universität wien

Master‘s Thesis

Scalability of Modern Scatterplot Visualizations for Large Image Datasets

by

Sebastian Klaassen

Advisor

Torsten Möller

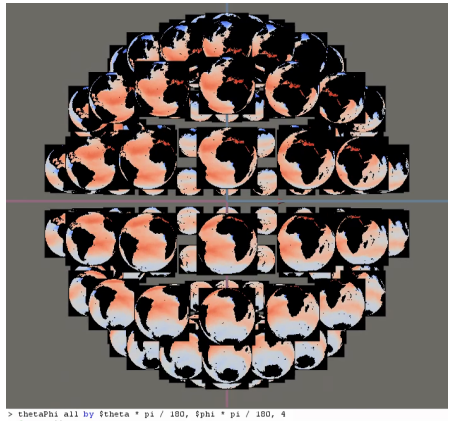
Motivation

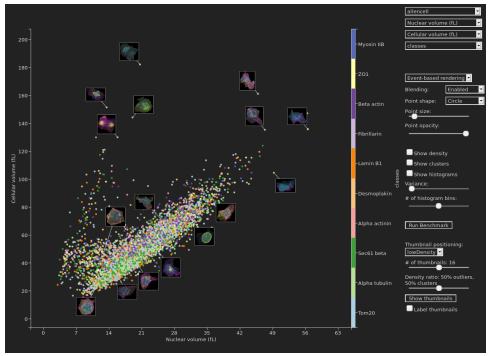
- The scatterplot (est. 1833) ↔ Modern large-scale image datasets
- Scalability of modern scatterplot visualizations for large image datasets
  - Performance scalability: What is feasible?
  - Information scalability: What is reasonable?

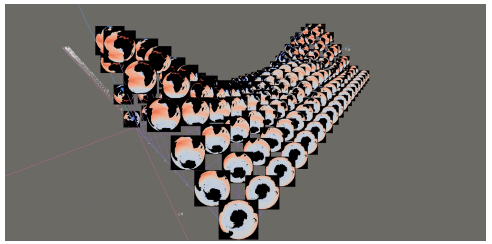
- Contributions
  - Software
    - 2 scatterplot-based applications
  - Theory
    - Exploration of visual mappings
    - Analysis of thumbnail placement strategies
    - Computation and usage of density maps

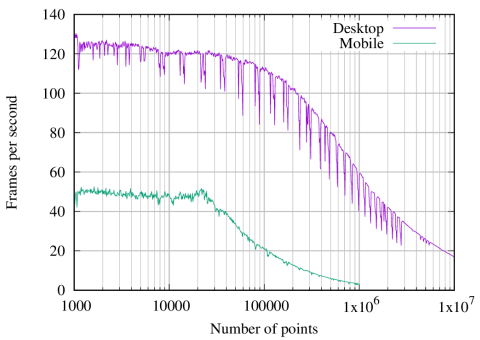
Outline

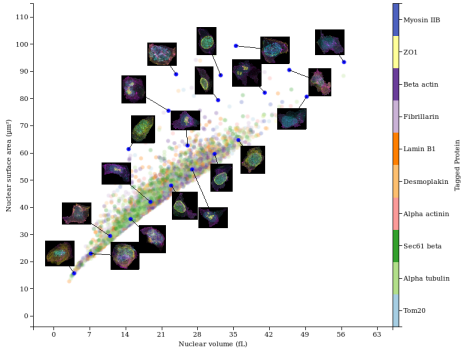
- Motivation
- Application 1: Global View
- Application 2: Interactive Cell Plot
- Visual Mapping
- Density Maps
- Labeling
- Evaluation

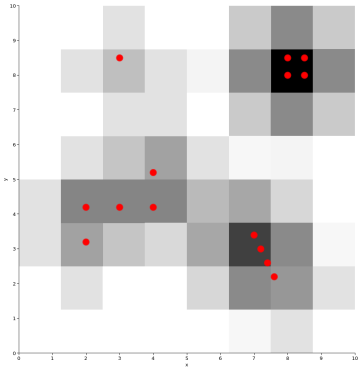






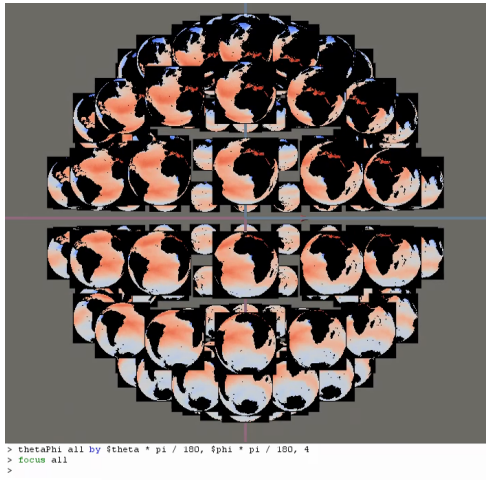






Application 1 → Outline

- Motivation
- Application 1: Global View
  - Requirements
  - Introduction
  - MPAS Dataset
- Application 2: Interactive Cell Plot
- Visual Mapping
- Density Maps
- Labeling
- Evaluation



Application 1 → Requirements

- Requirements
  - The Cinema data format assumes a dense input space
    - Datasets are assumed to contain images for all combinations of input parameters.
    - This restriction has been removed with “Dietrich” spec. of Cinema.
  - Performance Scalability
  - Information Scalability
  - Flexibility of Visual Mappings
  - Platform Independence

Application 1 → Requirements

- Requirements
  - The Cinema data format assumes a dense input space
  - Performance Scalability
    - In-situ image databases contain densely sampled domains with high resolution images. → database size > 100GB
    - We use a background thread to load/unload images on demand.
  - Information Scalability
  - Flexibility of Visual Mappings
  - Platform Independence

Application 1 → Requirements

- Requirements
  - The Cinema data format assumes a dense input space
  - Performance Scalability
  - Information Scalability
    - How many dimensions/images can be reasonably presented to the user?
    - We explore different ways of presenting the MPAS-Ocean dataset.
  - Flexibility of Visual Mappings
  - Platform Independence

Application 1 → Requirements

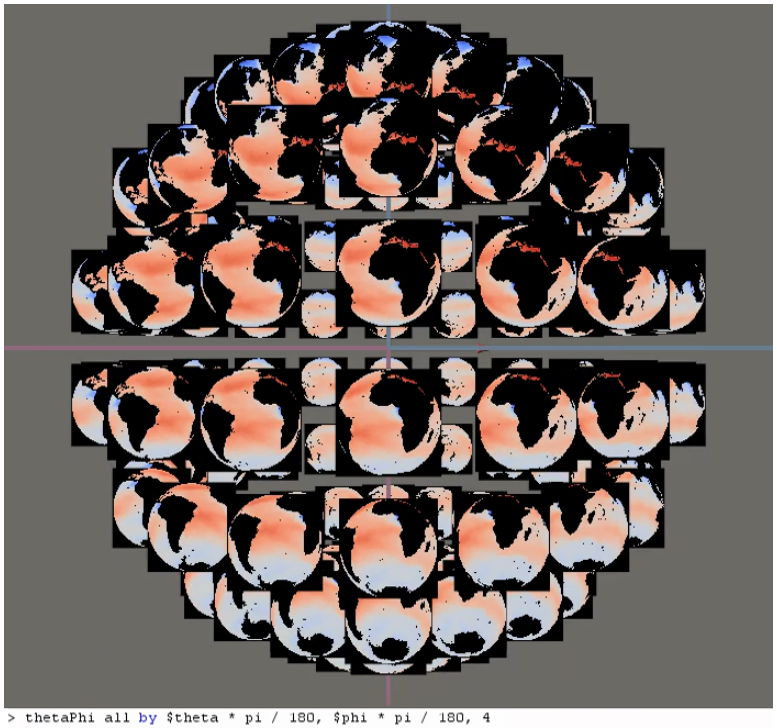
- Requirements
  - The Cinema data format assumes a dense input space
  - Performance Scalability
  - Information Scalability
  - Flexibility of Visual Mappings
    - Information scalability requires exploration of visual mappings.
    - We implement a scripting interface to apply vis. mappings on the fly.
  - Platform Independence

Application 1 → Requirements

- Requirements
  - The Cinema data format assumes a dense input space
  - Performance Scalability
  - Information Scalability
  - Flexibility of Visual Mappings
  - Platform Independence
    - To be considered a versatile open source image viewer, GV should be a cross-platform application
    - We implement GV in Mono (cross-platform C#)
    - We use a minimalistic user interface to minimize platform-dependent code
    - We render the scatterplot with OpenGL

Application 1 → Introduction

- The **Global View** (GV) is designed as an overview module for a new Cinema viewer.
- We created GV as a desktop application with a minimal user interface.
- The UI consists of a textual input window and an OpenGL output window.



[1] … © 2017 Allen Institute for Cell Science. Interactive Plotting: http://www.allencell.org/interactive-plotting.html
[2] … The full source code is available on GitHub: https://github.com/RcSepp/GlobalView.js
* … After the writing of this thesis we decided to rename the library from “GlobalView.js” to “ExaPlot”.

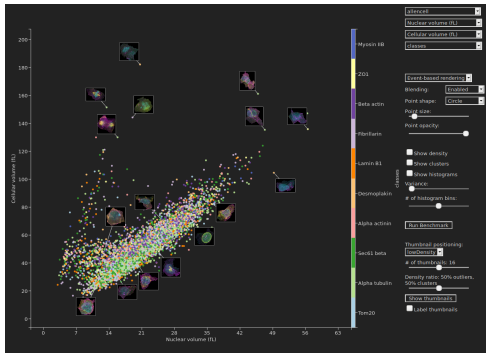
Application 1 → MPAS Dataset

- The MPAS dataset is a Cinema[1] database of in-situ generated images from a worldwide simulation of oceanic currents (Okubo-Weiss[2]).
- The simulation was run using the Model for Prediction Across Scales (MPAS), a climate modeling framework developed by the Los Alamos National Laboratory in cooperation with the National Center for Atmospheric Research.
- The database was created using the ParaView Catalyst pipeline[3].
- Dataset size
  - 5400 rows (images)
  - 3 columns: theta (18 views), phi (10 views), time (30 frames)
  - Image dimensions: 1024x1024px
  - Total size: 4.6GB

[1] … © 2017 Los Alamos National Laboratory. Cinema Science: http://cinemascience.org/
[2] … Sean Williams, Mark Petersen, Peer-Timo Bremer, Matthew Hecht, Valerio Pascucci, James Ahrens, Mario Hlawitschka, and Bernd Hamann. Adaptive extraction and quantification of geophysical vortices. IEEE Transactions on Visualization and Computer Graphics, 17(12):2088–2095, 2011.
[3] … Utkarsh Ayachit, Andrew Bauer, Berk Geveci, Patrick O'Leary, Kenneth Moreland, Nathan Fabian, and Jeffrey Mauldin. Paraview catalyst: Enabling in situ data analysis and visualization. In Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization, 25–29, 2015.

Application 2 → Outline

- Motivation
- Application 1: Global View
- Application 2: Interactive Cell Plot
  - Requirements
  - Introduction
  - Cell Dataset
  - Efficient Point Rendering
- Visual Mapping
- Density Maps
- Labeling
- Evaluation



Application 2 → Requirements

- Requirements
  - Deployment as a web application
    - ICP is a successor to the Interactive Plotting web application by the Allen Institute for Cell Science
    - By benchmarking a synthetic dataset of 100x the size of the current cell dataset, we prove that browser based rendering is sufficient for a future-proof web viewer
  - Annotation of images with thumbnails
  - Performance Scalability

Application 2 → Requirements

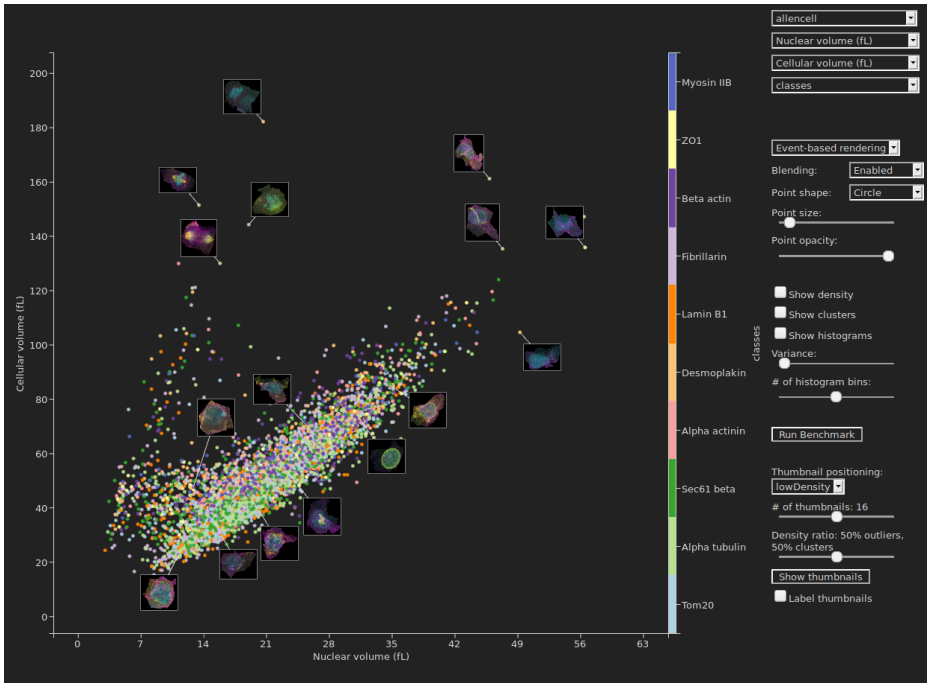
- Requirements
  - Deployment as a web application
  - Annotation of images with thumbnails
    - Information scalability of web applications is limited by the small screens of mobile web viewers.
    - As a result, we render cells as points, not images
    - We communicate cell images by annotating them with thumbnails
  - Performance Scalability

Application 2 → Requirements

- Requirements
  - Deployment as a web application
  - Annotation of images with thumbnails
  - Performance Scalability
    - Web application performance is limited by the smaller video memory and lower performance of mobile devices and by the execution speed limitations of JavaScript over compiled languages.
    - We use WebGL to achieve the highest possible rendering speed.
    - We minimize the video-memory footprint, by storing the dataset as a single continuous buffer.

Application 2 → Introduction

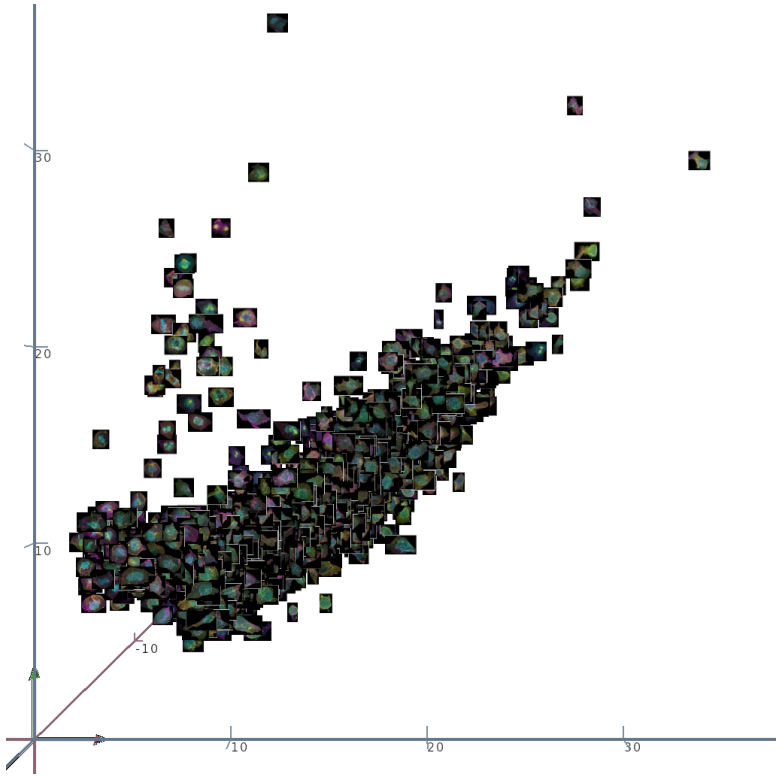
- The **Interactive Cell Plot** (ICP) is designed to replace the current Interactive Plotting application[1] on the website of the Allen Institute.
- We created ICP along with a JavaScript library[2] for efficiently rendering scatterplots of large image datasets on a web page*.
- For the purpose of development, we implement a sandbox interface:



[1] … © 2017 Allen Institute for Cell Science. Interactive Plotting: http://www.allencell.org/interactive-plotting.html
[2] … The full source code is available on GitHub: https://github.com/RcSepp/GlobalView.js
* … After the writing of this thesis we decided to rename the library from “GlobalView.js” to “ExaPlot”.

Application 2 → Cell Dataset

- Version 1.5 of the cell dataset of the Allen Institute for Cell Science[1].
- Cell images and features from the Wild Type C (WTC) human induced pluripotent stem cell line, produced by Bruce Conklin[2].
- Dataset size
  - 6077 rows (cells)
  - 7 columns (6 cell properties and the cell class)
  - 1 image per cell (width, height ≤ 128px; total size = 100MB)



[1] … Brock Roberts, Amanda Haupt, Andrew Tucker, Tanya Grancharova, Joy Arakaki, Margaret A. Fuqua, Angelique Nelson, Caroline Hookway, Susan A. Ludmann, Irina M. Mueller, Ruian Yang, Alan R. Horwitz, Susanne M. Rafelski, and Ruwanthi N. Gunawardane. Systematic gene tagging using CRISPR/Cas9 in human stem cells to illuminate cell organization. bioRxiv, 2017.

[2] … Faith R. Kreitzer, Nathan Salomonis, Alice Sheehan, Miller Huang, Jason S. Park, Matthew J. Spindler, Paweena Lizarraga, William A. Weiss, Po-Lin So, and Bruce R. Conklin. A robust method to derive functional neural crest cells from human pluripotent stem cells. American Journal of Stem Cells, 2(2):119, 2013.

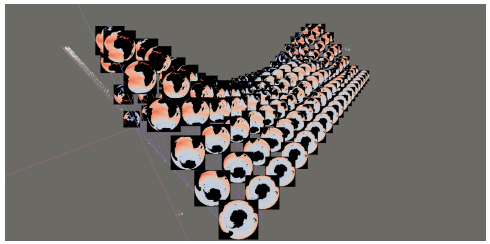
Application 2 → Fast Point Rendering

universität wien

- Requirements
  - Efficient OpenGL code requires **minimal communication** with the graphics driver. This applies even more for WebGL.
  - Data should be stored in a **single**, **static**, **compact** buffer.
- Limitations of WebGL*
  - No geometry shader
  - No line width property
  - No support for 1D textures or floating-point textures
- Implications
  - Storing the data table in memory as one continuous buffer
  - Compiling shaders on-the-fly
  - No hardware acceleration for density maps

* … Limitations in comparison with the current OpenGL standard; Only limitations relevant to this thesis.

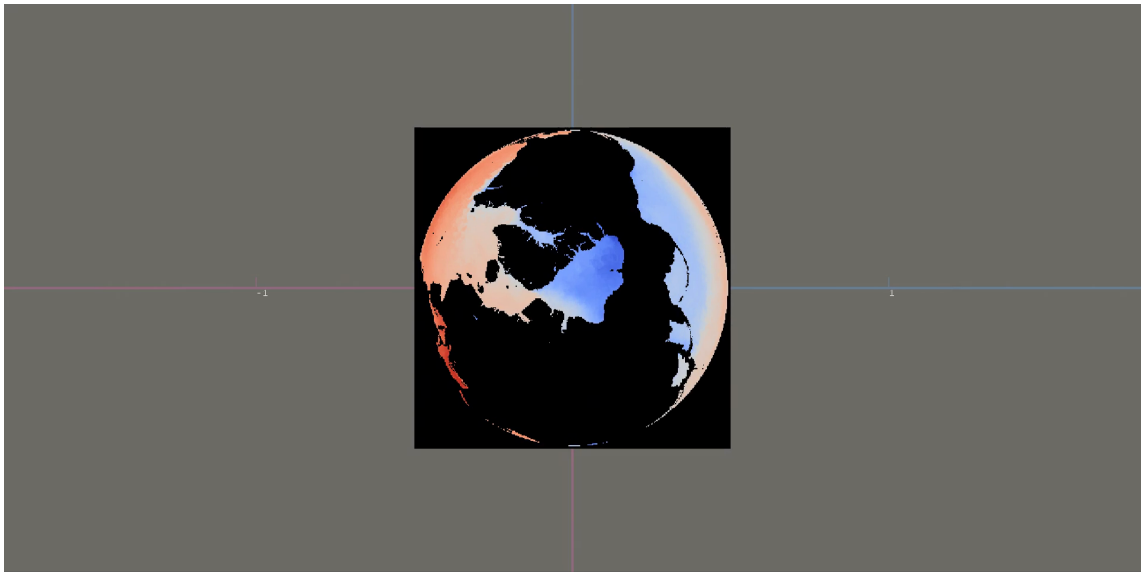
Visual Mapping → Outline



- Motivation
- Application 1: Global View
- Application 2: Interactive Cell Plot
- Visual Mapping
  - Introduction
  - View Mappings
  - Infovis Mappings
- Density Maps
- Labeling
- Evaluation

Visual Mapping → Introduction

- In the realm of multidimensional scatterplots a visual mapping is defined as a variable-to-axes mapping[1].
- In GV the initial visual mapping is an identity matrix (see below).
- The visual mapping matrix is transformed by formulating visual mapping transformations with an SQL-like scripting language.

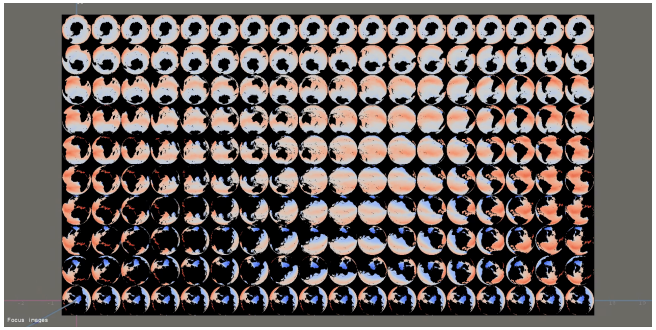


Identity Mapping

[1] … E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In IEEE Symposium on Information Visualization, pages 69–75. IEEE, 2000.

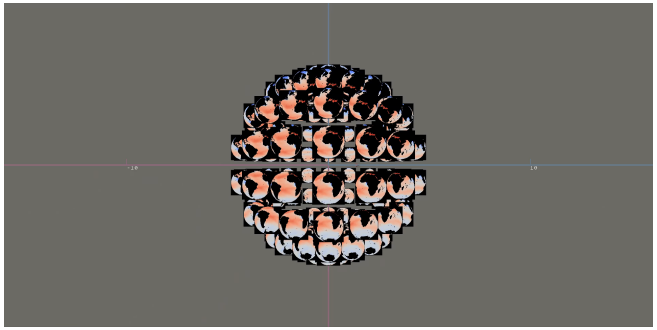
Visual Mapping → View Mappings

- We compare 3 visual mappings to encode the dimensions **theta** and **phi**.



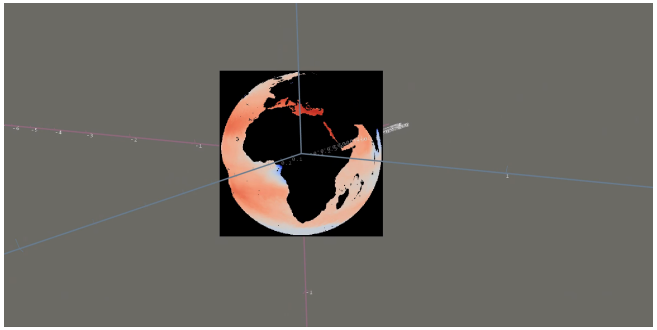
- Cartesian Mapping

  Theta is mapped to x
  Phi is mapped to y



- Spherical Mapping

  Every image is drawn at the point from which the view has been rendered in the simulation.



- Observer Mapping

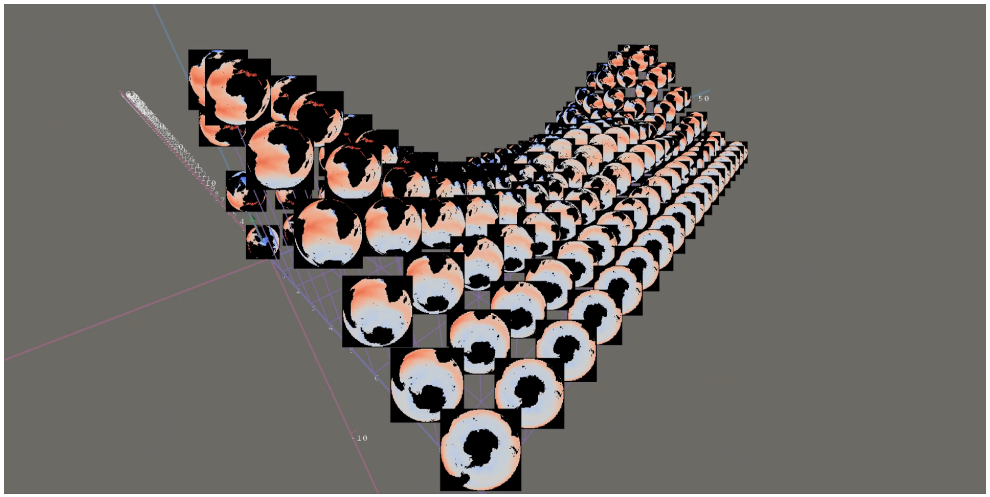
  The displayed view is the view whose geo. coordinates most closely resemble the viewing angle within GV.

Least intuitive.....Most intuitive

Showing all spatial views..... Showing local neighborhood..... Showing 1 view at a time

Visual Mapping → Infovis Mappings

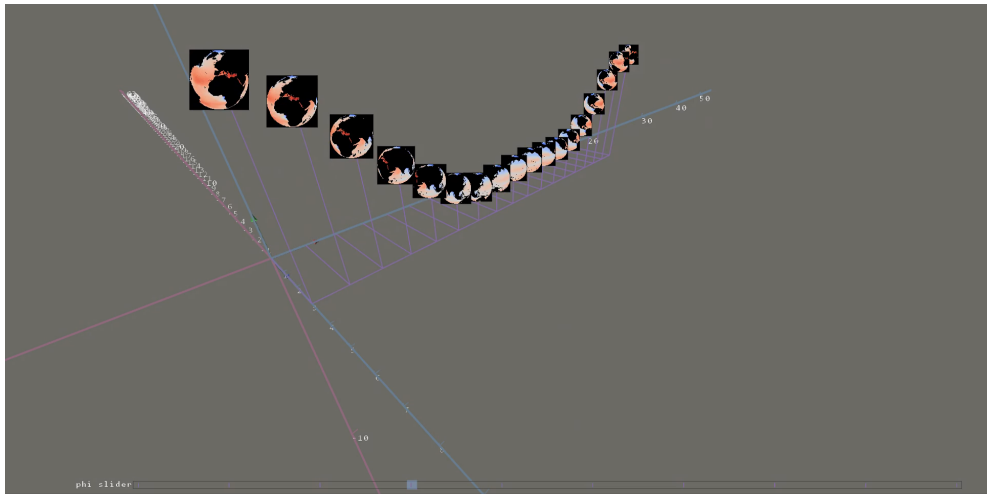
- The third dimension of a Cartesian mapping can be used to display meta data. The mappings below display average salinity.



- Plot Mapping

  The plot mapping places images like data points in a plot.

  To aid cognitive location of images in 3D space, the plot mapping draws purple lines between images and axes.



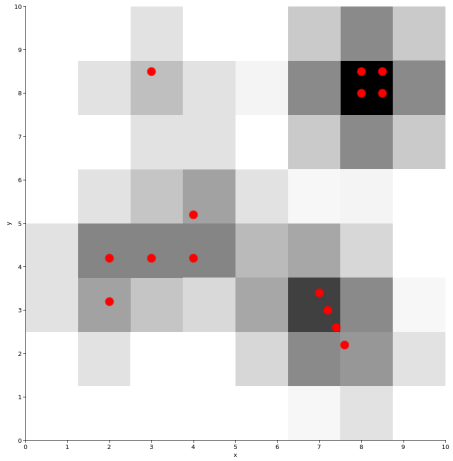
- Slider Mapping

  The slider mapping hides all but one slice of the dataset.

  The slice is selected interactively through a slider control at the bottom of the screen.

Density Maps → Outline

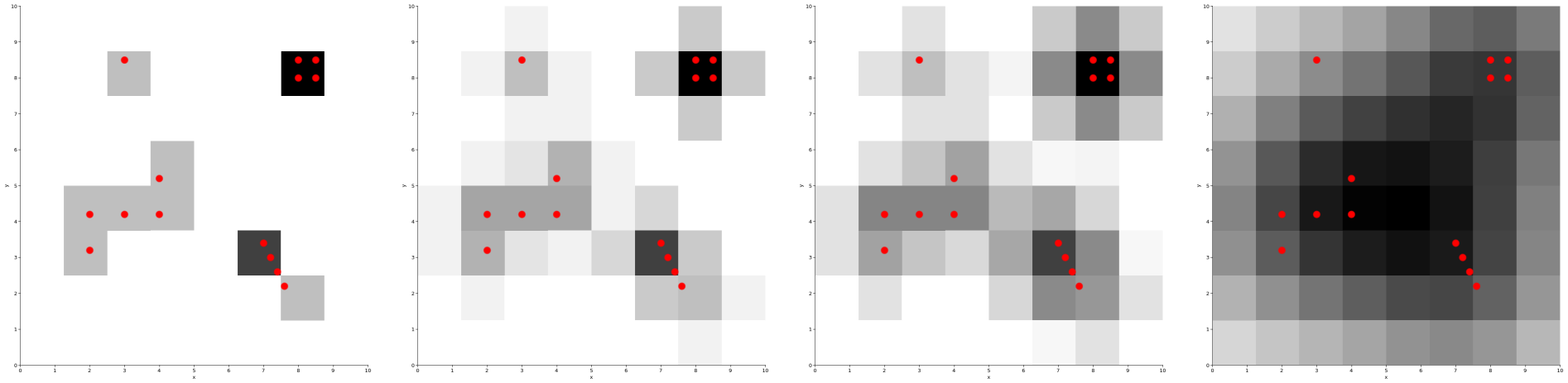
- Motivation
- Application 1: Global View
- Application 2: Interactive Cell Plot
- Visual Mapping
- Density Maps
  - Introduction
  - Use Cases
  - Runtime Estimation
- Labeling
- Evaluation



Density Maps → Introduction

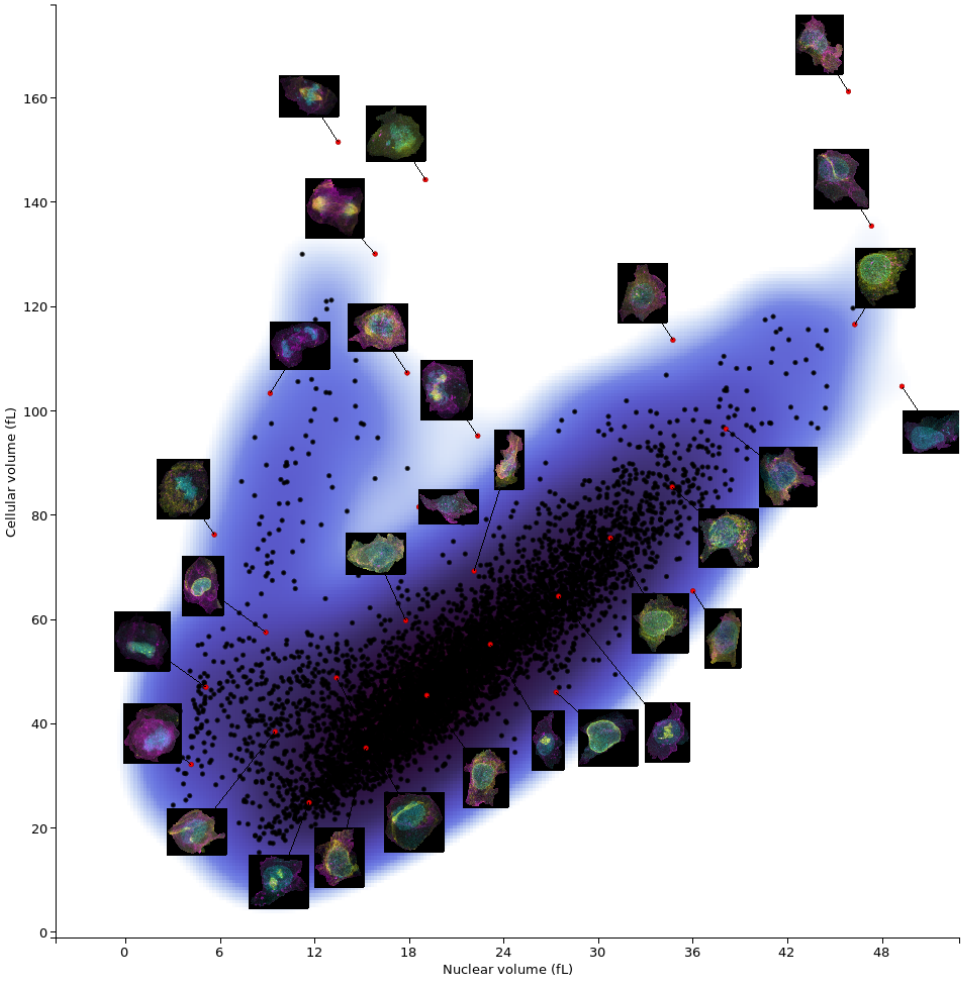
- A nonparametric way to estimate the probability density function of a random variable is known as the kernel density estimate (KDE).
- Definitions:
  i. A density map is a discretized KDE.
  ii. A density map is a histogram with non-zero variance.

- Example
  - 4 density maps of size 8x8 with increasing variance

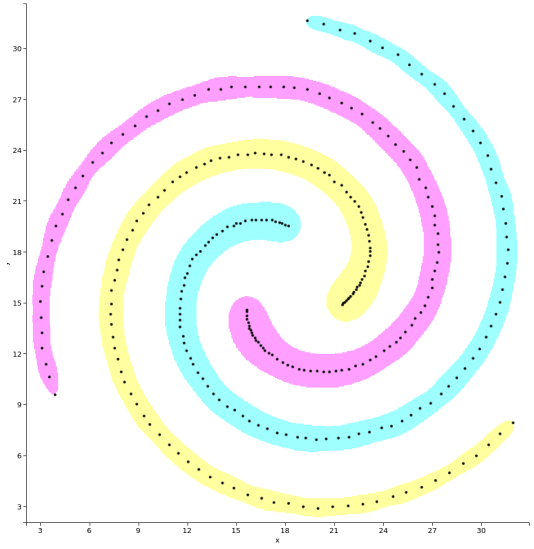


Density Maps → Use Cases

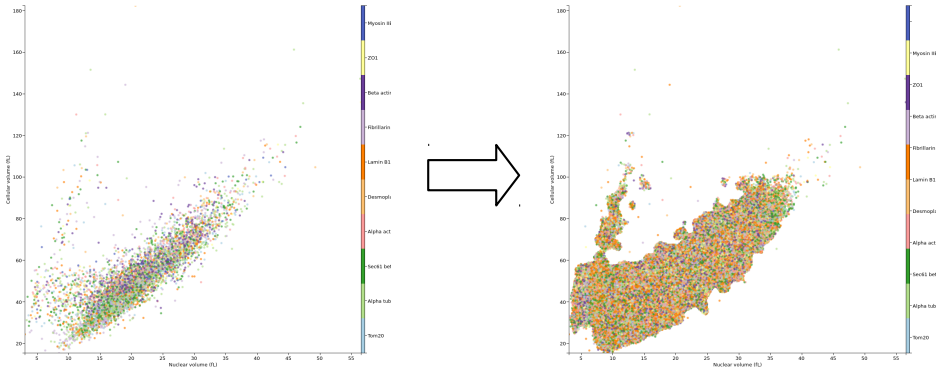
- Characteristic point detection
- Density based labeling



- Sample generation

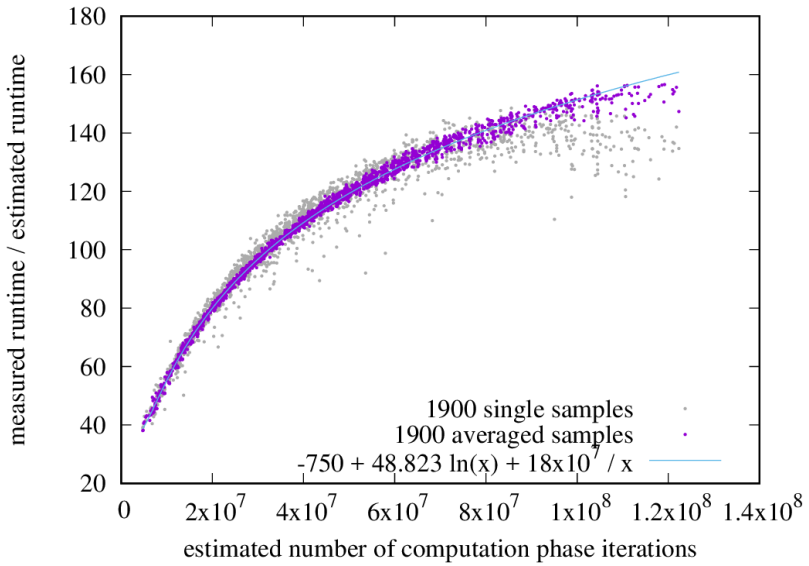


- Clustering



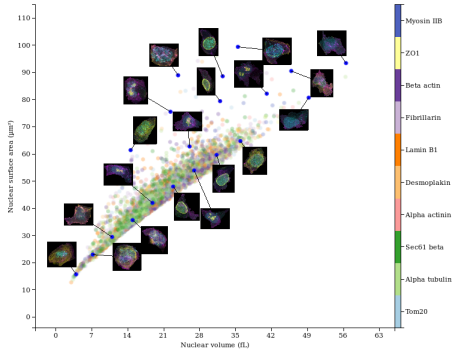
Density Maps → Runtime Estimation

- Computing a density map of size *s* by *s* pixels takes $O(n + s^4)$
- Exact runtime depends on distribution of points
- Solution: Estimate expected runtime in $O(s^2)$ and reduce *s* if the expected runtime is too high.
- Computation runtime ≈ estimation runtime * *f(*estimated # of iterations*)*
- Function *f()* is found empirically:



Labeling → Outline

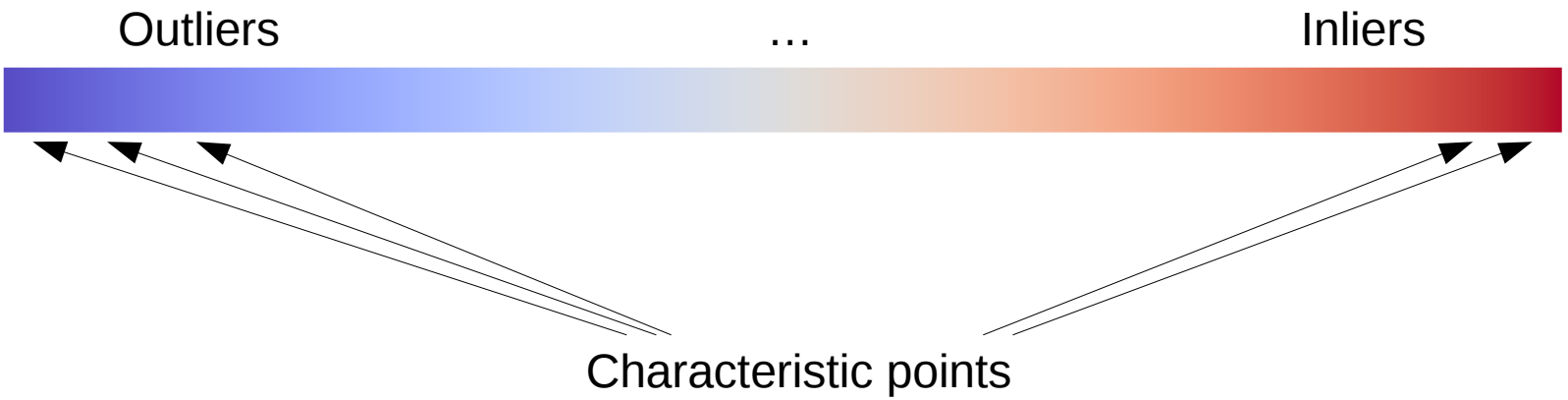
- Motivation
- Application 1: Global View
- Application 2: Interactive Cell Plot
- Visual Mapping
- Density Maps
- Labeling
  - Thumbnail Selection
  - Thumbnail Placement
- Evaluation



Labeling → Thumbnail Selection

- How many thumbnails?
  - Subjective
  - Data dependent
  - Answered in a user study

- Which points should be annotated?
  - Characteristic points = outliers + inliers

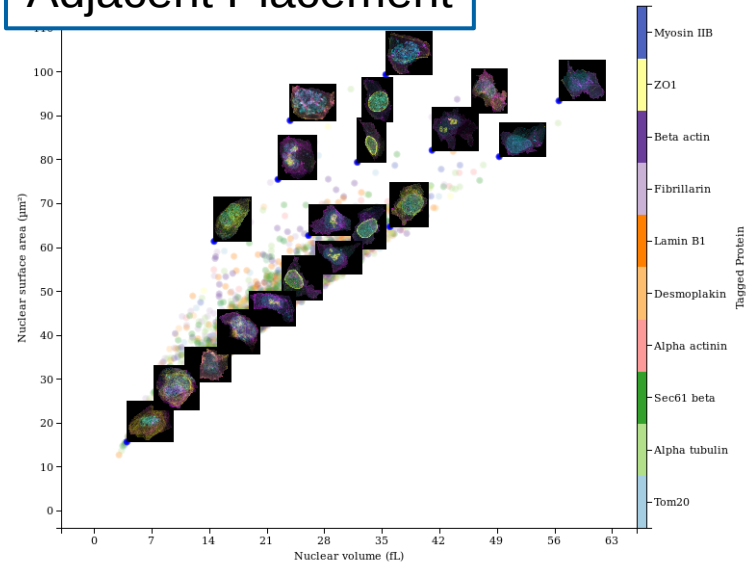


Labeling → Thumbnail Placement

- A thumbnail should be placed near to its site without occluding important parts of the plot.
- Trade-off: **Plot occlusion vs. label distance**

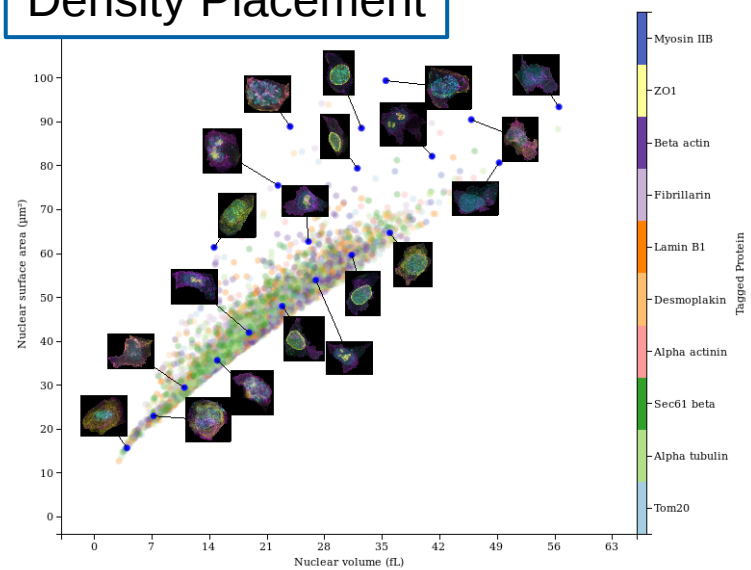
- We implement 5 different strategies:

Adjacent Placement

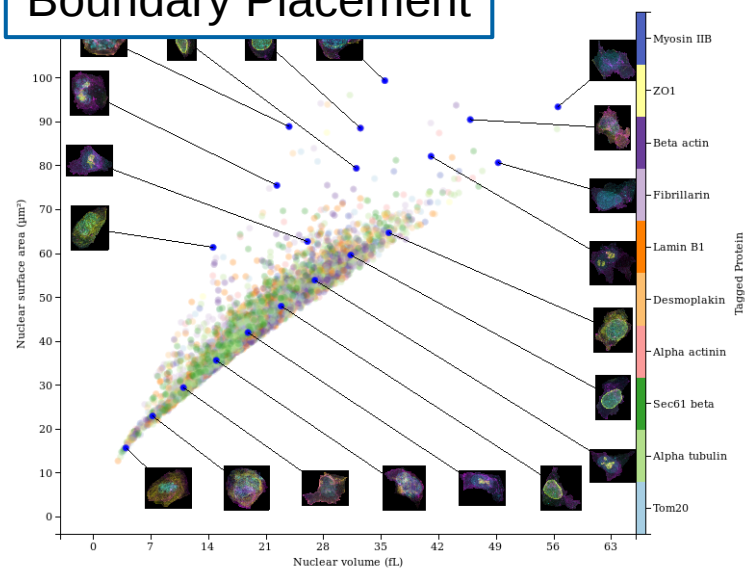


Labeling → Thumbnail Placement

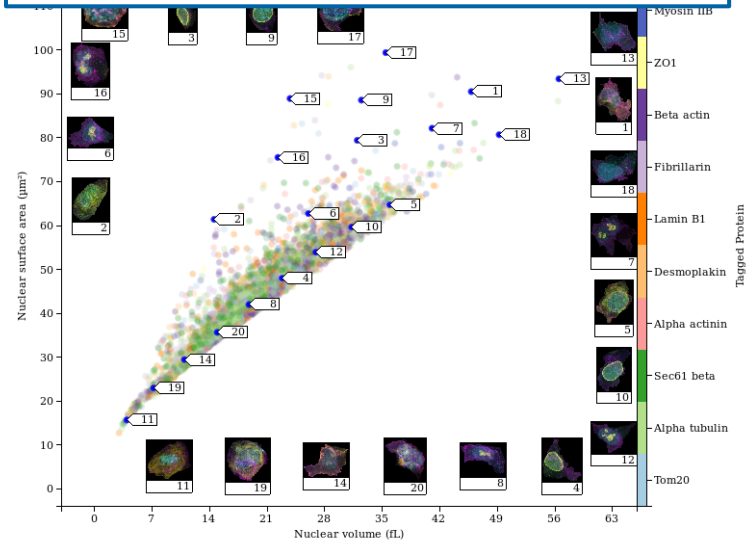
Density Placement



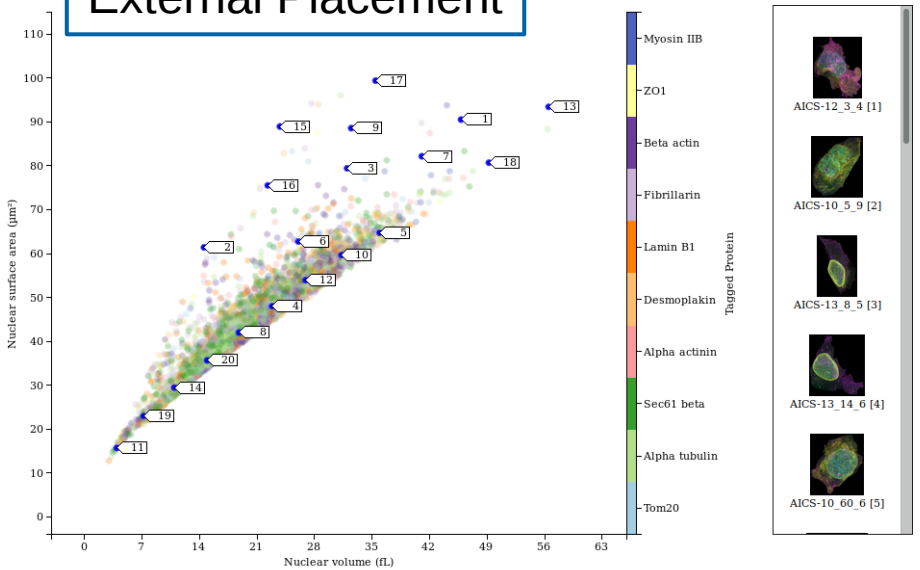
Boundary Placement



Numbered Boundary Placement

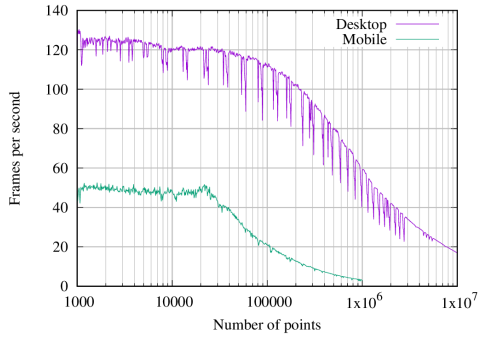


External Placement



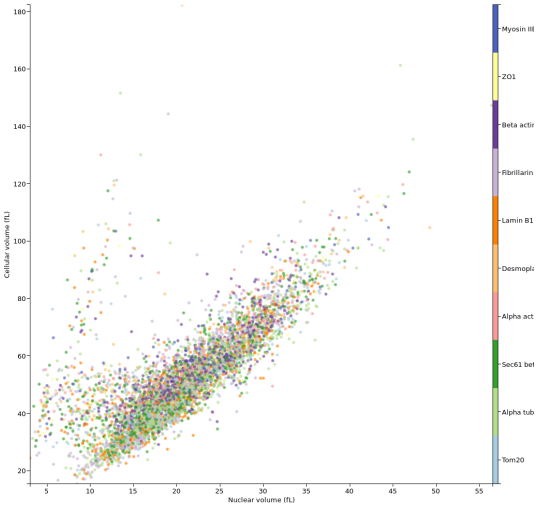
Evaluation → Outline

- Motivation
- Application 1: Global View
- Application 2: Interactive Cell Plot
- Visual Mapping
- Density Maps
- Labeling
- Evaluation
  - Scalability
  - Performance
  - User Study

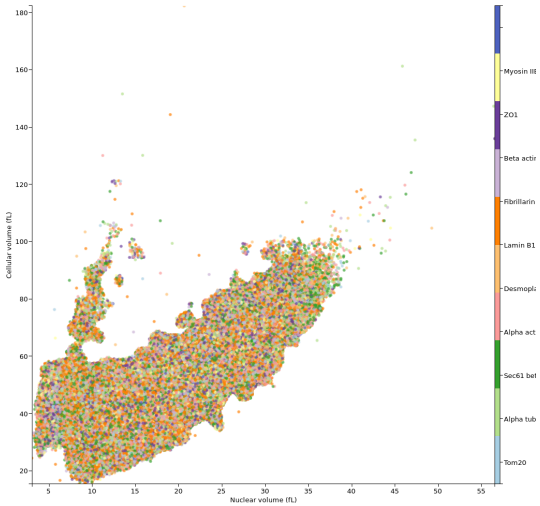


Evaluation → Scalability

- Cell dataset scalability
  - One key requirement of ICP is to ensure that the cell dataset can still be rendered as many more cells are added in the future.
  - We create identically distributed synthetic datasets of up to 100x the number of points (607,700 cells) by rejection sampling the density map.
  - ICP achieves interactive frame rates on all tested datasets.



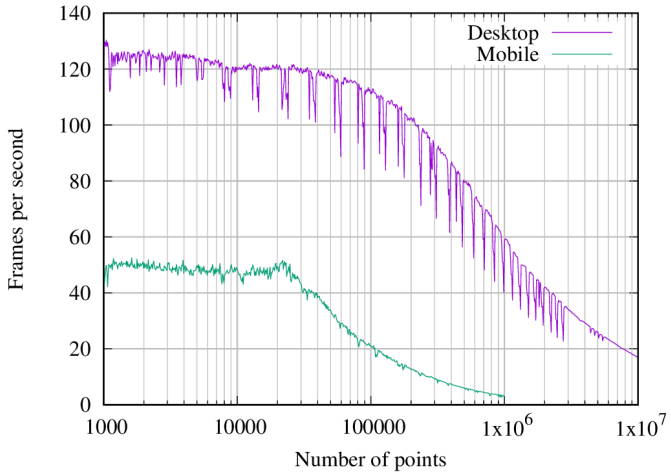
The cell dataset (6077 cells)



A synthetic dataset with 607,700 cells

Evaluation → Performance

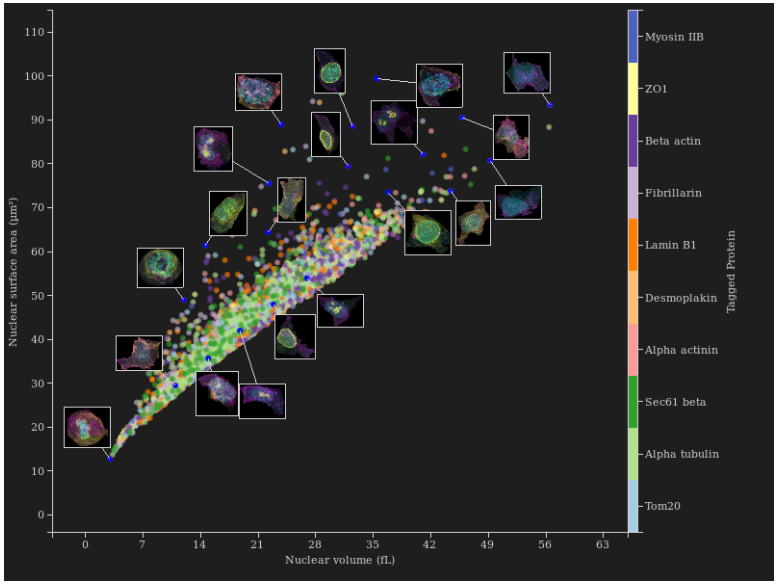
- Performance
  - We measure rendering performance by drawing uniformly distributed 2D points and measuring how many frames can be rendered in 10 seconds.
  - Higher-dimensional points should yield similar results, because only visible dimensions are passed to our on-the-fly compiled vertex shader.

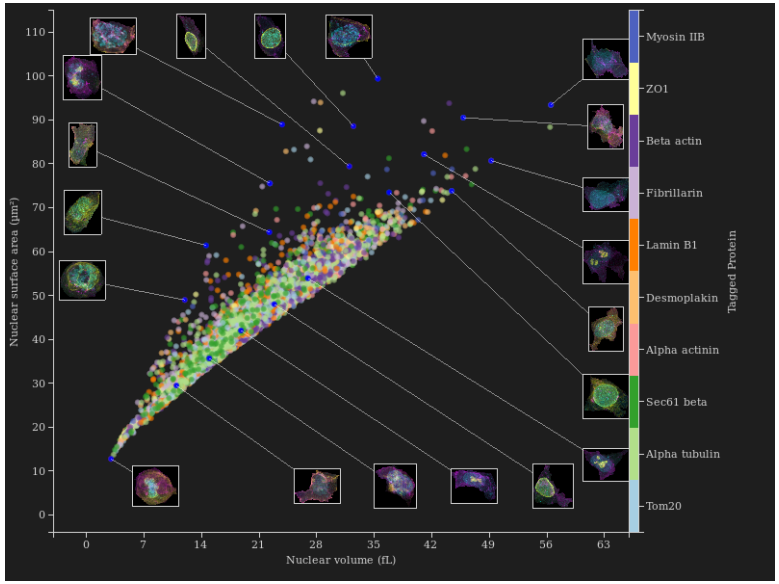


Evaluation → User Study

- Qualitative user study with 6 participants
- Questions
  - Visual parameters
    - Color scheme
    - Point size/shape
    - Should we visualize point density?
  - Labeling
    - How many thumbnails?
    - Which points should be annotated?
    - Where should thumbnails be placed?
  - Additional questions
    - Which parameters should be editable?
    - Do we need to allow manual thumbnail placement?

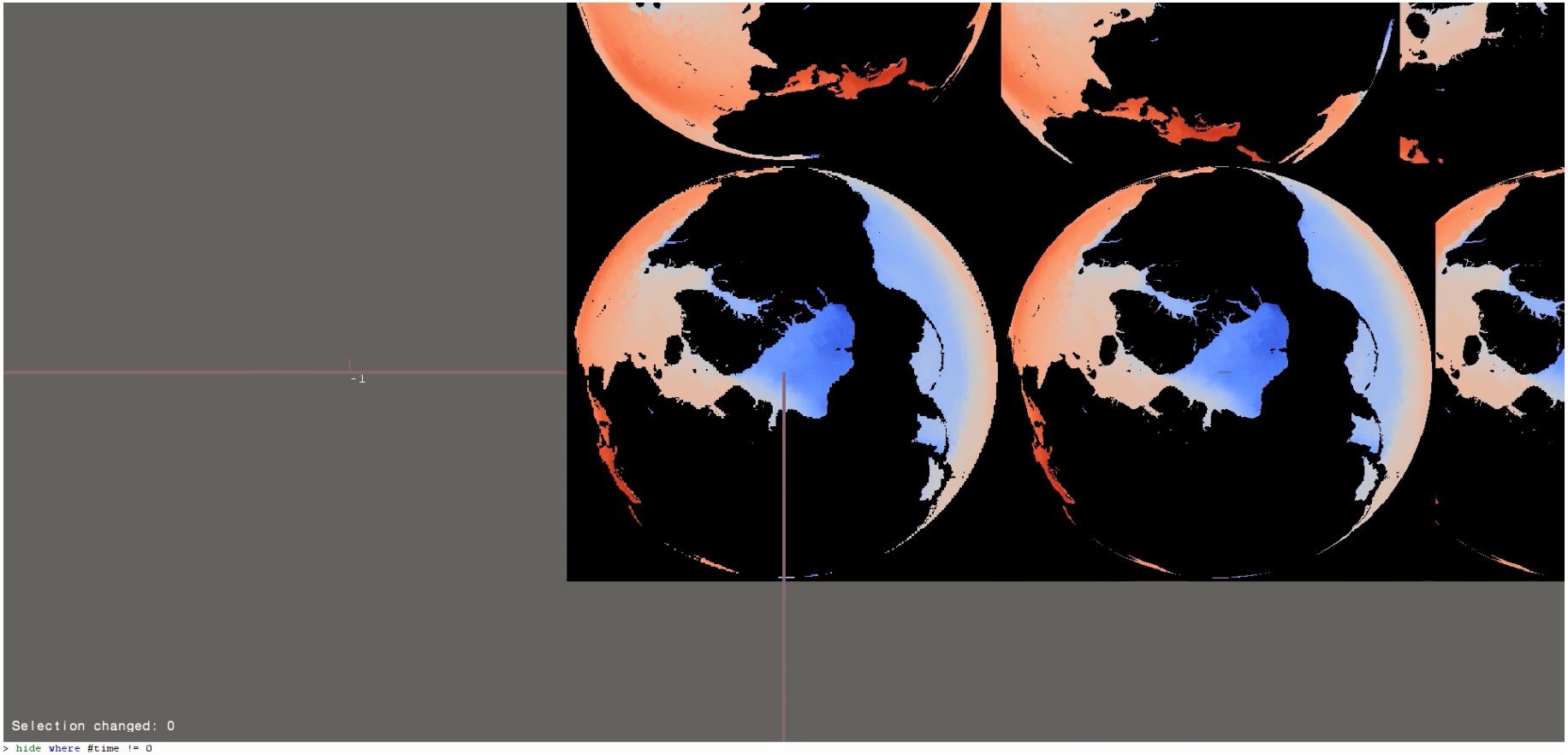
Evaluation → User Study Results



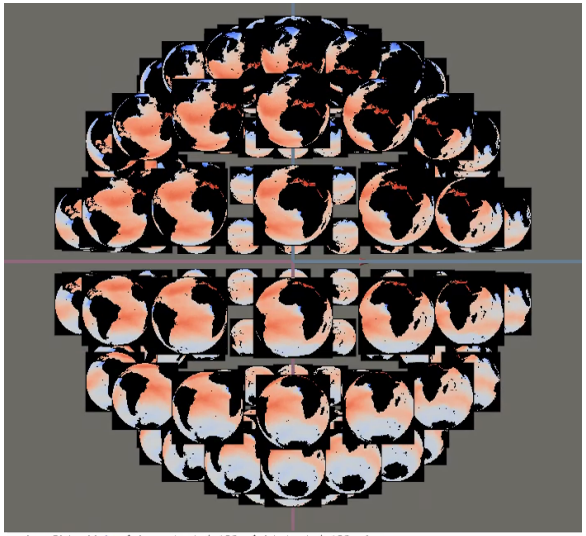


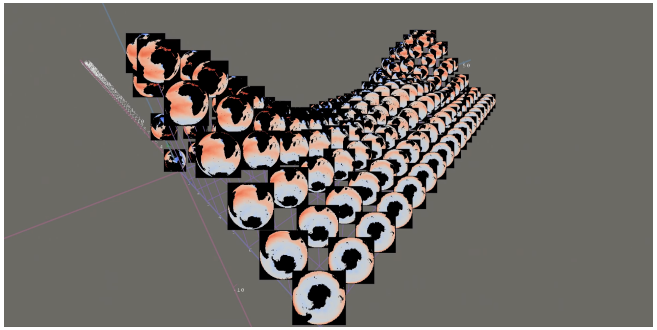
- How many points should be annotated with thumbnails? 20
- Which points should be annotated with thumbnails? 75% outliers
- Where should the thumbnails be placed? Density- or boundary placement
- Allow manual selection of thumbnails, but not manual repositioning.
- Expose controls for visualization parameters only on demand.

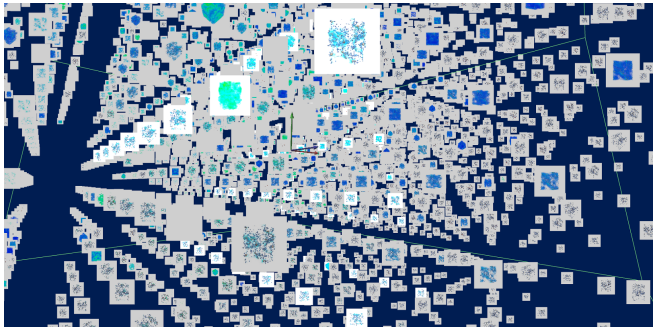
Questions

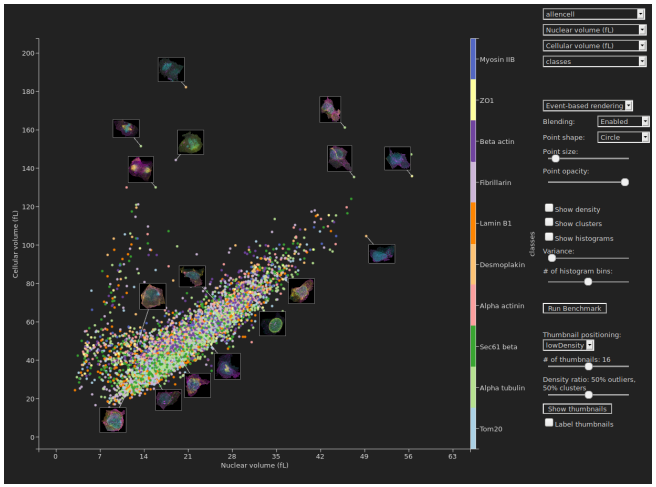

Selection changed: 0
> hide where #time != 0
> focus all
> star all by #theta, #phi
>

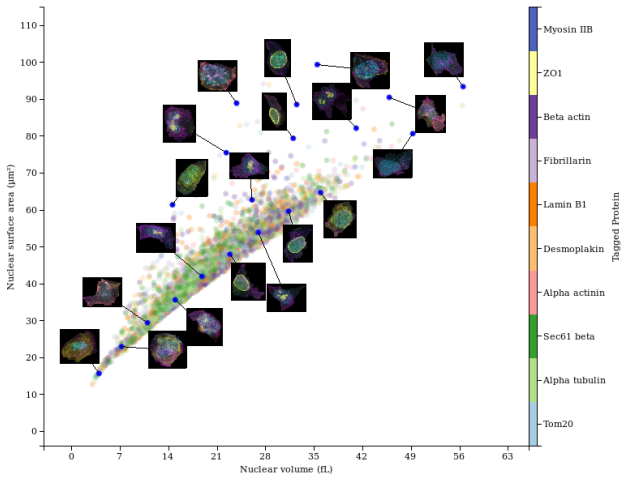
Questions

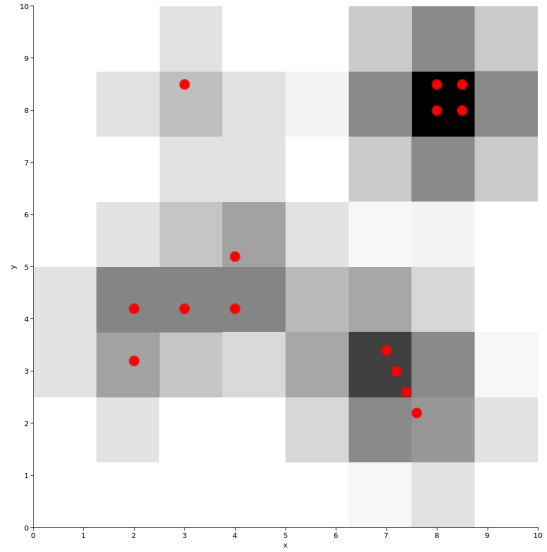

> thetaPhi all by $theta * pi / 180, $phi * pi / 180, 4
> focus all
>






allencell
Nuclear volume (fL)
Cellular volume (fL)
classes
Myosin IIB
ZO1
Beta actin
Fibrillarin
Lamin B1
Desmoplakin
Alpha actinin
Sec61 beta
Alpha tubulin
Tom20
Event-based rendering
Blending:
Enabled
Point shape:
Circle
Point size:
Point opacity:
Show density
Show clusters
Show histograms
Variance:
# of histogram bins:
Run Benchmark
Thumbnail positioning:
lowDensity
# of thumbnails: 16
Density ratio: 50% outliers, 50% clusters
Show thumbnails
Label thumbnails
Cellular volume (fL)
Nuclear volume (fL)


Nuclear surface area (µm²)
Nuclear volume (fL)
Myosin IIB
ZO1
Beta actin
Fibrillarin
Lamin B1
Desmoplakin
Alpha actinin
Sec61 beta
Alpha tubulin
Tom20
Tagged Protein


y
x