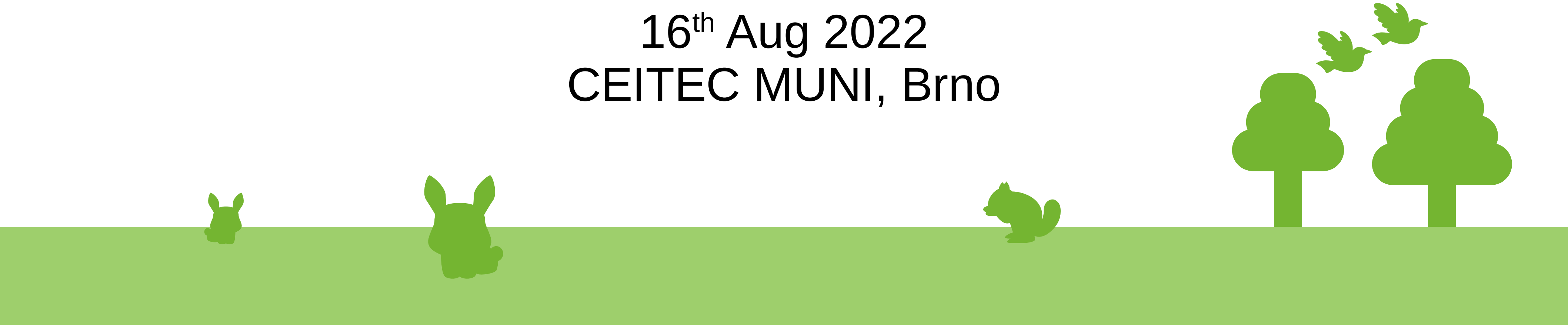
3D Data Visualization

Vladimír (Vlado) Ulman

EMBO Course on Lightsheet Microscopy

16th Aug 2022
CEITEC MUNI, Brno



About me

- Applied Computer Scientists & Open-source SW believer
- Image processing & analysis & vizu, big images in parallel
- Algorithms benchmarking (synth. data)
- Support for DL methods training (silver ground-truth)

- Central European Institute of Technology
  (CEITEC, Masaryk University, Brno)

- Centre for Biomedical Image Processing
  (CBIA, Faculty of Informatics, MU, Brno)
- CellTrackingChallenge.net

- IT4Innovations National Supercomuting Center
  (IT4I, VSB – Technical University of Ostrava, Ostrava)
- HPC Workflow Manager (for Fiji) + DataStore

About the talk

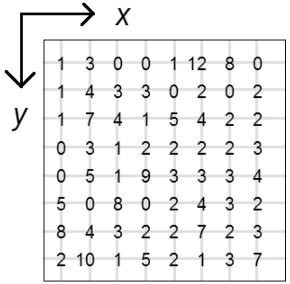
- Main theme:
  - 3D data vizu is actually mostly about
    - Some form of 3D-to-2D projection
    - Some form of information reduction
  - Principles used under the hood

- Outline:
  - Slice rendering
  - Volume rendering
  - Cartographic projections
  - Graphical representations

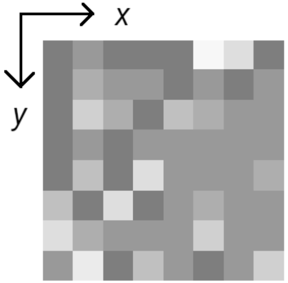
- Download: **https://www.fi.muni.cz/~xulman/files/EMBO_LS2022.pdf**

Terminology

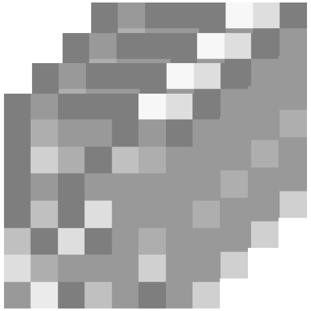
- 2D image
  - Regularly displaced (scalar) values on an orthogonal 2D grid
  - Aka. frame, slice or section

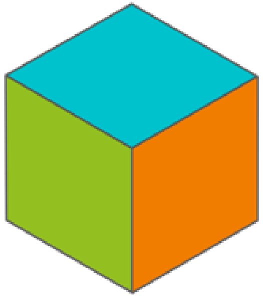


1	3	0	0	1	12	8	0
1	4	3	3	0	2	0	2
1	7	4	1	5	4	2	2
0	3	1	2	2	2	2	3
0	5	1	9	3	3	3	4
5	0	8	0	2	4	3	2
8	4	3	2	2	7	2	3
2	10	1	5	2	1	3	7



- 3D image
  - A sequence of 2D images → 3D grid
  - Aka. stack or volume
  - No triangles & textures (computer games)
  - Raw data

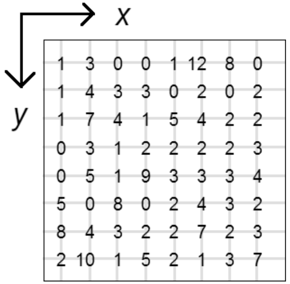




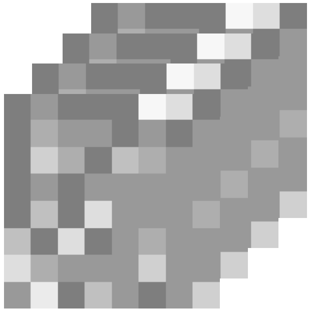
- Tabular data that include x,y,z coordinates
  - Processed data, e.g. Point cloud

Terminology

- Dimensionality increases
  - By 1D when time-lapse
  - By 1D with every imaged channel
  - By 1D with every view angle



1	3	0	0	1	12	8	0
1	4	3	3	0	2	0	2
1	7	4	1	5	4	2	2
0	3	1	2	2	2	2	3
0	5	1	9	3	3	3	4
5	0	8	0	2	4	3	2
8	4	3	2	2	7	2	3
2	10	1	5	2	1	3	7



- Picture element → **Pixel**, Volumetric pixel → **Voxel**

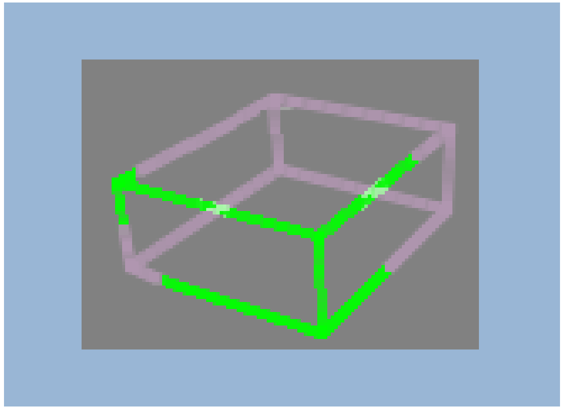
- Pixel consumes memory:
  - 8 *or* **16 bits** (1 *or* 2 Bytes, integers, 0-255 *or* 0-65535)
  - 32 *or* 64 bits (4 *or* 8 Bytes, floating-point, single *or* double precision)

- Pixel/Voxel represents a physical area/volume
  - “Microns per pixel” along each axis → Resolution
  - Same sizes in all axes = **isotropic resolution**

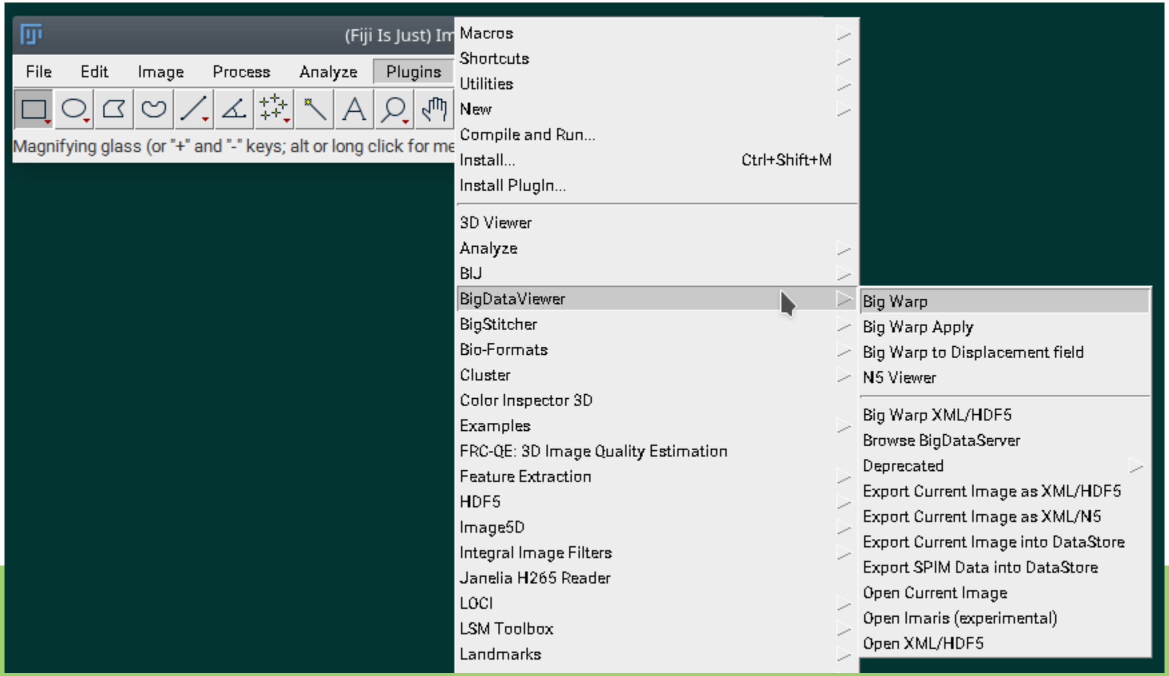
Slice Rendering

Idea: Show the voxels at the intersection of an user-given plane with the Volume.

Orthogonal views are a special case of this.



Example SW: BigDataViewer → **BDV**

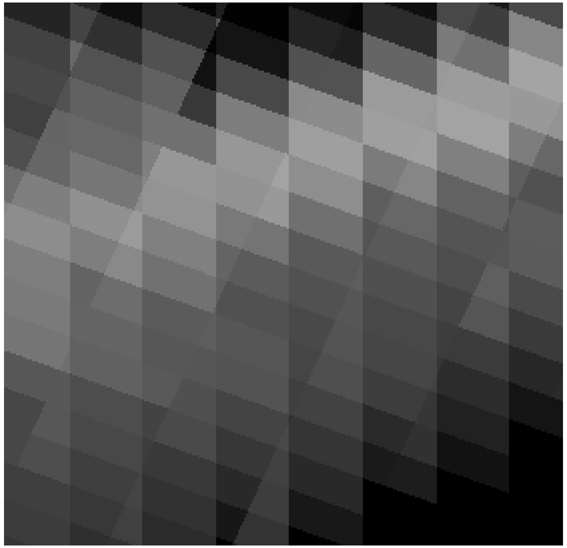

(Fiji Is Just) Im

Magnifying glass (or "+" and "-" keys; alt or long click for me

...available in Fiji

Slice Rendering

Idea: Show the voxels at the intersection of an user-given plane with the Volume.



BigDataViewer (BDV):

- Considers Voxels only (even for 2D images)
- Considers **16 bits** only!
- Fast to determine voxels that are hit by the plane
- Fast to compute (and display) their intersection polygons
- **Fast to fetch their values**
  …when using an intelligent data storage: chunks + pyramids

- Since all is fast, BDV can afford displaying multiple images at once

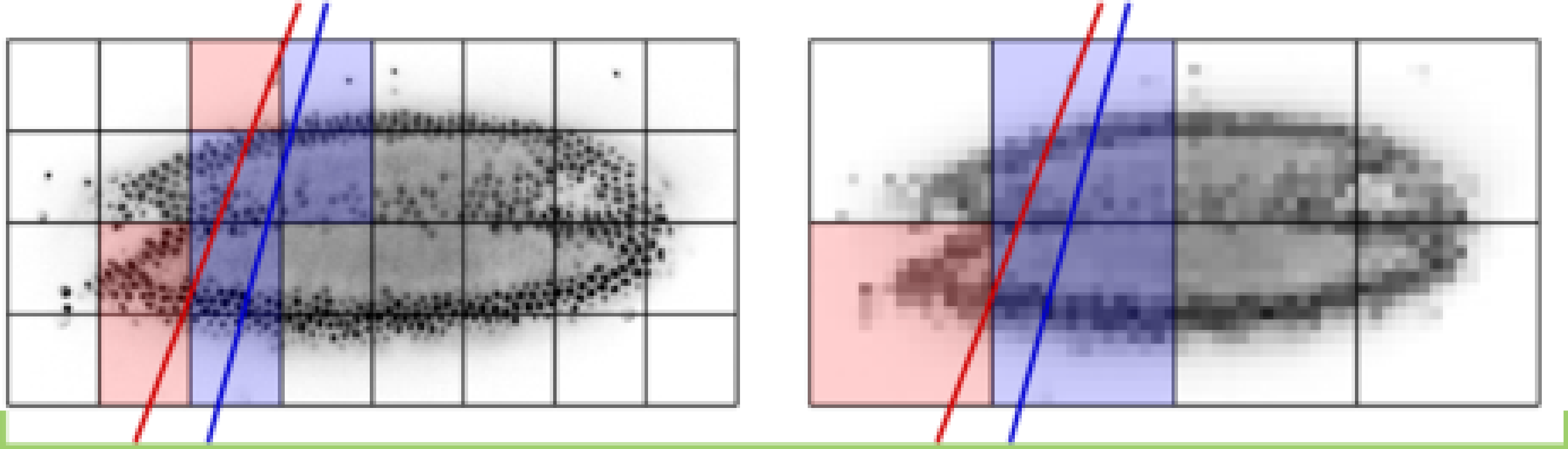
Slice Rendering

BigDataViewer:

- ***Fast to fetch their values***

  …when using an intelligent data storage: chunks + pyramids

- Chunks – to read a voxel, only an including small chunk of Bytes needs to be loaded
- Pyramids – small copies of the image at increasingly lower resolution are available



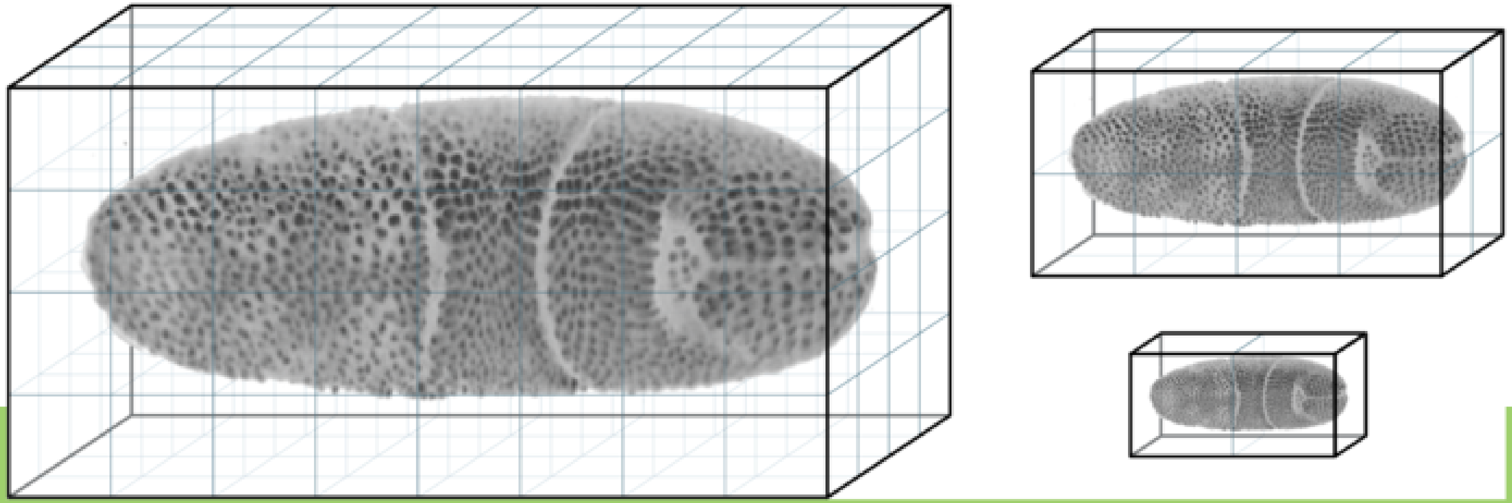
Slice Rendering

BigDataViewer:

- ***Fast to fetch their values***

  …when using an intelligent data storage: chunks + pyramids

- Chunks – to read a voxel, only an including small chunk of Bytes needs to be loaded
- Pyramids – small copies of the image at increasingly lower resolution are available



Slice Rendering

BigDataViewer:

- ***Fast to fetch their values***

  …when using an intelligent data storage: chunks + pyramids

- Chunks – to read a voxel, only an including small chunk of Bytes needs to be loaded
- Pyramids – small copies of the image at increasingly lower resolution are available

- Google Maps store (and show) maps in the same way
- **Requires appropriate image file format**
  - Baseline: BDV.HDF5
  - Rising star: OME.Zarr (Dialect.GenericContainer)

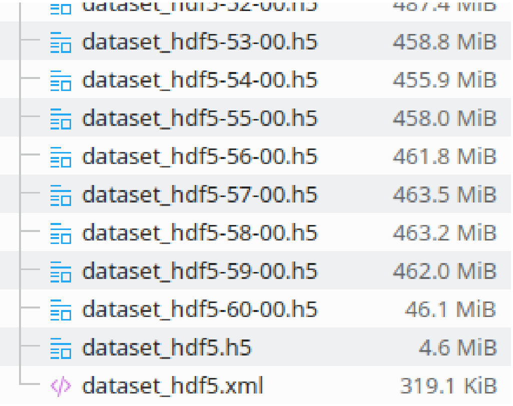
Slice Rendering

BigDataViewer:

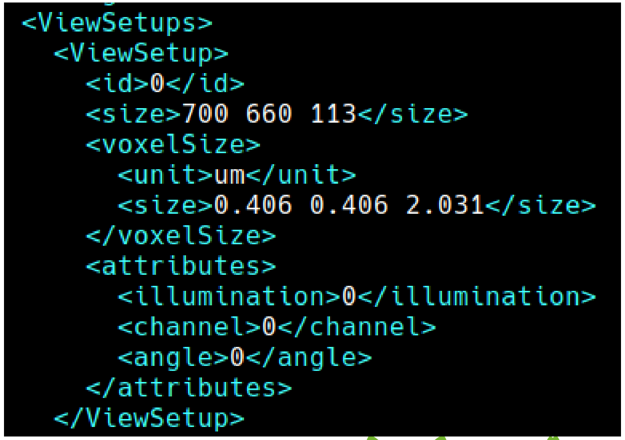
- *Requires appropriate image file format*
  - *Baseline: BDV.HDF5*

- Traditionally: **dataset.xml** plus some container(s)
- **.xml** holds conveniently metadata about the image
  - Small, human-readable, editable
  - Also includes a pointer on the container

- BDV “flattens” dimensionality
  to 4D: x, y, z, **source = ViewSetup**



dataset_hdf5-52-00.h5	457.4 MiB
dataset_hdf5-53-00.h5	458.8 MiB
dataset_hdf5-54-00.h5	455.9 MiB
dataset_hdf5-55-00.h5	458.0 MiB
dataset_hdf5-56-00.h5	461.8 MiB
dataset_hdf5-57-00.h5	463.5 MiB
dataset_hdf5-58-00.h5	463.2 MiB
dataset_hdf5-59-00.h5	462.0 MiB
dataset_hdf5-60-00.h5	46.1 MiB
dataset_hdf5.h5	4.6 MiB
dataset_hdf5.xml	319.1 KiB



```
<ViewSetups>
  <ViewSetup>
    <id>0</id>
    <size>700 660 113</size>
    <voxelSize>
      <unit>um</unit>
      <size>0.406 0.406 2.031</size>
    </voxelSize>
    <attributes>
      <illumination>0</illumination>
      <channel>0</channel>
      <angle>0</angle>
    </attributes>
  </ViewSetup>
```

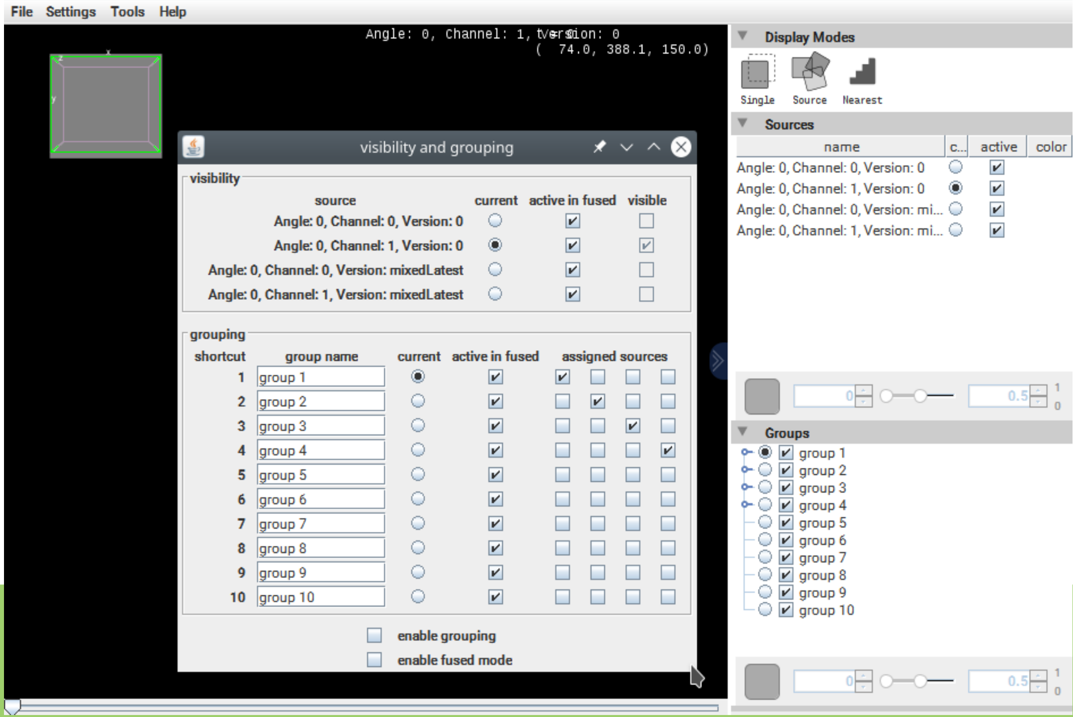
Slice Rendering

BigDataViewer:

- *Since all is fast, BDV can afford displaying multiple images at once*

- Recall: BDV considers Voxels only (even for 2D images)
- Recall: Dimensionality increases
  - By 1D when time-lapse
  - By 1D with every imaged channel
  - By 1D with every view angle

- BDV “flattens” dimensionality
  to 4D: x, y, z, **source = ViewSetup**



```
Angle: 0, Channel: 1, version: 0
(  74.0, 388.1, 150.0)
```

visibility and grouping

visibility

source	current	active in fused	visible
Angle: 0, Channel: 0, Version: 0	○	☑	□
Angle: 0, Channel: 1, Version: 0	●	☑	☑
Angle: 0, Channel: 0, Version: mixedLatest	○	☑	□
Angle: 0, Channel: 1, Version: mixedLatest	○	☑	□

grouping

shortcut	group name	current	active in fused	assigned sources			
1	group 1	●	☑	☑	□	□	□
2	group 2	○	☑	□	☑	□	□
3	group 3	○	☑	□	□	☑	□
4	group 4	○	☑	□	□	□	☑
5	group 5	○	☑	□	□	□	□
6	group 6	○	☑	□	□	□	□
7	group 7	○	☑	□	□	□	□
8	group 8	○	☑	□	□	□	□
9	group 9	○	☑	□	□	□	□
10	group 10	○	☑	□	□	□	□

□ enable grouping

□ enable fused mode

Sources

name	c...	active	color
Angle: 0, Channel: 0, Version: 0	○	☑	
Angle: 0, Channel: 1, Version: 0	●	☑	
Angle: 0, Channel: 0, Version: mi...	○	☑	
Angle: 0, Channel: 1, Version: mi...	○	☑	

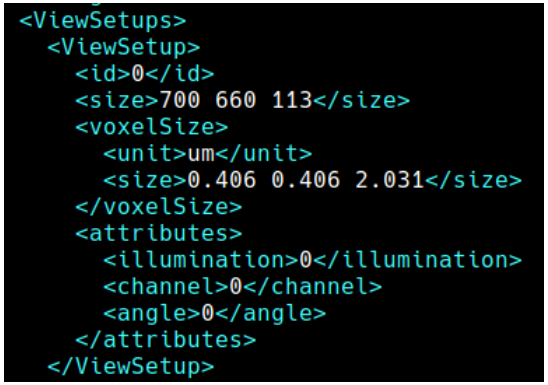
0 0.5 1 0

Groups

- ● ☑ group 1
- ○ ☑ group 2
- ○ ☑ group 3
- ○ ☑ group 4
- ○ ☑ group 5
- ○ ☑ group 6
- ○ ☑ group 7
- ○ ☑ group 8
- ○ ☑ group 9
- ○ ☑ group 10

0 0.5 1 0

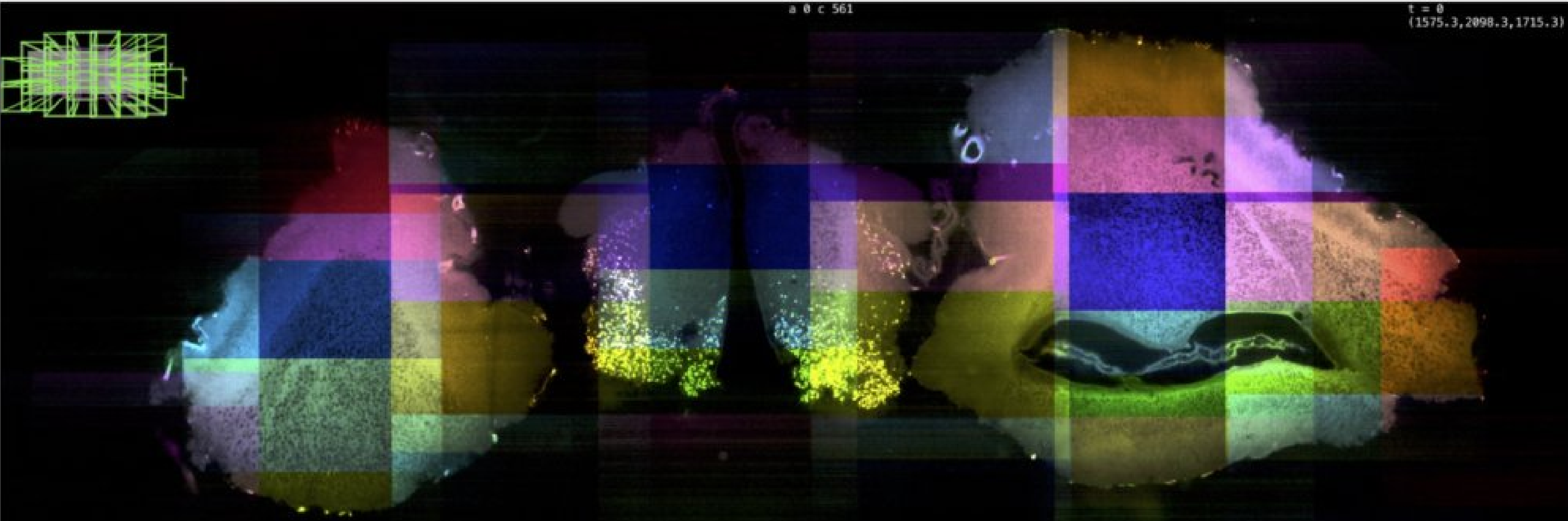
Slice Rendering



```
<ViewSetups>
  <ViewSetup>
    <id>0</id>
    <size>700 660 113</size>
    <voxelSize>
      <unit>um</unit>
      <size>0.406 0.406 2.031</size>
    </voxelSize>
    <attributes>
      <illumination>0</illumination>
      <channel>0</channel>
      <angle>0</angle>
    </attributes>
  </ViewSetup>
```

BigDataViewer inside other SWs:

- BDV “flattens” dimensionality
  to 4D: x, y, z, **source = ViewSetup**
- Example from BigStitcher

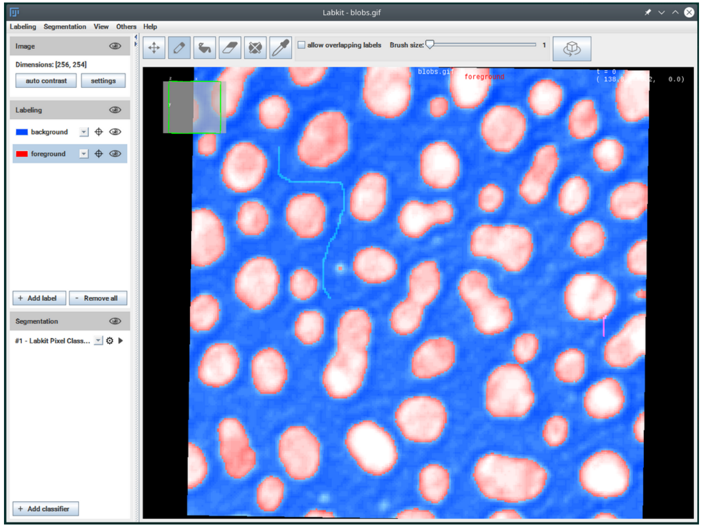


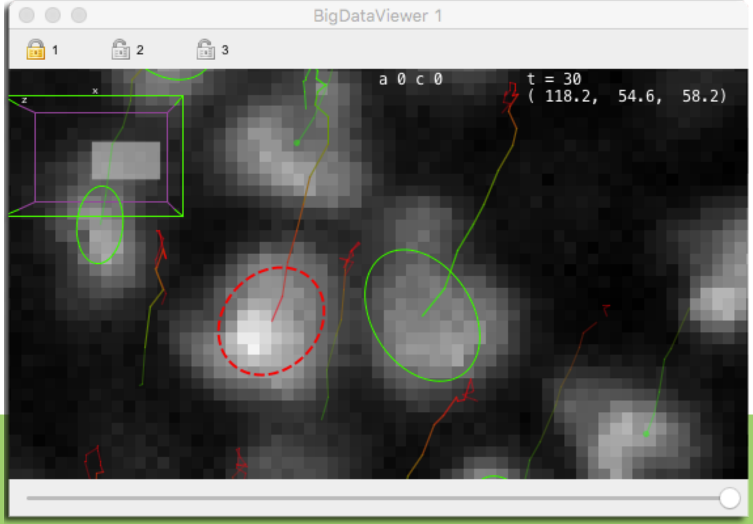
Slice Rendering

BigDataViewer as a core image viewer inside other SWs:

- LabKit, LabelEditor (Jug lab)
- MaMuT, Mastodon (Pasteur + CBG)
- knip (KNIME image processing)
- Paintera, BigWarp, BigCAT (Saalfeld lab)
- BigStitcher (Preibisch lab)
- BigDataProcessor2 (Tischi EMBL)
- MoBIE (Tischi EMBL)
- Mostly by MPI-CBG alumni or friends

- We will exercise BDV later today...

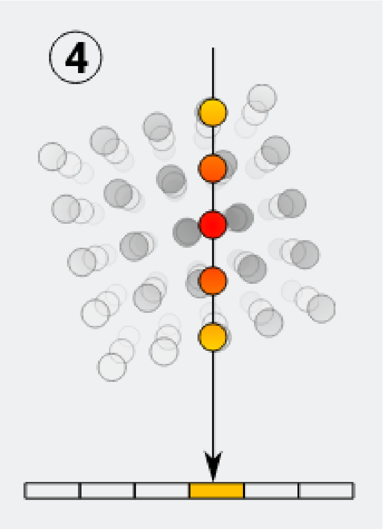




Volume Rendering

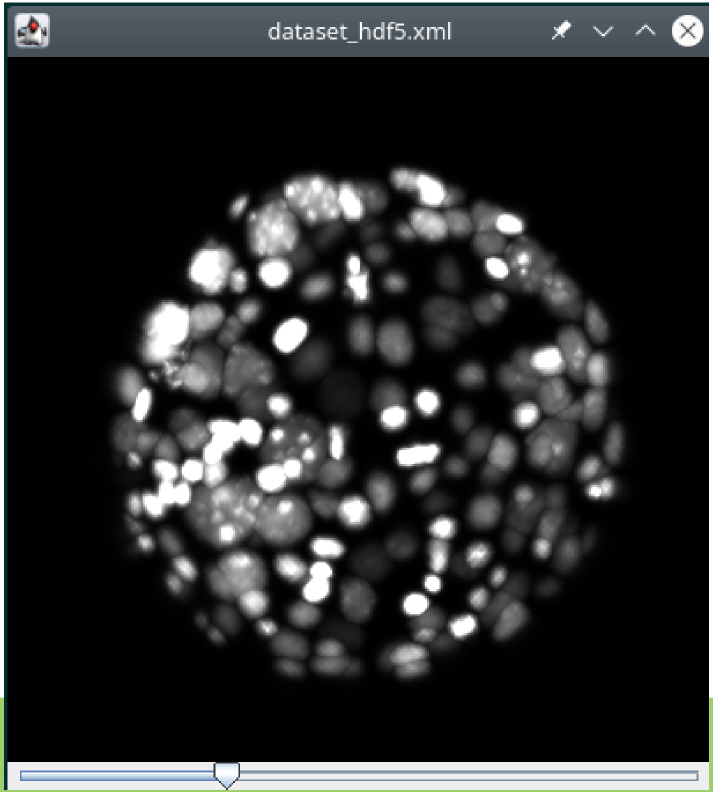
Idea: Collect voxels along lines of sight (rays), and show cummulated value.

Maximum intensity projection is a special case of this.



Example SW freely in Fiji:

- **BigVolumeViewer** (BVV) →
  - Consumes the same **dataset.xml**
  - Similar to BDV
- SciView
  - Uses BVV
  - Future Fiji viewer(?)



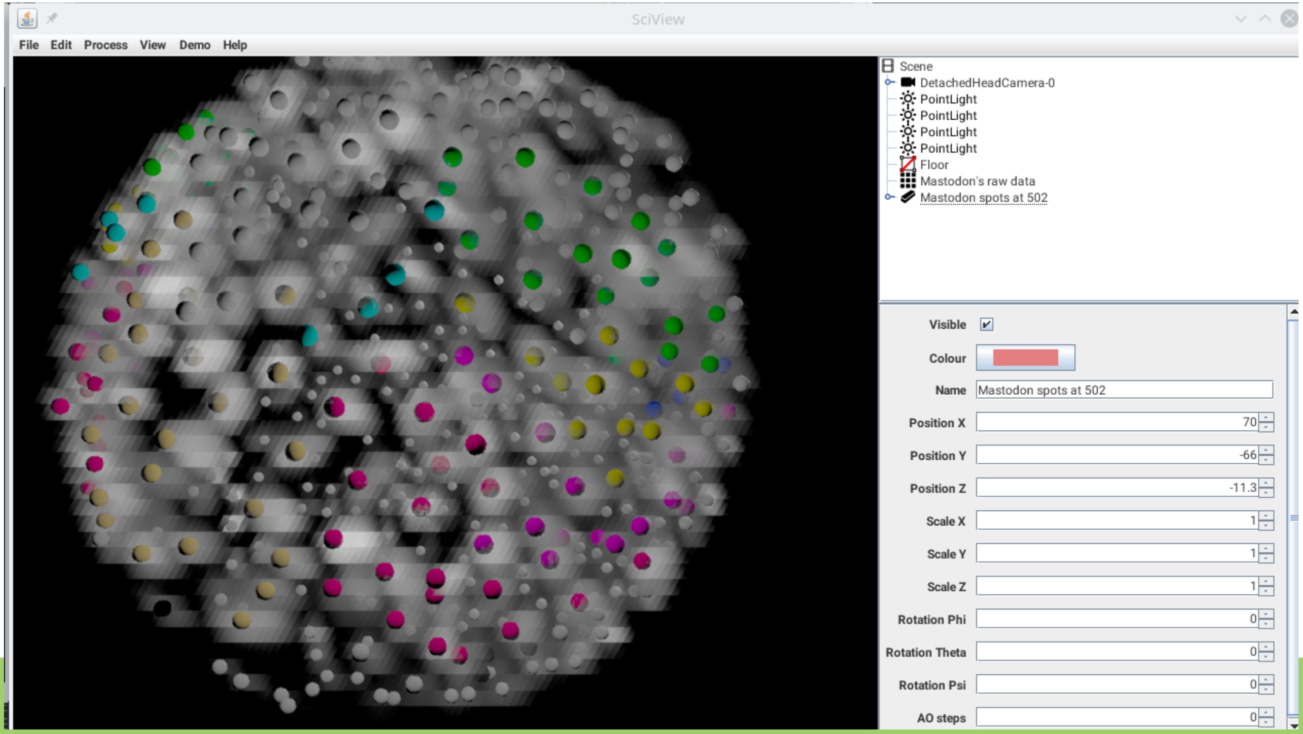
Volume Rendering

Idea: Collect voxels along lines of sight (rays), and show cummulated value.

Both available via the *SciView* update site...

Example SW freely in Fiji:

- BigVolumeViewer (BVV)
  - Consumes the same **dataset.xml**
  - Similar to BDV
- **SciView** → → →
  - Uses BVV
  - Future Fiji viewer(?)

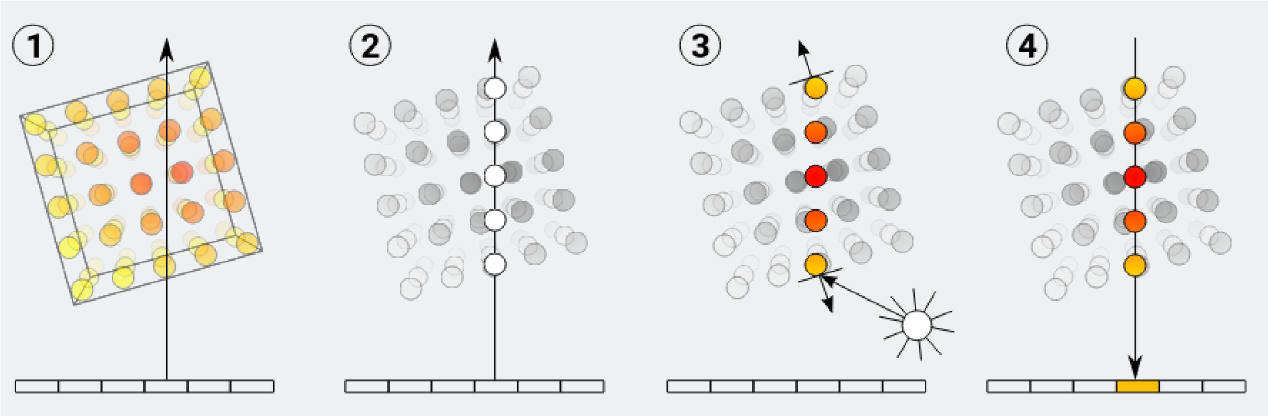


Visible	☑
Colour	🟥
Name	Mastodon spots at 502
Position X	70
Position Y	-66
Position Z	-11.3
Scale X	1
Scale Y	1
Scale Z	1
Rotation Phi	0
Rotation Theta	0
Rotation Psi	0
AO steps	0

Volume Rendering

Idea: Collect voxels along line**s** of sight, show cummulated value.

1) Cast ray to get value for every screen pixel
2) Fetch (off-grid) voxel values along the ray
3) Assign color considering coloring scheme, lighting conditions, transfer function
4) Composite colors to a final shown one



A true rendering is expensive:

- **Large** viewing **window** → **more rays** to be cast… and inspected
- **Large volume** → **each ray** needs to visit **more voxels** to obtain the display value
- New camera position → **recompute all over**

Volume Rendering

A true rendering is expensive:

- ***Large*** *viewing* ***window*** → ***more rays*** *to be cast… and inspected*
- ***Large volume*** → ***each ray*** *needs to visit* ***more voxels*** *to obtain the display value*
- *New camera position* → ***recompute all over***
- But requires no data understanding

Consider:

- Requesting smaller window to display the rendering
- Downscaling the displayed volume

Idea B: Decide what is foreground and background in the image, fit 3D surface mesh to the foreground, triangles take color from image, do standard rendering of triangles

- Actually, particularly popular solution…

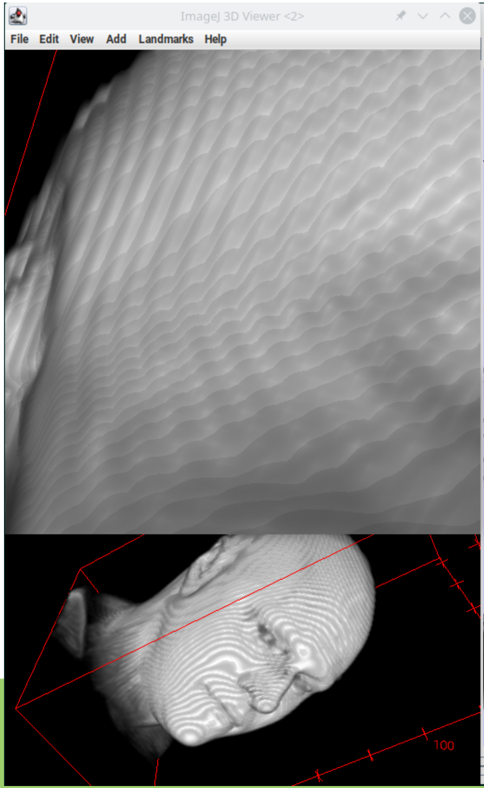
Volume Rendering

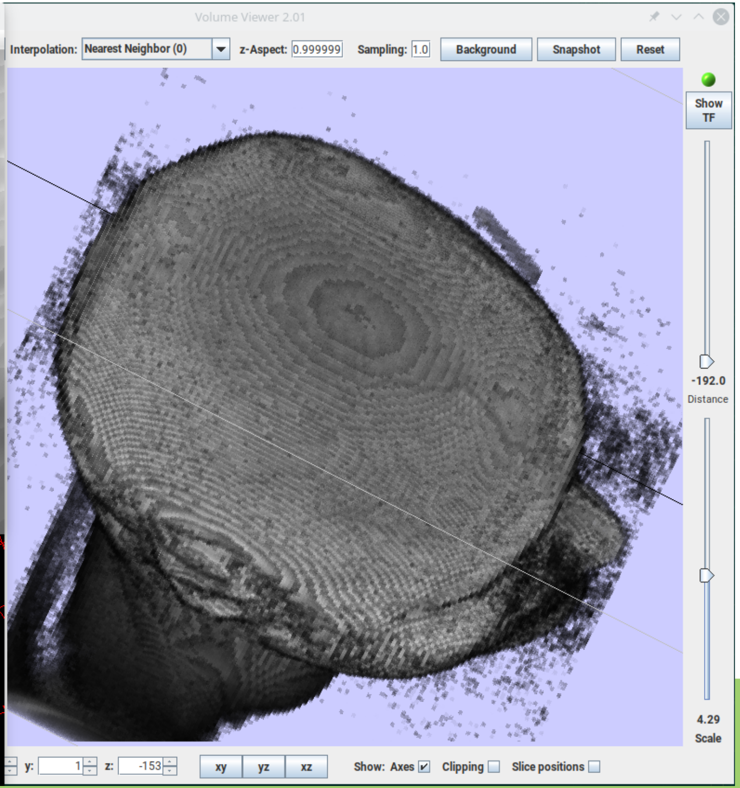
Idea B: Decide what is foreground and background in the image, fit 3D surface mesh to the foreground, triangles take color from image, standard rendering of triangles

- Actually, particularly popular solution…

Example SW freely in Fiji:

- **3D Viewer** →
- **Volume Viewer** →
- 3D script
  - Rendering post-processed
  - Video exports
  - Friendly animation narrator





Interpolation: Nearest Neighbor (0) z-Aspect: 0.999999 Sampling: 1.0

-192.0 Distance

4.29 Scale

y: 1 z: -153

Show: Axes ☑ Clipping □ Slice positions □

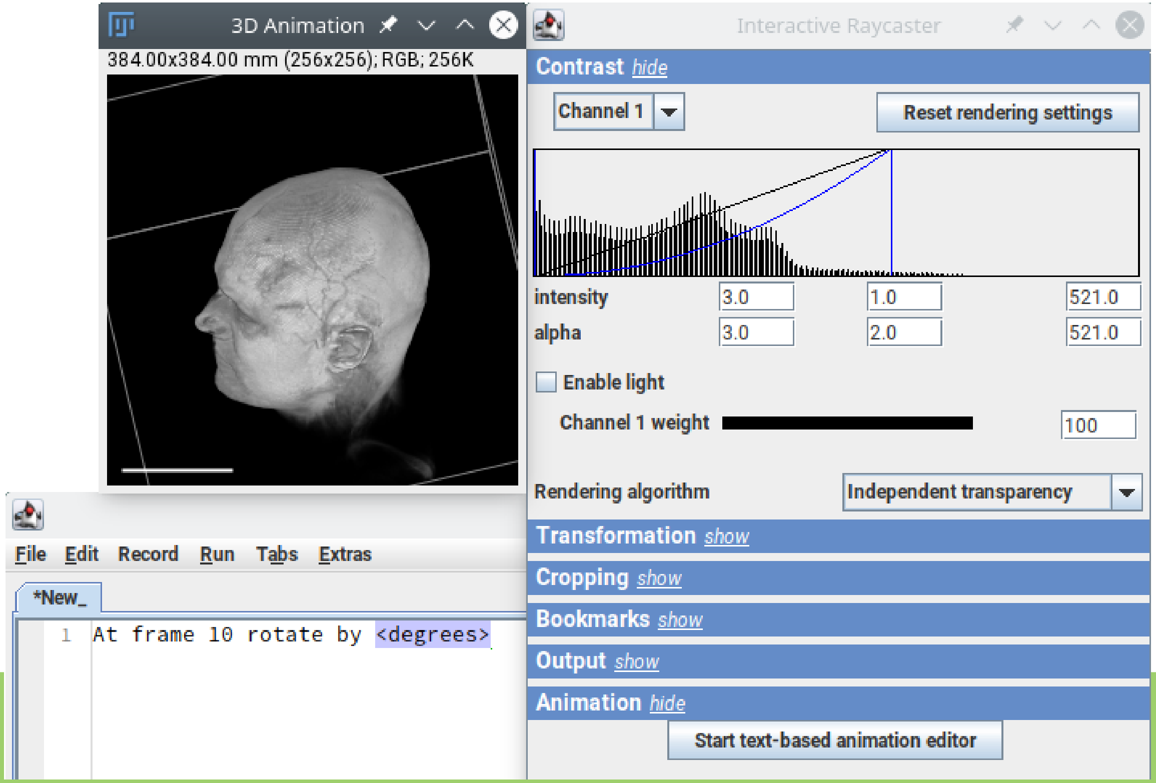
Volume Rendering

Idea B: Decide what is foreground and background in the image, fit 3D surface mesh to the foreground, triangles take color from image, standard rendering of triangles

- Actually, particularly popular solution…

Example SW freely in Fiji:

- 3D Viewer
- Volume Viewer
- **3D script** → → →
  - Rendering post-processed
  - Video exports
  - **Friendly animation narrator** →



3D Animation

384.00x384.00 mm (256x256); RGB; 256K

Interactive Raycaster

Contrast hide

Channel 1

intensity	3.0	1.0	521.0
alpha	3.0	2.0	521.0

☐ Enable light

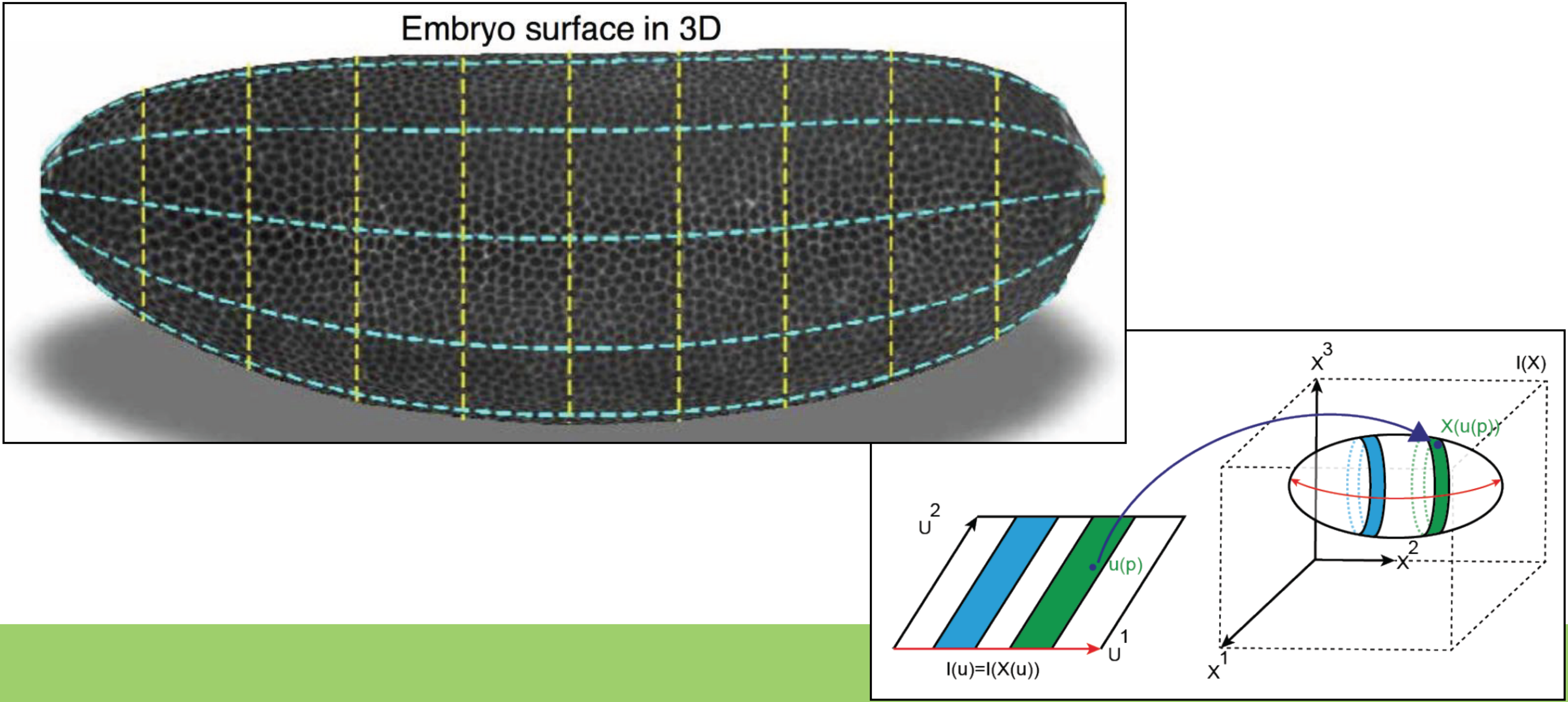
Channel 1 weight 100

Rendering algorithm: Independent transparency

```
At frame 10 rotate by <degrees>
```

Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object? Unfold!

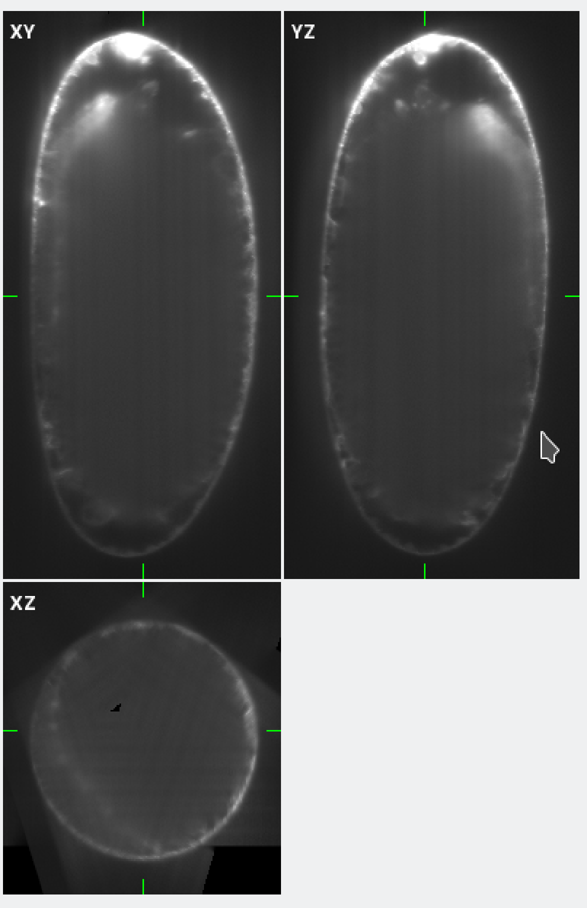


Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object? Unfold!

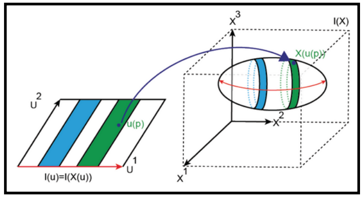
“Classical”

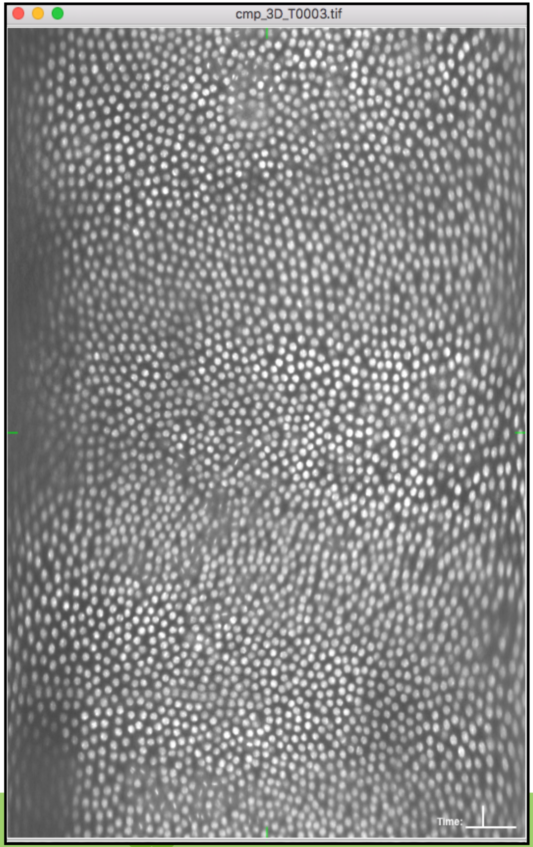
3D volume



Cartographic

2D slice

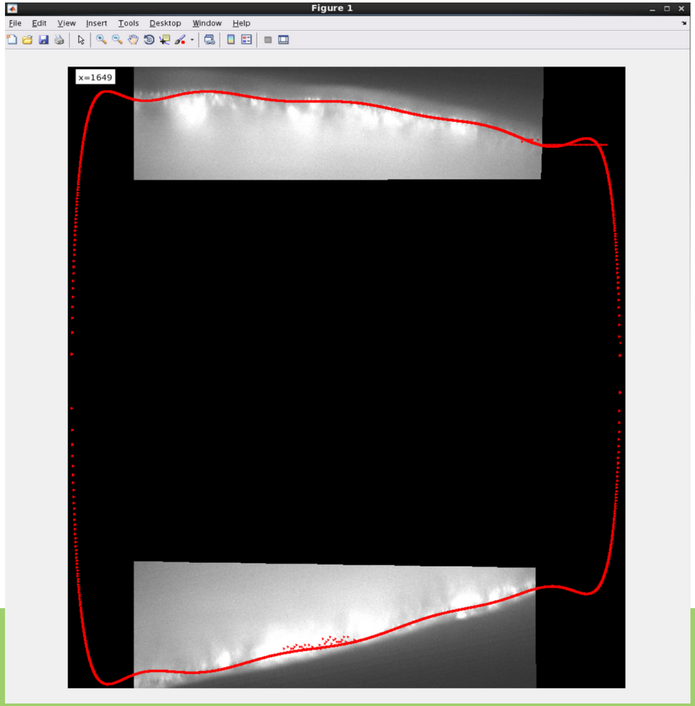


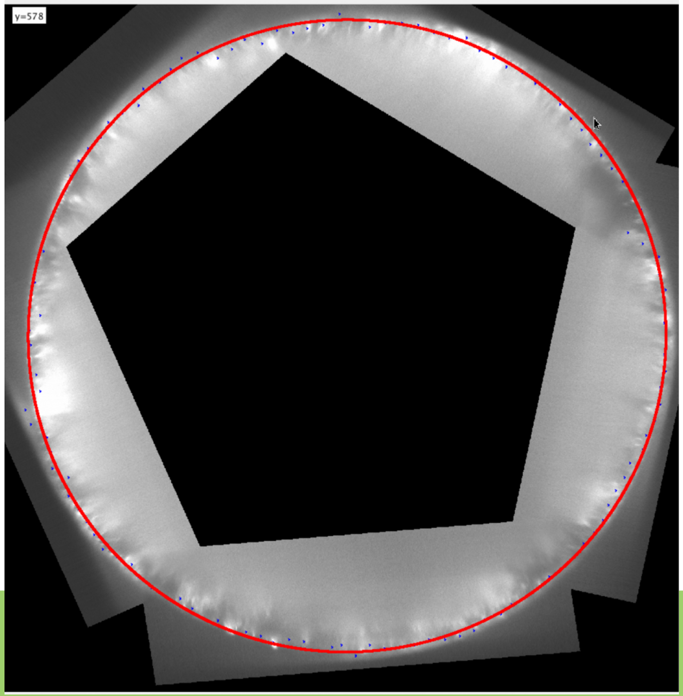


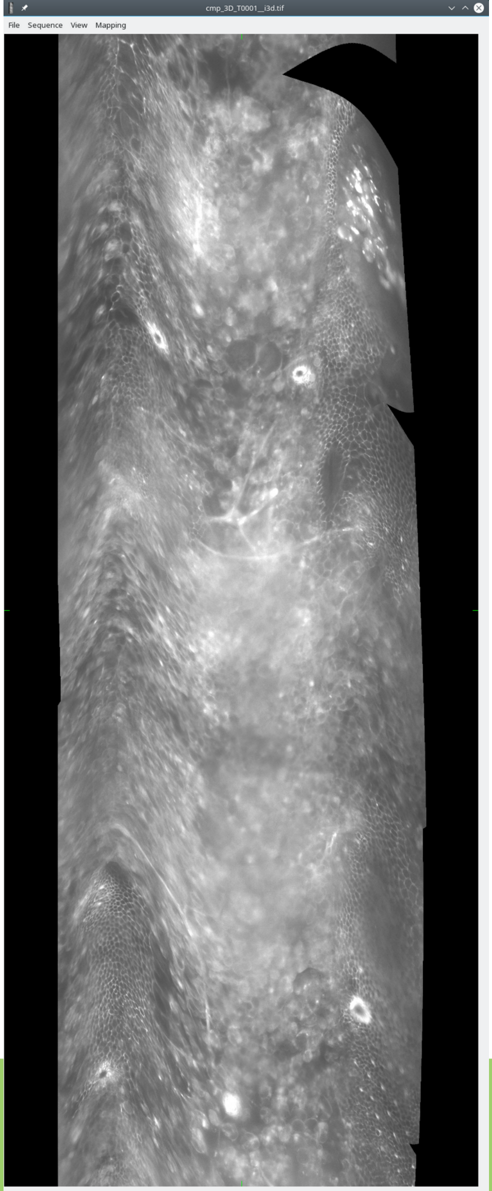
Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object?

- Requires precise boundary detection…
- Reduces data, here to 1000x
  3D 42.5 GB (2990 x 2536 x 3011) → 2D 41.8 MB (2517 x 8716)

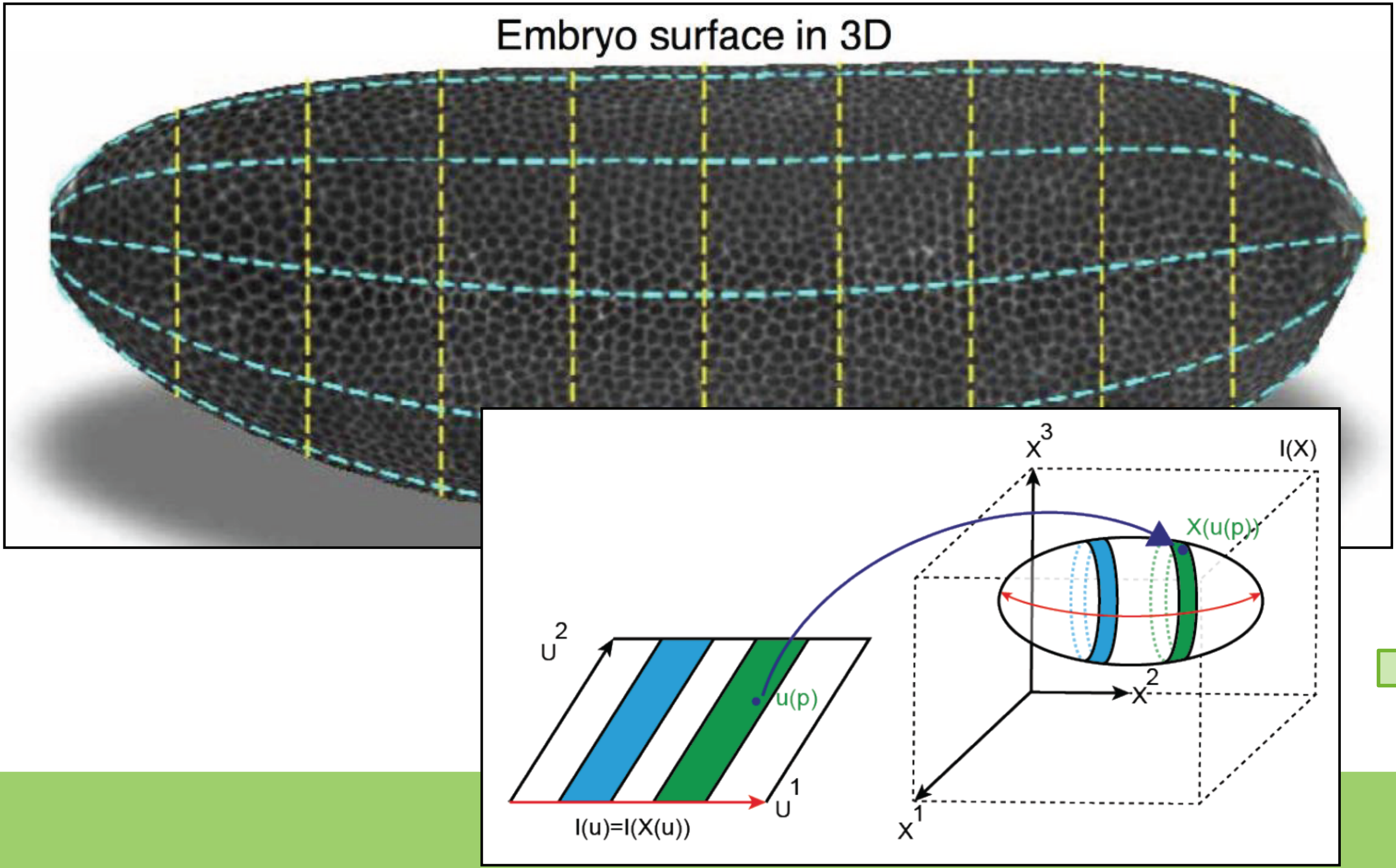




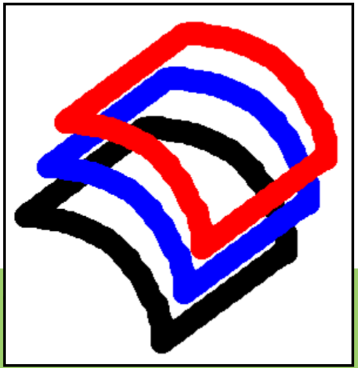


Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object? Unfold!



“Onion” layers

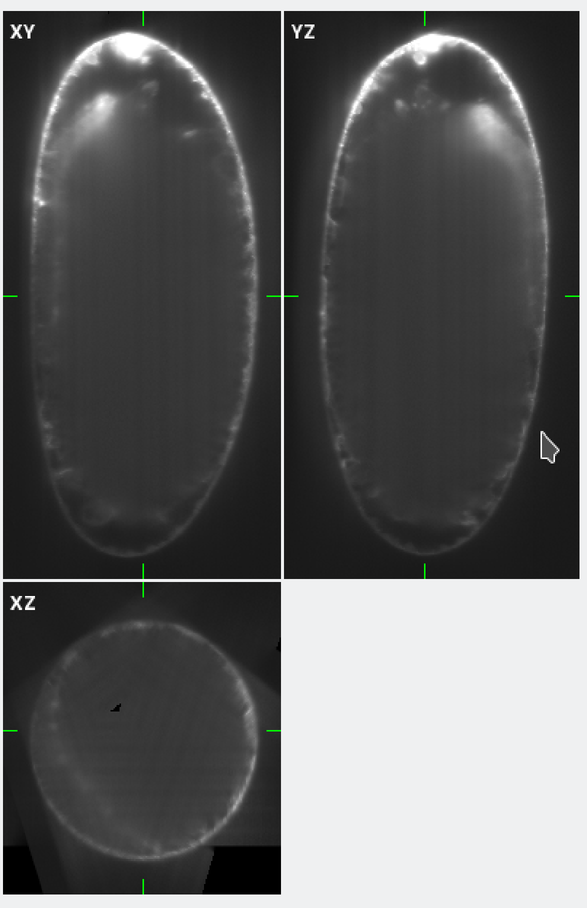


Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object? Unfold!

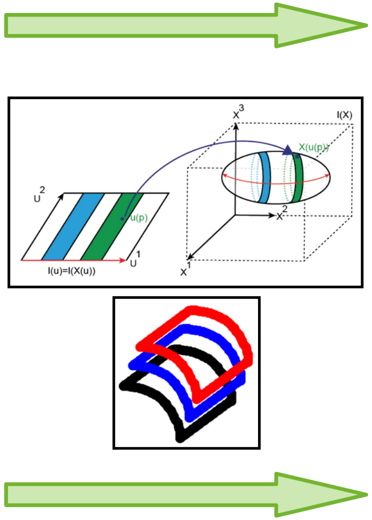
“Classical”

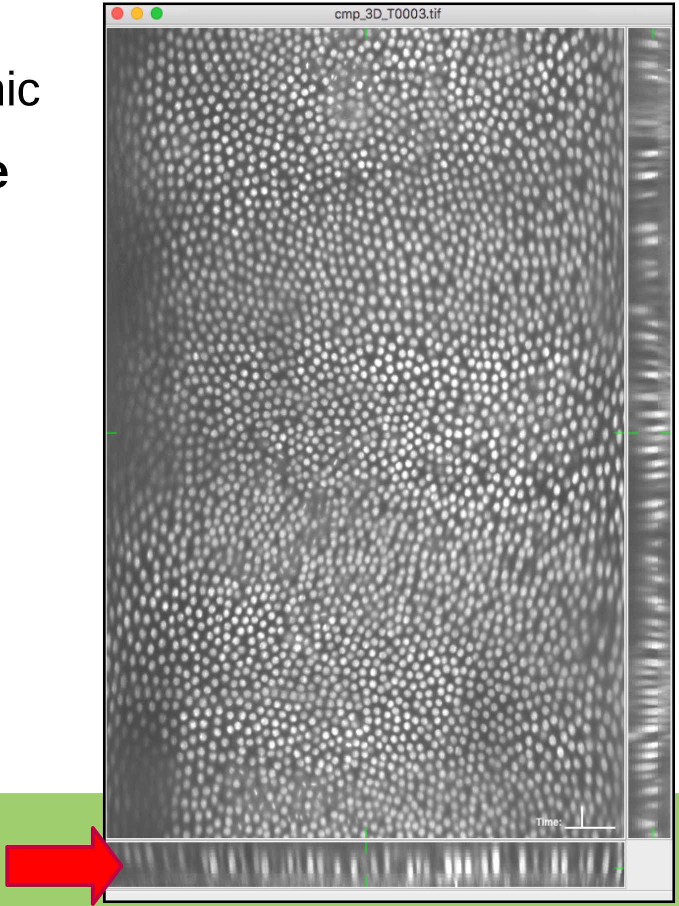
3D volume



Cartographic

3D volume



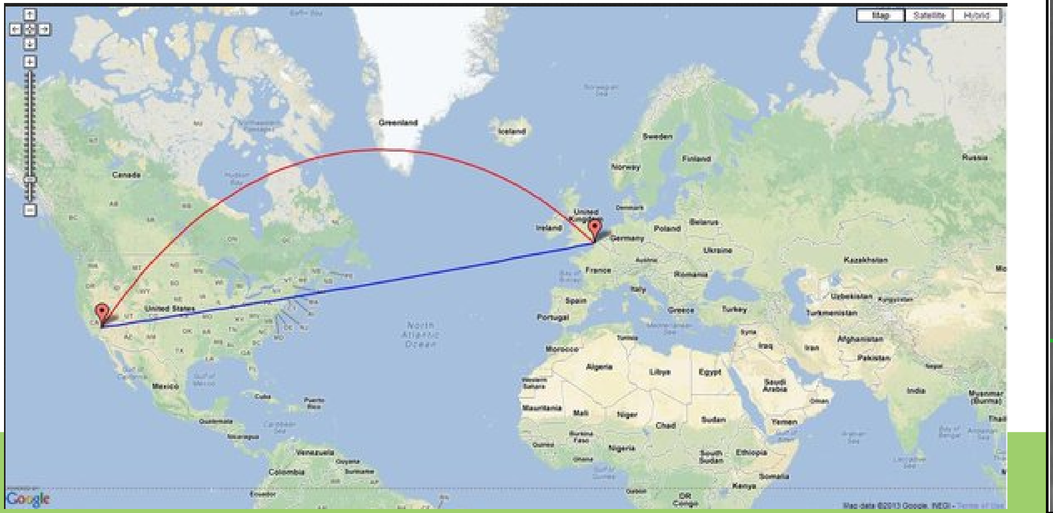


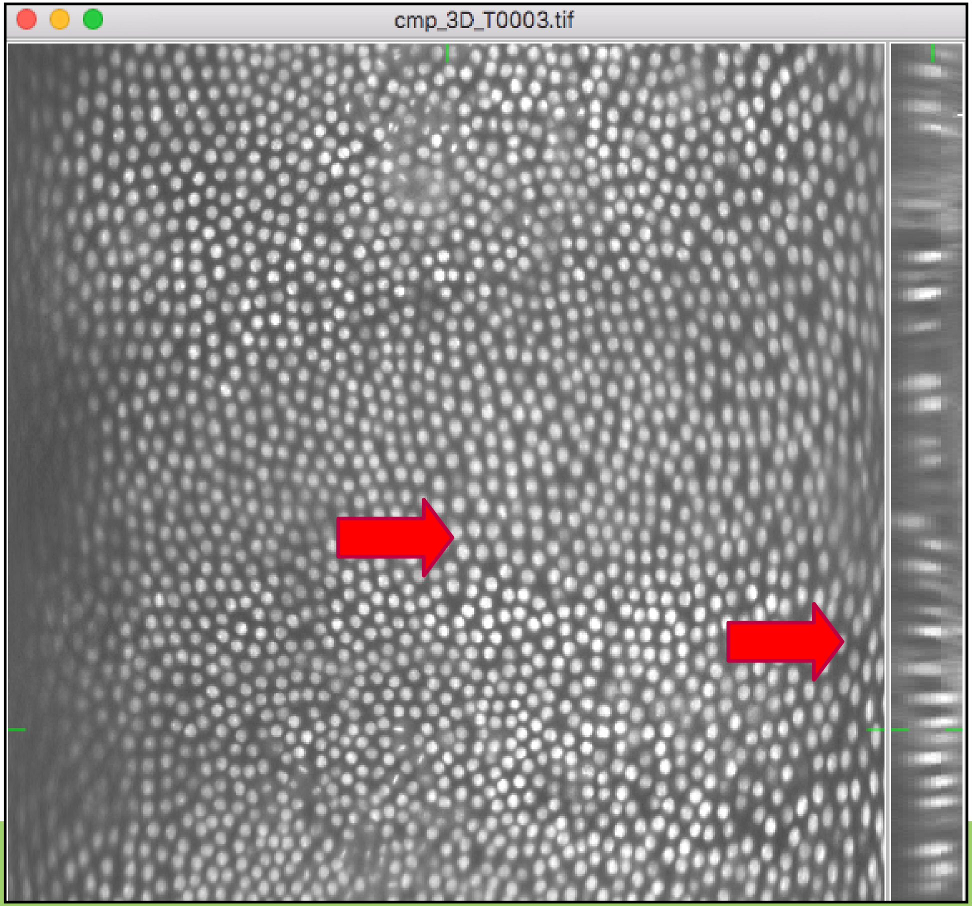
Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object?

- *Requires precise boundary detection…*
- *Reduces data…*

- **Distorts data**
  Is the flight route EU→US a direct line?



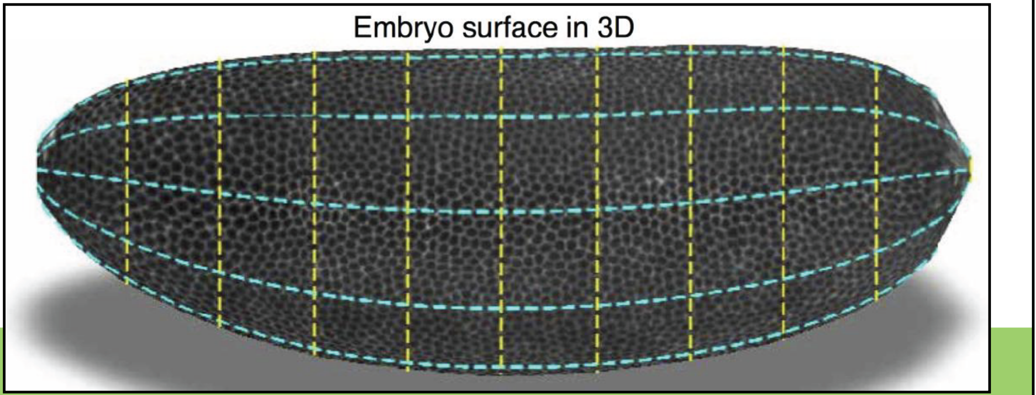


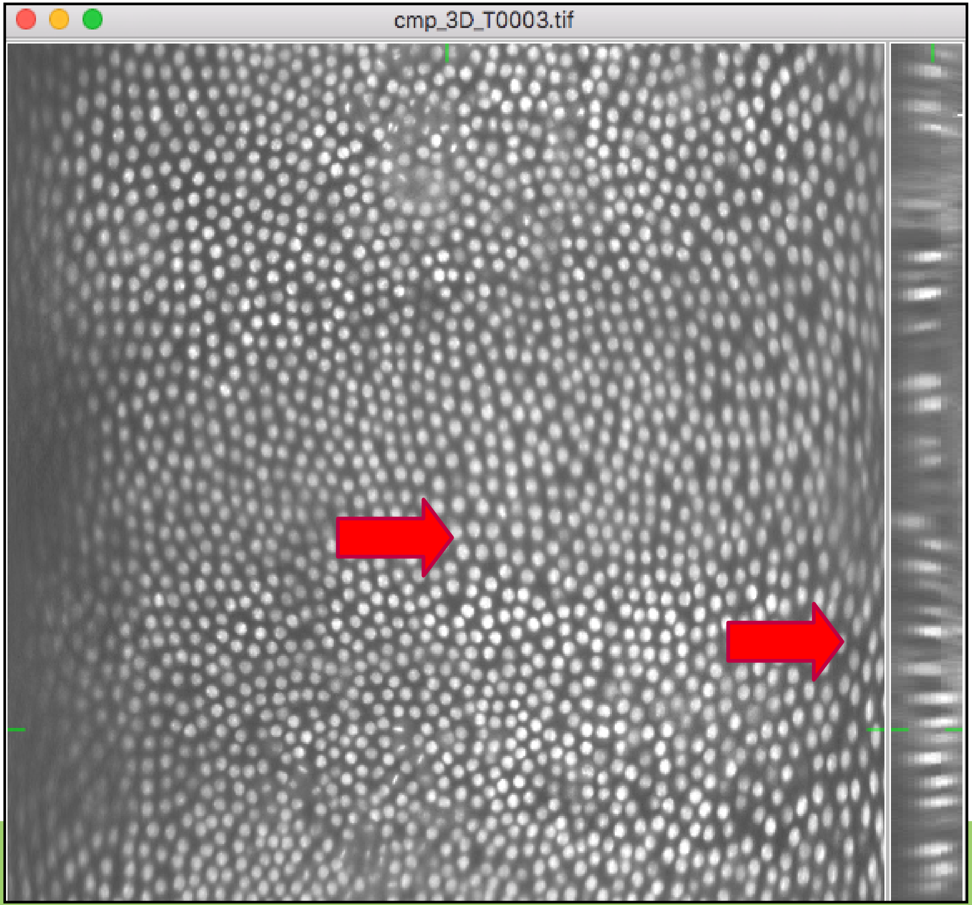
Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object?

- *Requires precise boundary detection…*
- *Reduces data…*

- **Distorts data** (**SW:** ImSAnE, Matlab)
  - Maps to a “horizontal” cylinder
  - Voxels at poles are stretched



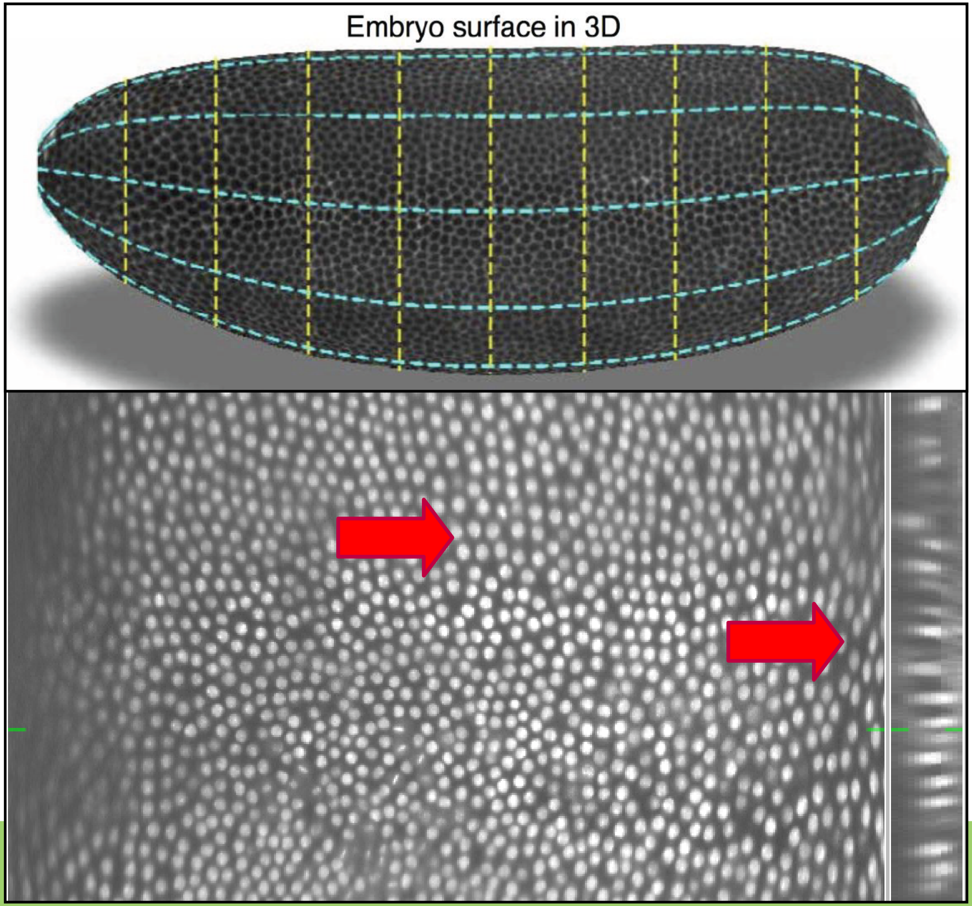


Cartographic Projections

Idea: (Interesting) Data is mostly on a mathematical-ish object?

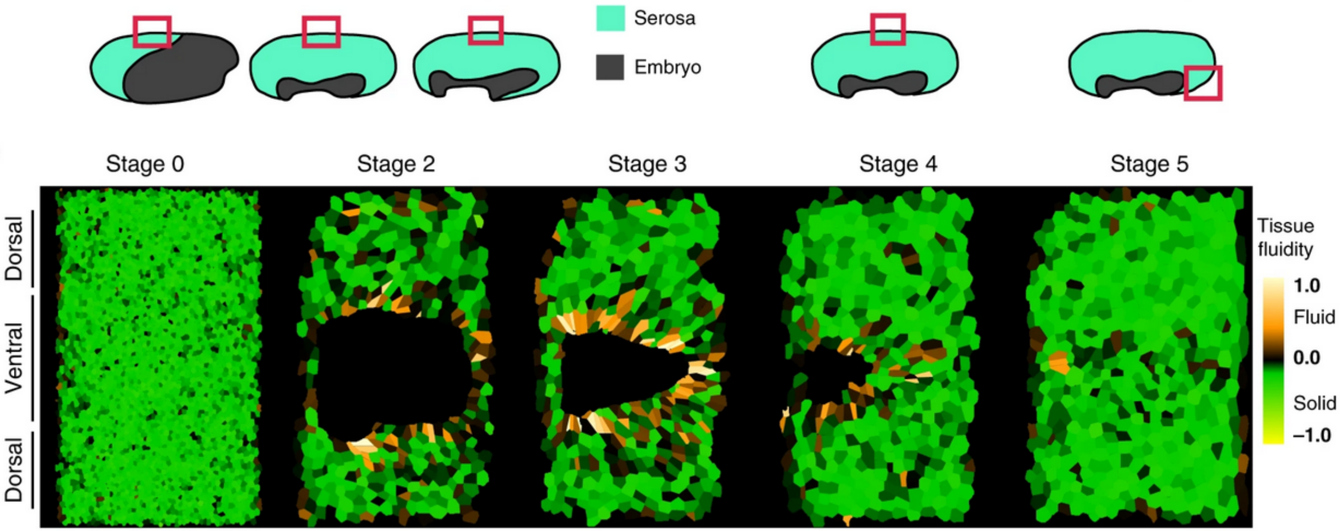
- *Requires precise boundary detection…*
- *Reduces data…*

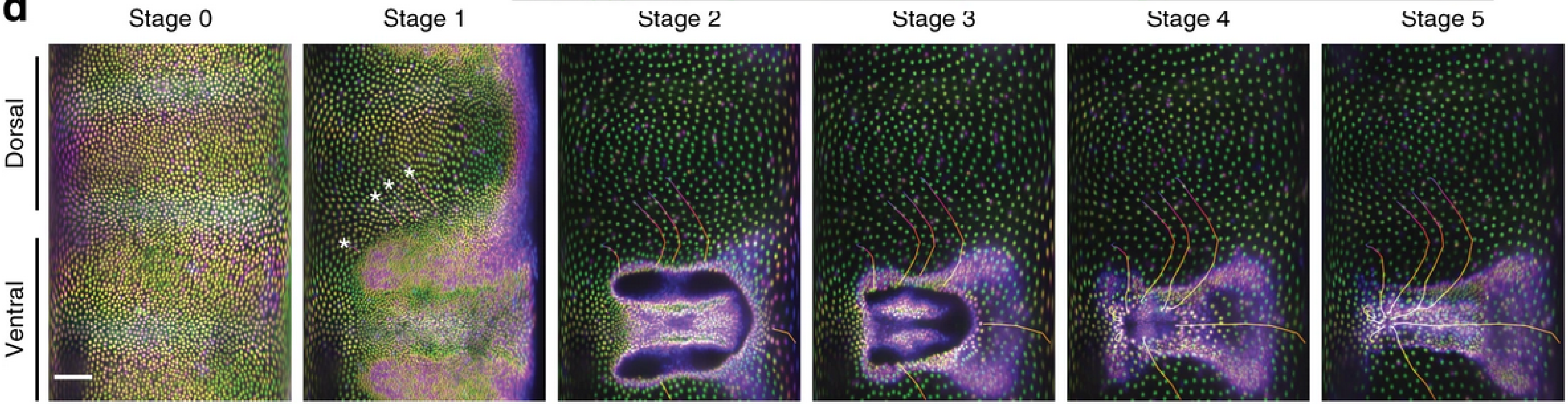
- **Distorts data** (**SW:** ImSAnE, Matlab)
  - Maps to a “horizontal” cylinder
  - Voxels at poles are stretched
  - Spatially varies
    - Real dist. between carto-pixels
    - Real area a carto-pixel represents
  - Known surface provides **correction maps**
  - **Adapted** image processing routines



Cartographic Projections

- From Jain et al. 2020:
- Development of cell properties
  - Fluidity → →
  - Position:





Graphical Representations

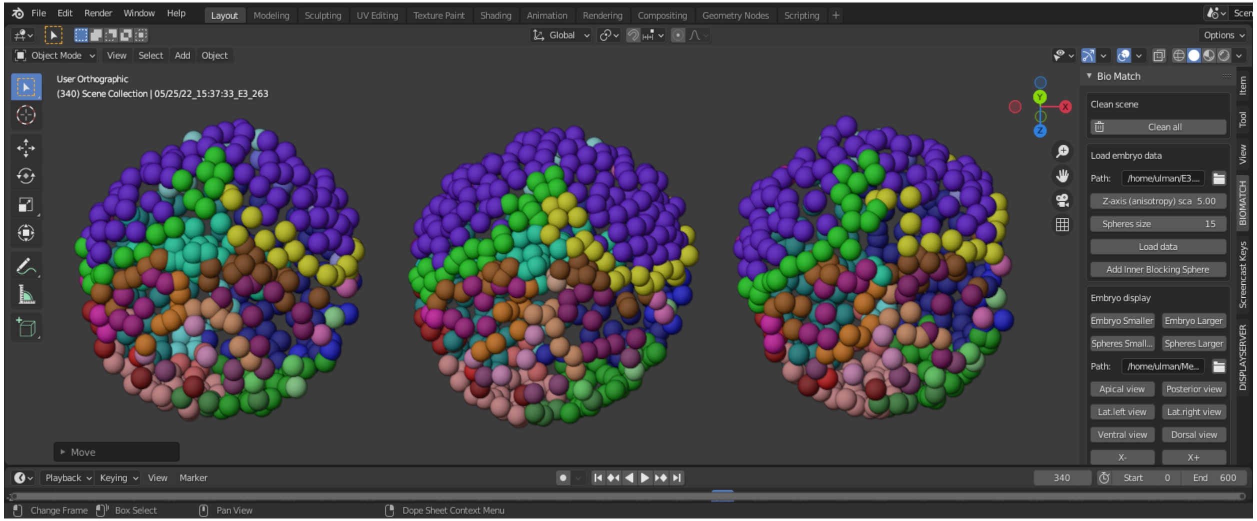
Idea: Show (3D) tabular data with computer graphics primitives.

➡ Suprissingly informative desptite tremendous information reduction.

Example SW

- SciView in Fiji
  - Uses BVV
  - Future Fiji viewer(?)

- Blender → → →
  - https://www.blender.org/
  - *”Blender is Free and Open Source software, forever.”*



User Orthographic
(340) Scene Collection | 05/25/22_15:37:33_E3_263

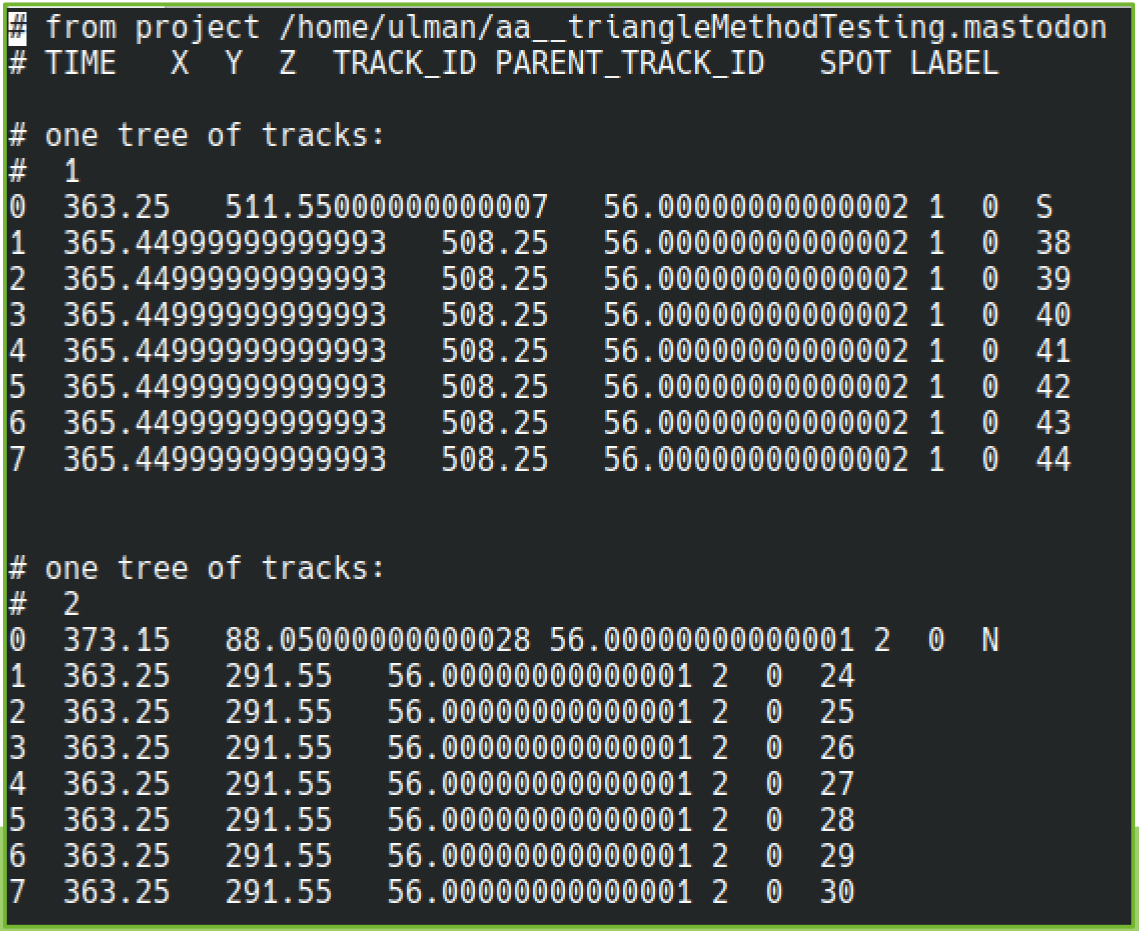
Path: /home/ulman/E3...
Z-axis (anisotropy) sca 5.00
Spheres size 15
Path: /home/ulman/Me...

340 Start 0 End 600

Graphical Representations

Idea: Show (3D) tabular data with computer graphics primitives.

→ Suprissingly informative desptite tremendous information reduction.



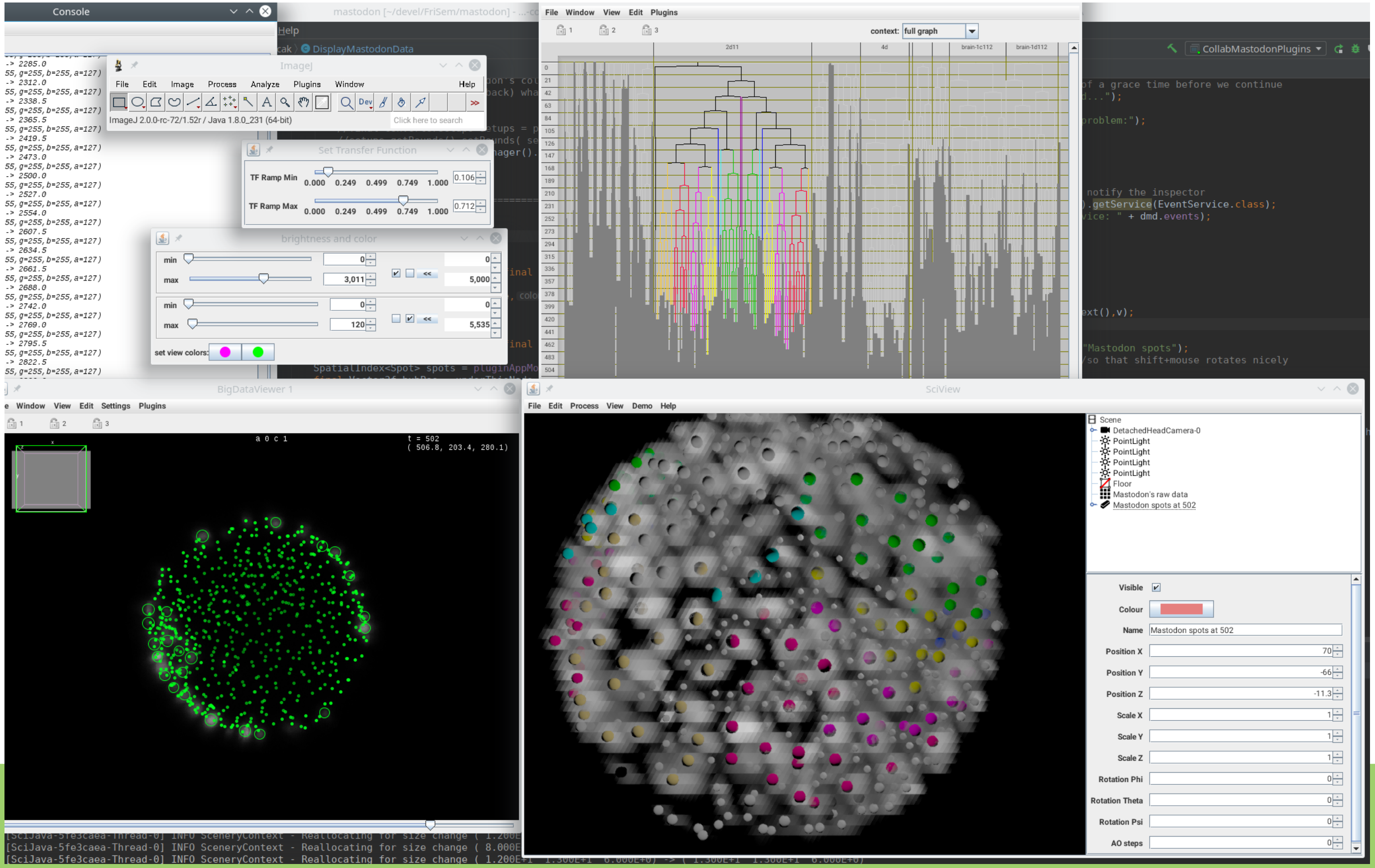
```
# from project /home/ulman/aa__triangleMethodTesting.mastodon
# TIME   X  Y  Z  TRACK_ID PARENT_TRACK_ID   SPOT LABEL

# one tree of tracks:
#  1
0  363.25   511.55000000000007   56.00000000000002 1  0  S
1  365.44999999999993   508.25   56.00000000000002 1  0  38
2  365.44999999999993   508.25   56.00000000000002 1  0  39
3  365.44999999999993   508.25   56.00000000000002 1  0  40
4  365.44999999999993   508.25   56.00000000000002 1  0  41
5  365.44999999999993   508.25   56.00000000000002 1  0  42
6  365.44999999999993   508.25   56.00000000000002 1  0  43
7  365.44999999999993   508.25   56.00000000000002 1  0  44


# one tree of tracks:
#  2
0  373.15   88.05000000000028 56.00000000000001 2  0  N
1  363.25   291.55   56.00000000000001 2  0  24
2  363.25   291.55   56.00000000000001 2  0  25
3  363.25   291.55   56.00000000000001 2  0  26
4  363.25   291.55   56.00000000000001 2  0  27
5  363.25   291.55   56.00000000000001 2  0  28
6  363.25   291.55   56.00000000000001 2  0  29
7  363.25   291.55   56.00000000000001 2  0  30
```

Graphical Representations

- SciView visualization (2020): featuring tag colors, **vol. rendering**, spots, trajectories, adjustable vizu, interconnected!



```
-> 2285.0
55, g=255, b=255, a=127)
-> 2312.0
55, g=255, b=255, a=127)
-> 2338.5
55, g=255, b=255, a=127)
-> 2365.5
55, g=255, b=255, a=127)
-> 2419.5
55, g=255, b=255, a=127)
-> 2473.0
55, g=255, b=255, a=127)
-> 2500.0
55, g=255, b=255, a=127)
-> 2527.0
55, g=255, b=255, a=127)
-> 2554.0
55, g=255, b=255, a=127)
-> 2607.5
55, g=255, b=255, a=127)
-> 2634.5
55, g=255, b=255, a=127)
-> 2661.5
55, g=255, b=255, a=127)
-> 2688.0
55, g=255, b=255, a=127)
-> 2742.0
55, g=255, b=255, a=127)
-> 2769.0
55, g=255, b=255, a=127)
-> 2795.5
55, g=255, b=255, a=127)
-> 2822.5
55, g=255, b=255, a=127)
```

ImageJ 2.0.0-rc-72/1.52r / Java 1.8.0_231 (64-bit)

TF Ramp Min 0.106

TF Ramp Max 0.712

min 0 max 3,011 5,000

min 0 max 120 5,535

```
of a grace time before we continue
d...");

problem:");

 notify the inspector
).getService(EventService.class);
vice: " + dmd.events);

ext(),v);

"Mastodon spots");
/so that shift+mouse rotates nicely
```

```
[SciJava-5fe3caea-Thread-0] INFO SceneryContext - Reallocating for size change ( 8.000E
[SciJava-5fe3caea-Thread-0] INFO SceneryContext - Reallocating for size change ( 1.200E+1  1.300E+1  6.000E+0) -> ( 1.300E+1  1.300E+1  6.000E+0)
```

Name: Mastodon spots at 502
Position X: 70
Position Y: -66
Position Z: -11.3
Scale X: 1
Scale Y: 1
Scale Z: 1
Rotation Phi: 0
Rotation Theta: 0
Rotation Psi: 0
AO steps: 0

also: little fragile, HW heavy

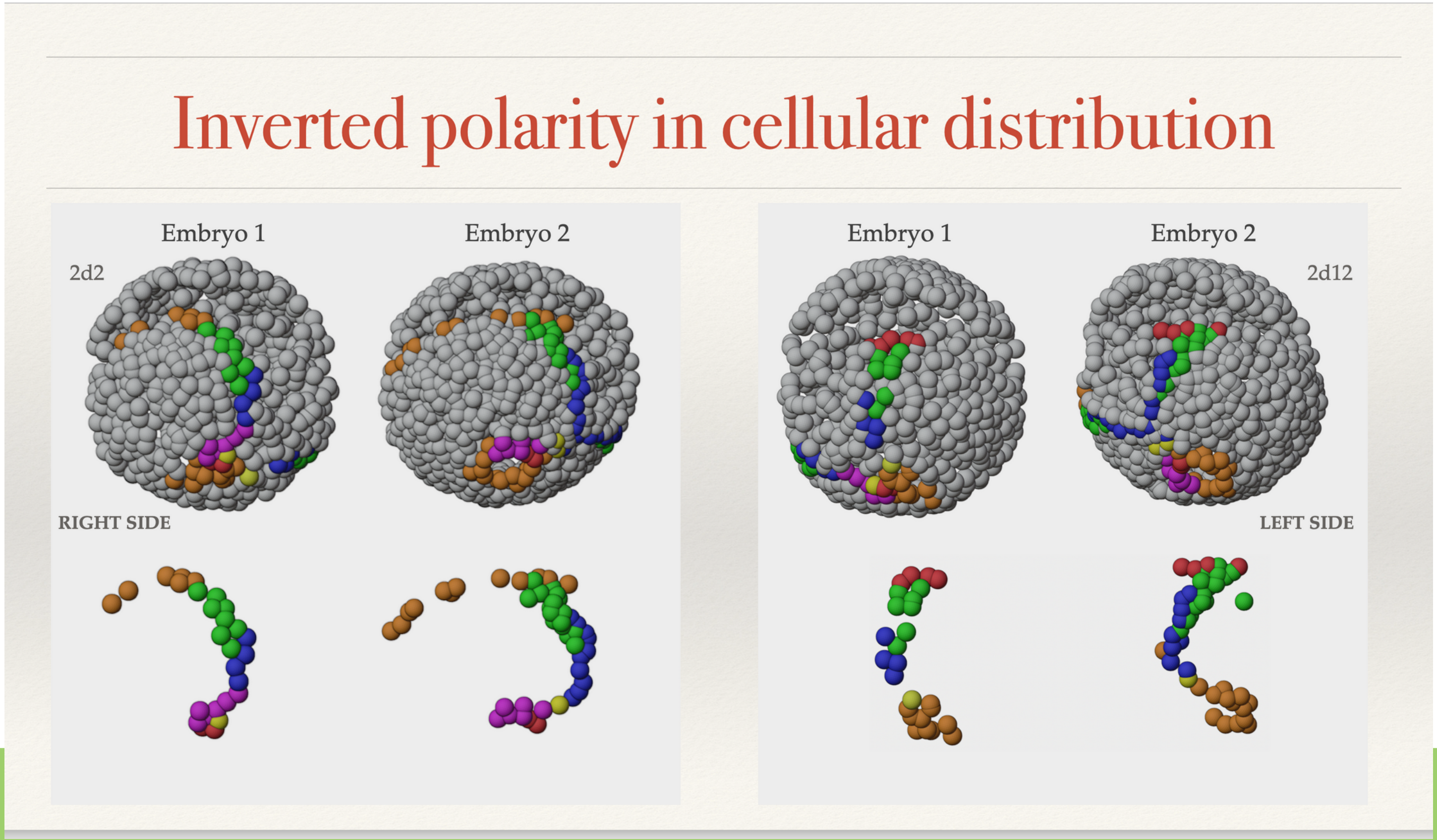
Graphical Representations

- Blender visualization (newest, 2021):

  stable, known, advanced…

  overwhelming, google-able

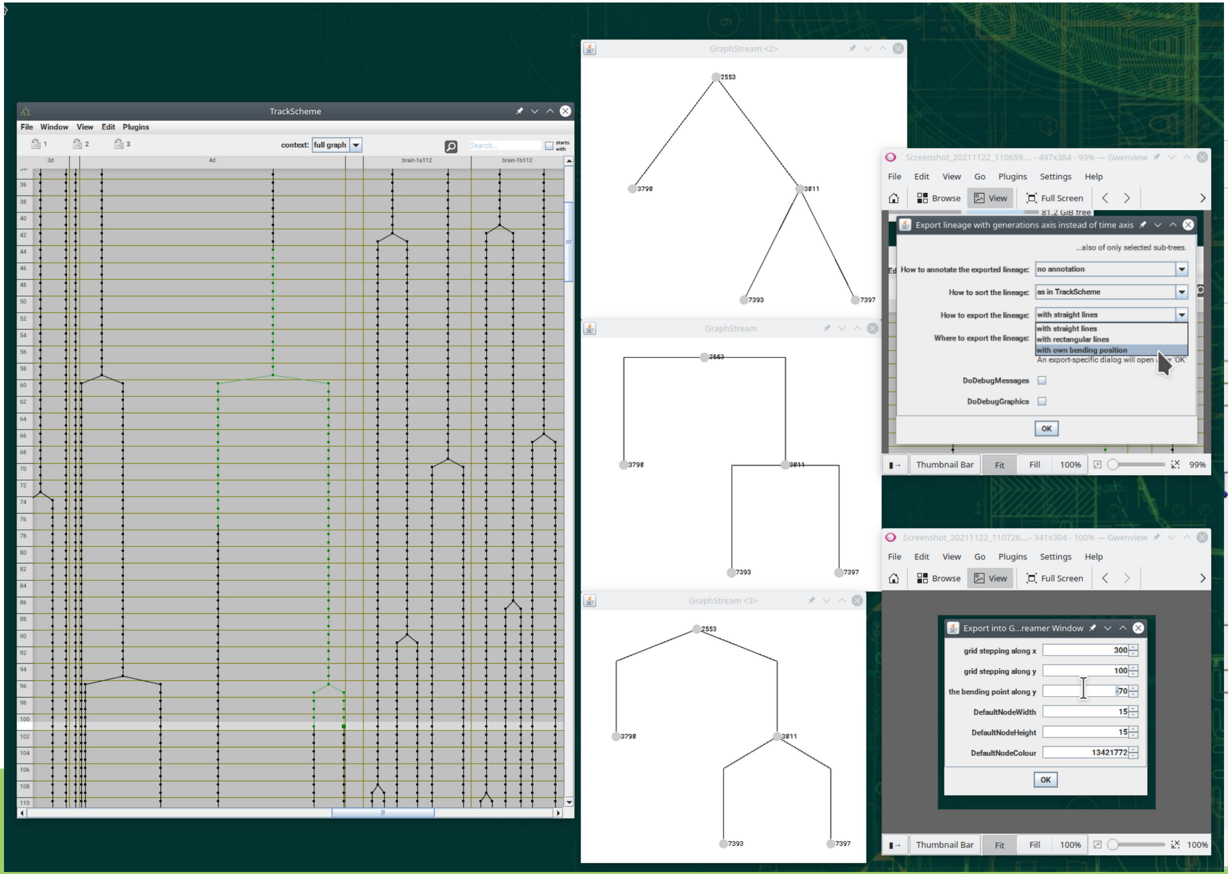
  capable, performant



Inverted polarity in cellular distribution

Graphical Representations

- Blender visualization (newest, 2021): lineage vizu (options)


TrackScheme
File Window View Edit Plugins
context: full graph
GraphStream <2>
2553
3798
3811
7393
7397
GraphStream
GraphStream <3>
Screenshot_20211122_110659.... - 497x384 - 99% — Gwenview
File Edit View Go Plugins Settings Help
Browse View Full Screen
Thumbnail Bar Fit Fill 100% 99%
Screenshot_20211122_110726....- 341x304 - 100% — Gwenview
Thumbnail Bar Fit Fill 100% 100%

Export lineage with generations axis instead of time axis

...also of only selected sub-trees.

How to annotate the exported lineage: no annotation

How to sort the lineage: as in TrackScheme

How to export the lineage: with straight lines

- with straight lines
- with rectangular lines
- with own bending position

Where to export the lineage:

An export-specific dialog will open after 'OK'

DoDebugMessages □

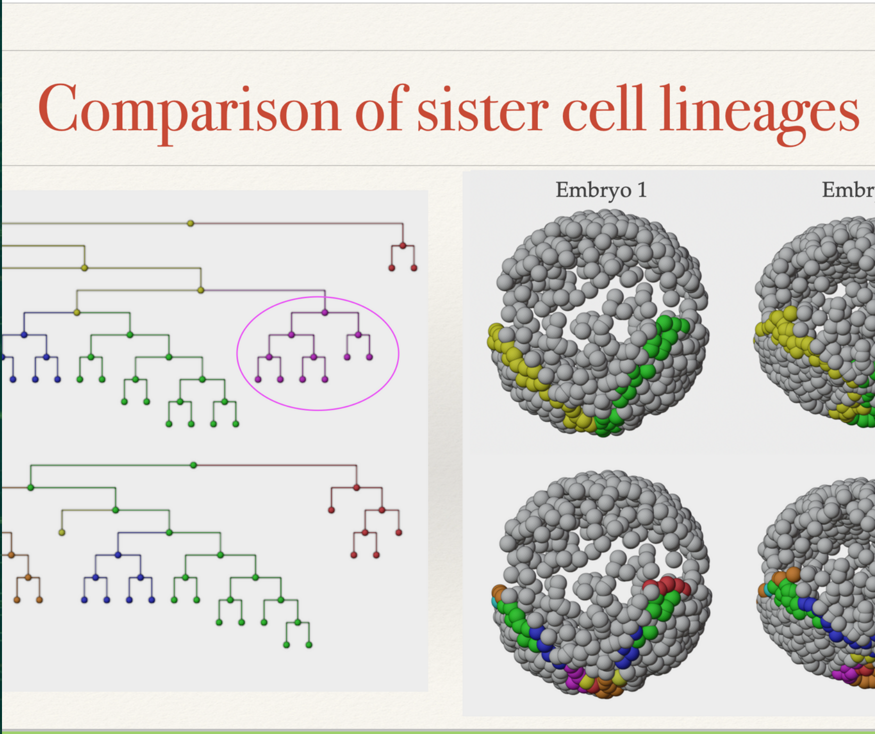
DoDebugGraphics □

OK

Export into G...reamer Window

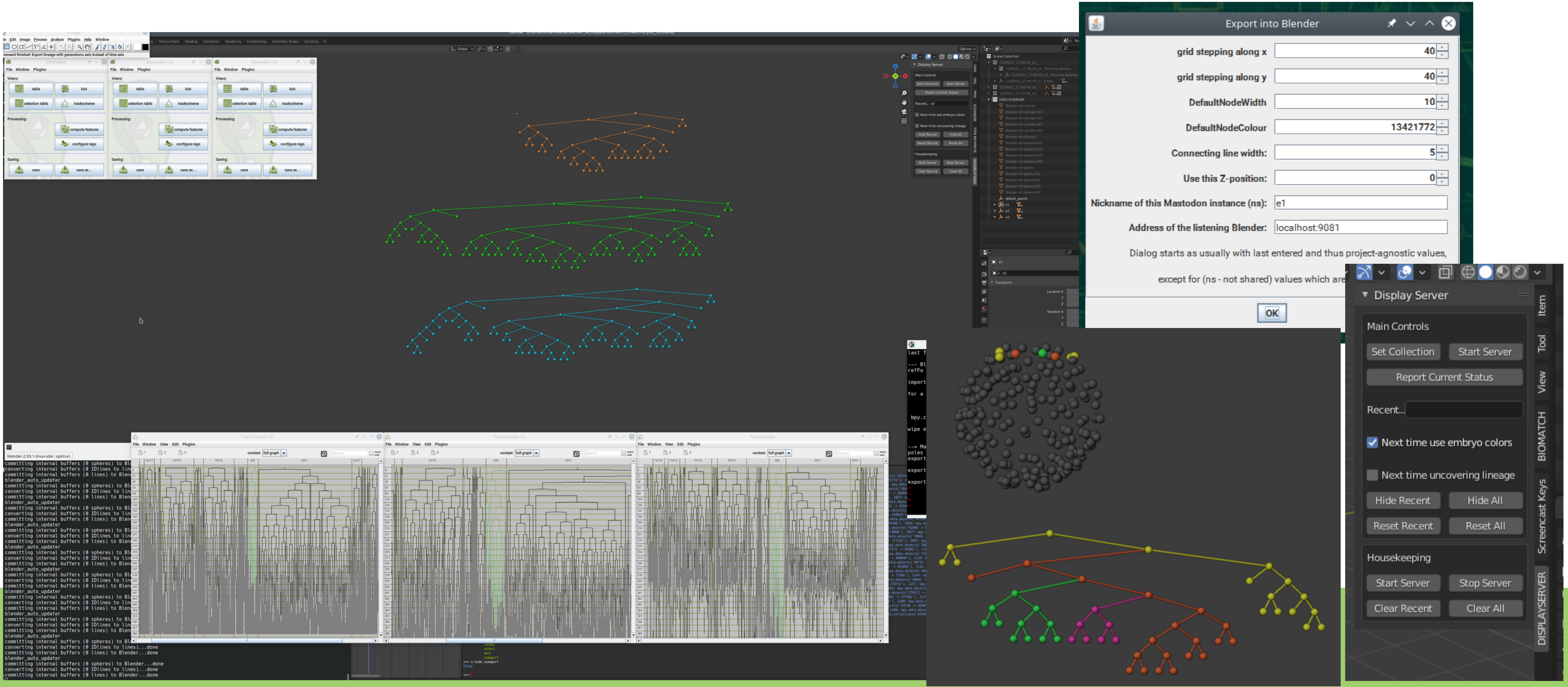
grid stepping along x	300
grid stepping along y	100
the bending point along y	-70
DefaultNodeWidth	15
DefaultNodeHeight	15
DefaultNodeColour	13421772

OK


Comparison of sister cell lineages

Graphical Representations

- Blender visualization (newest, 2021): comparing trees (4K LCD)



Export into Blender

grid stepping along x: 40

grid stepping along y: 40

DefaultNodeWidth: 10

DefaultNodeColour: 13421772

Connecting line width: 5

Use this Z-position: 0

Nickname of this Mastodon instance (ns): e1

Address of the listening Blender: localhost:9081

Dialog starts as usually with last entered and thus project-agnostic values,

except for (ns - not shared) values which are

OK

Display Server

Main Controls

Set Collection | Start Server

Report Current Status

Recent...

☑ Next time use embryo colors

□ Next time uncovering lineage

Hide Recent | Hide All

Reset Recent | Reset All

Housekeeping

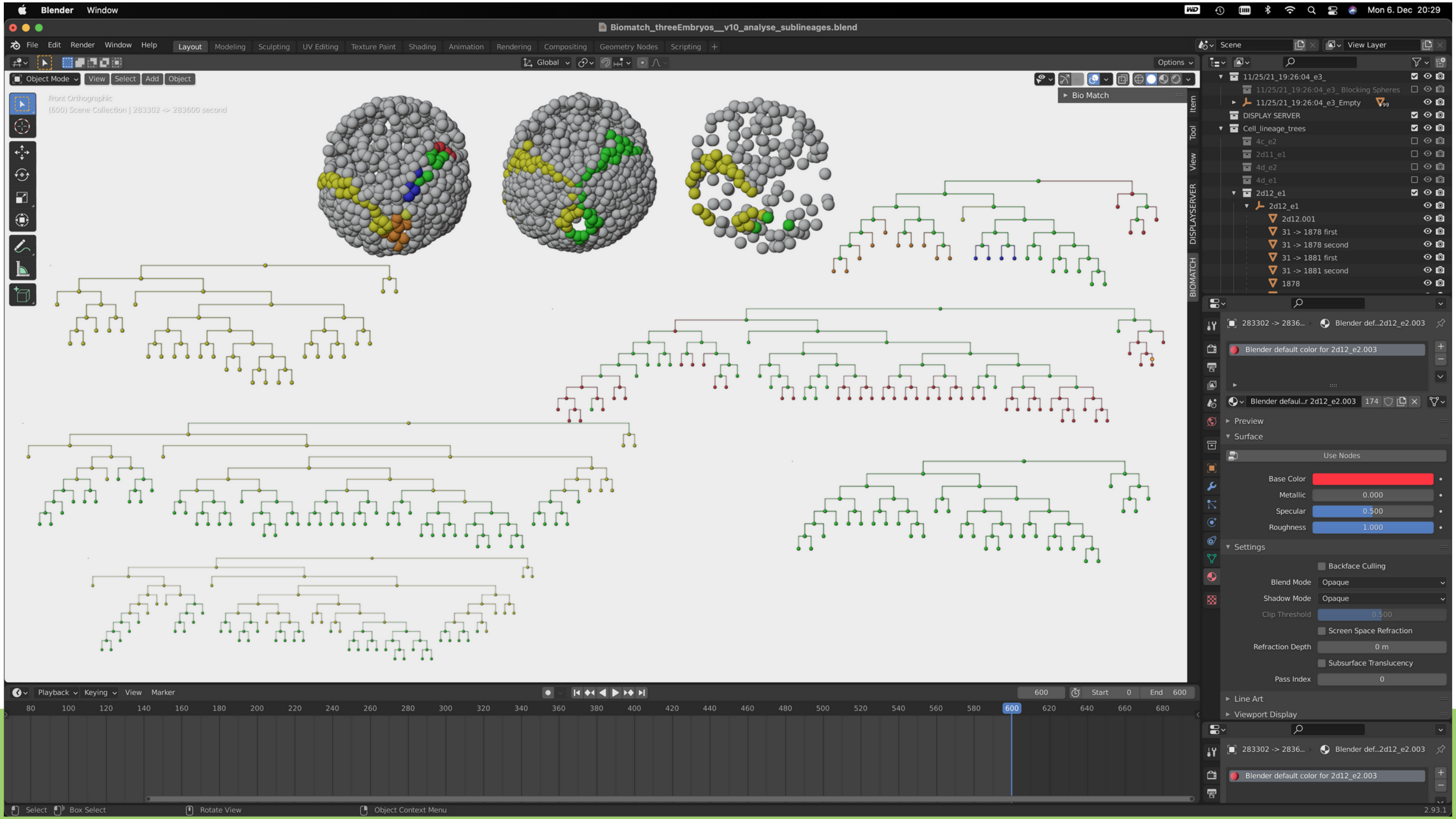
Start Server | Stop Server

Clear Recent | Clear All

```
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lin
committing internal buffers (0 lines) to Blen
blender_auto_updater
committing internal buffers (0 spheres) to Bl
converting internal buffers (0 IDlines to lines)...done
committing internal buffers (0 lines) to Blender...done
blender_auto_updater
committing internal buffers (0 spheres) to Blender...done
converting internal buffers (0 IDlines to lines)...done
committing internal buffers (0 lines) to Blender...done
```

Graphical Representations

- Blender visualization (newest, 2021):



Biomatch_threeEmbryos__v10_analyse_sublineages.blend

Front Orthographic
(600) Scene Collection | 283302 -> 283600 second

11/25/21_19:26:04_e3_
11/25/21_19:26:04_e3_ Blocking Spheres
11/25/21_19:26:04_e3_Empty
DISPLAY SERVER
Cell_lineage_trees
4c_e2
2d11_e1
4d_e2
4d_e1
2d12_e1
2d12_e1
2d12.001
31 -> 1878 first
31 -> 1878 second
31 -> 1881 first
31 -> 1881 second
1878

283302 -> 2836... Blender def...2d12_e2.003

Blender default color for 2d12_e2.003

Blender defaul...r 2d12_e2.003 174

Preview

Surface

Use Nodes

Base Color
Metallic 0.000
Specular 0.500
Roughness 1.000

Settings

□ Backface Culling
Blend Mode Opaque
Shadow Mode Opaque
Clip Threshold 0.500
□ Screen Space Refraction
Refraction Depth 0 m
□ Subsurface Translucency
Pass Index 0

Line Art

Viewport Display

283302 -> 2836... Blender def...2d12_e2.003

Blender default color for 2d12_e2.003

600 Start 0 End 600

80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400 420 440 460 480 500 520 540 560 580 600 620 640 660 680

flexibility,

animations,

interactivity,

ATM: very

much a

prototype

THANK YOU

- I thank all my colleagues and supporters

- I’m available both weeks, full time
- I’m here to help
  ….and also to experiment ;-)
- Please, don’t hesitate to approach me

- Download: **https://www.fi.muni.cz/~xulman/files/EMBO_LS2022.pdf**