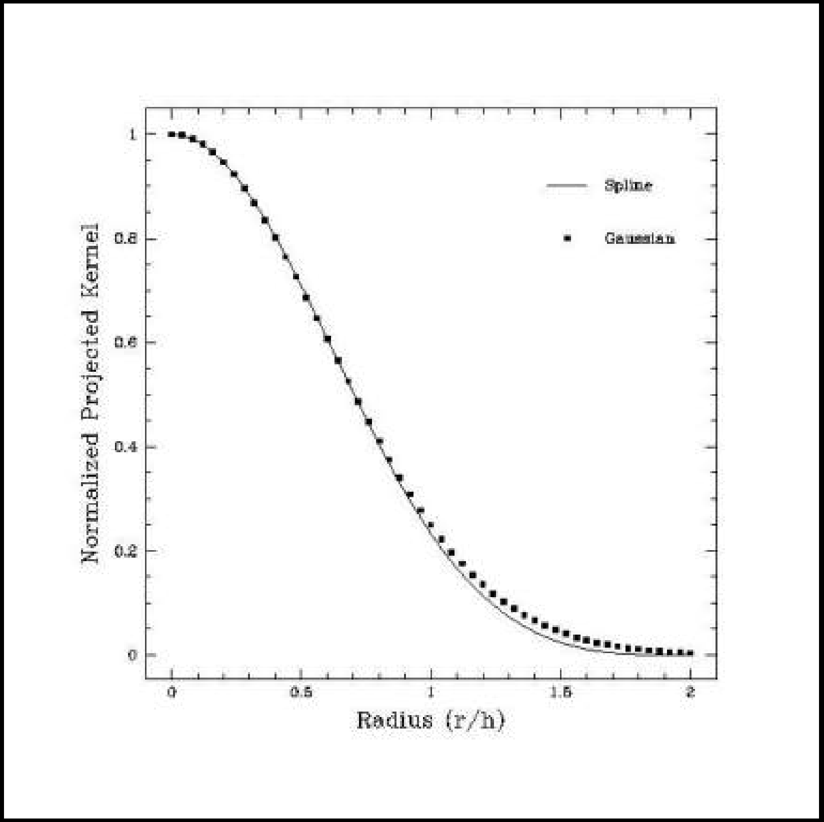
Shading Example: Teapot

N-body & SPH Simulations

- N-body simulations
  – particle based gravity
  – gravity is “softened” on small scales
- Smoothed Particle Hydrodynamics
  – particles represent gas clouds
  – smoothing kernel – density profile
  – adapts over space and time
- Work well together
  – stars, galaxies, cosmology sims

SPH Shader

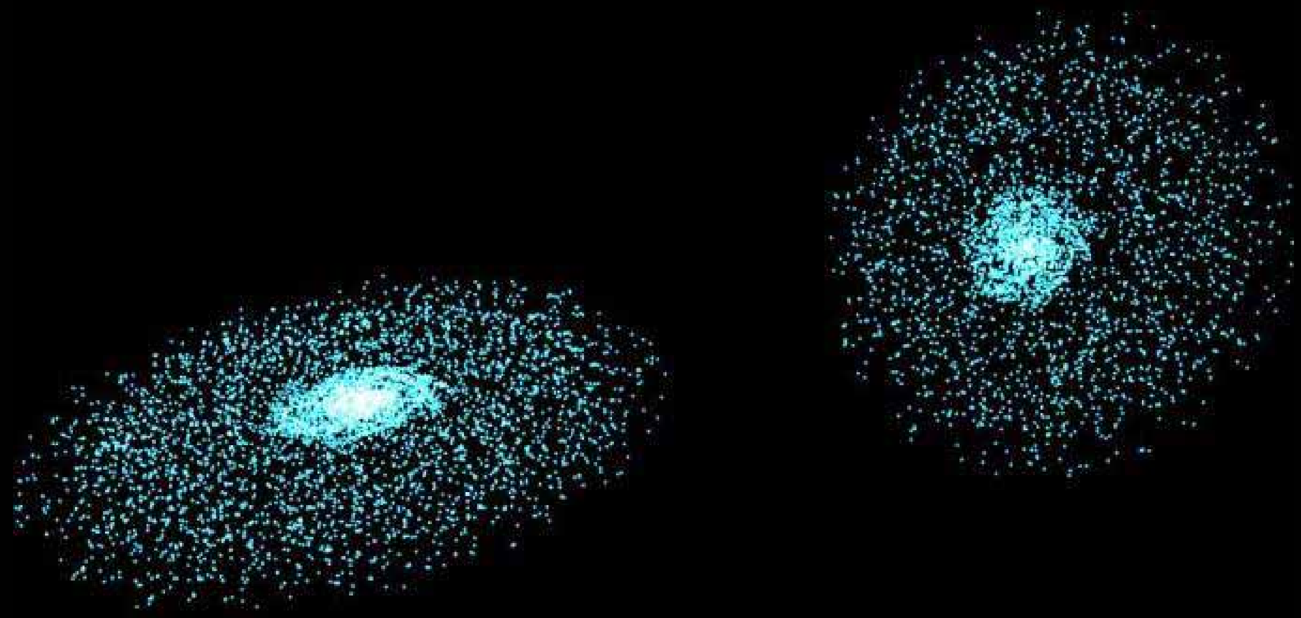
- Disk geometry
- Shade with 2D projection of smoothing kernel
  - Gaussian splat
  - Semi-transparent
- Can use softening length for gravity or calculate smoothing
- Near exact visual representation



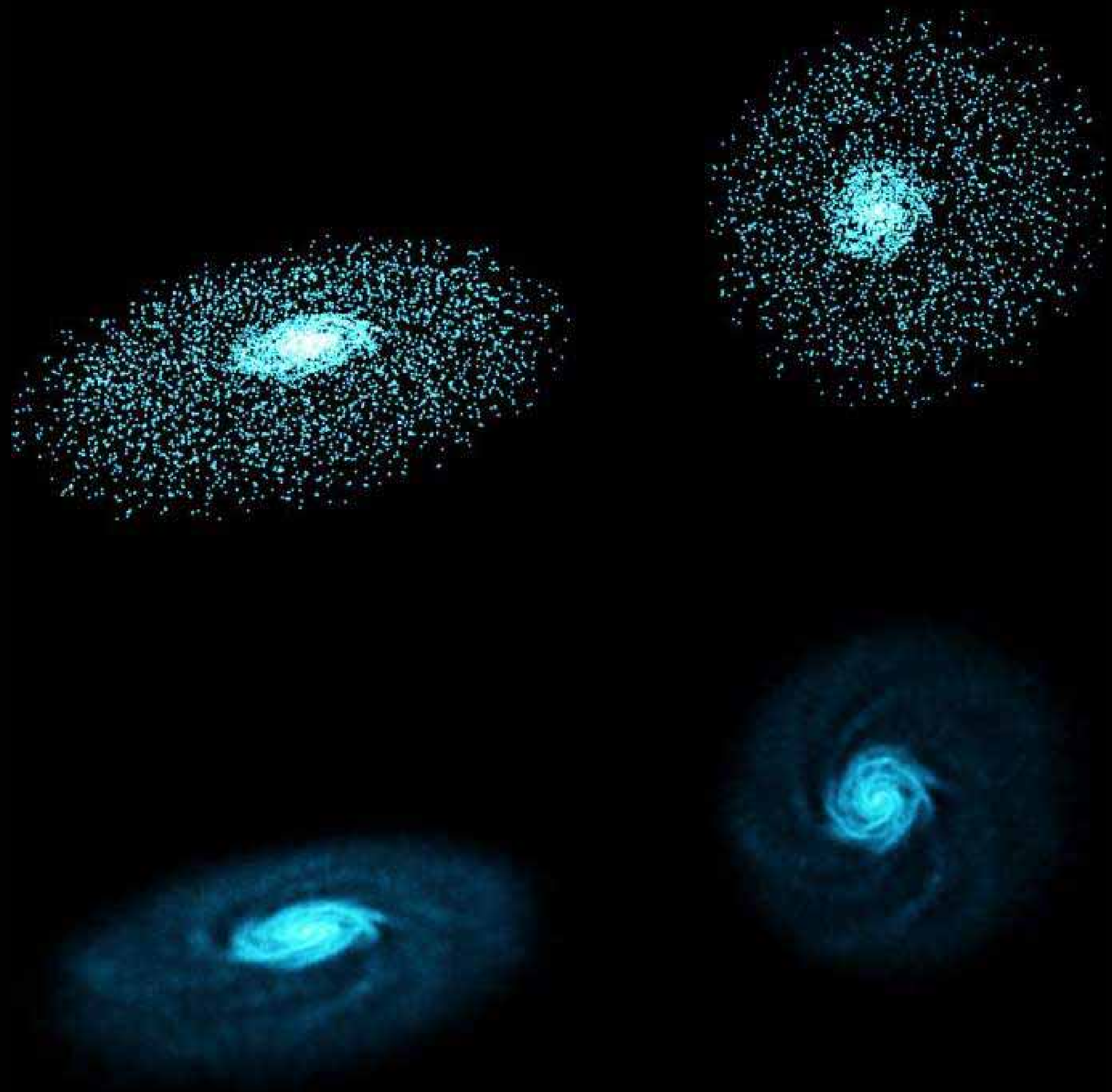
```
/* splat.sl */
Surface splat(  … variables … ) {
  … variables …

/* Set radius */
  r = (1.0 - t)*radius;

/* Set contribution from gaussian */
  if (percent_g > 0.0) {
    value_g = amplitude_g*exp(-r*r/(sigma_g*sigma_g));
    value_g = value_g*exposure;
    if (value_g > 1.0) value_g = 1.0;
  } else {
    value_g = 0.0;
  }

/* Set contribution from exponential */
  percent_e = 1.0 - percent_g;

  if (percent_e > 0.0) {
    value_e = amplitude_e*exp(-r/sigma_e);
    value_e = value_e*exposure;
    if (value_e > 1.0) value_e = 1.0;
  } else {
    value_e = 0.0;
  }
```

```
/* Calculate sum of contributions */
  value = percent_g*value_g + percent_e*value_e;

/* Set opacity */
  if (value >= 1.0) {
    opacity = color(1, 1, 1);
  } else {
    opacity = color(value, value, value);
  }

/* Set output values */
  Oi = opacity;
  Ci = splatcolor*value;
}
```

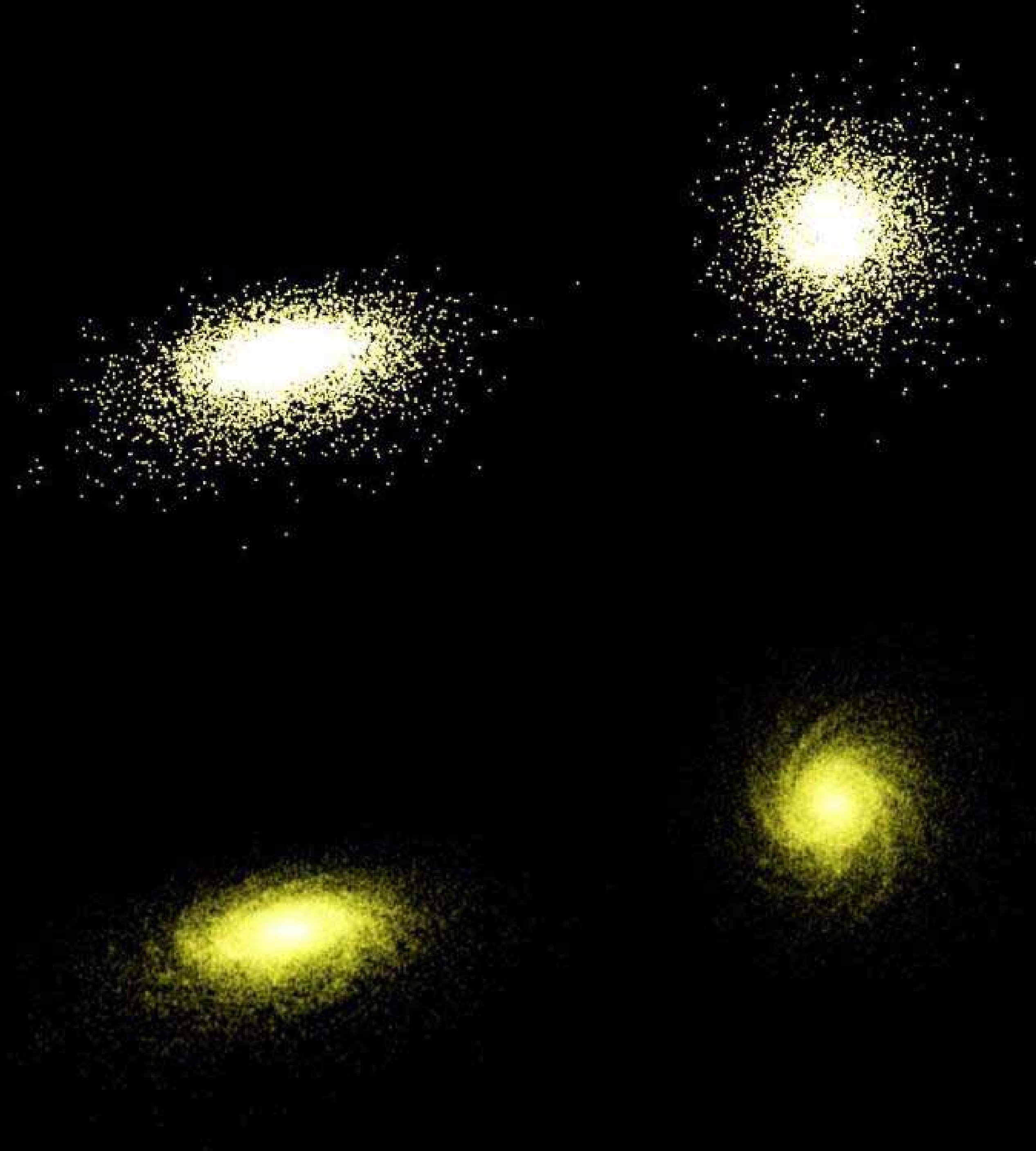
gas



gas



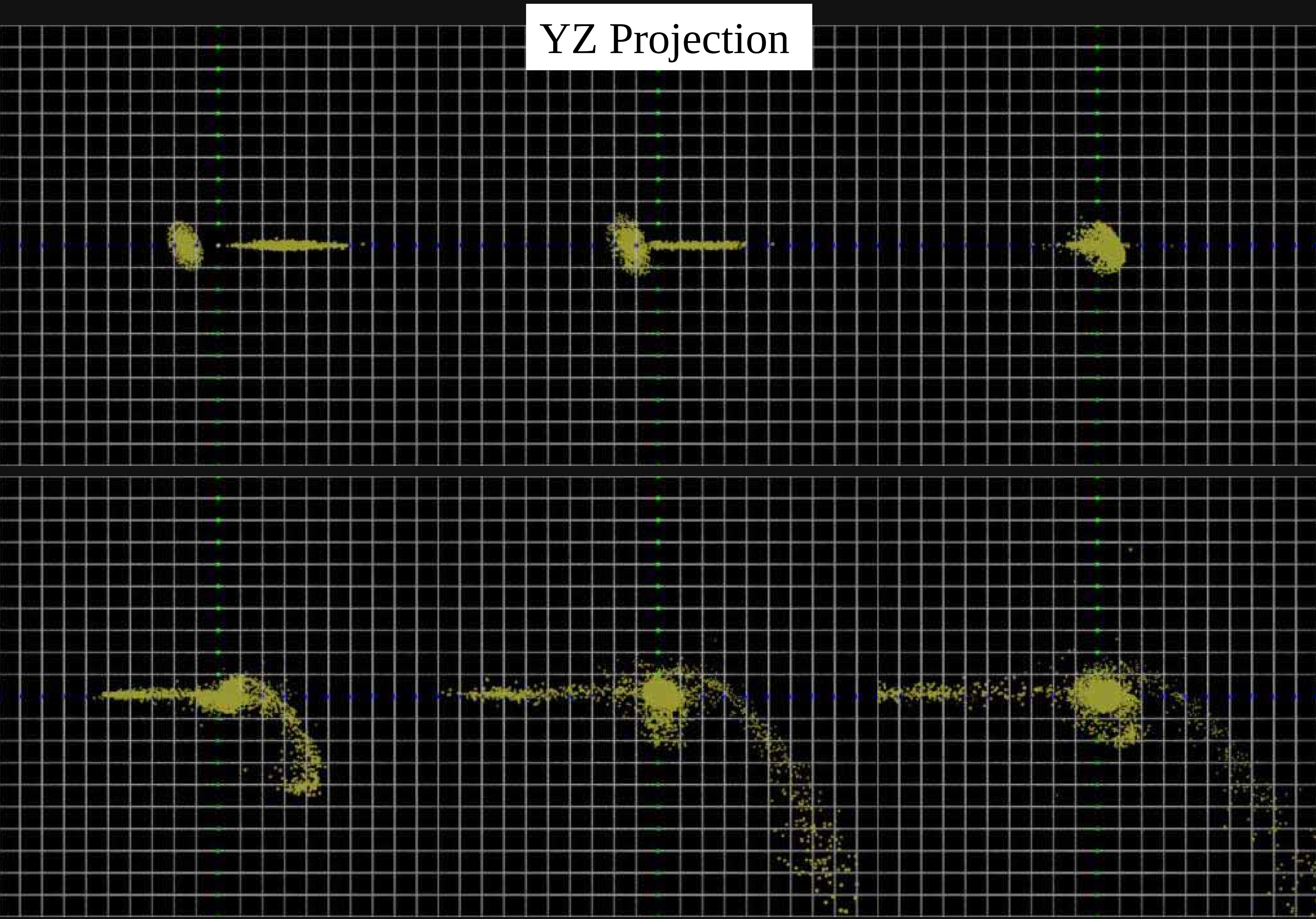
stars



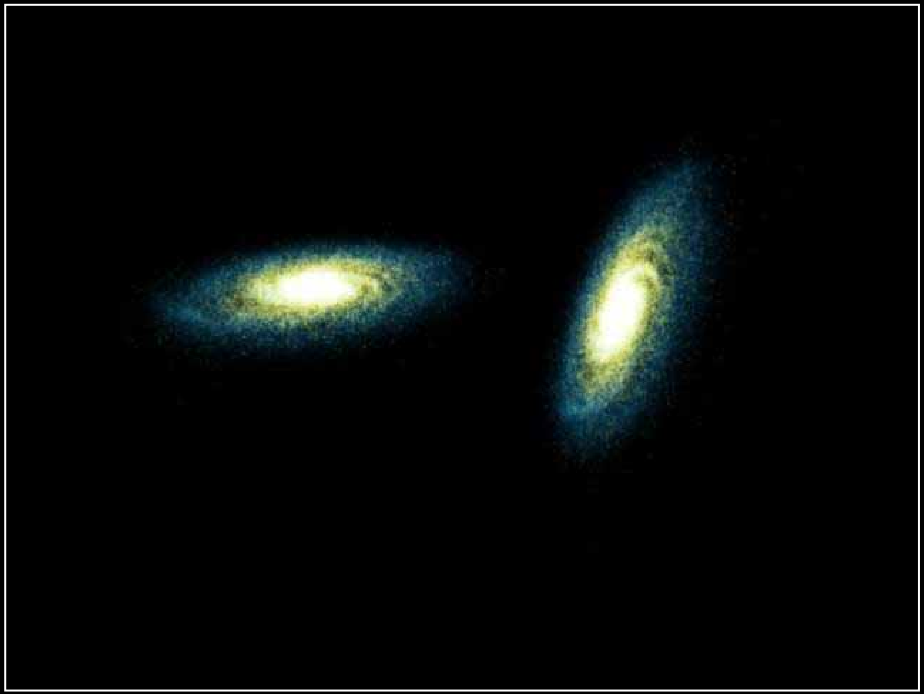
Galaxy Collision Movie

Choreography

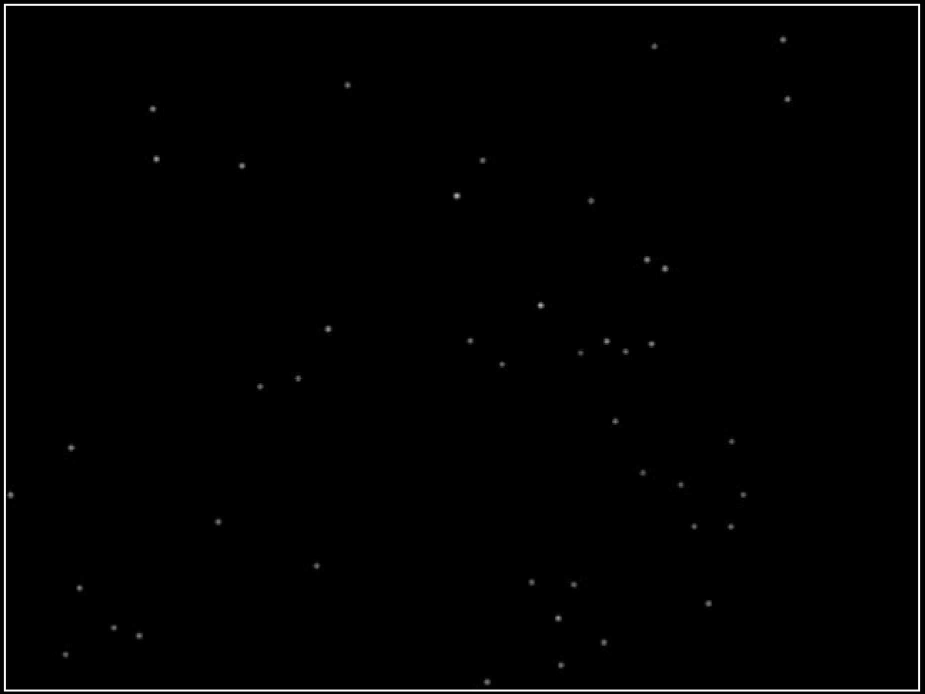
- Camera motion
- Invaluable for giving 3D feel
- Missing from most science animations


YZ Projection

Compositing



Galaxies



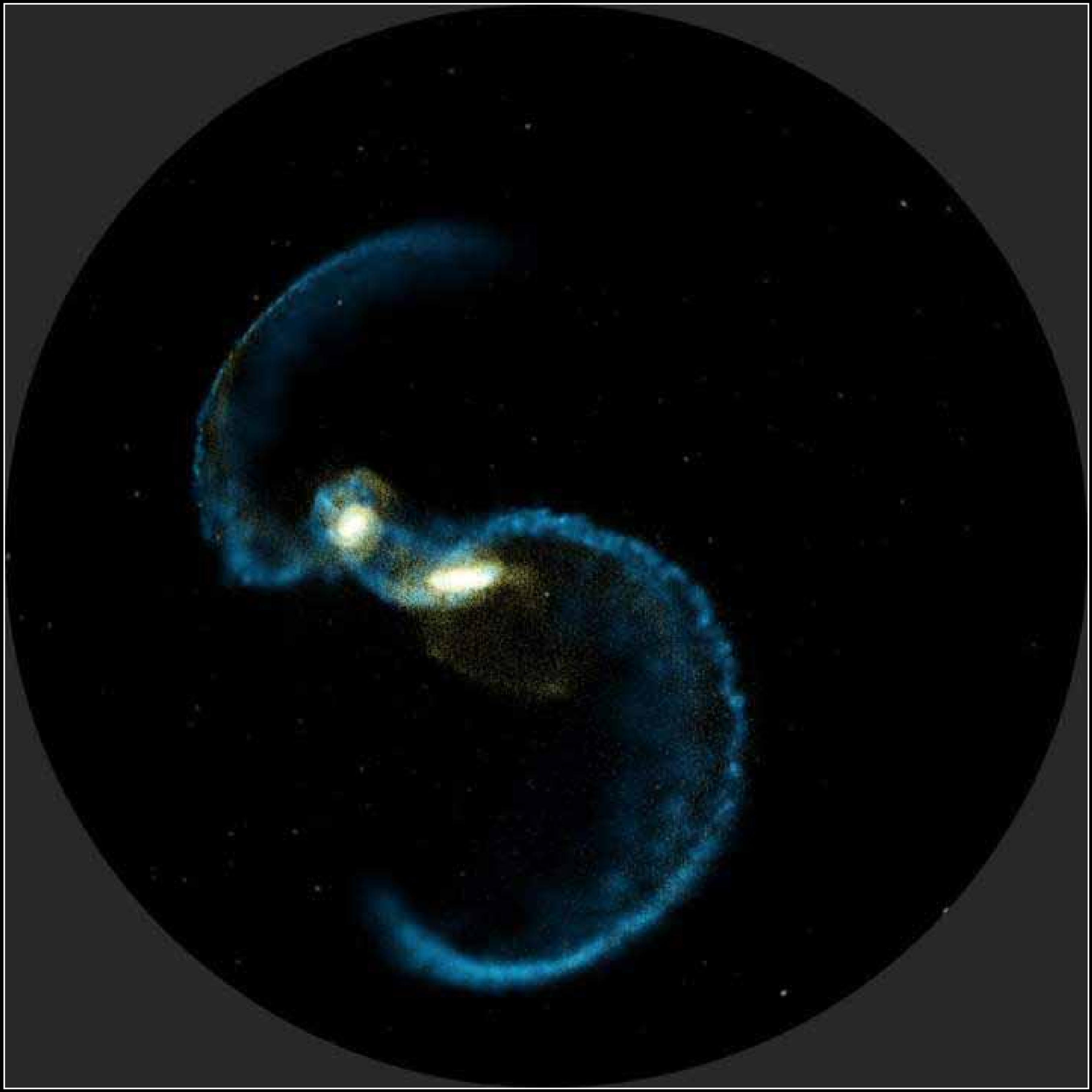
Background

Fisheye Lens Shader

- Shader can re-direct light path with a ray-tracing renderer
- Insert fisheye shader in front of scene to produce planetarium dome master

```
/*----- fisheyelens.sl
 *  Procedural shader applied to a RiPolygon which ray traces a fisheye lens from the origin.
 */
surface fisheyelens ( float lens_angle = 180.0; float scale = 10.0; )
{
  color blackcolor = color(0.0,0.0,0.0);
  varying float ss = s*scale;
  varying float tt = t*scale;
  varying float r = sqrt(ss*ss + tt*tt);

  if (r > 0.5) {
    Ci = blackcolor;
  } else {
    float polar_angle = radians(lens_angle)*r;
    float z = cos(polar_angle);
    float x = sin(polar_angle)*ss/r;
    float y = sin(polar_angle)*tt/r;
    varying vector tracedir = vector "camera" (x, y, z);
    varying point startpoint = point "camera" (0, 0, 0);
    Ci = trace(startpoint, tracedir);
  }
}
```

trace function requires raytracer



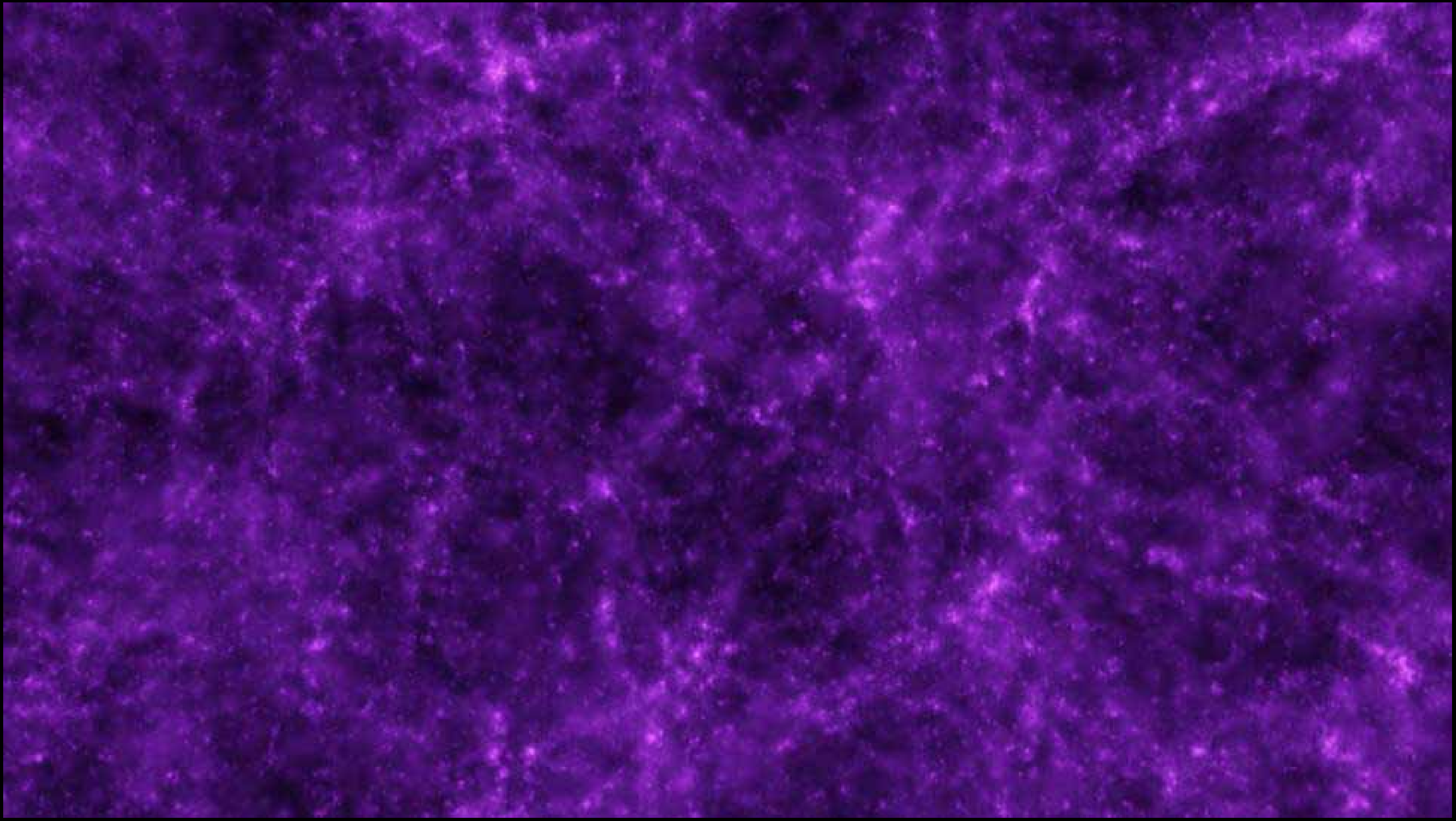
Custom Software

Cosmology

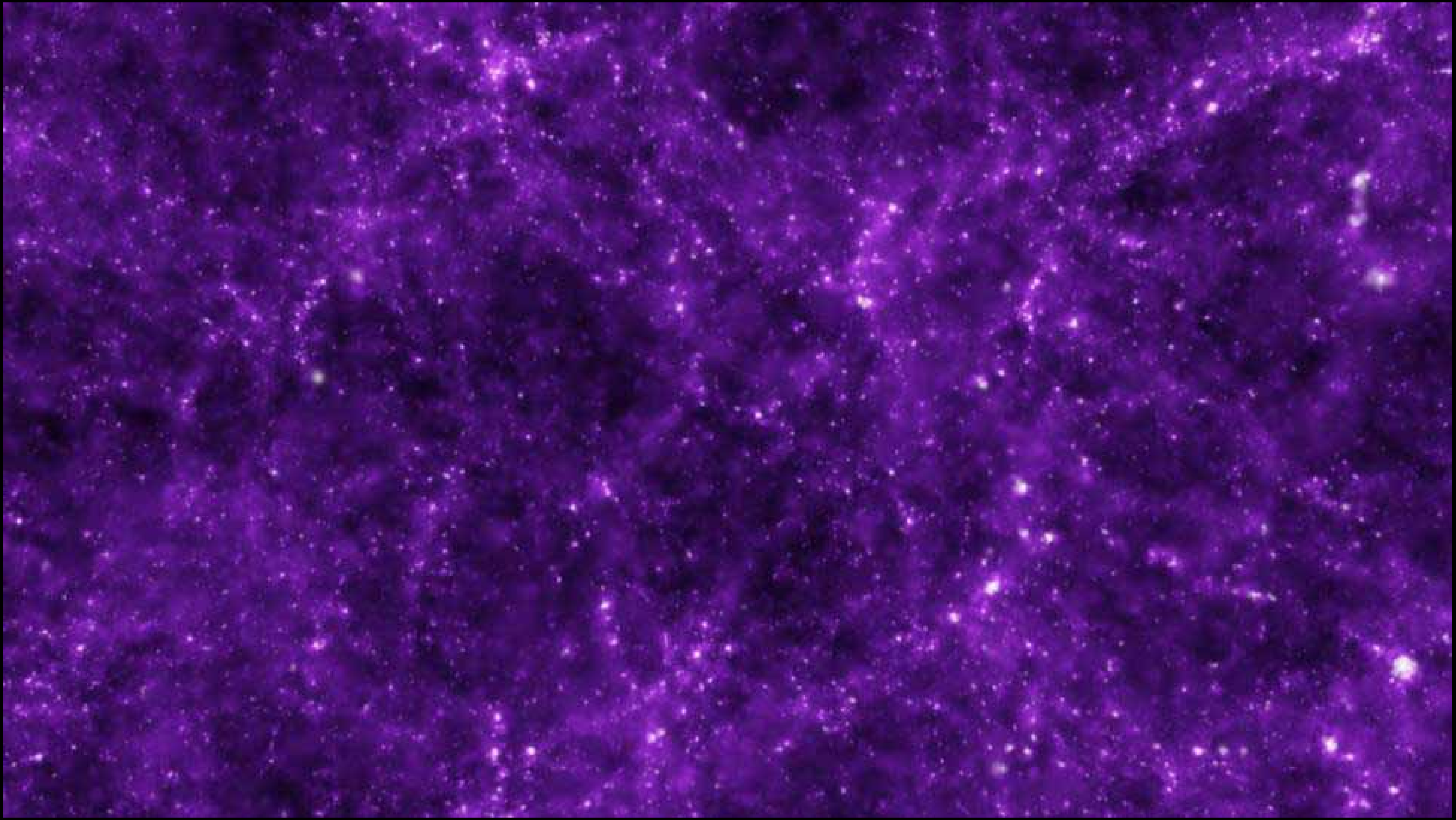
- Large scale structure of the universe
- SPH - high density gas shows galaxies
- N-body - dark matter shows mass

galaxies

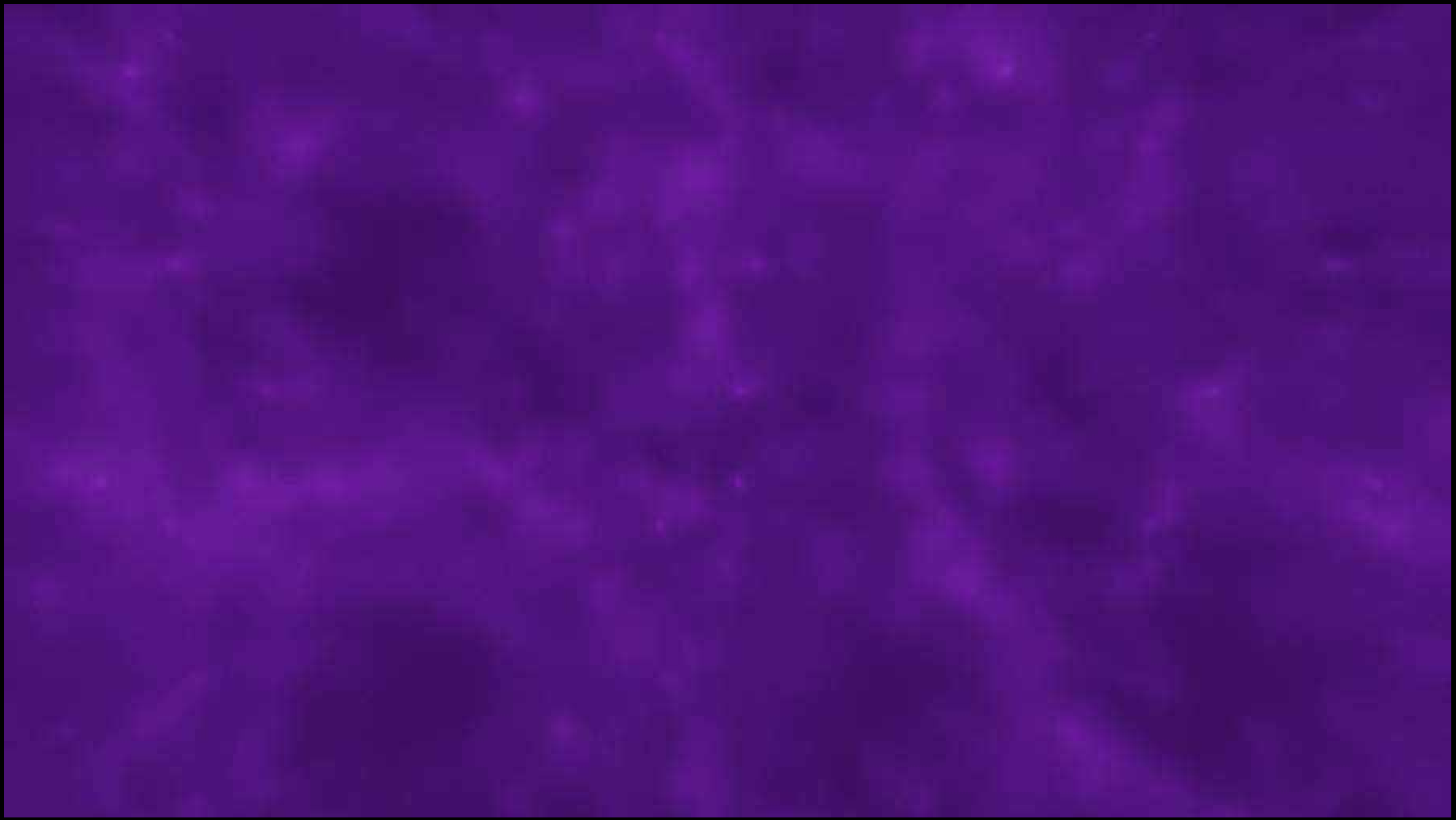
dark matter



galaxies & dark matter



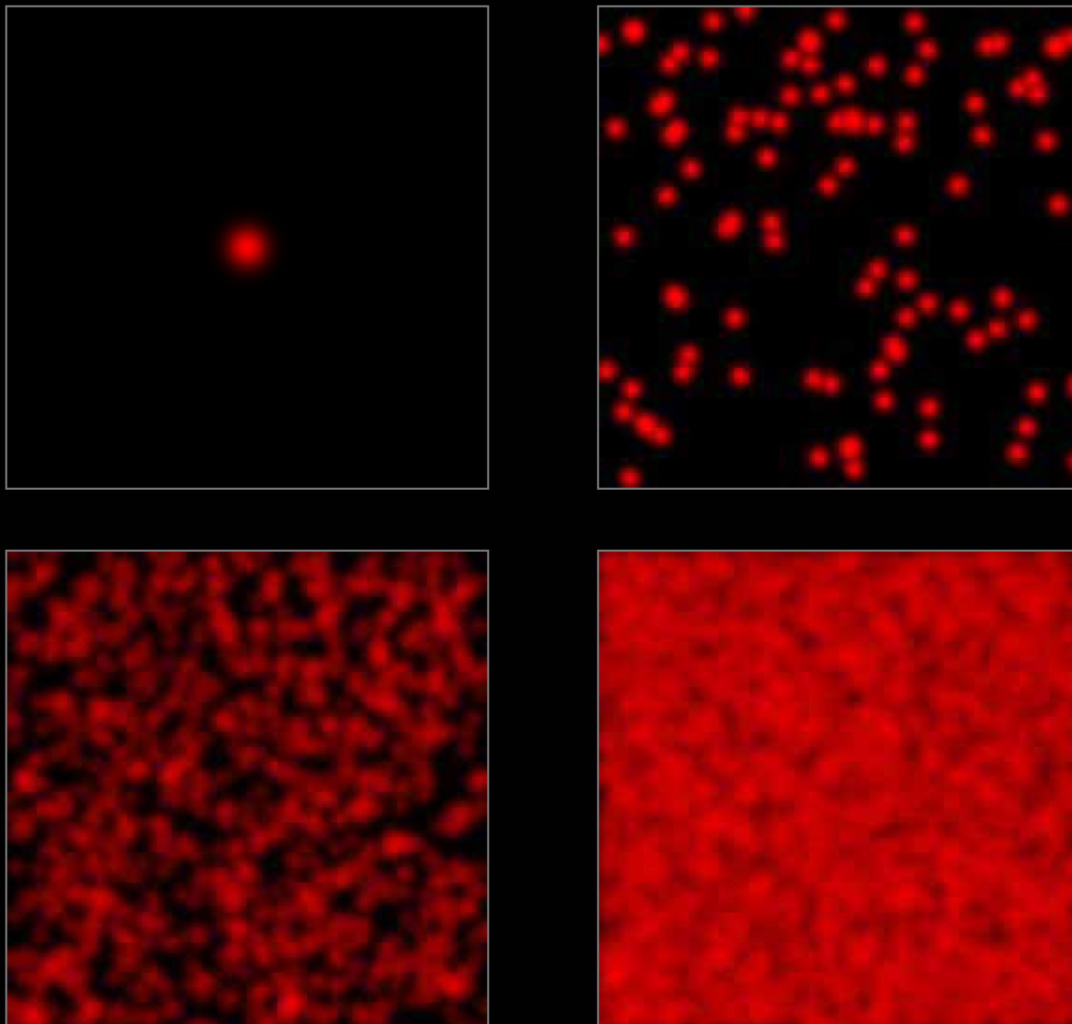
Galaxy Formation



Test render - 650,000 particles
1280 x 720 pixels – 29 hours

New Solution?

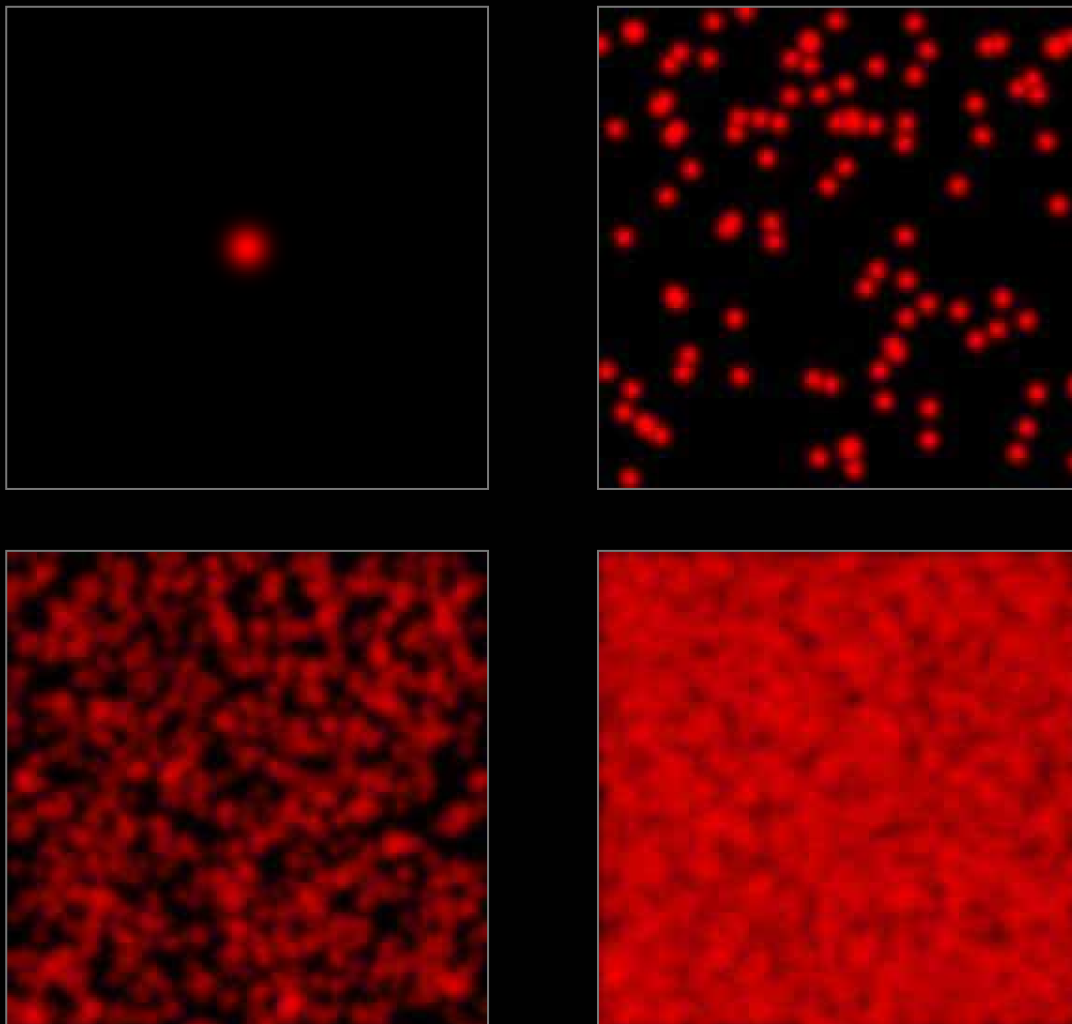
- Current: BMRT (ray tracer)
- Scanline rendering much worse
- Other geometric primitives
- Texture mapping vs shading
- Hardware renderer
- Custom software



- Test cases
  - 2 GHz CPU
  - 1k – 28 sec
  - 10k - 303 sec
  - 100k – 5784 sec

Hardware Rendering

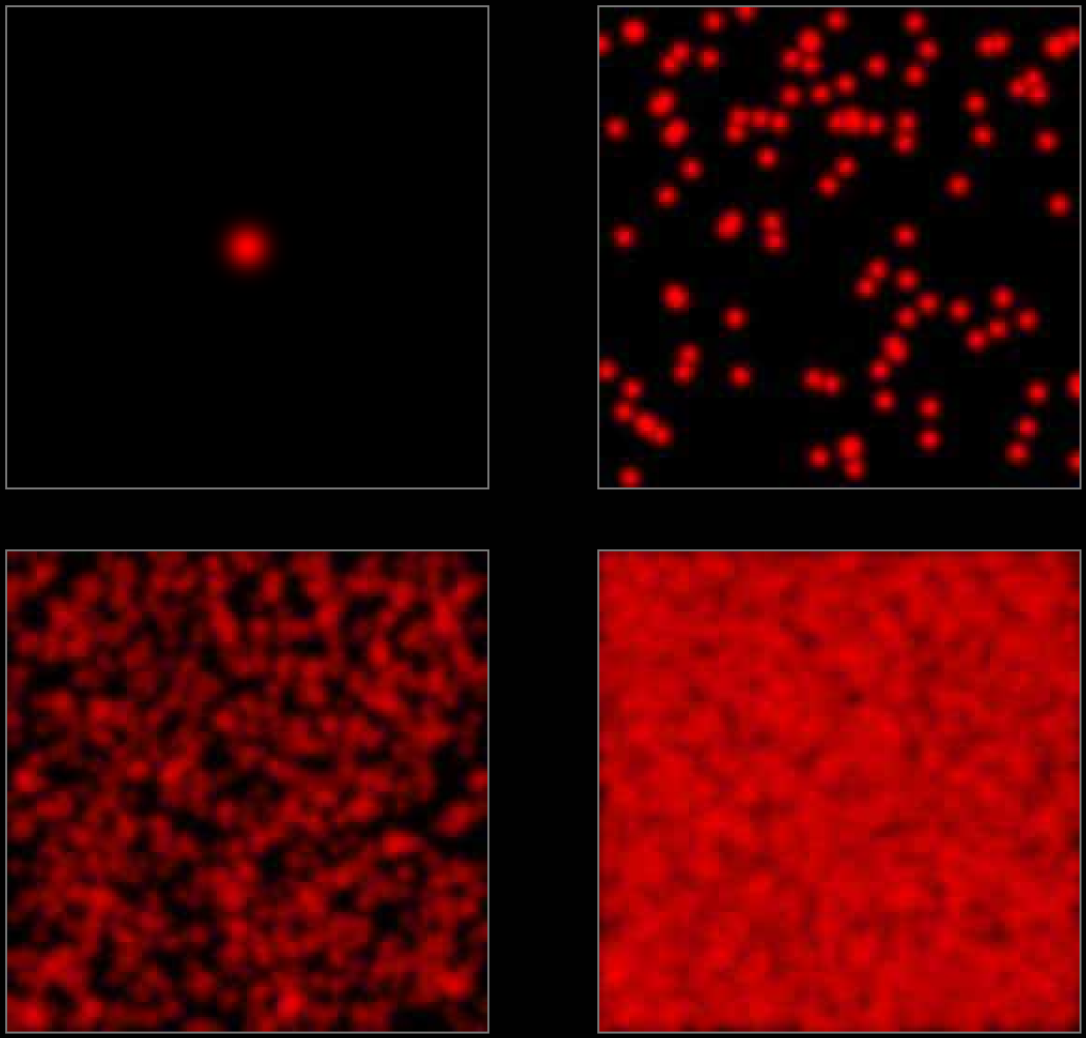
- Nvidia's Gelato
  - Larry Gritz
  - GPU in addition to CPU
  - Software to sell hardware
  - Similar to Renderman
  - Python API
- Tests on Quadro FX 1100 card



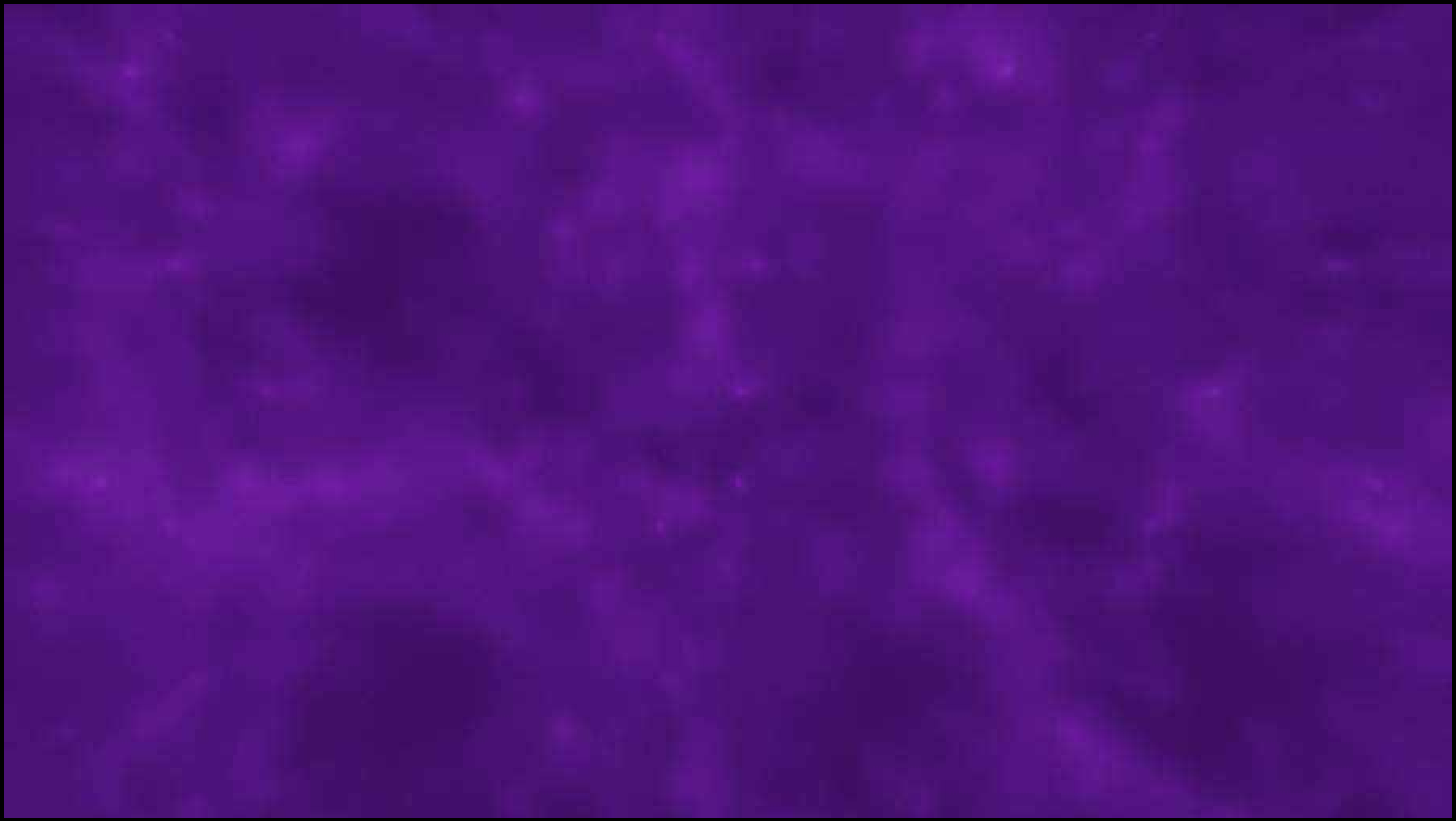
- BMRT
  - 1k – 28 sec
  - 10k - 303 sec
  - 100k – 5784 sec
- Gelato
  - 1k – 28 sec
  - 10k - 320 sec
  - 100k – 5810 sec

Custom C Code

- C code already used to read data, write RIB
- Splatting similar to kernel smoothing in SPH
- Painter's algorithm
  - Sort on Z depth
  - Paint splat on image with transparency
  - $N_{particles} \times (N_{pixels})^2$
- libtiff for writing images
- Independence vs feature set



- Quick and dirty test
- 10,000 splats
  - BMRT 303 sec
  - Gelato 320 sec
  - C code 9 sec

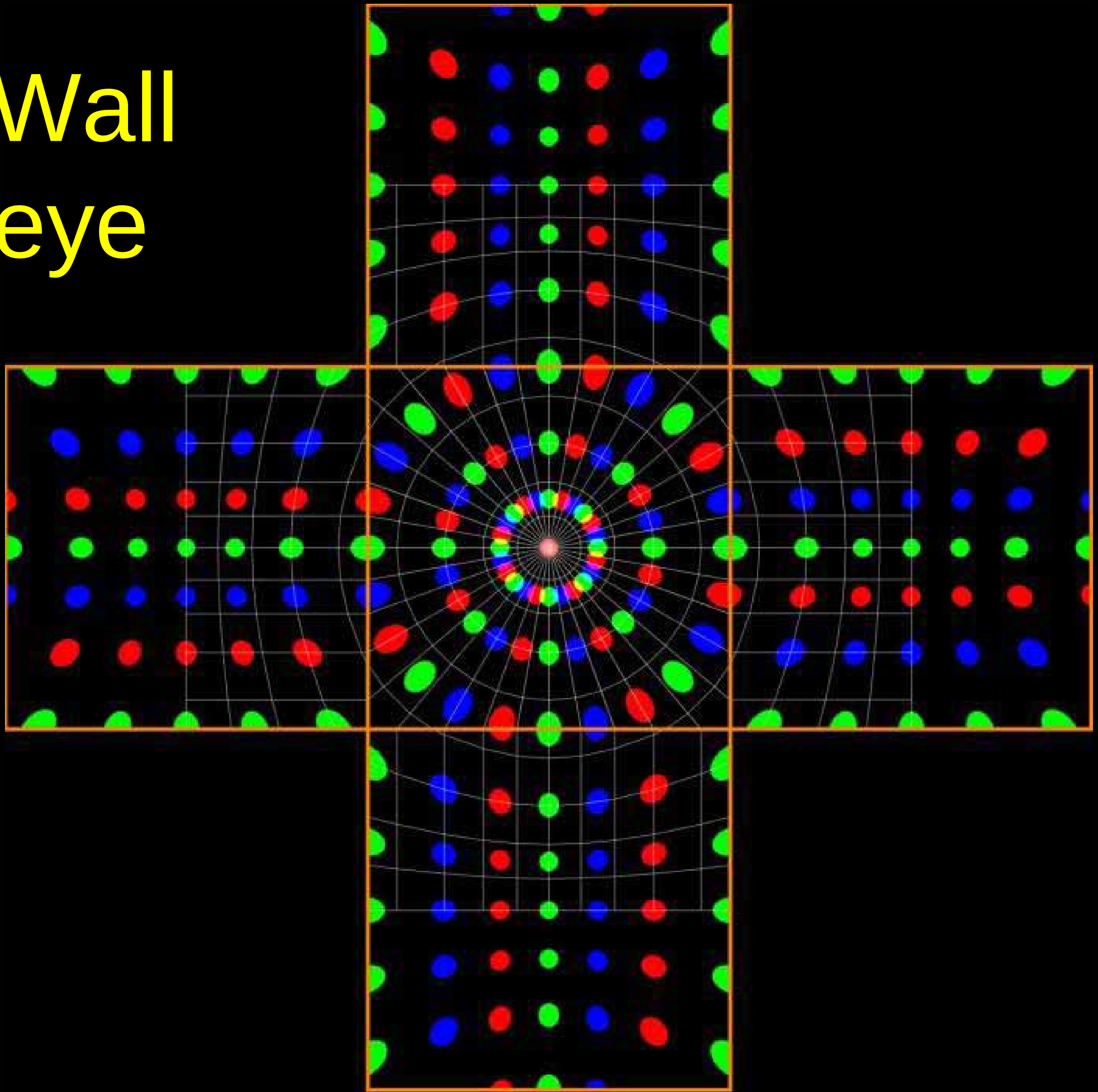


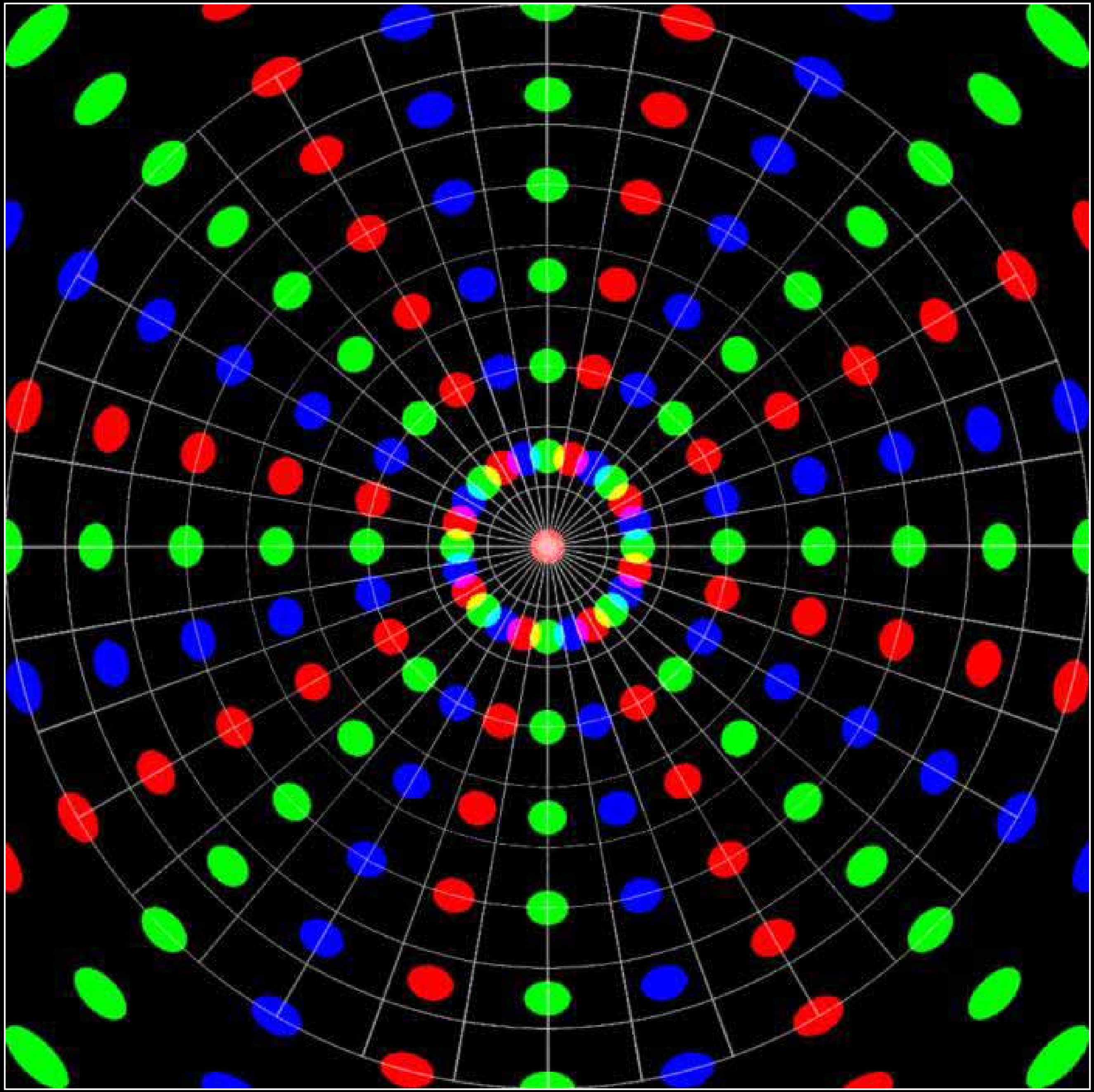
BMRT Test render - 650,000 particles
1280 x 720 pixels – 29 hours

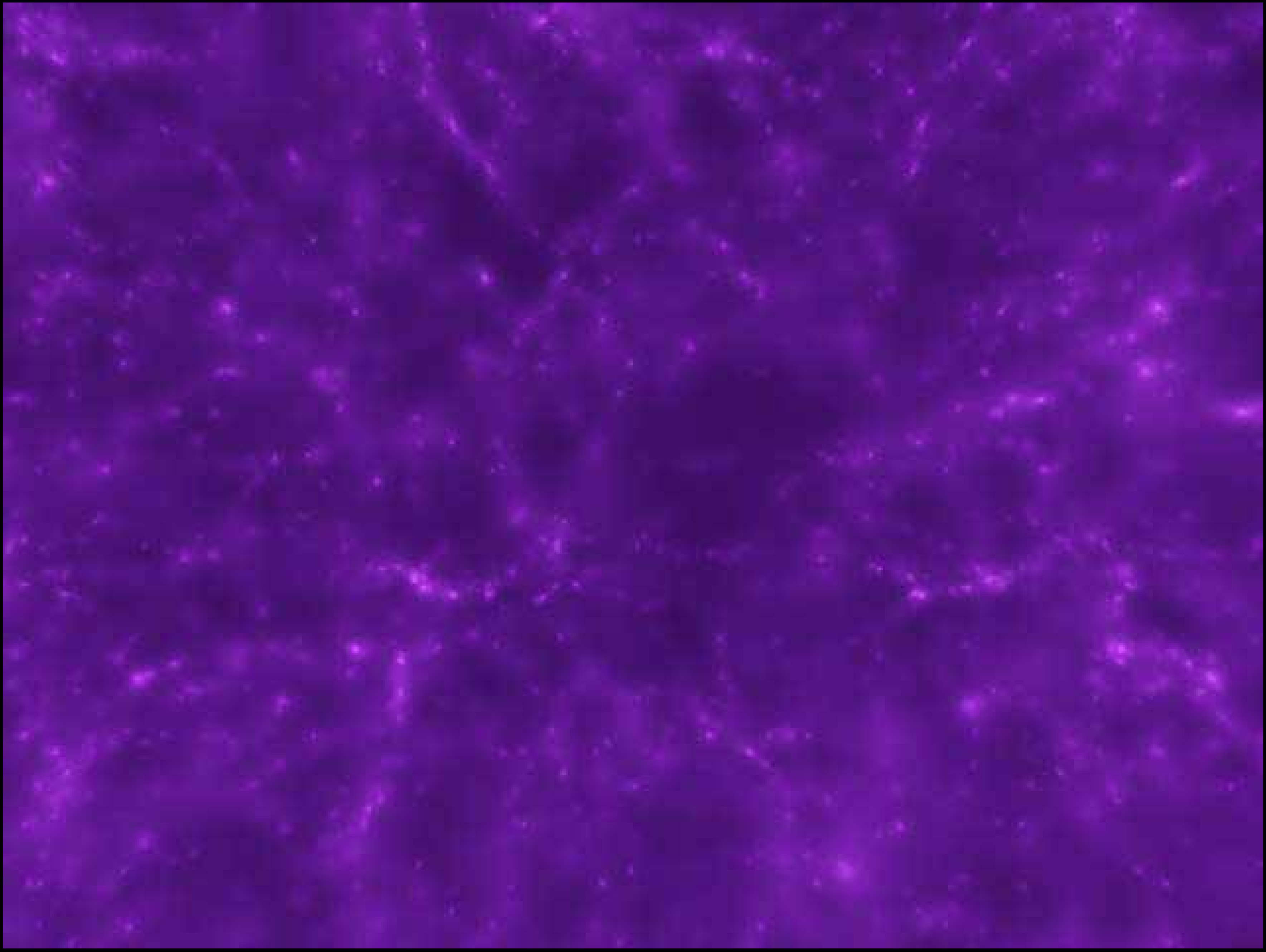
splat_particles Test render - 650,000 particles
1280 x 720 pixels – 27 minutes

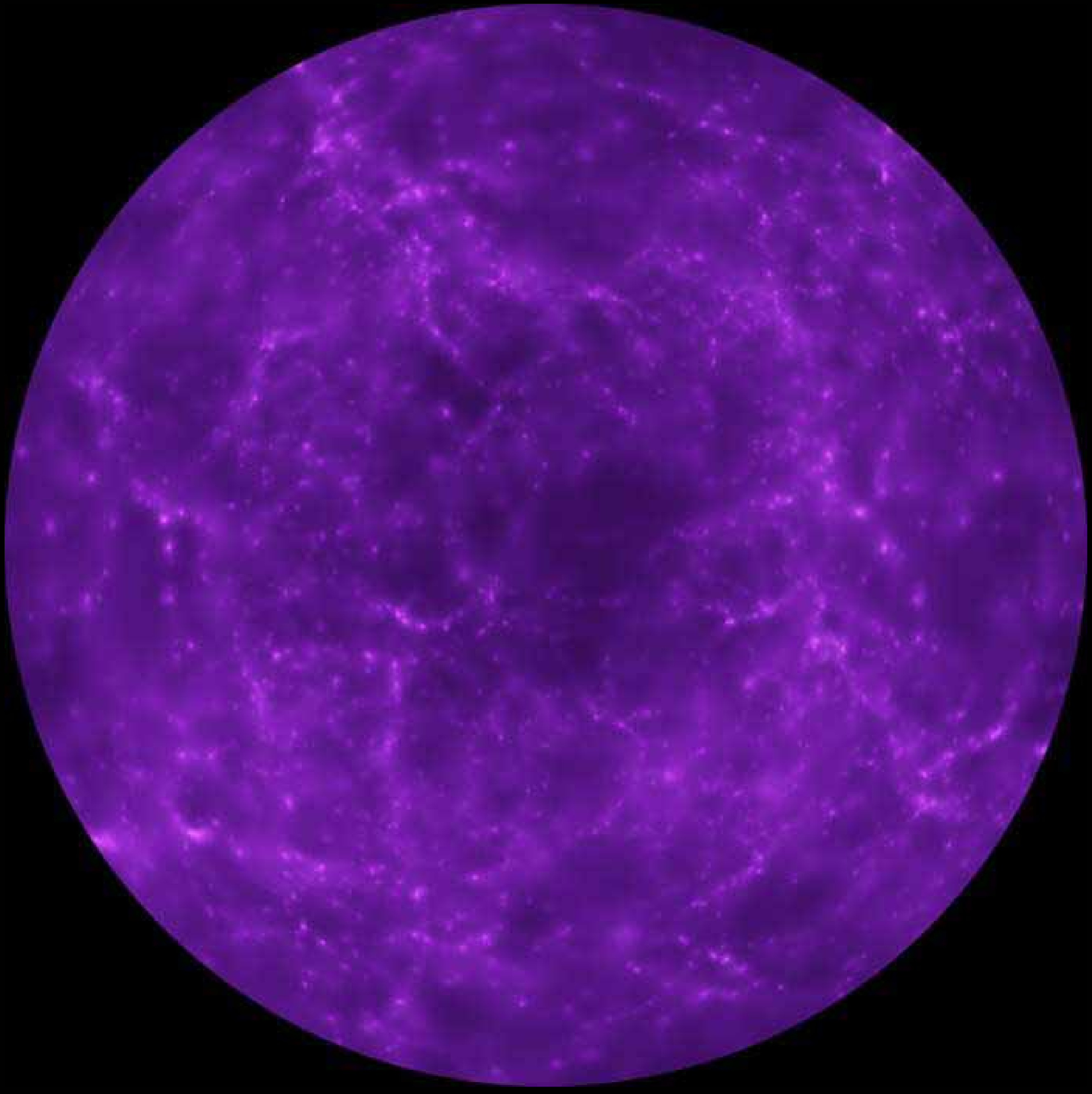
Galaxy Formation Movie

Five Wall Fisheye









Conclusions

- Accuracy - data and simulations
  – Visualization methods must preserve accuracy
- Aesthetics - graphics software
  – COTS, FOSS, Custom
  – Utilize programming interfaces
- Sci Viz benefits
  – Better data representation
  – Wider audience appeal
- Resources
  – Visuals on FJS web site, code not yet
  – astro-viz email list at groups.yahoo.com