

### +/- Message

```
## Loading required package: knitr
```

### +/- R Code

# Quantitative Big Imaging

author: Kevin Mader date: 27 March 2014 width: 1440 height: 900 css: ../template.css transition: rotate  
ETHZ: 227-0966-00L

## Analysis of Complex Objects

### Course Outline

- 20th February - Introductory Lecture
- 27th February - Filtering and Image Enhancement (A. Kaestner)
- 6th March - Basic Segmentation, Discrete Binary Structures
- 13th March - Advanced Segmentation
- 20th March - Analyzing Single Objects
- 27th March - **Analyzing Complex Objects**
- 3rd April - Spatial Distribution
- 10th April - Statistics and Reproducibility
- 17th April - Dynamic Experiments
- 8th May - Big Data
- 15th May - Guest Lecture - In-Operando Imaging of Batteries (V. Wood)
- 22th May - Project Presentations

## Literature / Useful References

### Books

- Jean Claude, Morphometry with R
- Online (<http://link.springer.com/book/10.1007%2F978-0-387-77789-4>) through ETHZ
- Buy it (<http://www.amazon.com/Morphometrics-R-Use-Julien-Claude/dp/038777789X>)
- John C. Russ, "The Image Processing Handbook", (Boca Raton, CRC Press)

- Available online (<http://dx.doi.org/10.1201/9780203881095>) within domain ethz.ch (or proxy.ethz.ch / public VPN)
- 

## Papers / Sites

- Thickness
  - [1] Hildebrand, T., & Ruegsegger, P. (1997). A new method for the model-independent assessment of thickness in three-dimensional images. *Journal of Microscopy*, 185(1), 67–75. doi:10.1046/j.1365-2818.1997.1340694.x
- Curvature
  - <http://mathworld.wolfram.com/MeanCurvature.html>  
(<http://mathworld.wolfram.com/MeanCurvature.html>)
  - [2] "Computation of Surface Curvature from Range Images Using Geometrically Intrinsic Weights", T. Kurita and P. Boulanger, 1992.

## Previously on QBI ...

- Image Enhancement
  - Highlighting the contrast of interest in images
  - Minimizing Noise
- Segmentation
  - Understanding value histograms
  - Dealing with multi-valued data
- Automatic Methods
  - Hysteresis Method, K-Means Analysis
- Regions of Interest
  - Contouring
- Component Labeling
- Single Shape Analysis

## Outline

- Motivation (Why and How?)
- What are Distance Maps?
- Skeletons
  - Tortuosity
- What are thickness maps?
  - Thickness with Skeletons

- Watershed Segmentation
  - Connected Objects
- Curvature
  - Characteristic Shapes

# Learning Objectives

## Motivation (Why and How?)

- How do we measure distances between many objects?
- How can we extract topology of a structure?
- How can we measure sizes in complicated objects?
- How do we measure sizes relevant for diffusion or other local processes?
- How do we identify separate objects when they are connected?
- How do we investigate surfaces in more detail and their shape?
- How can we compare shape of complex objects when they grow?
  - Are there characteristic shape metrics?

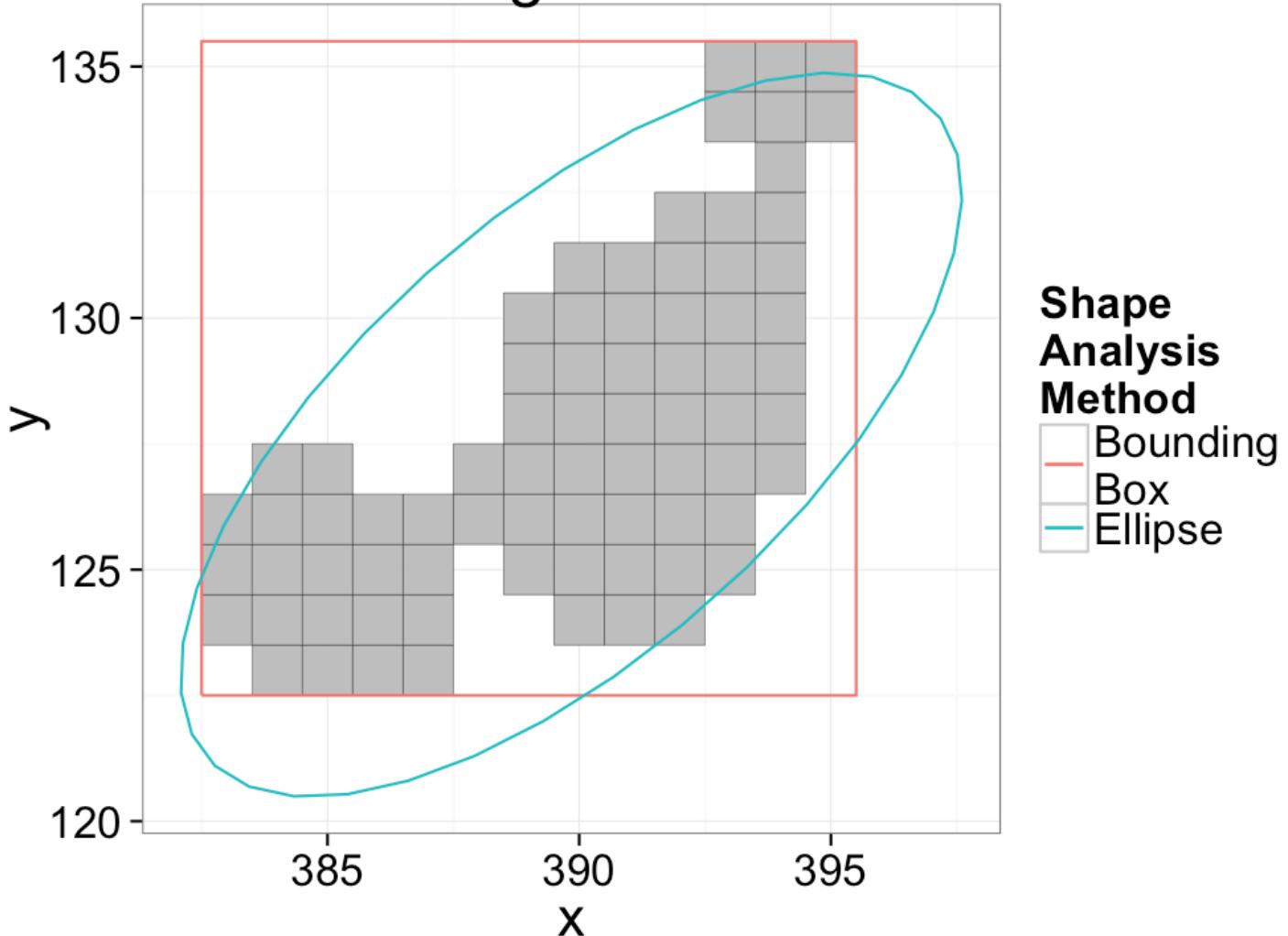
## What did we want in the first place

To simplify our data, but an ellipse model is **too** simple for many shapes

So while bounding box and ellipse-based models are useful for many objects and cells, they do a very poor job with the sample below.

+/- R Code

# Single Cell



## Why

- We assume an entity consists of connected pixels (wrong)
- We assume the objects are well modeled by an ellipse (also wrong)

## What to do?

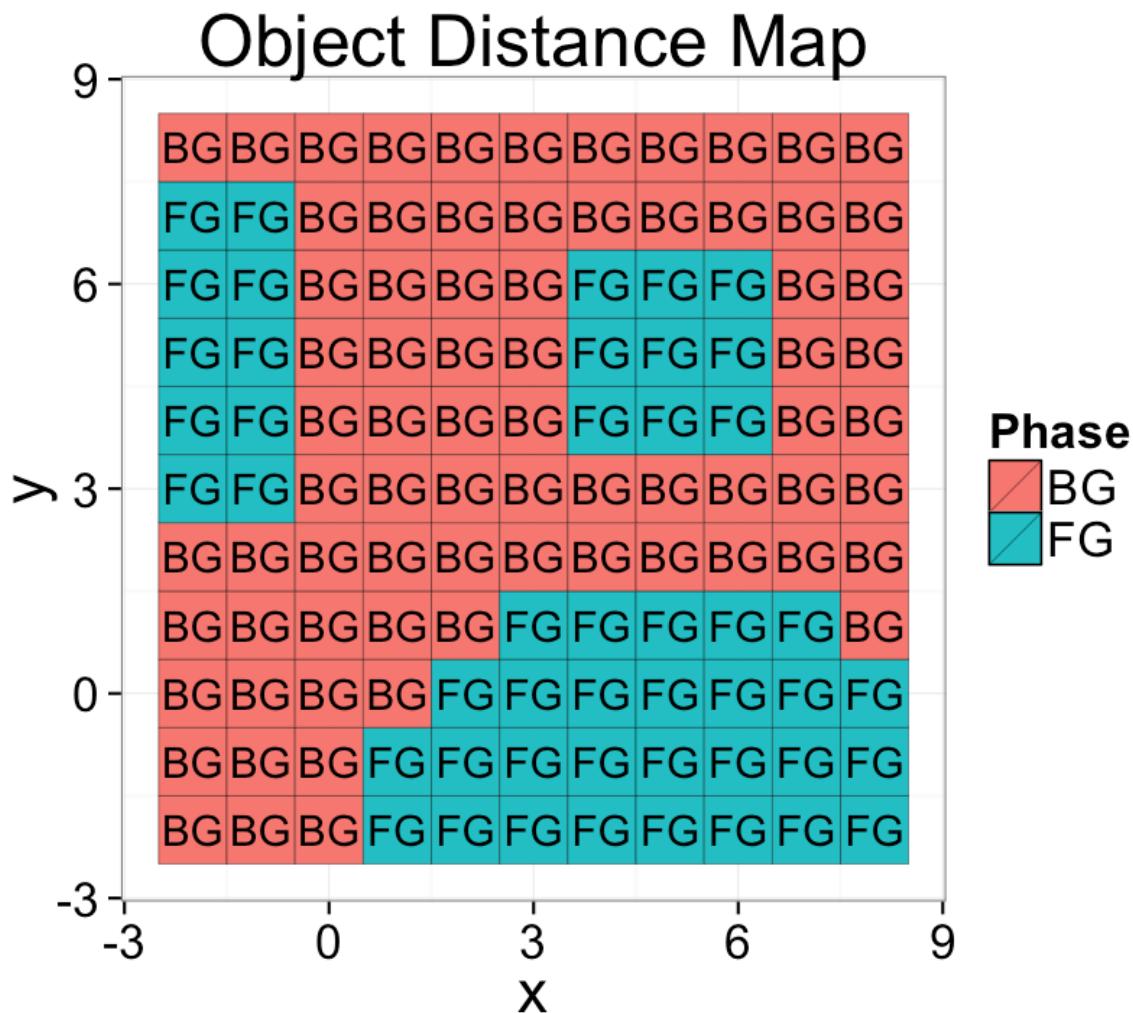
- Is it 3 connected objects which should all be analyzed separately?
- If we could **divide it**, we could then analyze each part as an ellipse

- Is it one network of objects and we want to know about the constrictions?
- Is it a cell or organelle with docking sites for cell?
- Neither extents nor anisotropy are very meaningful, we need a **more specific metric** which can characterize

# Distance Maps: What are they

+/- R Code

A map (or image) of distances. Each point in the map is the distance that point is from a given feature of interest (surface of an object, ROI, center of object, etc)



+/- R Code

## Definition

If we start with an image as a collection of points divided into two categories

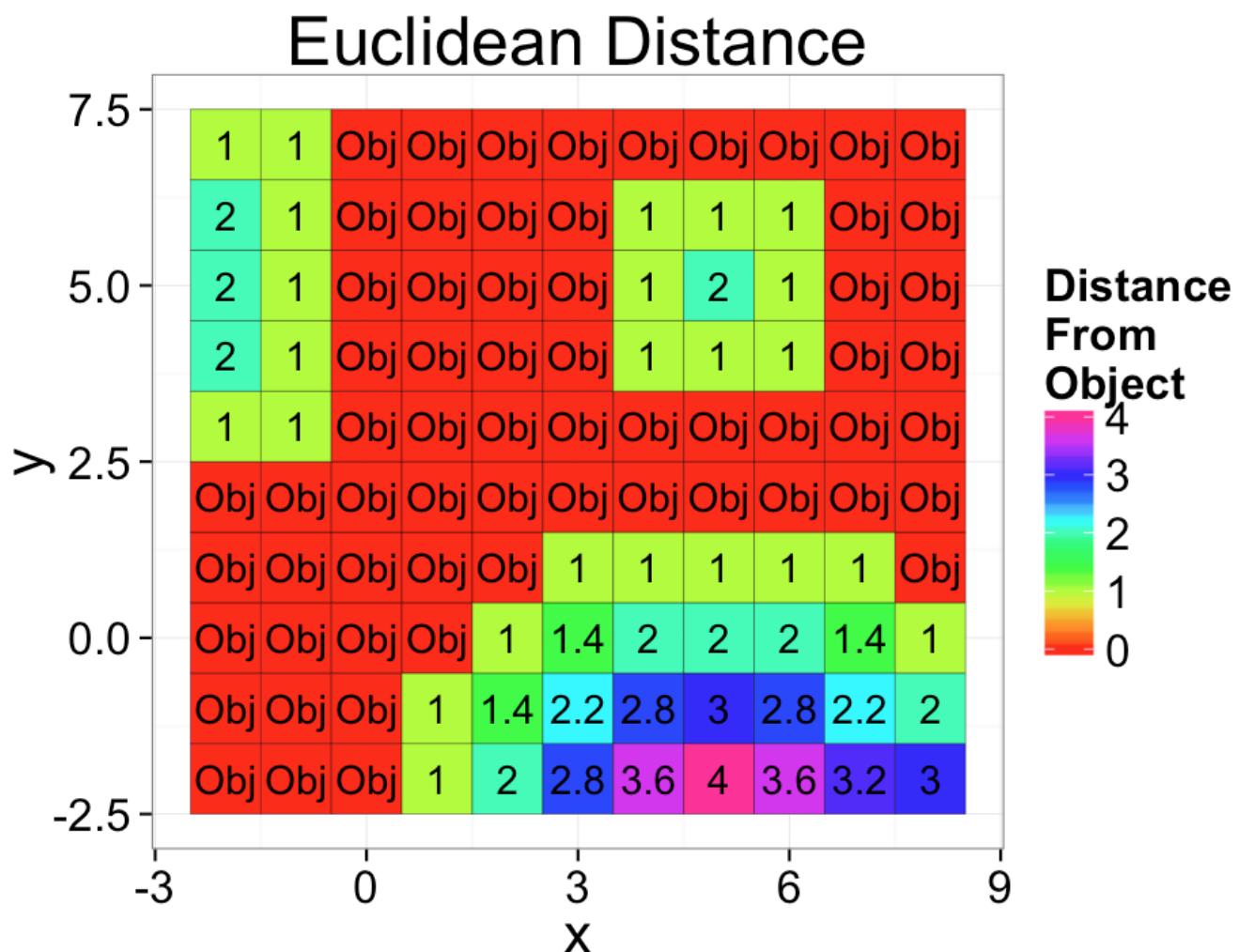
- $Im(x, y) = \{\text{Foreground, Background}\}$
- We can define a distance map operator ( $dist$ ) that transforms the image into a distance map

$$dist(\vec{x}) = \min(\|\vec{x} - \vec{y}\| \forall \vec{y} \in \text{Background})$$

We will use Euclidean distance  $\|\vec{x} - \vec{y}\|$  for this class but there are other metrics which make sense when dealing with other types of data like Manhattan/City-block or weighted metrics.

## Distance Maps: Types

Using this rule a distance map can be made for the euclidean metric

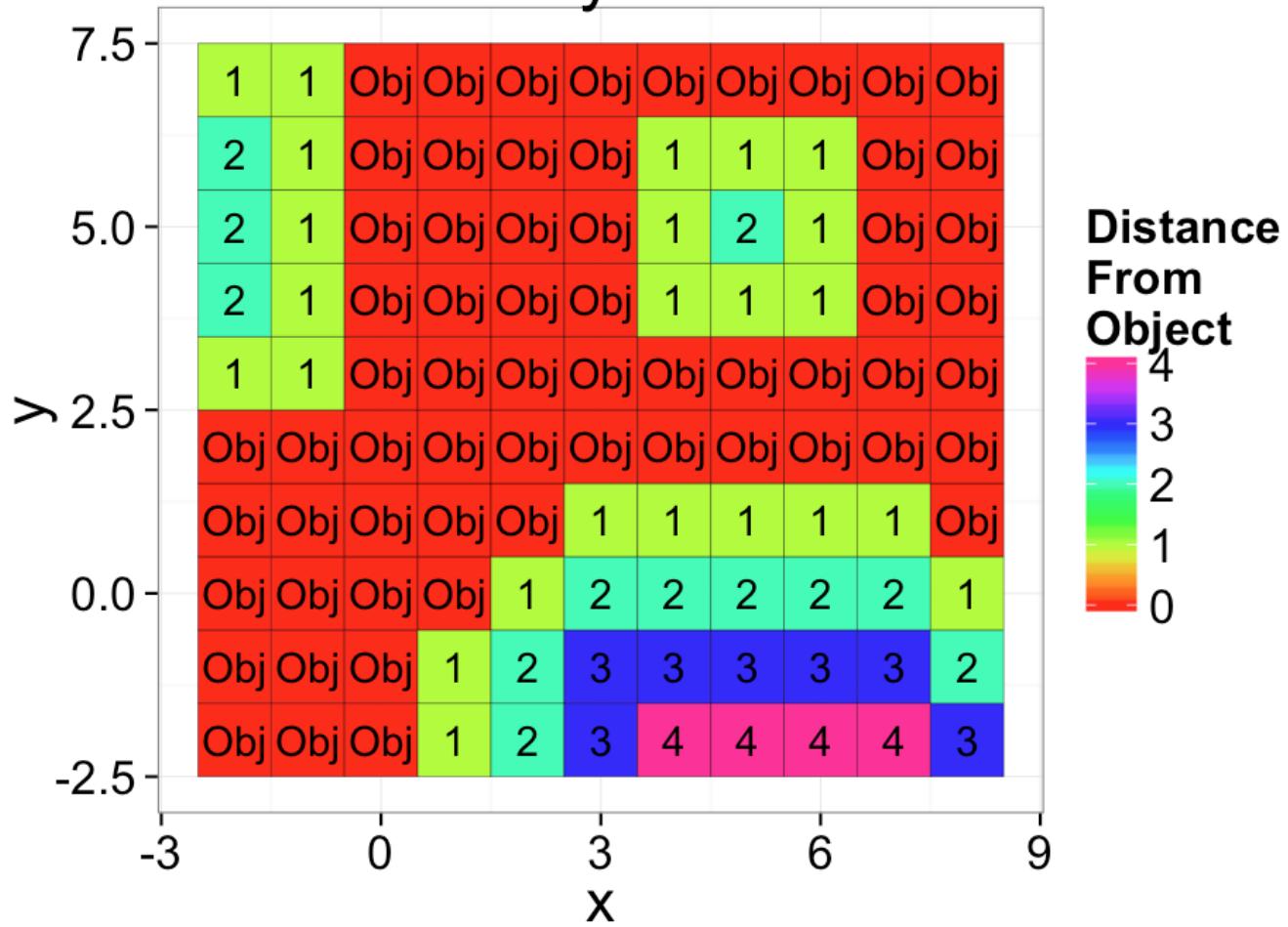


+/- R Code

Similarly the Manhattan or city block distance metric can be used where the distance is defined as

$$\sum_i |\vec{x} - \vec{y}|_i$$

## Manhattan / City Block Distance



+/- R Code

## Distance Maps: Precaution

type: alert

The distance map is one of the critical points where the resolution of the imaging system is important.

- We measure distances computationally in pixels or voxels
- but for them to have a meaning physically they must be converted
  - Isotropic imaging ( $1 \mu\text{m} \times 1 \mu\text{m} \times 1 \mu\text{m}$ ) is **fine**

## Anisotropic

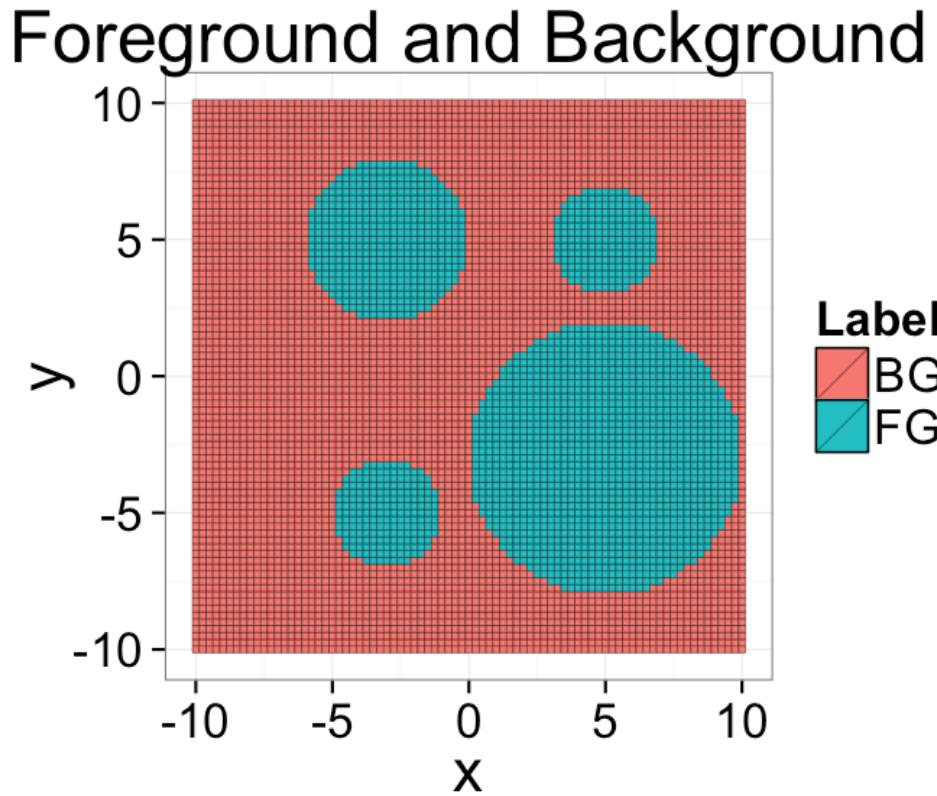
Ideally

- as part of filtering, resample and convert to an isotropic scale.

If that is not possible

- custom distance map algorithms which use the side-lengths of the voxels to calculate distance rather than assuming  $1 \times 1 \times 1$

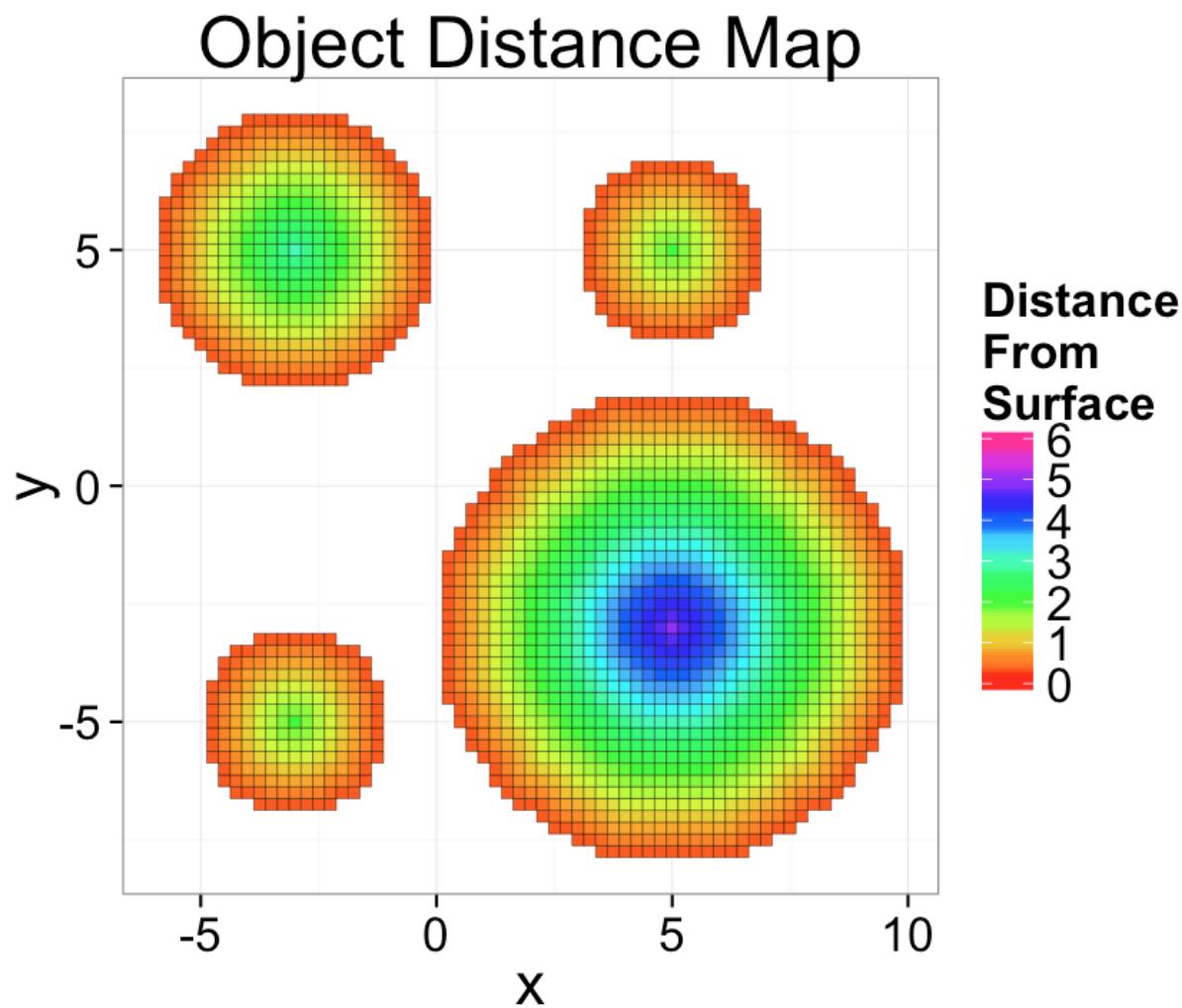
## Distance Maps



+/- R Code

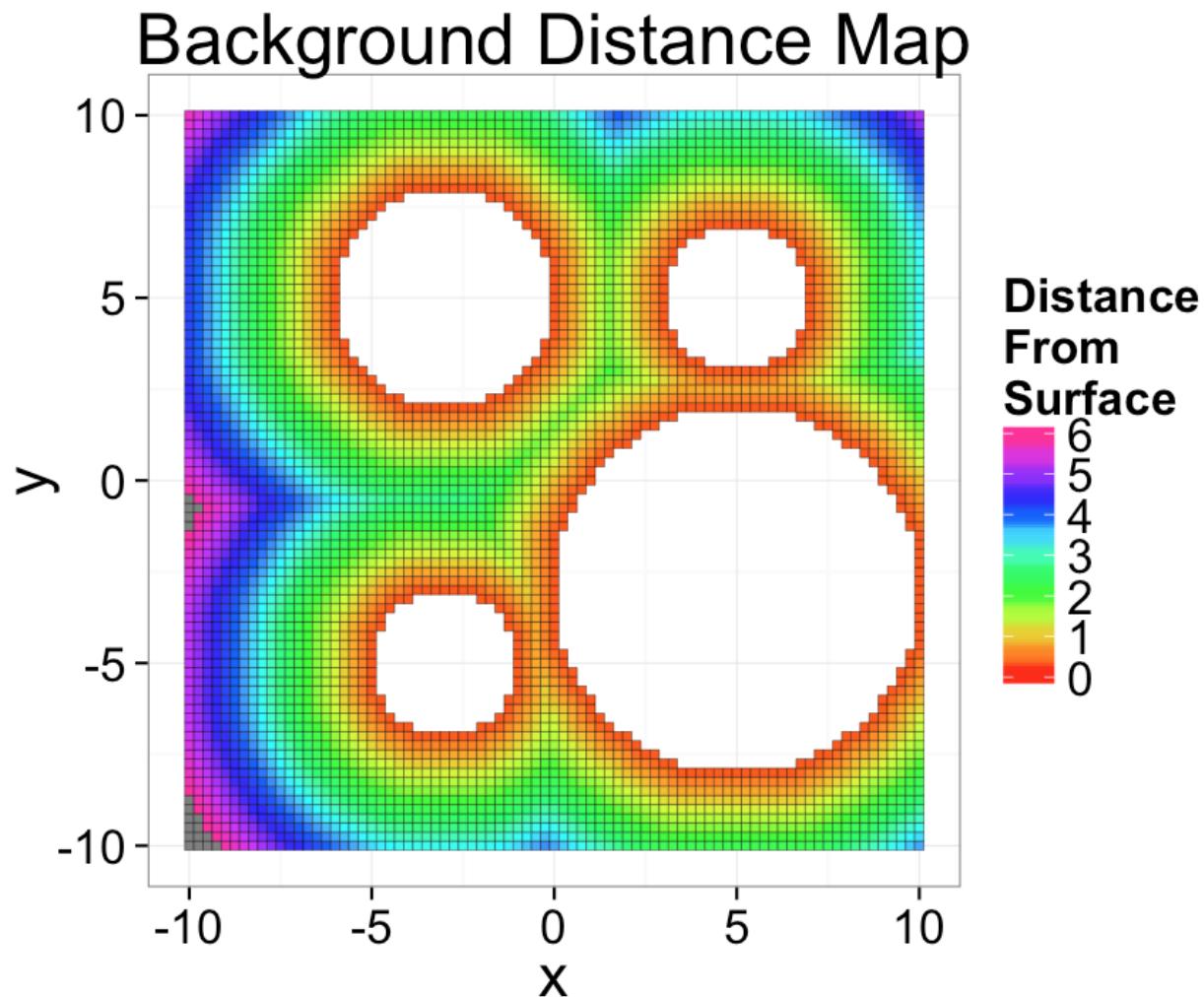
We can create 2 distance maps

1. Foreground → Background
  - Information about the objects size and interior
2. Background → Foreground
  - Information about the distance / space between objects



+/- R Code

+/- R Code



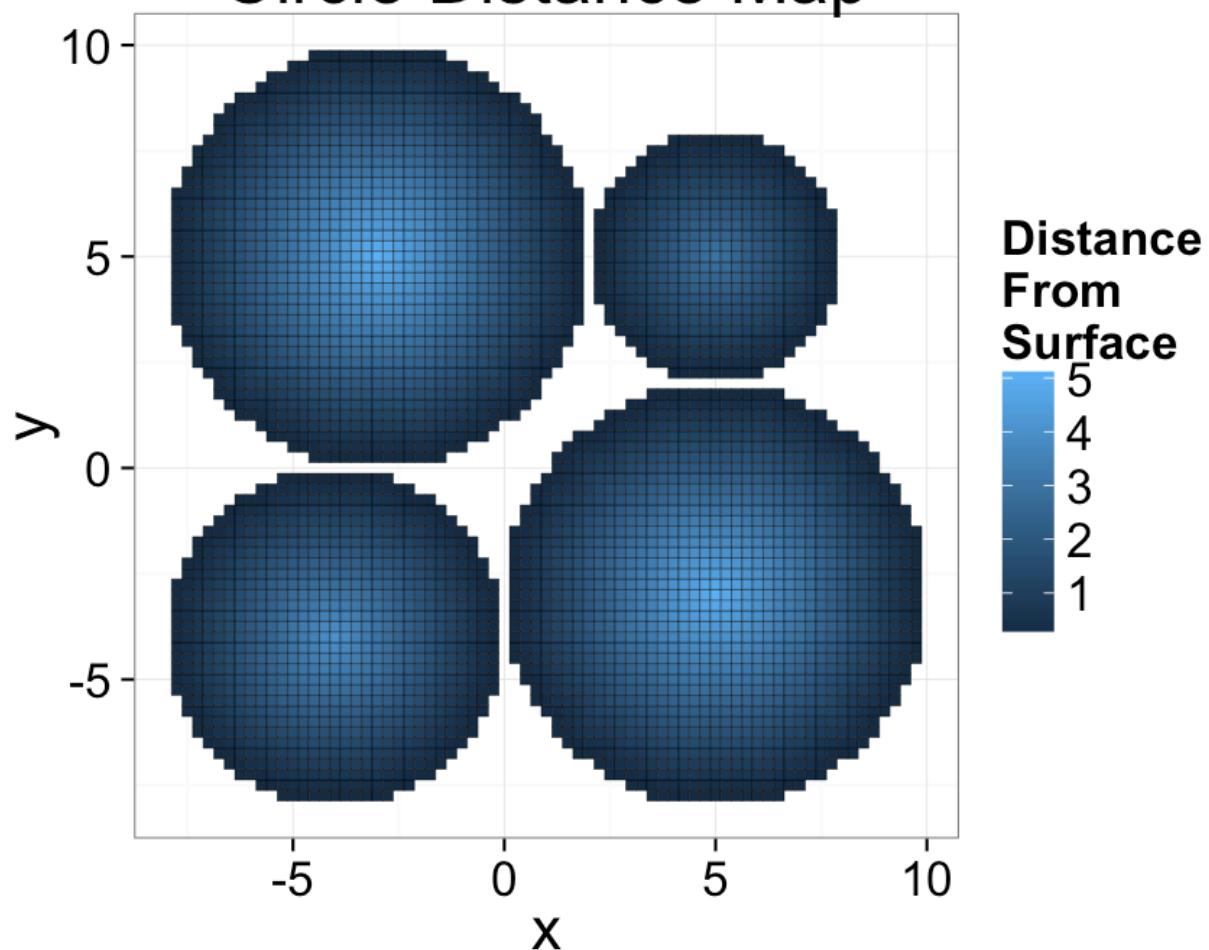
## Distance Map

One of the most useful components of the distance map is that it is *relatively* insensitive to small changes in connectivity.

- Component Labeling would find radically different results for these two images
  - One has 4 small circles
  - One has 1 big blob

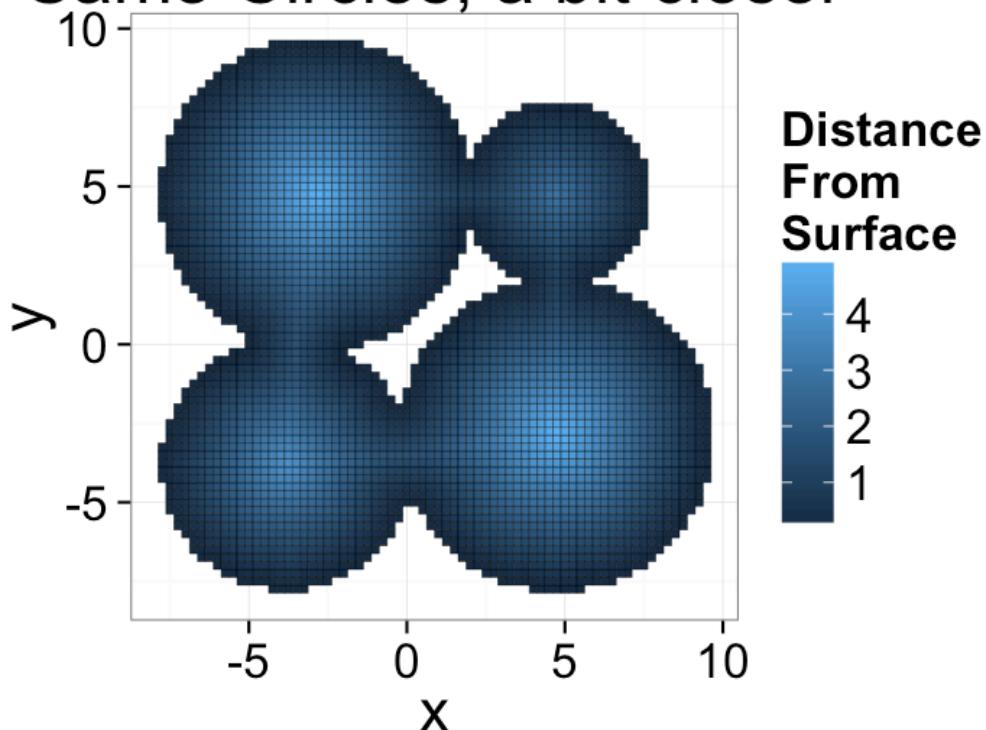
+/- R Code

## Circle Distance Map

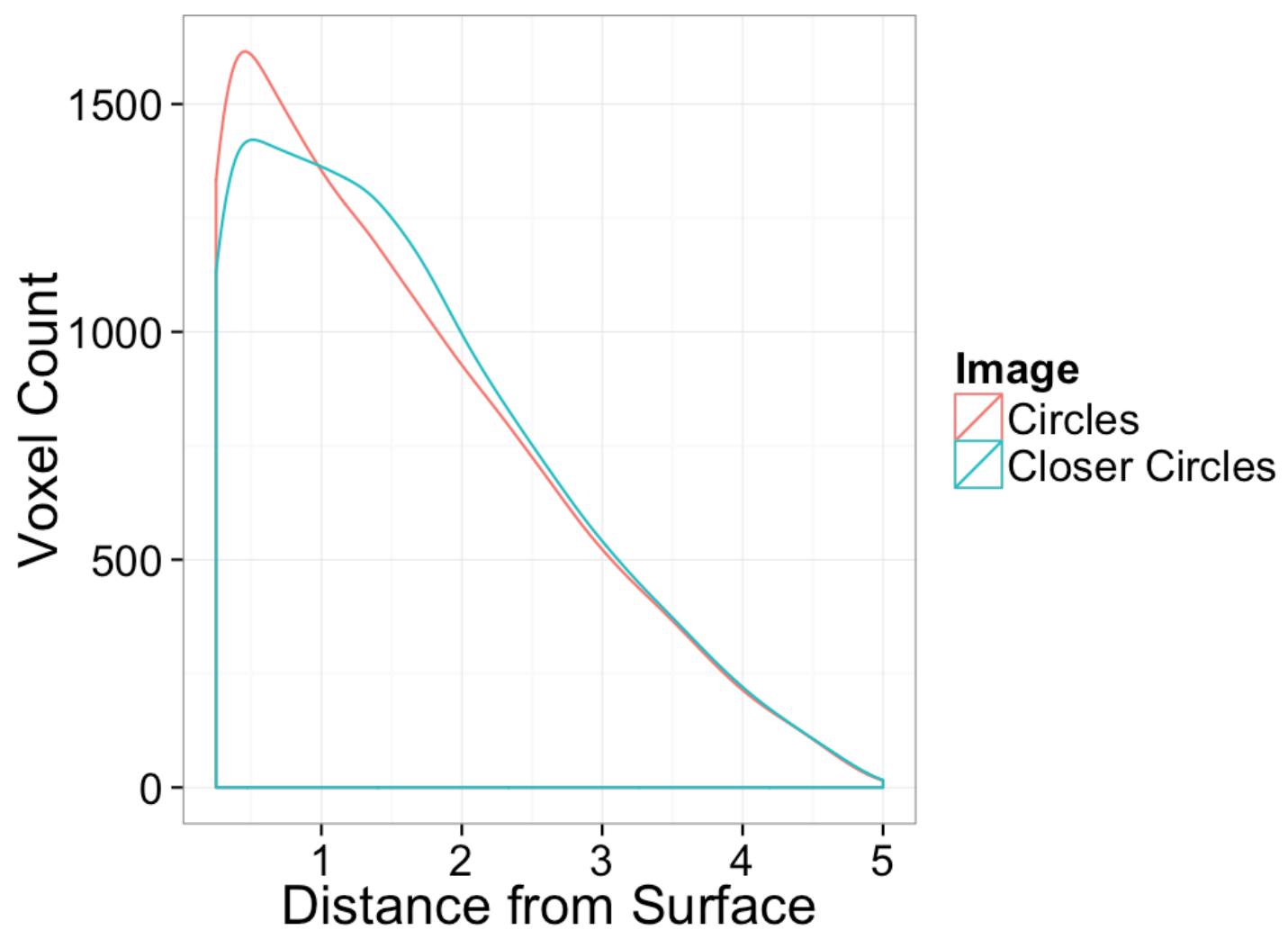


+/- R Code

## Same Circles, a bit closer



+/- R Code

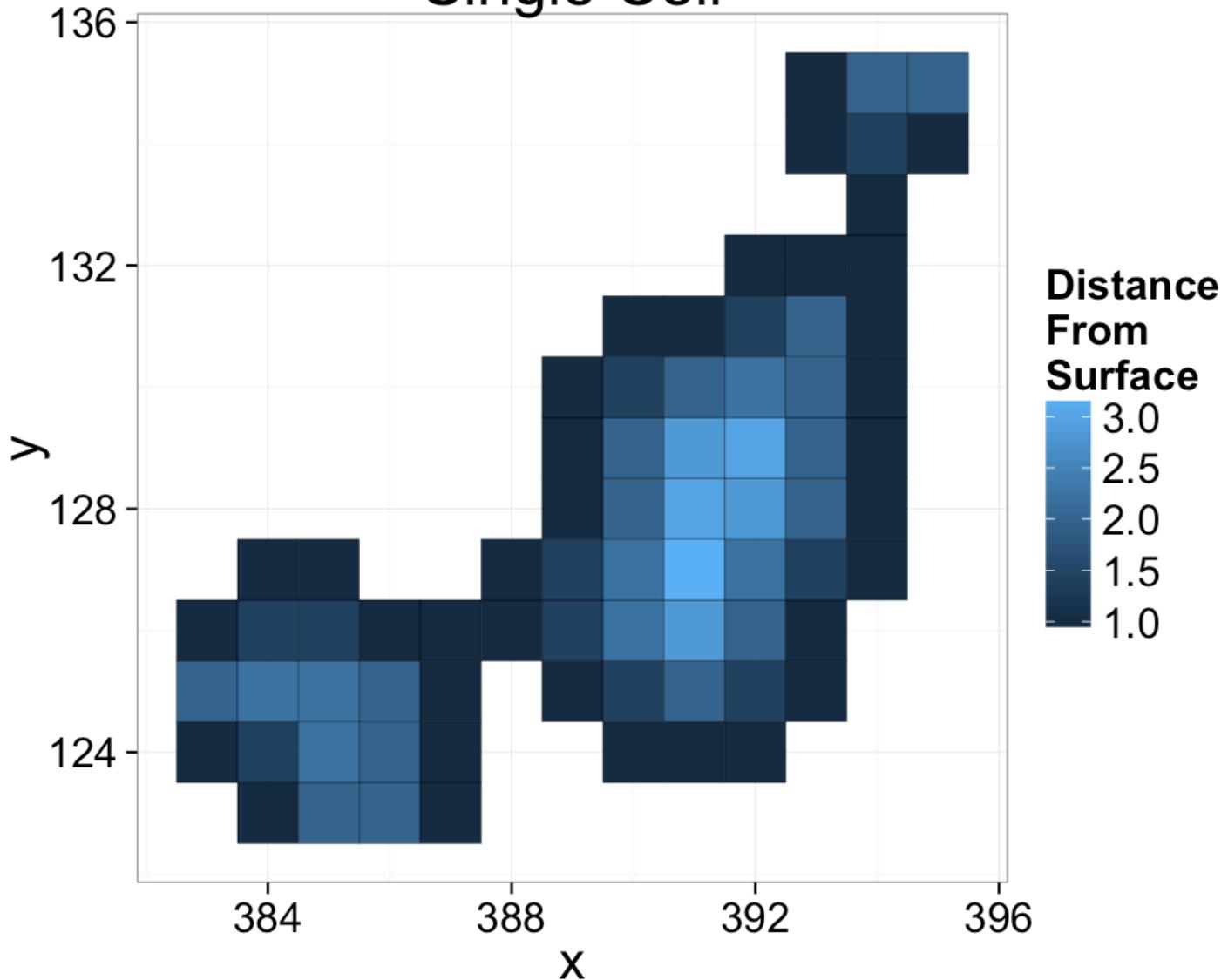


## Distance Map of Cell

Foreground

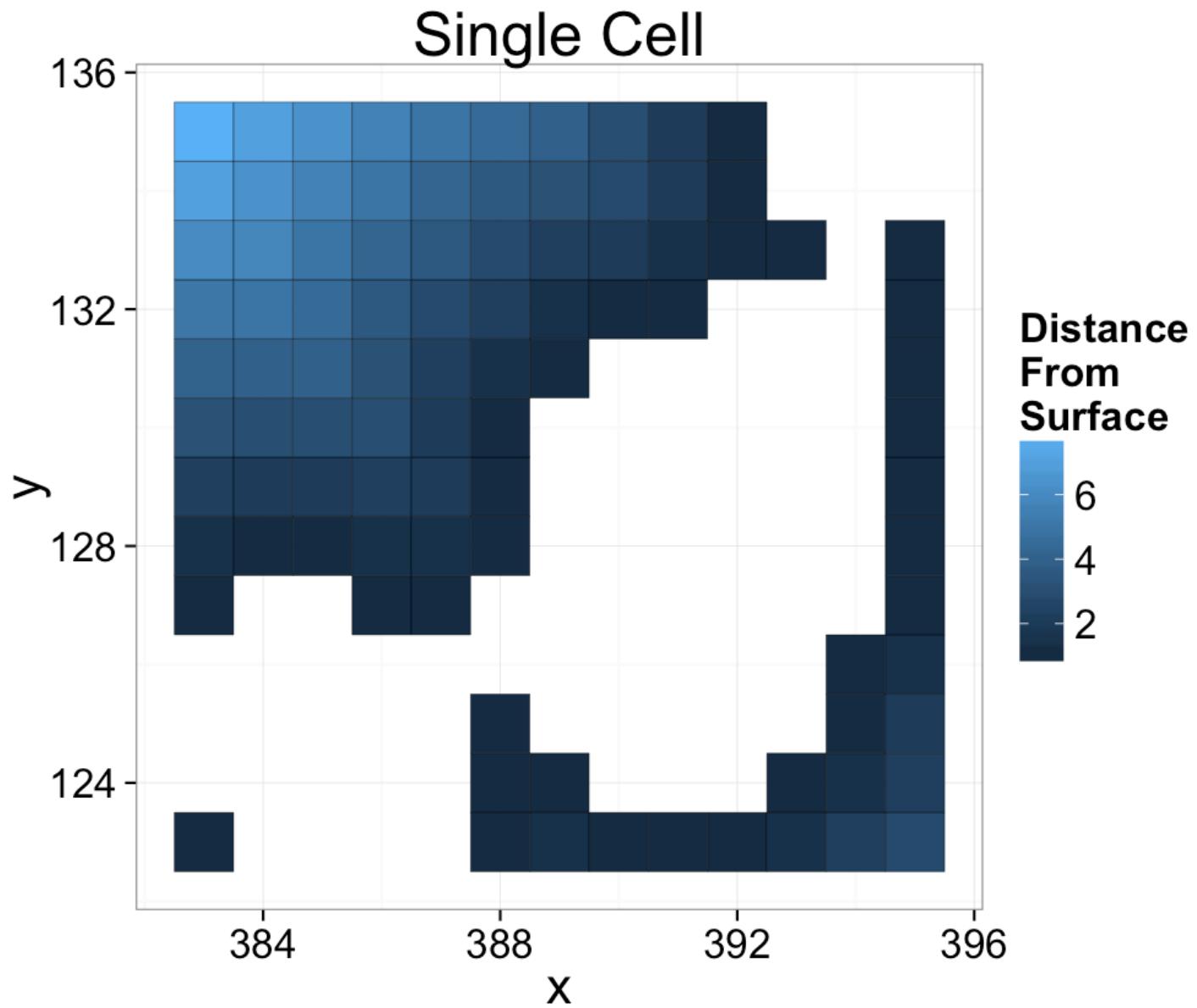
+/- R Code

## Single Cell



## Background

+/- R Code

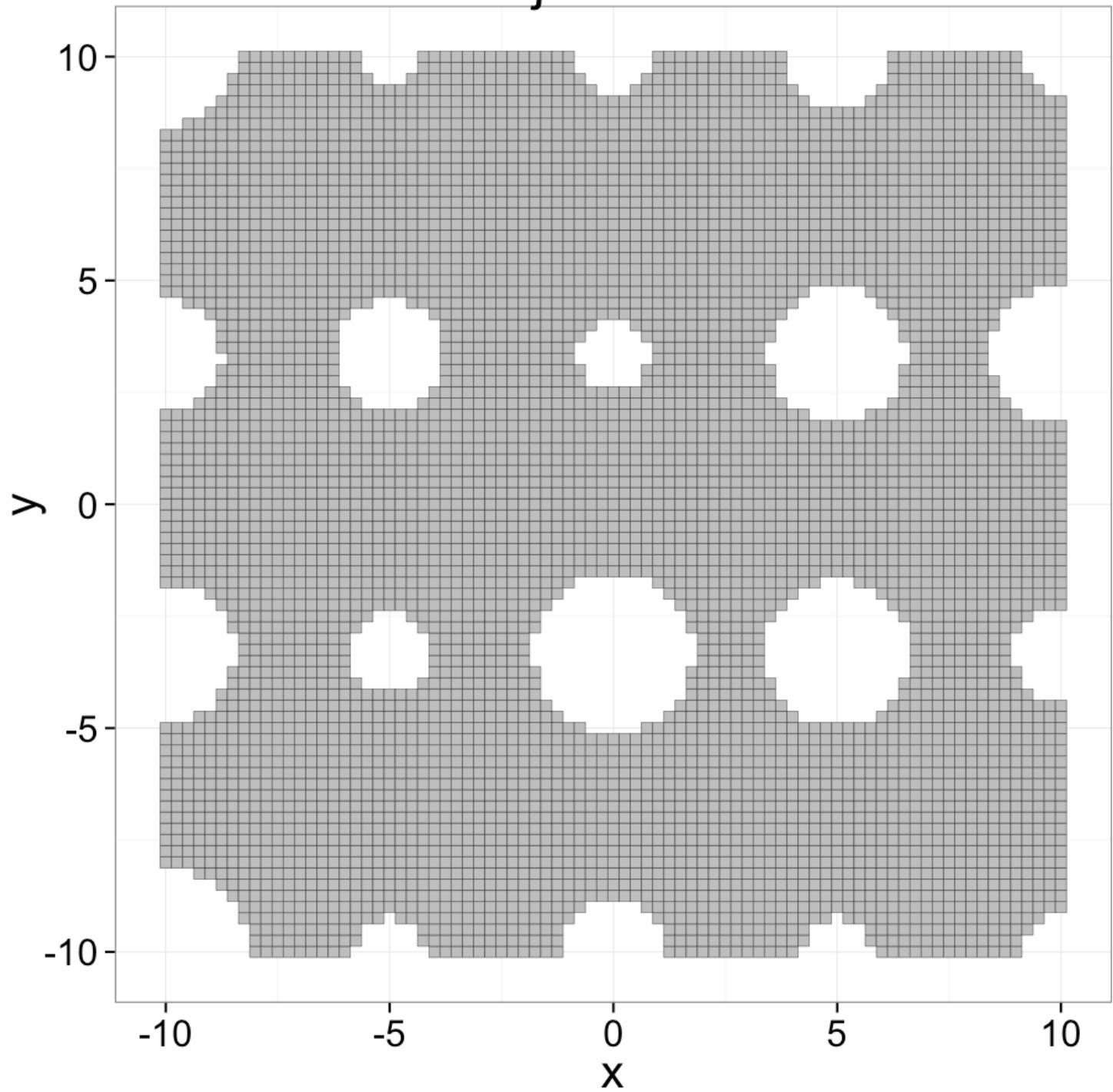


## Skeletonization / Networks

For some structures like cellular materials and trabecular bone, we want a more detailed analysis than just thickness. We want to know

- which structures are connected
- how they are connected
- express the network in a simple manner
  - quantify tortuosity
  - branching

## Object Mask



**+/- R Code**

# Skeletonization

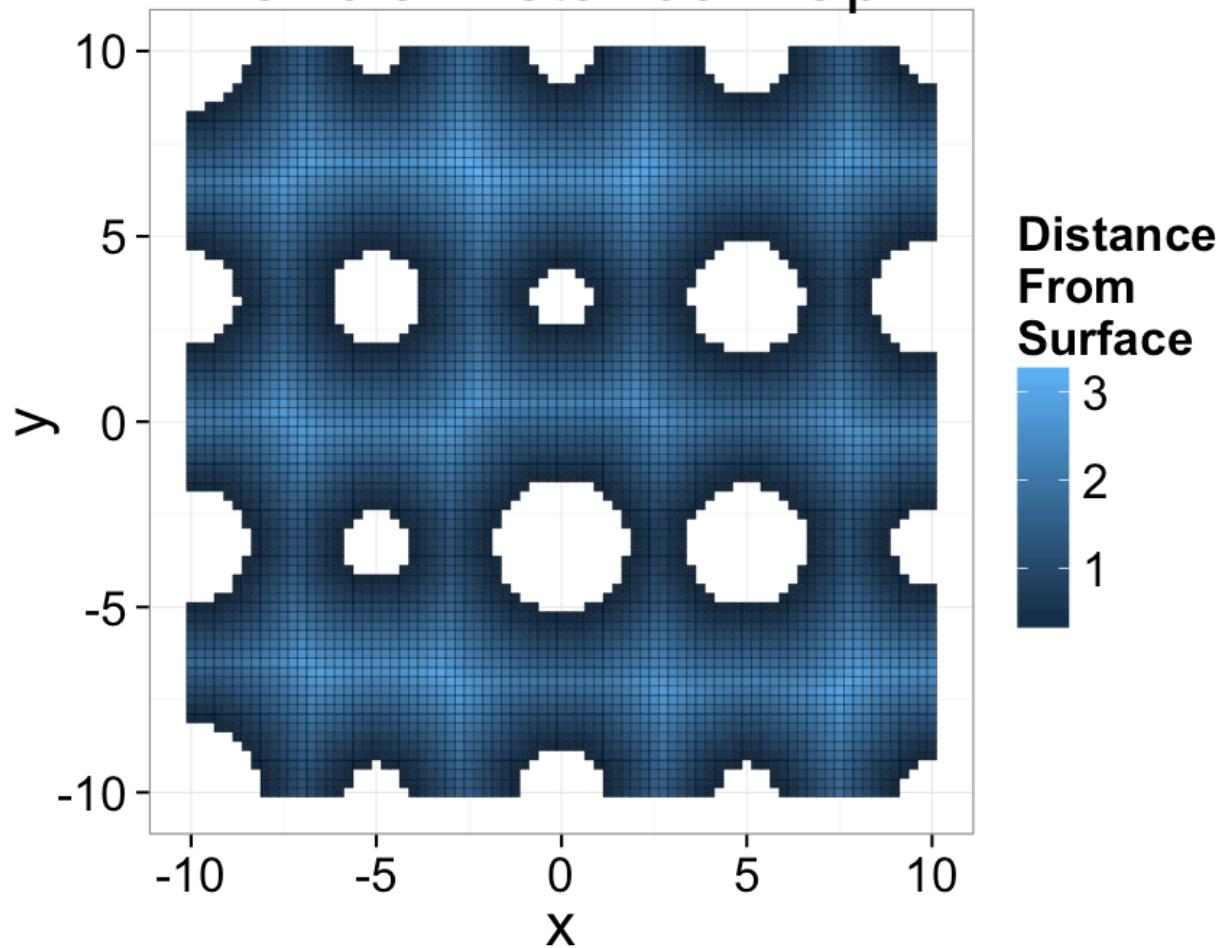
The first step is to take the distance transform the structure

$$I_d(x, y) = \text{dist}(I(x, y))$$

We can see in this image there are already local maxima that form a sort of backbone which closely maps to what we are interested in.

**+/- R Code**

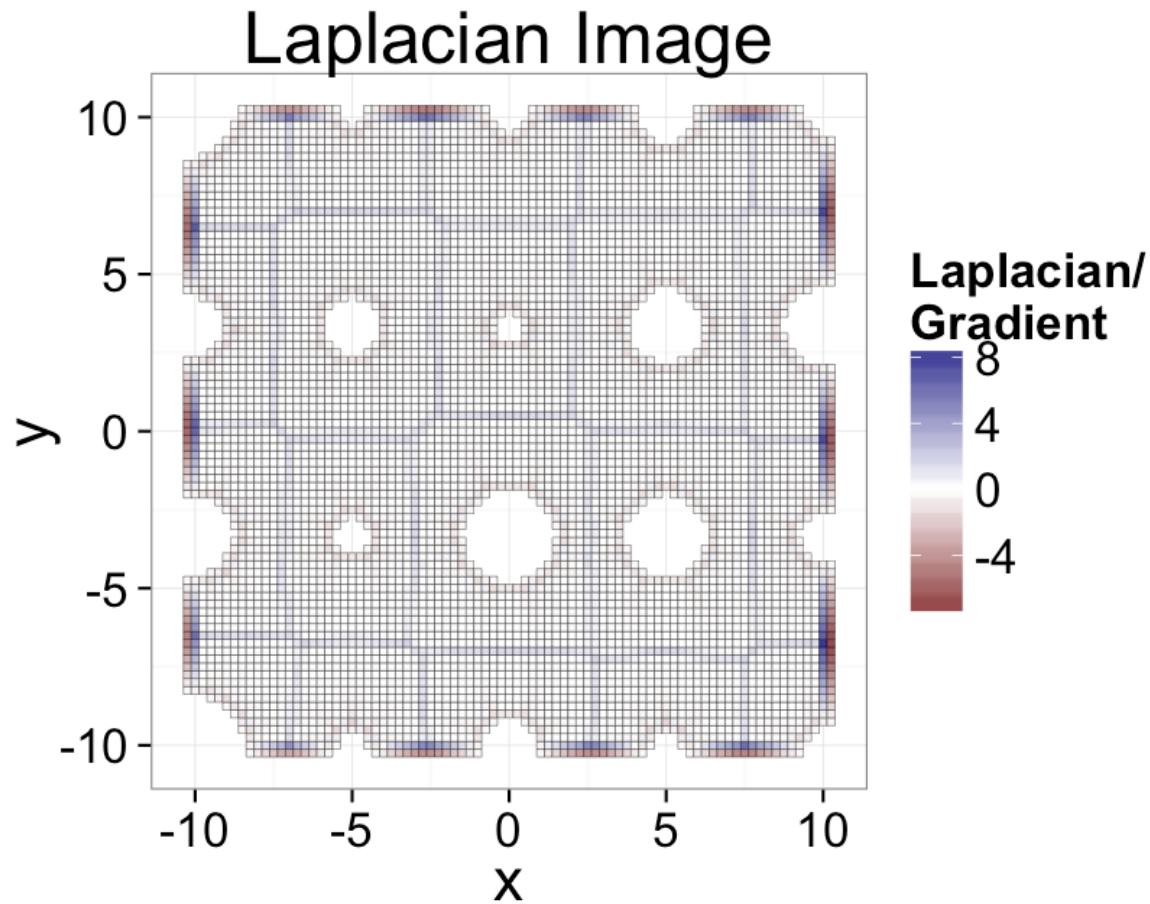
## Circle Distance Map



By using the laplacian filter as an approximate for the derivative operator which finds the values which have high local gradients.

$$\nabla I_d(x, y) = \left( \frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2} \right) I_d \approx \underbrace{\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}}_{\text{Laplacian Kernel}} \otimes I_d(x, y)$$

+/- R Code



## Creating the skeleton

We can locate the local maxima of the structure by setting a minimum surface distance

$$I_d(x, y) > \text{MIN} - \text{SLOPE}$$

and combining it with a minimum slope value

$$\nabla I_d(x, y) > \text{MIN} - \text{DIST}$$

## Thresholds

Harking back to our first lectures, this can be seen as a 2D threshold of the entire dataset.

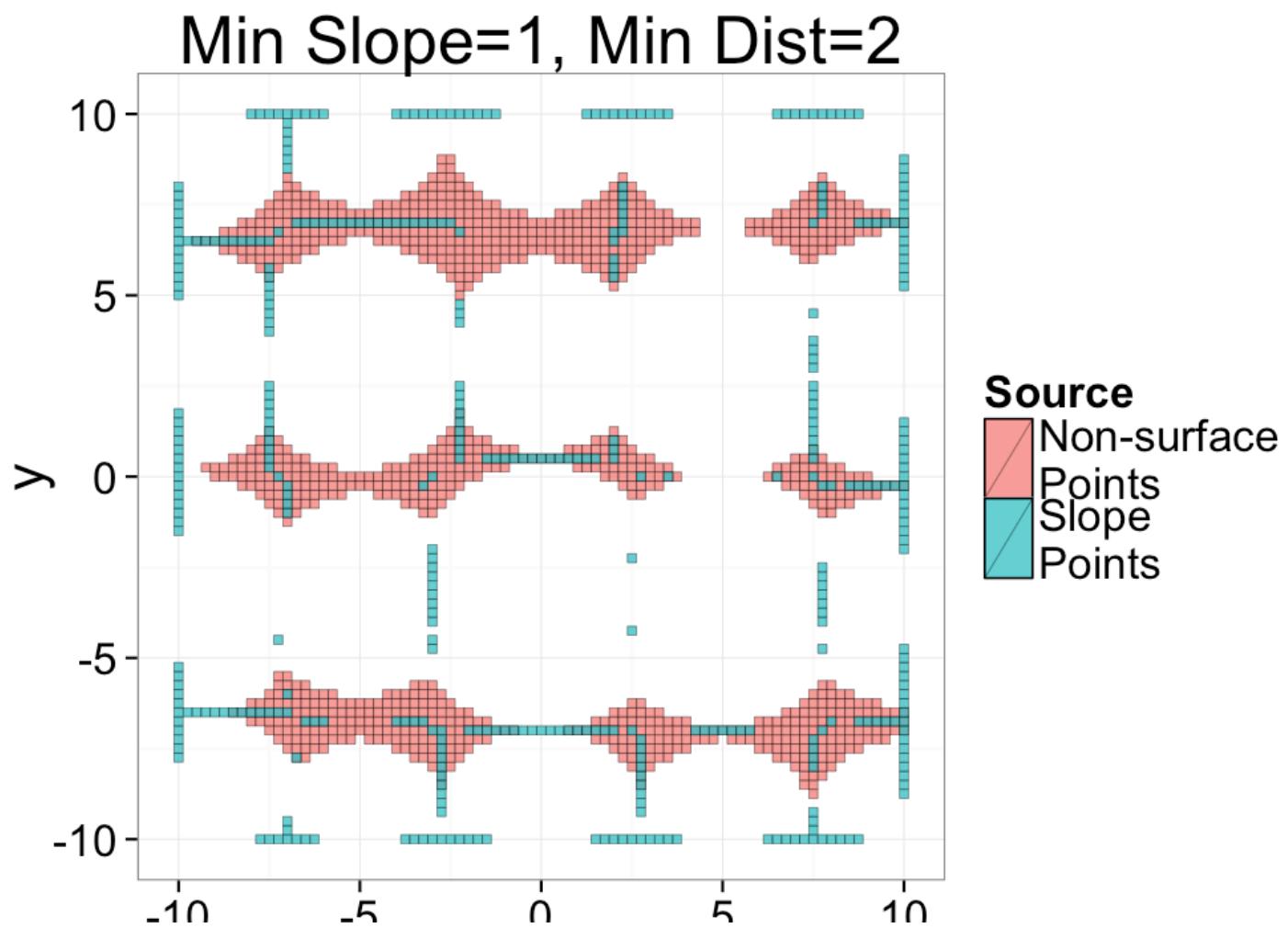
- We first make the dataset into a tuple

$$\text{cImg}(x, y) = \langle \underbrace{I_d(x, y)}_1, \underbrace{\nabla I_d(x, y)}_2 \rangle$$

$$\text{skelImage}(x, y) =$$

$$\begin{cases} 1, & \text{cImg}_1(x, y) \geq \text{MIN-DIST} \\ & \& \text{cImg}_2(x, y) \geq \text{MIN-SLOPE} \\ 0, & \text{otherwise} \end{cases}$$

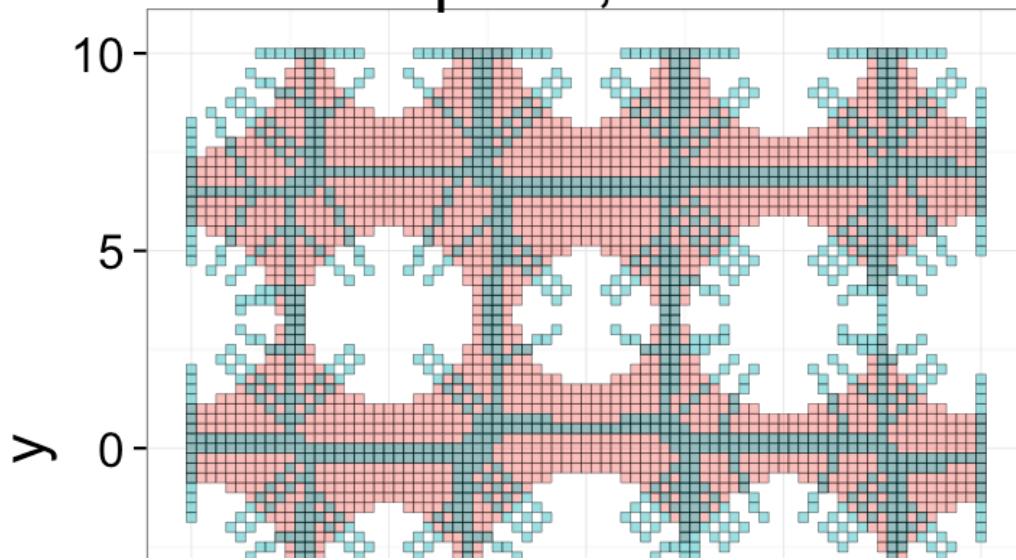
## Skeleton: Different Thresholds



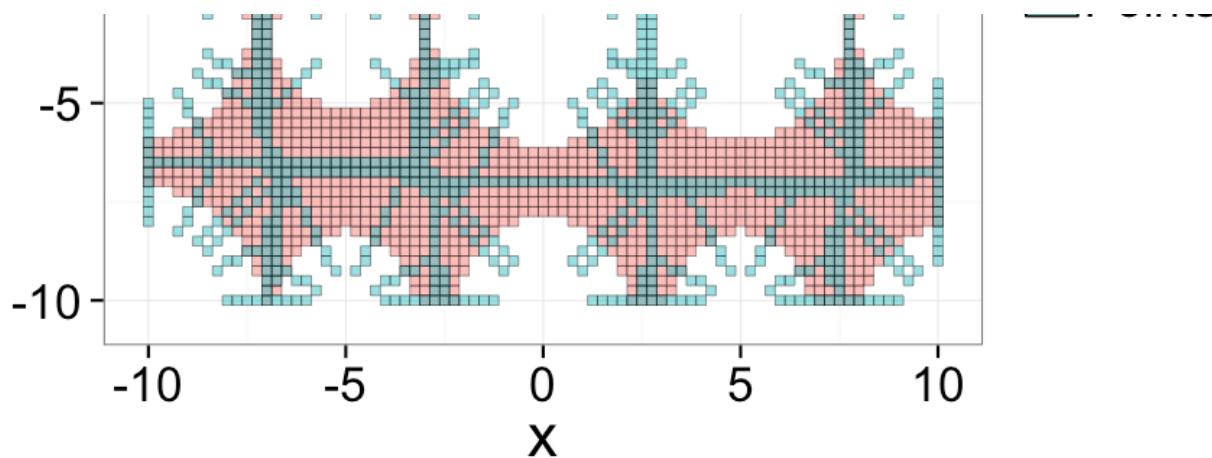
- ▾ ▾ ▾ ▾ X

+/- R Code

Min Slope=0, Min Dist=1



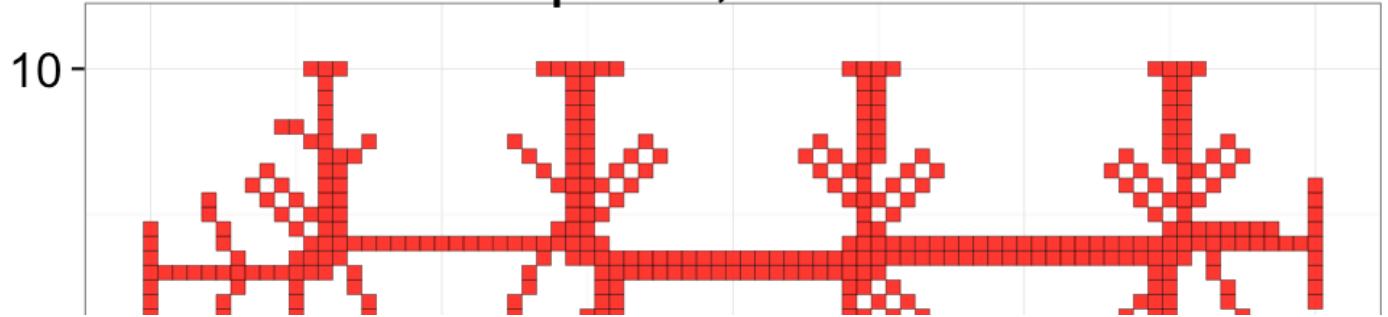
**Source**  
Non-surface  
Points  
Slope  
Points

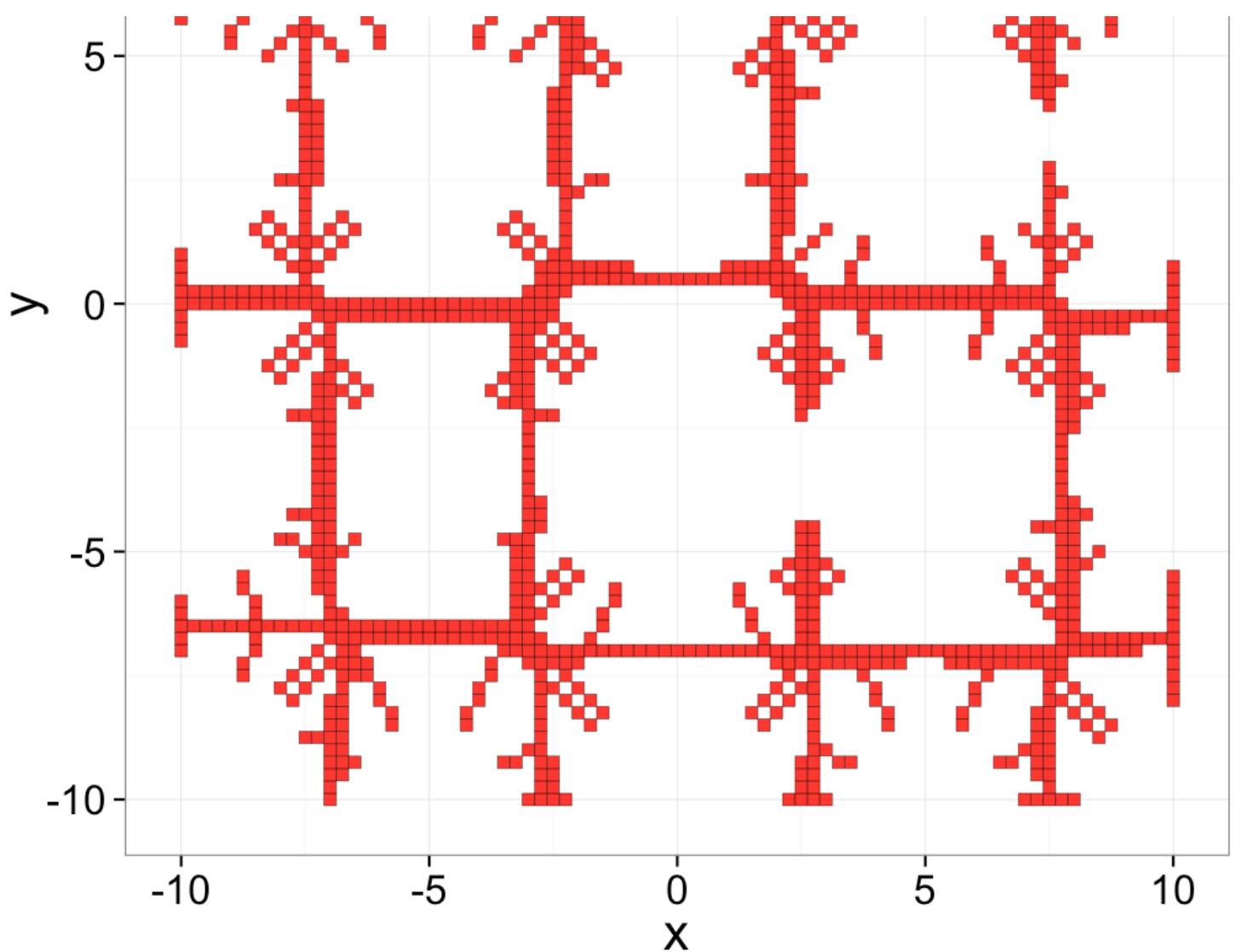


+/- R Code

## Skeleton: Pruning

Min Slope=0, Min Dist=1.5



**+/- R Code**

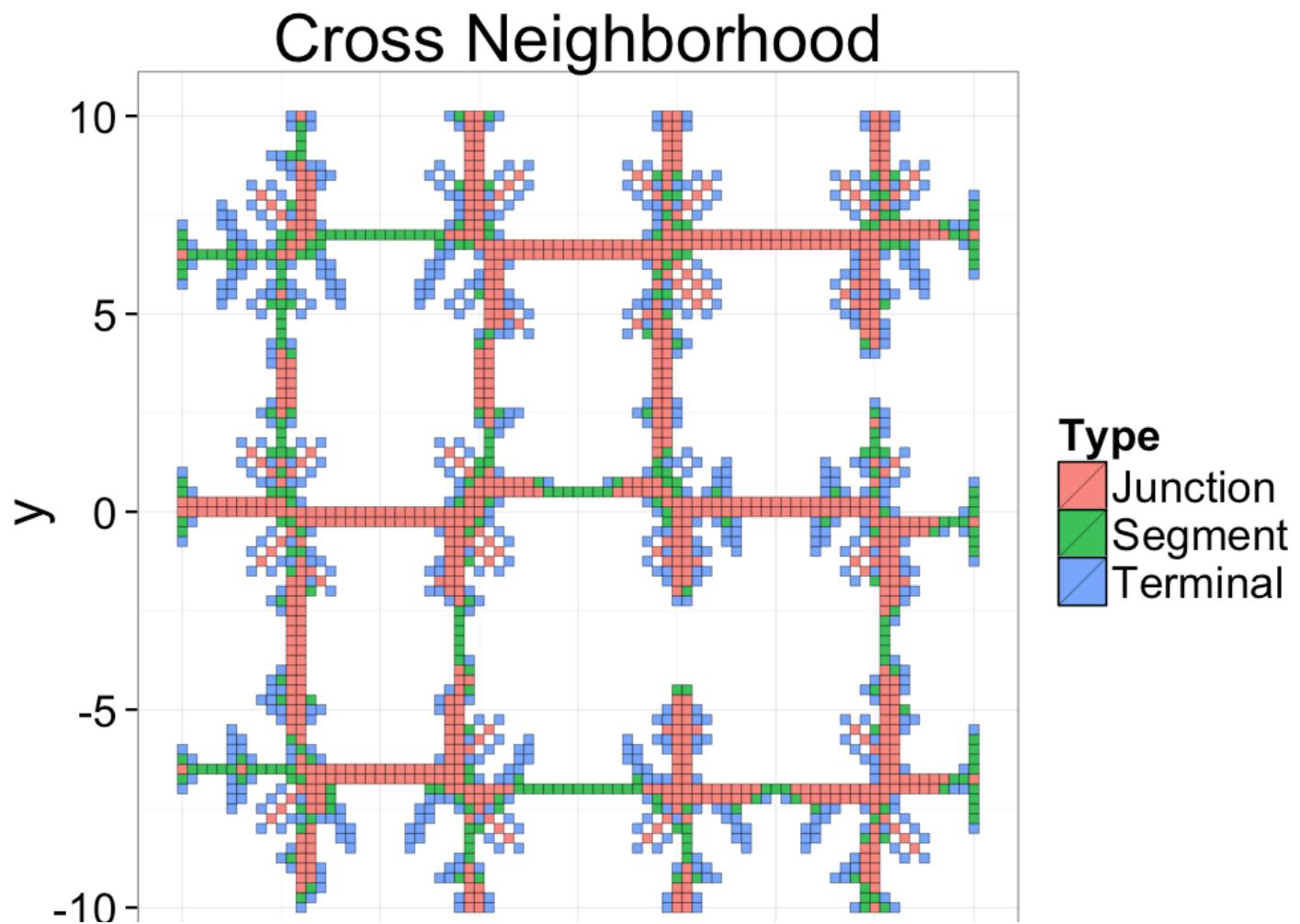
The structure is overgrown

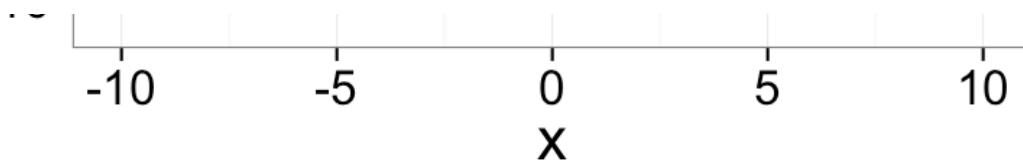
1. Stricter 'thresholds'
  - Sensitive to 'threshold' selection
  - Resolution dependent
2. Dedicated pruning algorithms

- Ideally model-based
- Minimum branch length
- Minimum branch width

# Skeleton: Junctions

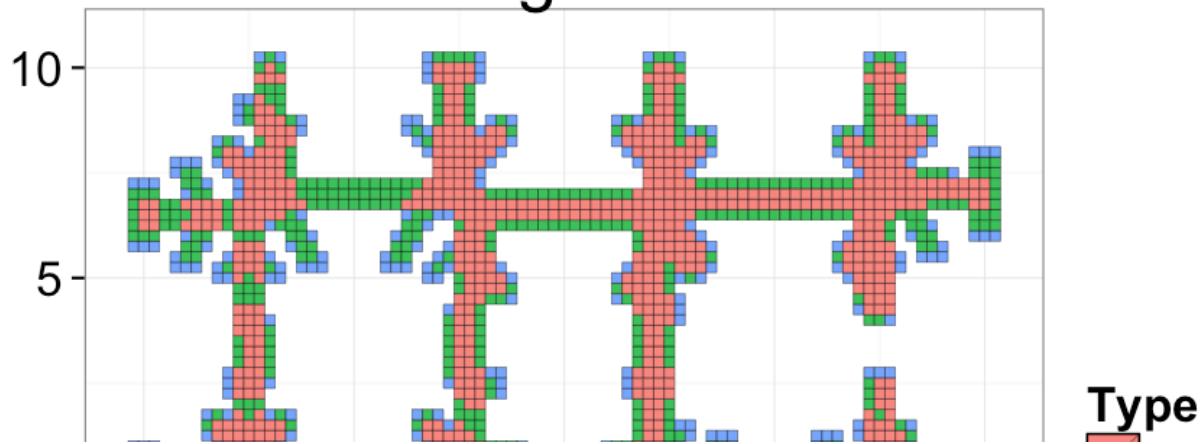
With the skeleton which is ideally one voxel thick, we can characterize the junctions in the system by looking at the neighborhood of each voxel.

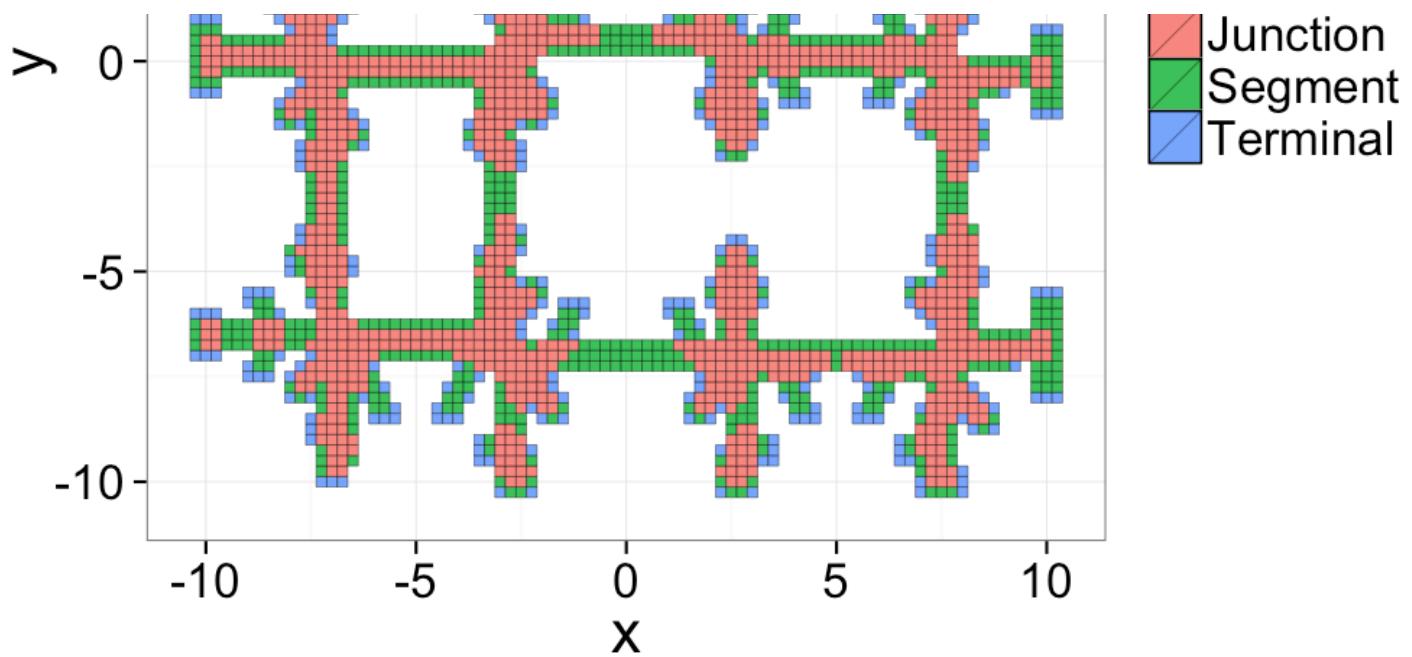


**+/- R Code**

As with *nearly* every operation, the neighborhood we define is important and we can see that if we use a box neighborhood vs a cross neighborhood (4 vs 8 adjacent) we count diagonal connections differently

## Box Neighborhood





+/- R Code

## Skeleton: Network Assessment

Once we have classified the skeleton into *terminals*, *segments*, and *junctions* we can analyze the various components and assess metrics like connectivity, branching, and many other.

The easiest way is to further process the image by

- component labeling the different segments and joints
- determining the average length of each segment
- number of incoming segments at a given joint

Most of the other metrics can be simply counted

## Cross Neighborhood

+/- R Code

| label    | x   |
|----------|-----|
| Junction | 798 |
| Segment  | 309 |
| Terminal | 611 |

## Box Neighborhood

+/- R Code

| label    | x    |
|----------|------|
| Junction | 1431 |
| Segment  | 816  |
| Terminal | 412  |

## Skeleton: Tortuosity

One of the more interesting ones in material science is called tortuosity and it is defined as the ratio between the arc-length of a *segment* and the distance between its starting and ending points.

$$\tau = \frac{L}{C}$$

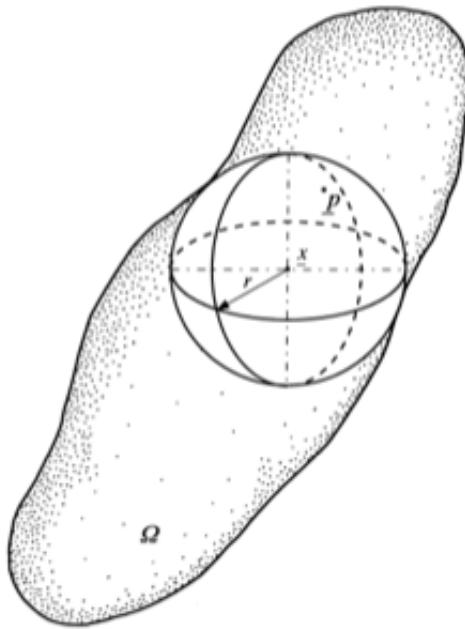
A high degree of tortuosity (<http://en.wikipedia.org/wiki/Tortuosity>) indicates that the network is convoluted and is important when estimating or predicting flow rates. Specifically

- in geology it is an indication that diffusion and fluid transport will occur more slowly
- in analytical chemistry it is utilized to perform size exclusion chromatography
- in vascular tissue it can be a sign of pathology.

## Thickness Map: What is it?

Thickness is a metric for assessing the size and structure of objects in a very generic manner. For every point  $\vec{x}$  in the image you find the largest sphere which:

- contains that point
- is entirely contained within the object



Taken from [1]

- The image shows a typical object
- The sphere centered at point  $p$  with a radius  $r$  is the largest that fits

## Application

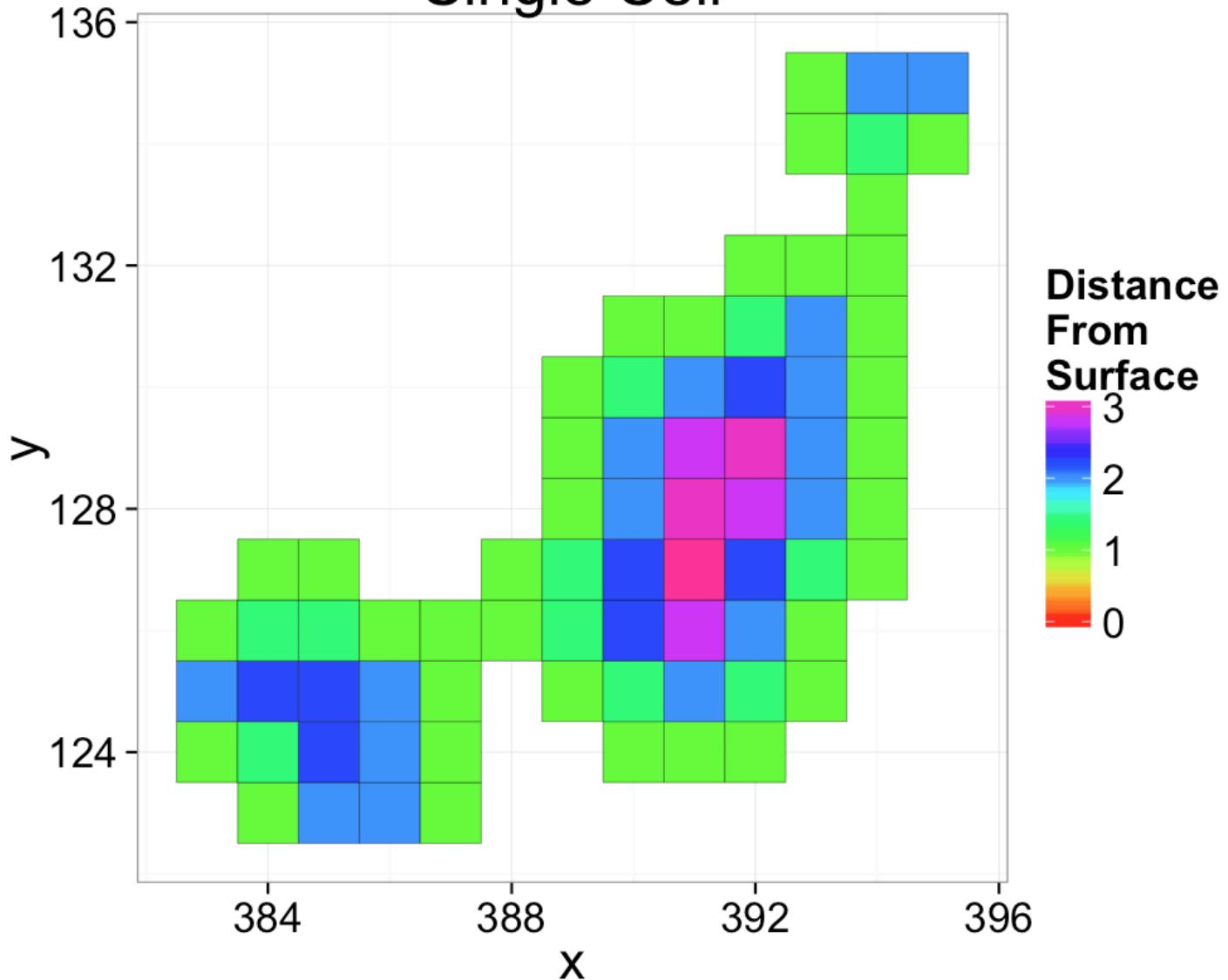
- Ideal for spherical and cylindrical objects since it shows their radius
- Also relevant for flow and diffusion since it can highlight bottlenecks in structure

## Thickness Map

+/- R Code

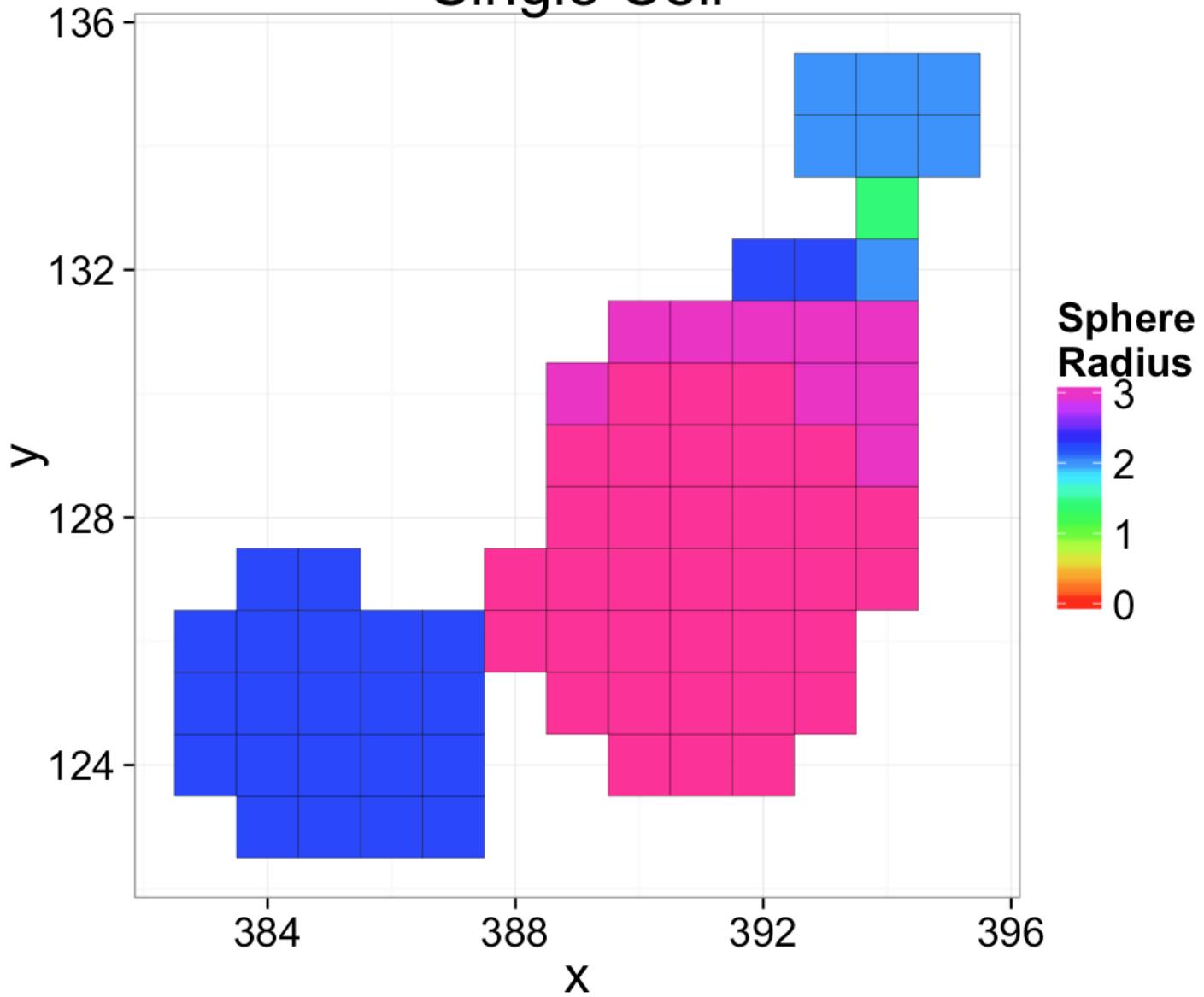
+/- R Code

# Single Cell



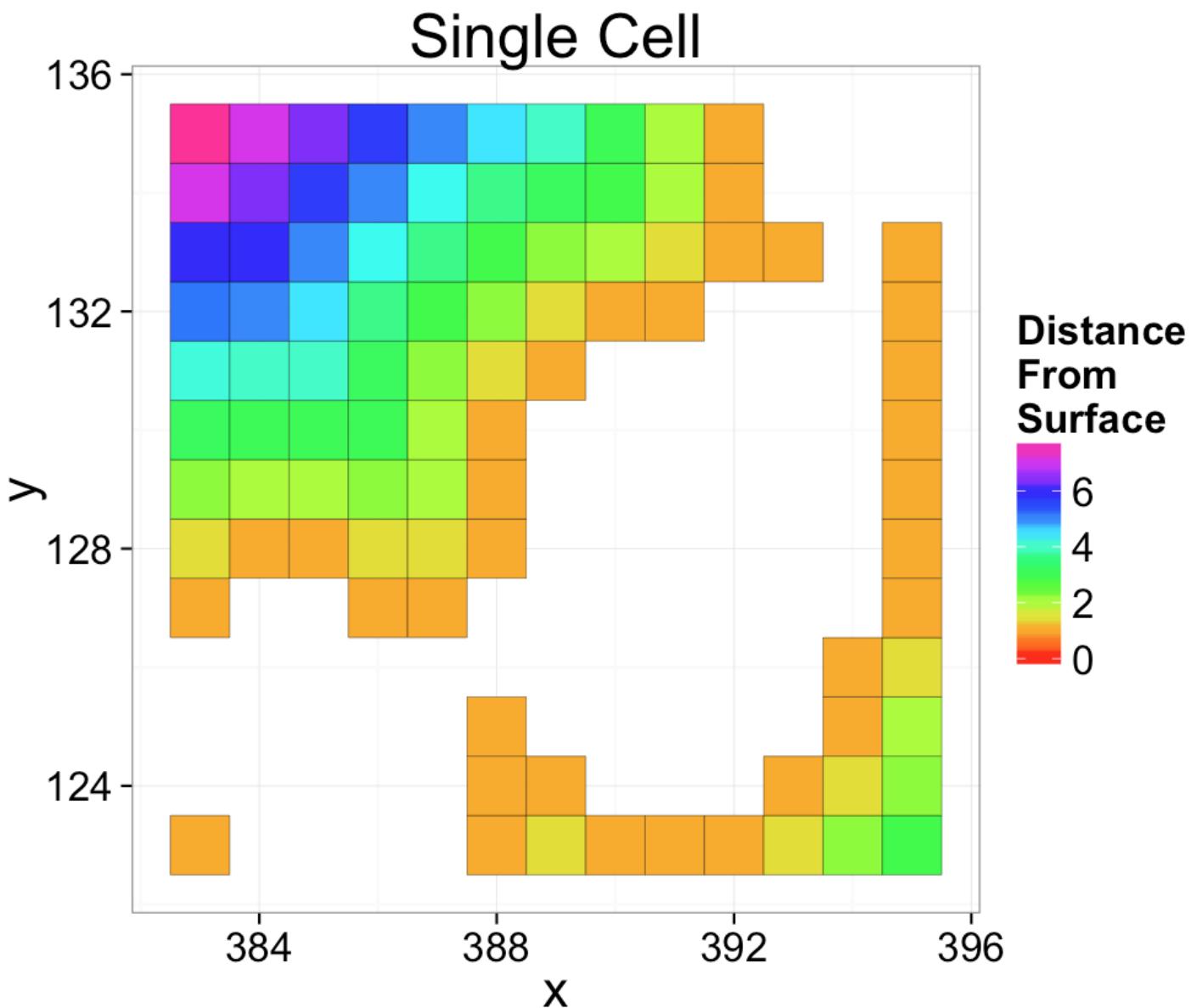
+/- R Code

## Single Cell

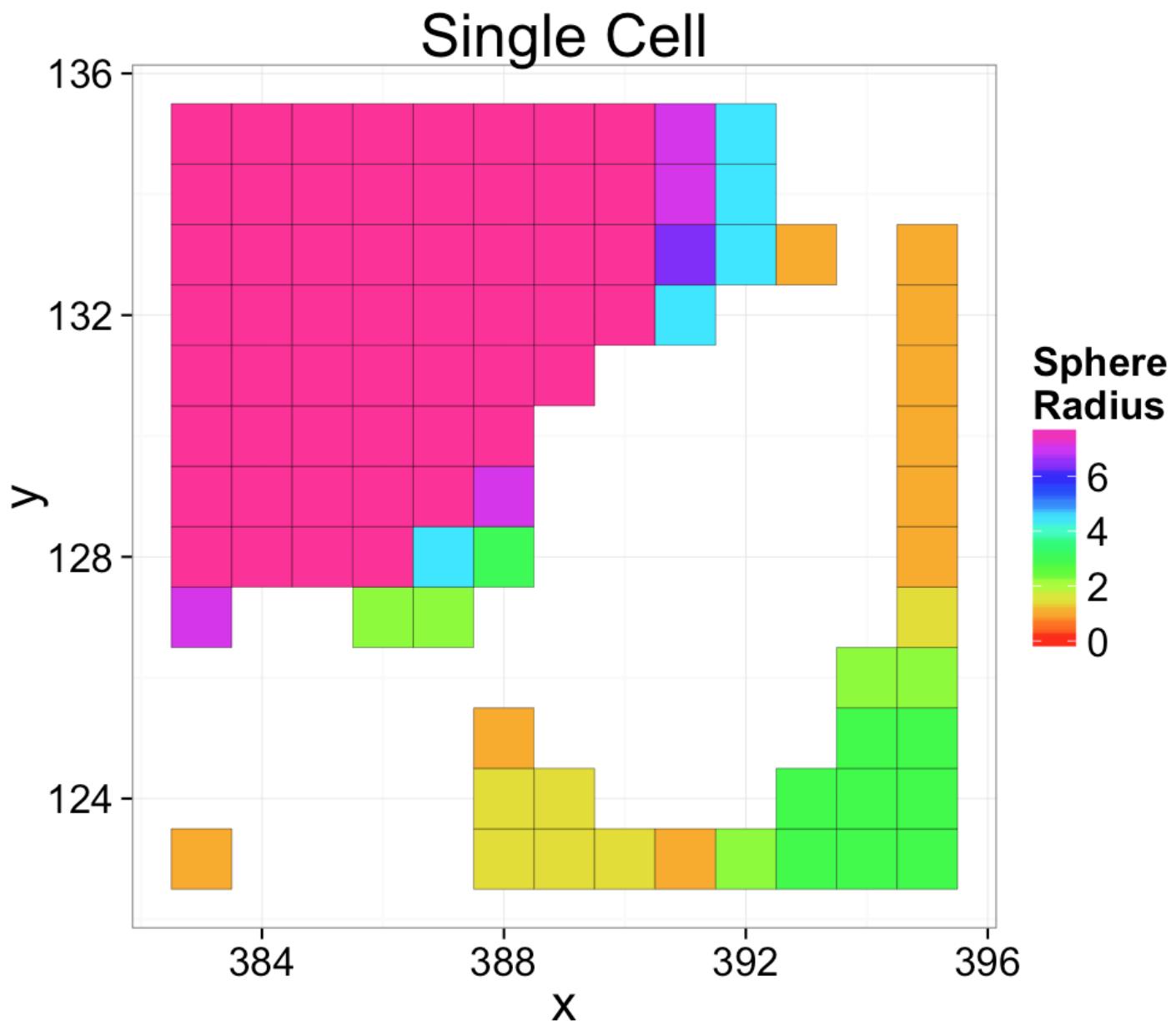


## Thickness Map

+/- R Code



+/- R Code



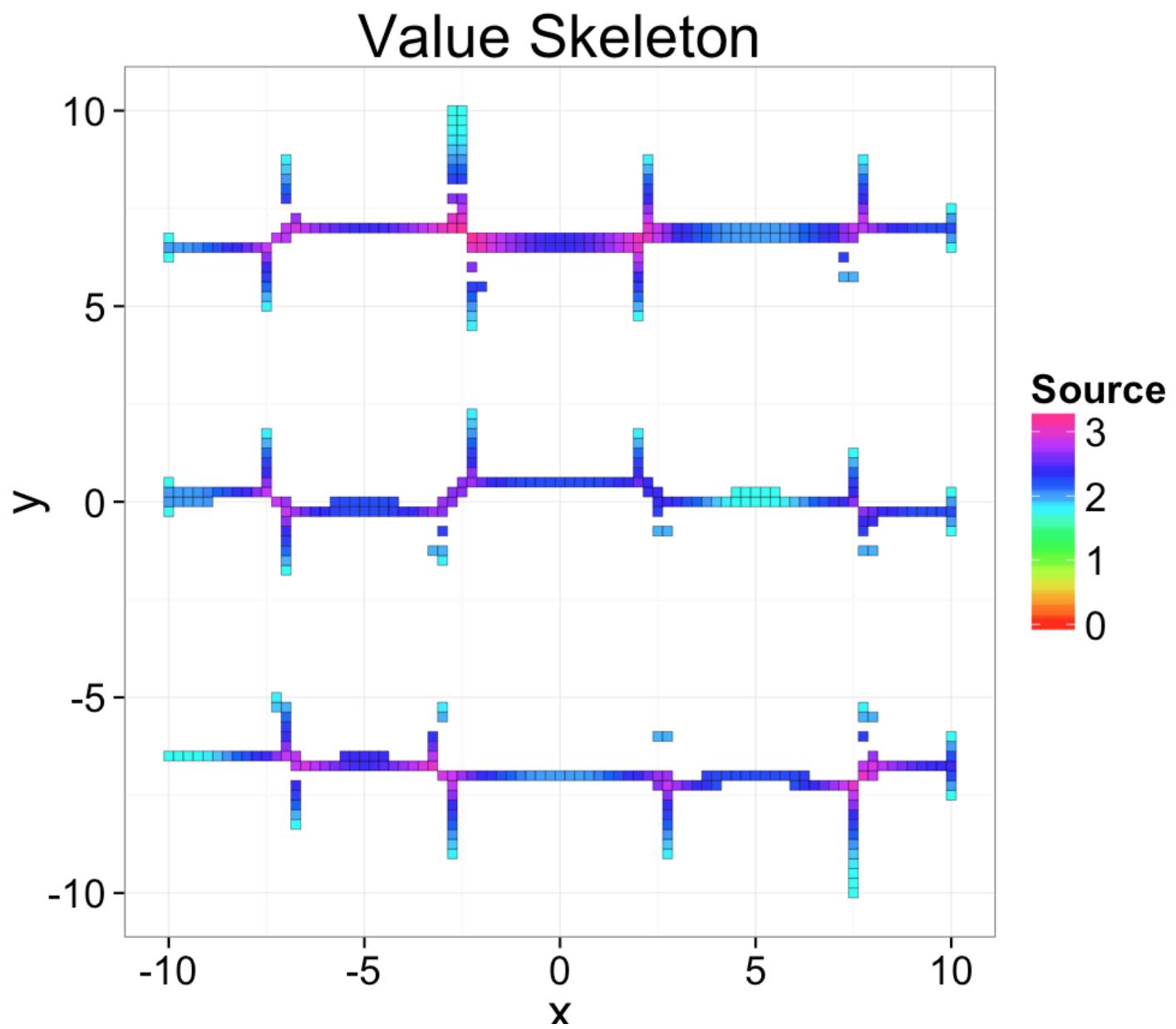
## Thickness Map From Skeleton

Calculating the distance map by drawing a sphere at each point is very time consuming ( $O(n^3)$ ).

- The skeleton (last section) is very closely related to the thickness.
- We found the local maxima in the image using the Laplace
- We can thus grow the Spheres from these points instead of all
- Start by instead of thresholding transforming the image to the distance at each point

$$\text{thSkelImage}(x, y) =$$

$$\begin{cases} \text{cImg}_1(x, y), & \text{cImg}_1(x, y) \geq \text{MIN} - \text{DIST} \\ & \& \text{cImg}_2(x, y) \geq \text{MIN} - \text{SLOPE} \\ 0, & \text{otherwise} \end{cases}$$



^

+/- R Code

# From Skeleton vs All Points

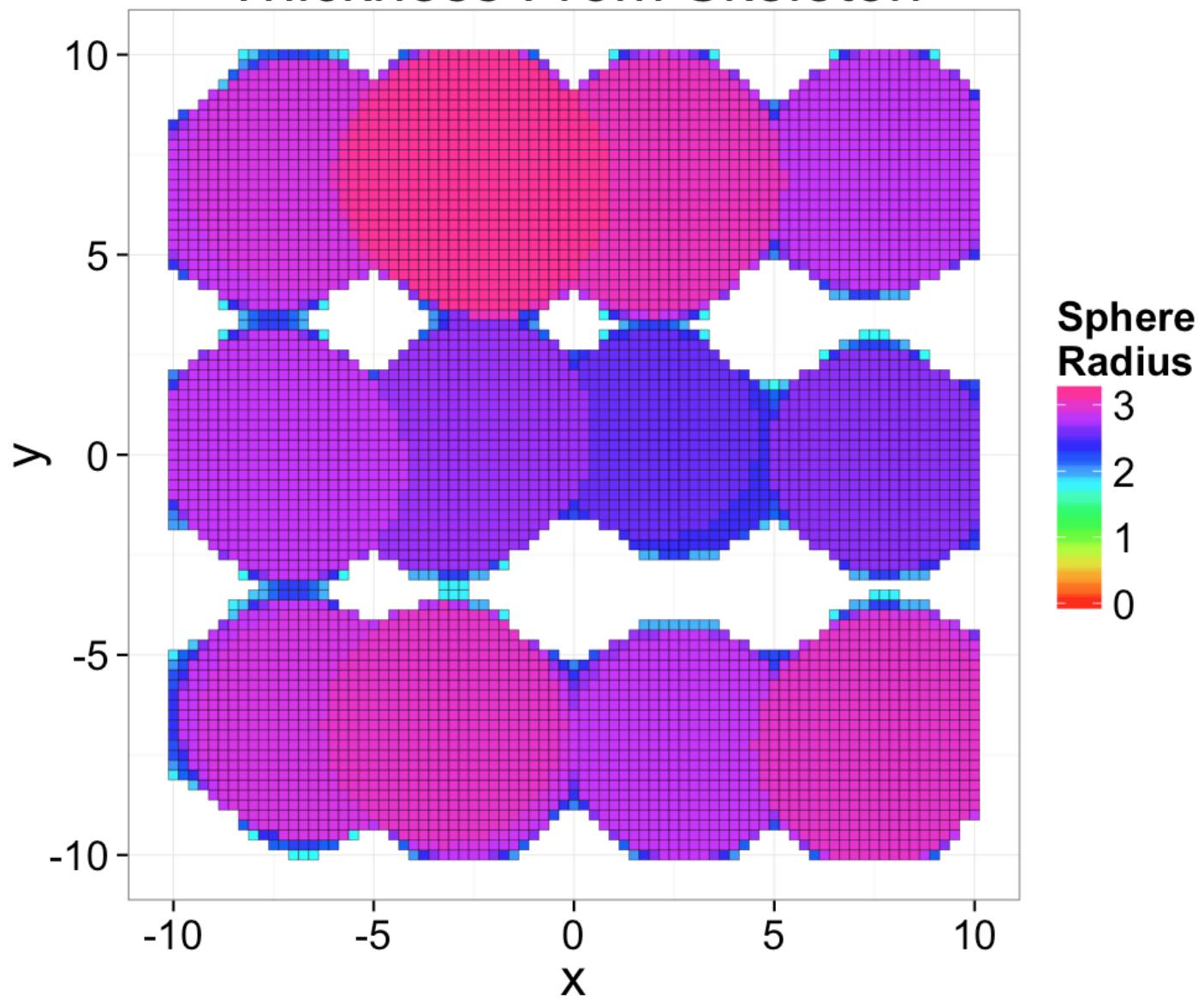
+/- R Code

+/- Output

```
##      user  system elapsed
##  4.686   0.077   4.787
```

+/- R Code

## Thickness From Skeleton

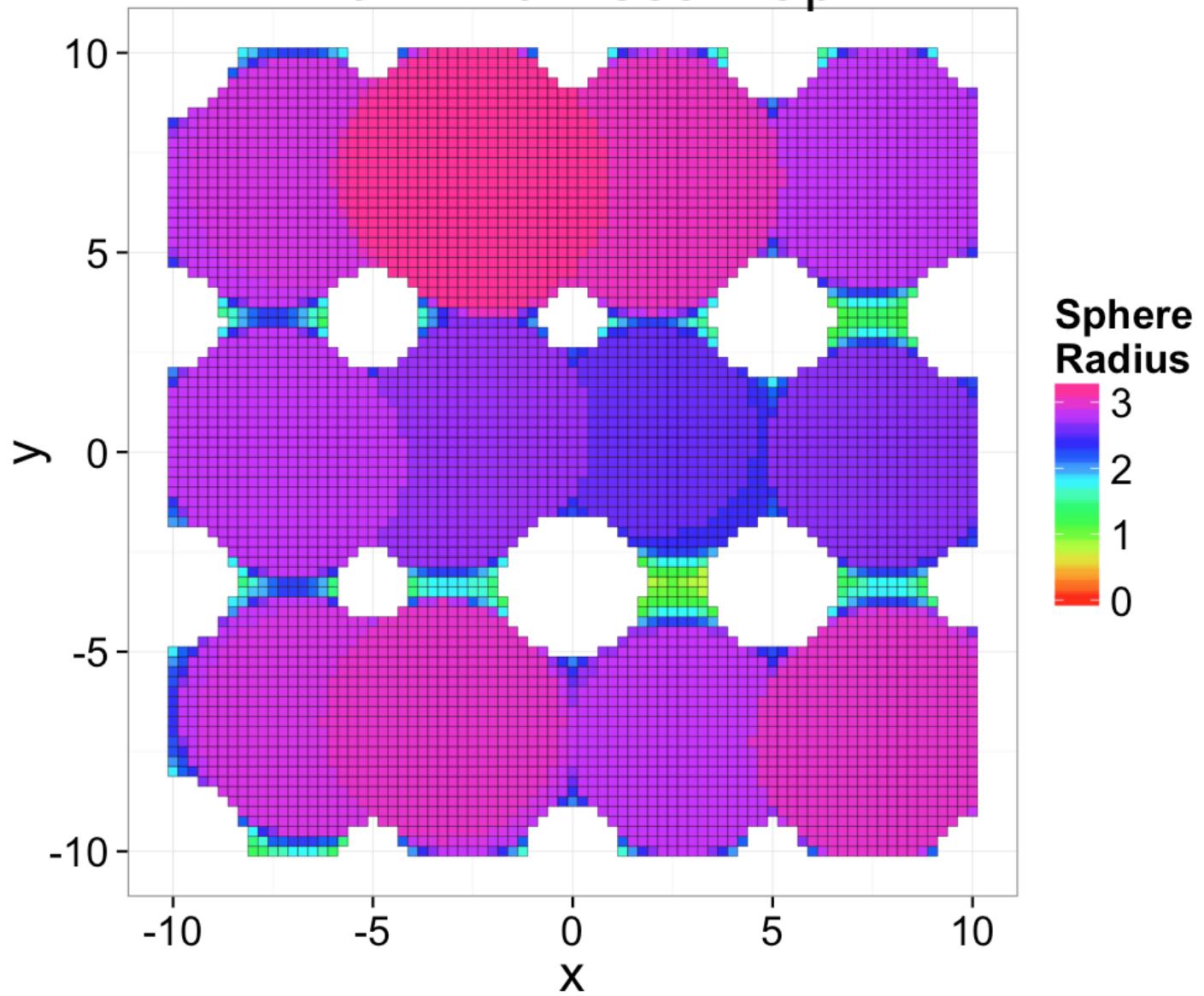


+/- R Code

+/- Output

```
##    user  system elapsed
##  17.610   0.527 18.438
```

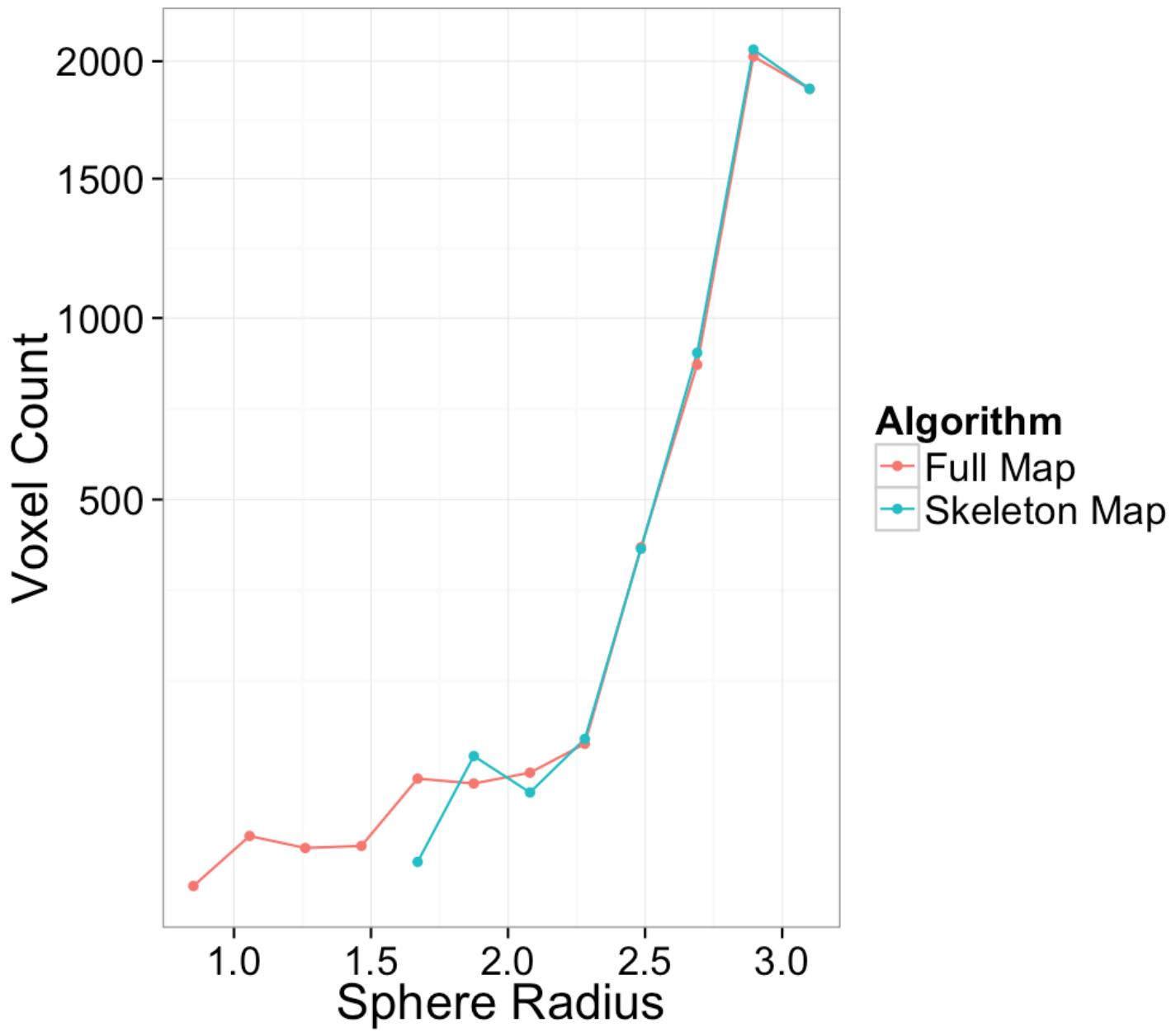
## Full Thickness Map



+/- R Code

## Statistically Does it Matter

+/- R Code



## It depends

- Small structures are lost
- They might not have been very important or *noisy* anyways
- Higher values are very similar

+/- R Code

id

Full Map

Skeleton Map

|         |      |      |
|---------|------|------|
| Min.    | 0.75 | 1.75 |
| 1st Qu. | 2.70 | 2.70 |
| Median  | 2.83 | 2.83 |
| Mean    | 2.82 | 2.84 |
| 3rd Qu. | 3.01 | 3.01 |
| Max.    | 3.20 | 3.20 |

# How much can we cut down

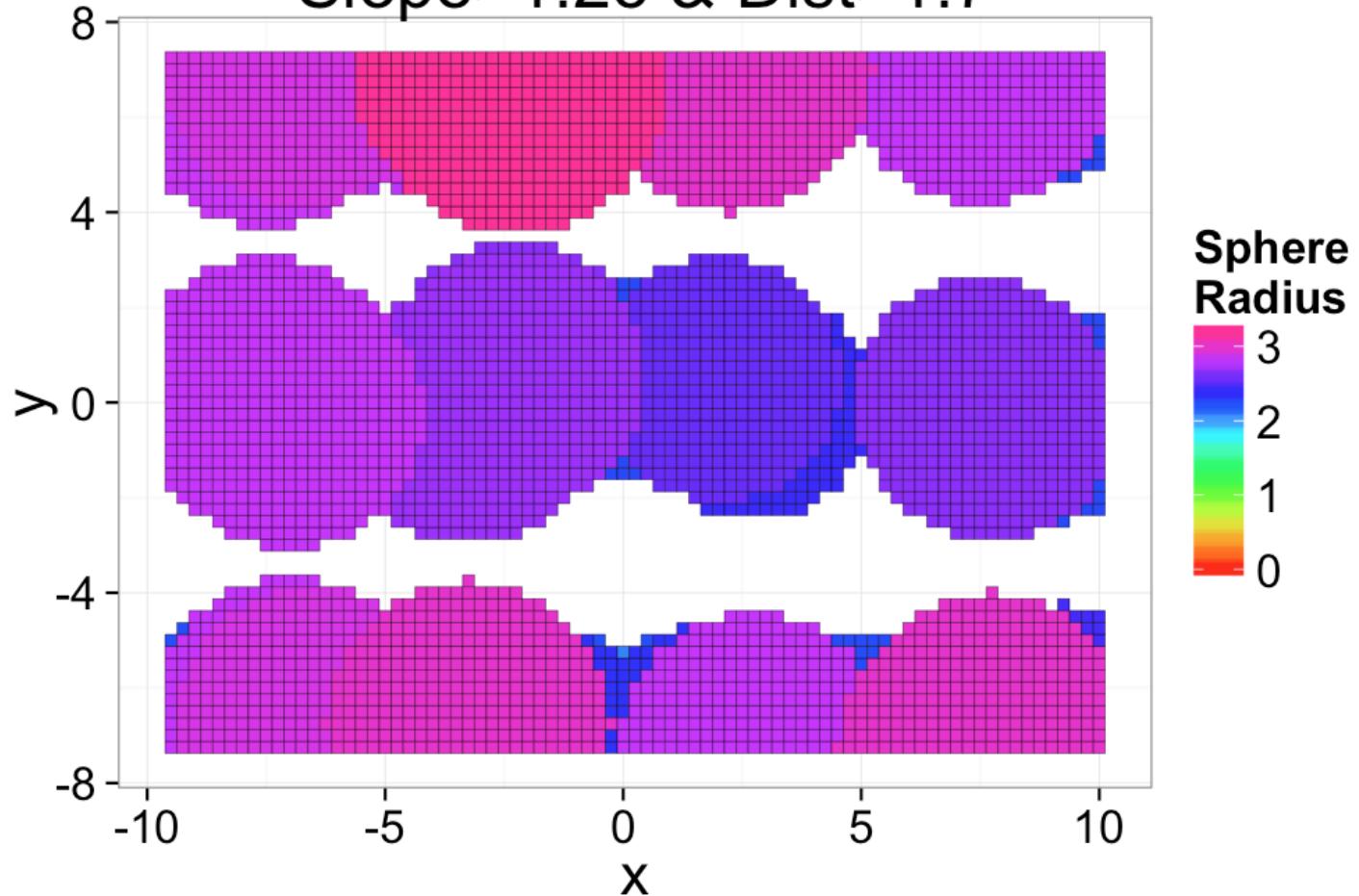
+/- R Code

+/- Output

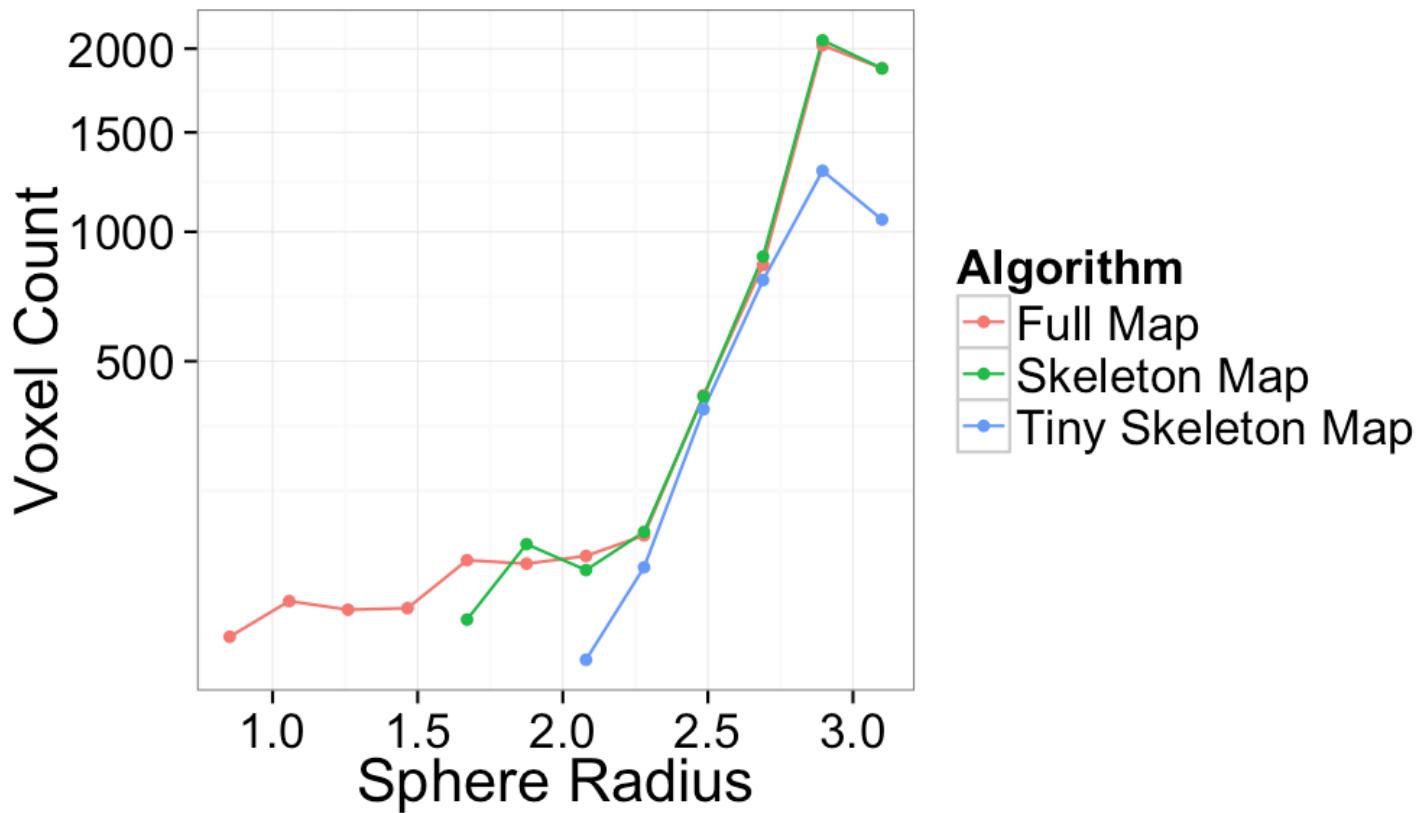
```
##    user  system elapsed
##  2.713   0.068   2.817
```

+/- R Code

## Thickness from Skeleton Slope>1.25 & Dist>1.7



+/- R Code

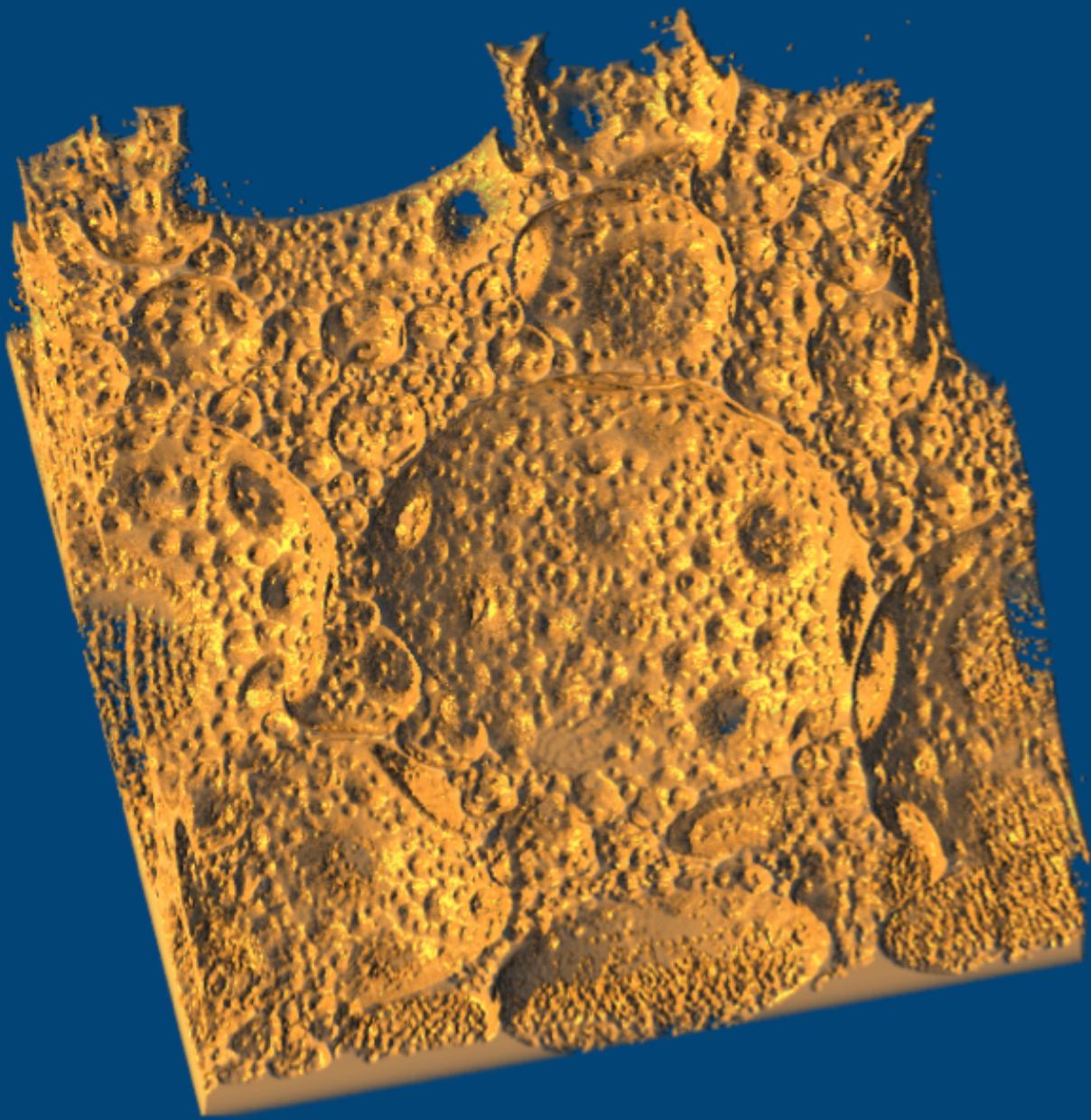


+/- R Code

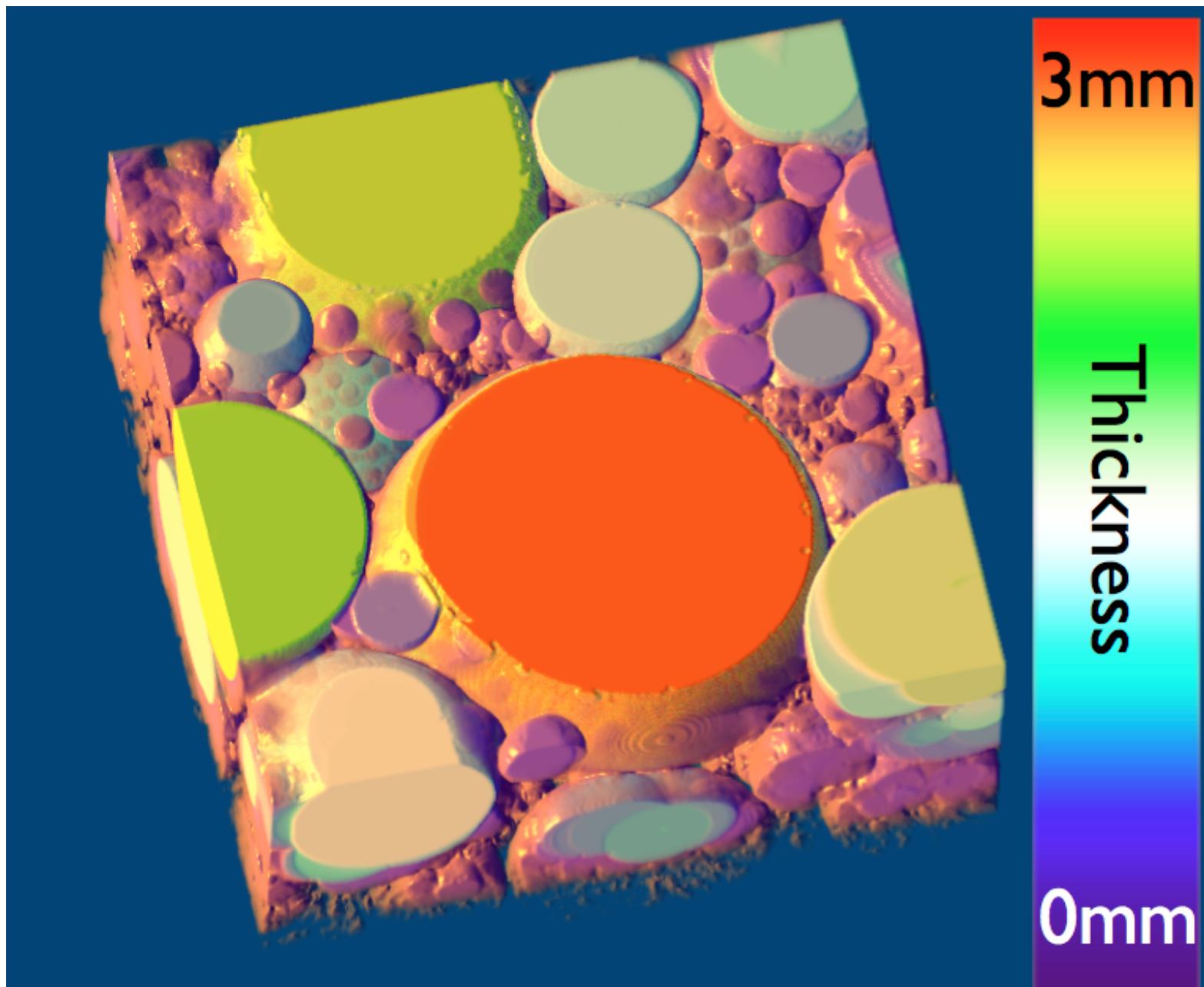
| id      | Full Map | Skeleton Map | Tiny Skeleton Map |
|---------|----------|--------------|-------------------|
| Min.    | 0.75     | 1.75         | 2.06              |
| 1st Qu. | 2.70     | 2.70         | 2.70              |
| Median  | 2.83     | 2.83         | 2.83              |
| Mean    | 2.82     | 2.84         | 2.84              |
| 3rd Qu. | 3.01     | 3.01         | 3.01              |
| Max.    | 3.20     | 3.20         | 3.20              |

## Thickness in 3D Images

While the examples and demonstrations so far have been shown in 2D, the same exact technique can be applied to 3D data as well. For example for this liquid foam structure



- The thickness can be calculated of the background (air) voxels in the same manner.
- With care, this can be used as a proxy for bubble size distribution in systems where all the bubbles are connected to it is difficult to identify single ones.

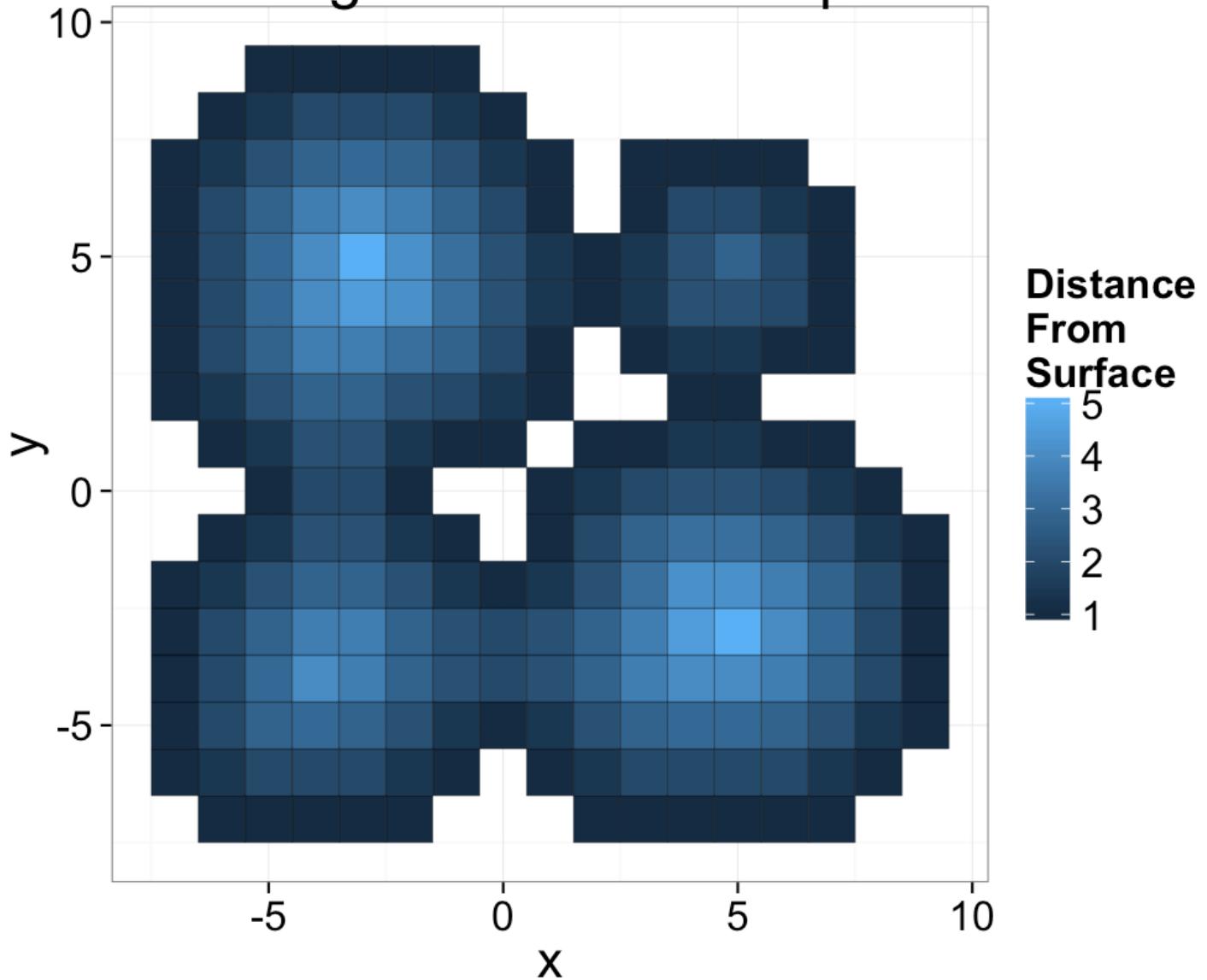


## Watershed

Watershed is a method for segmenting objects without using component labeling.

- It utilizes the shape of structures to find objects
- From the distance map we can make out substructures with our eyes
- But how do we find them?!

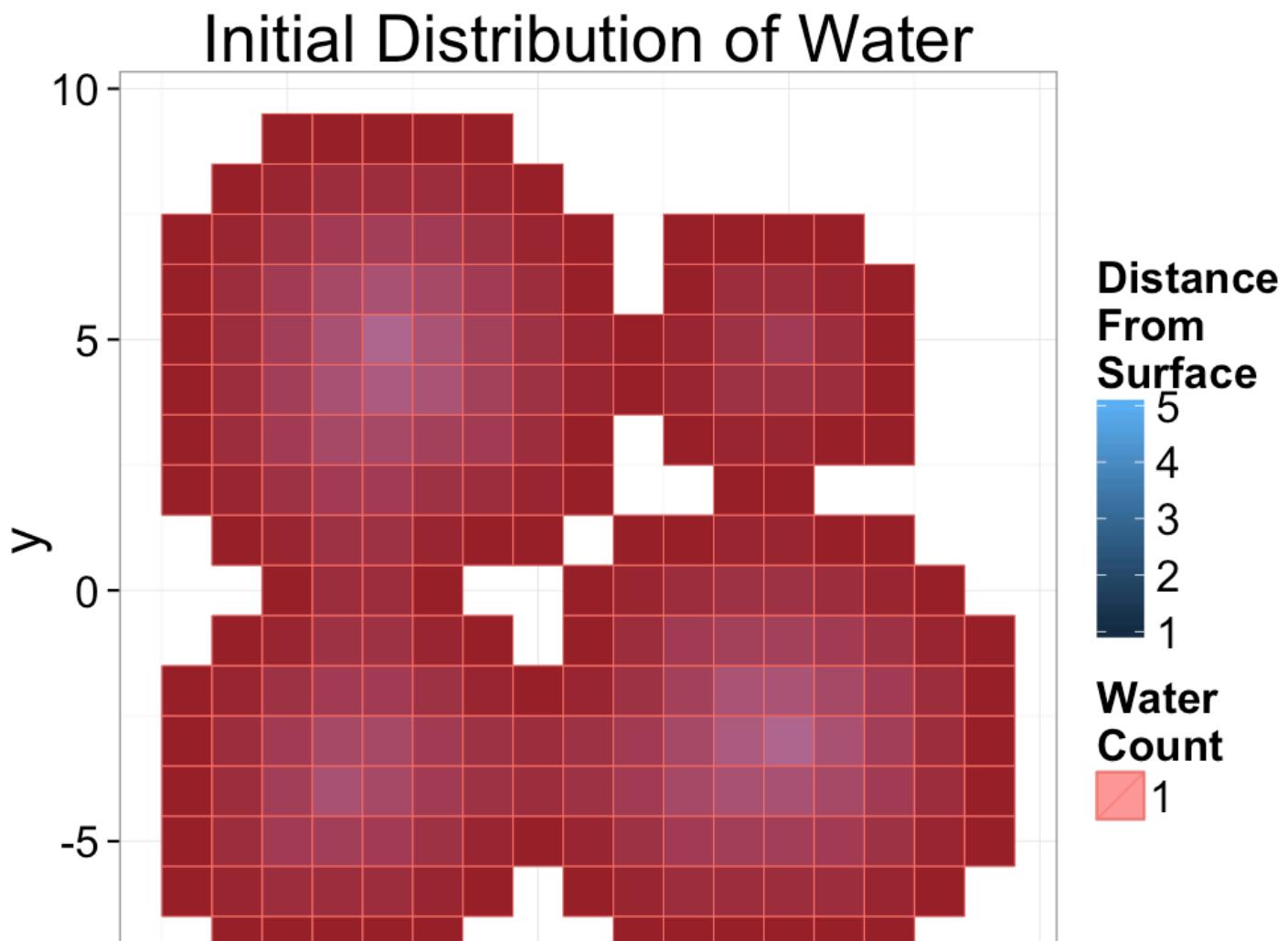
## Original Distance Map

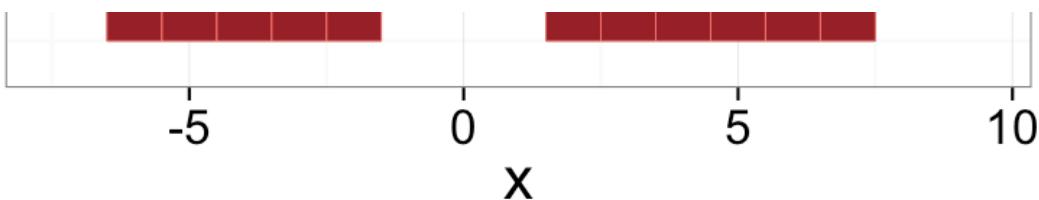


**+/- R Code**

# Watershed: Flowing Downhill

So we start by distributing pixels all across the image

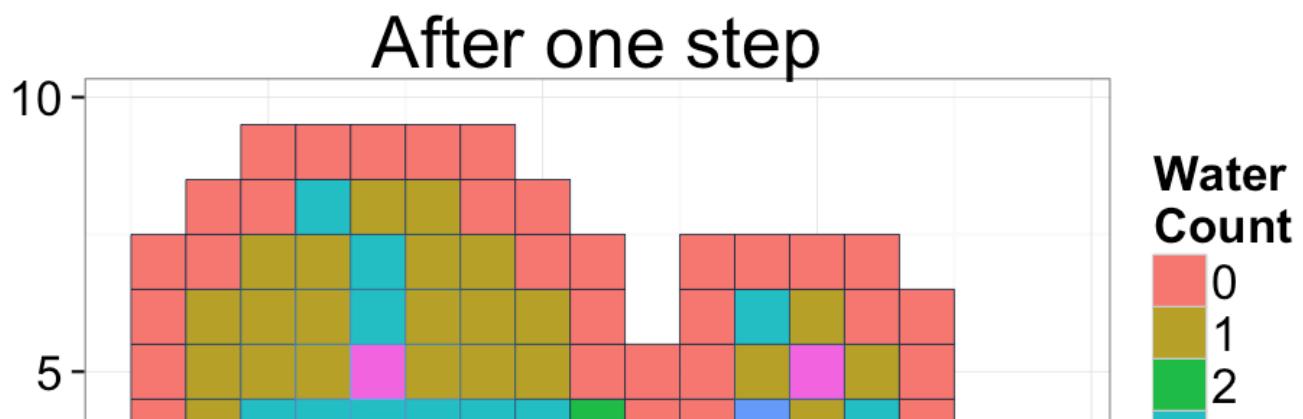


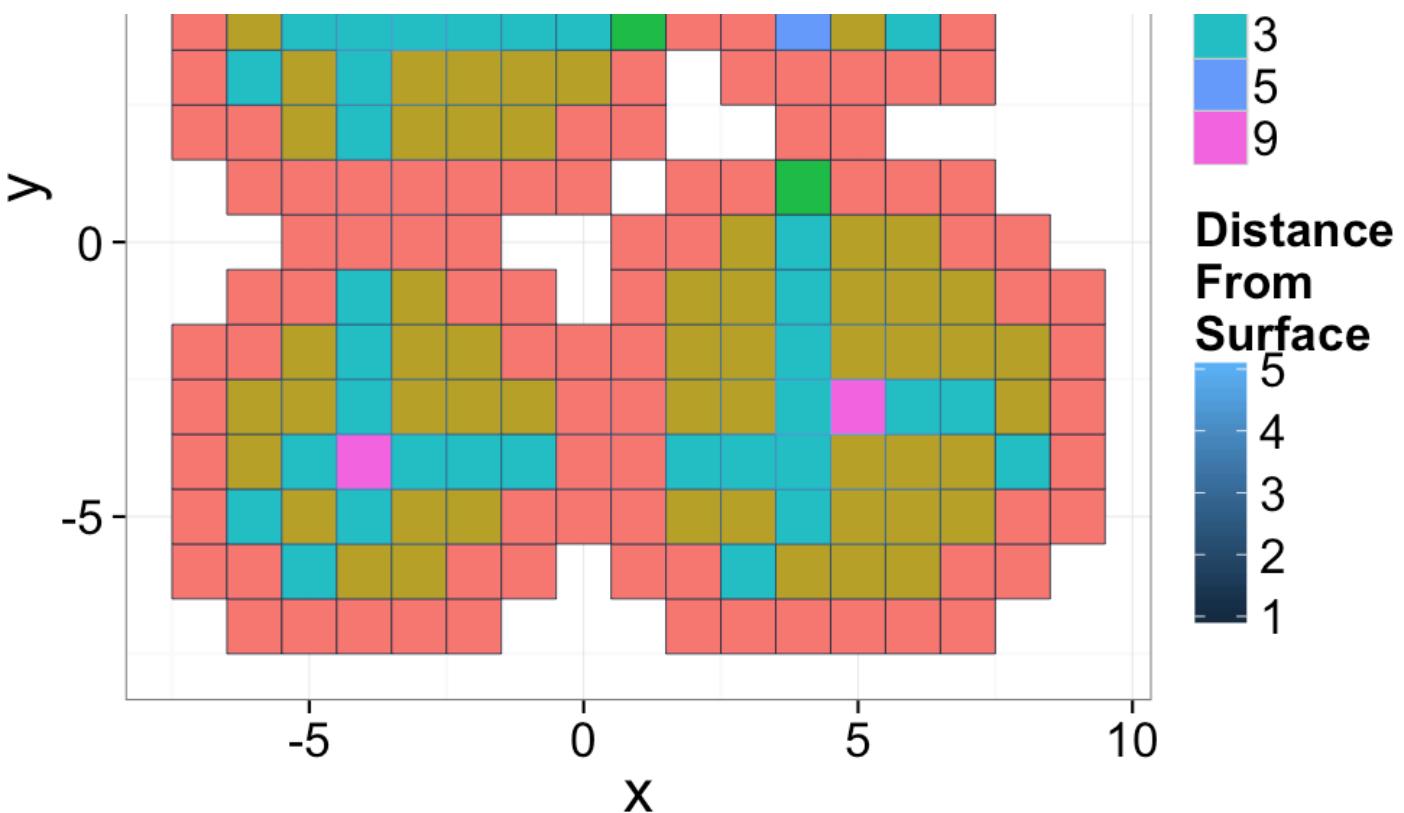


+/- R Code

If any of the neighbors have a higher distance (more downhill) then move to that position

+/- R Code





+/- R Code

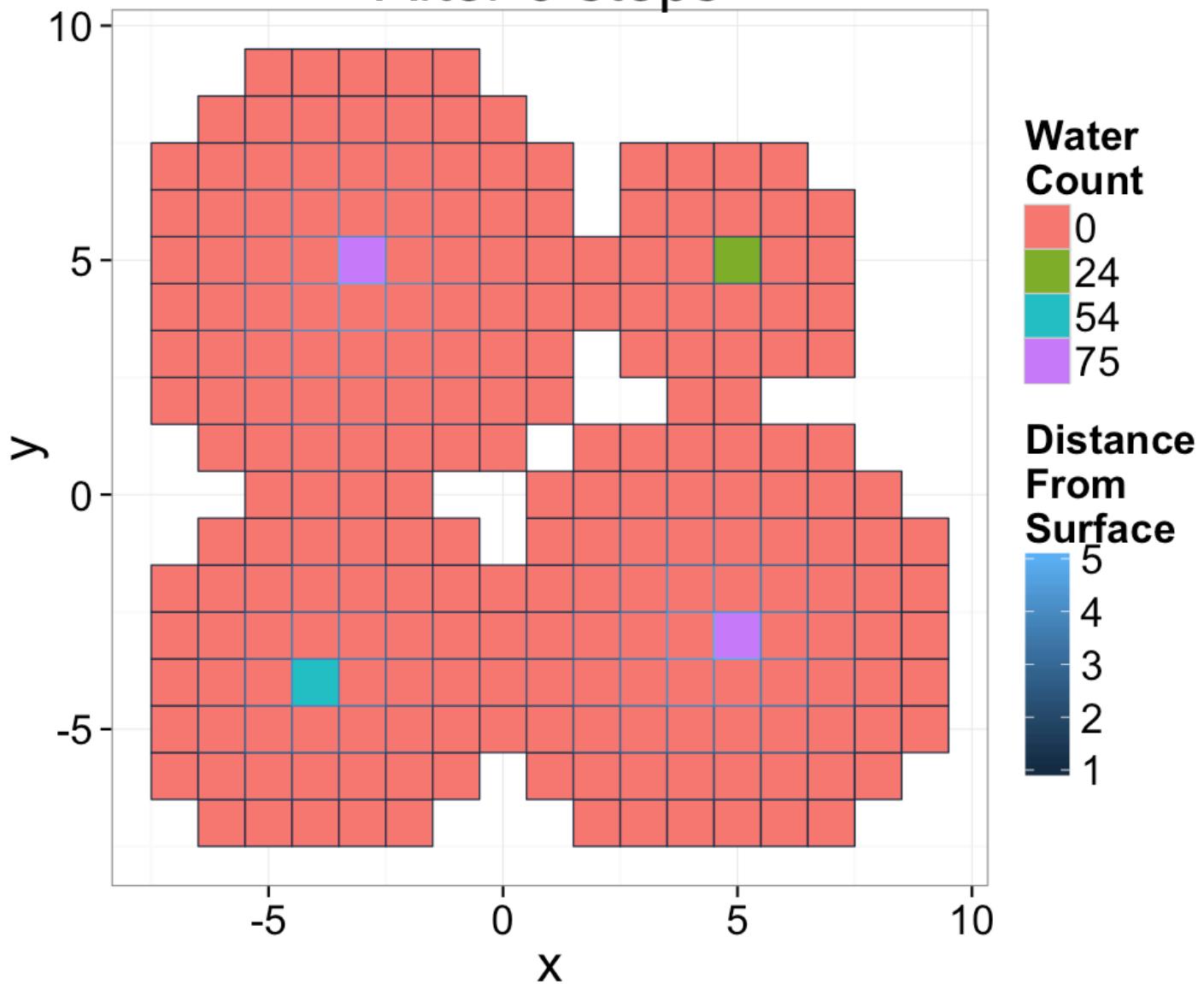
## Watershed: More flowing

## After 2 steps



+/- R Code

After 6 steps

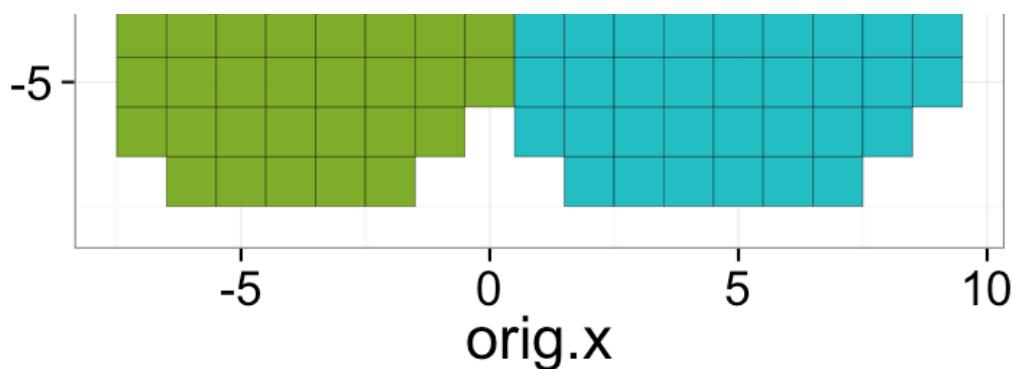


**+/- R Code**

# Watershed: Regrowing

We can then take the points from these basins and regrow them back to their original size and these represent the watershed regions of our image



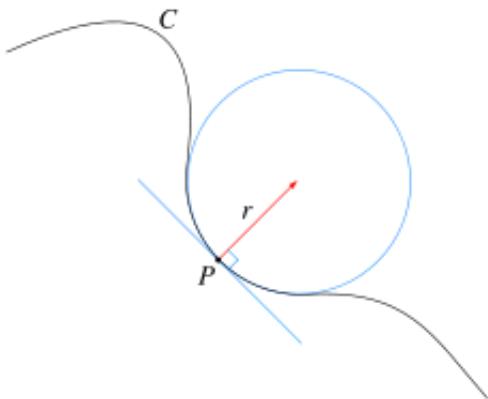


+/- R Code

## Curvature

Curvature is a metric related to the surface or interface between phases or objects.

- It is most easily understood in its 1D sense or being the radius of the circle that matches the local shape of a curve



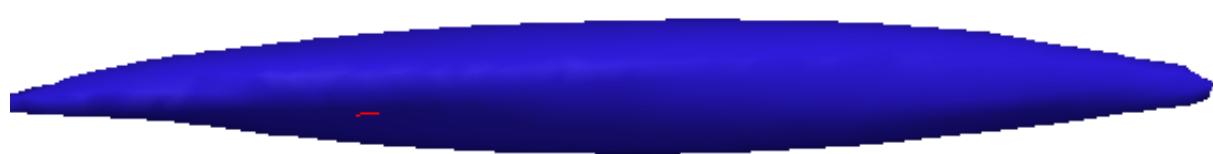
- Mathematically it is defined as

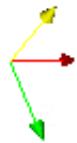
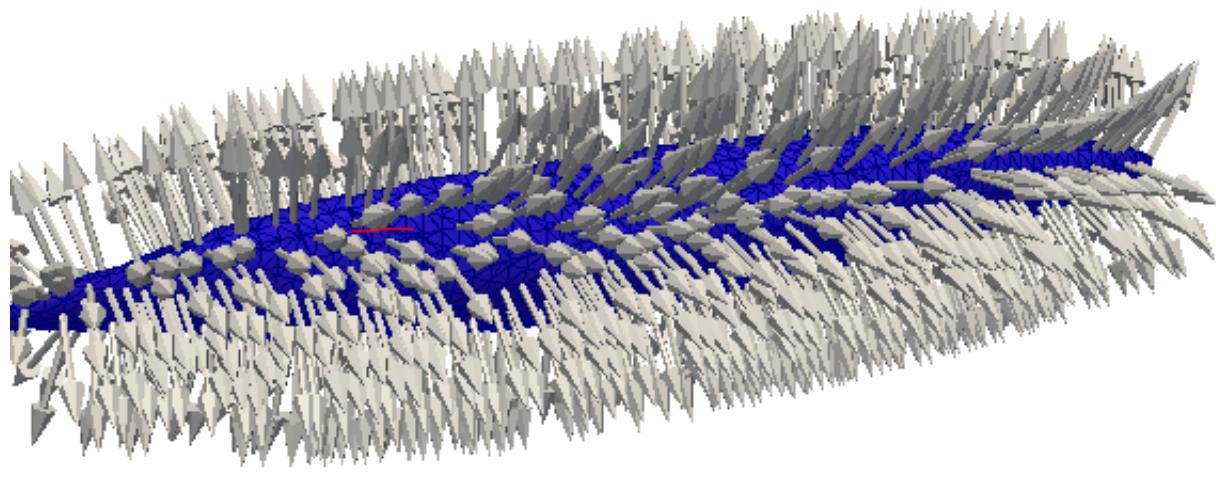
$$\kappa = \frac{1}{R}$$

- Thus a low curvature means the value means a very large radius and a small curvature means a very small radius

# Curvature: Surface Normal

In order to meaningfully talk about curvatures of surfaces, we first need to define a consistent frame of reference for examining the surface of an object. We thus define a surface normal vector as a vector oriented orthogonally to the surface away from the interior of the object →  $\vec{N}$





## Curvature: 3D

With the notion of surface normal defined ( $\vec{N}$ ), we can define many curvatures at point  $\vec{x}$  on the surface.

- This is because there are infinitely many planes which contain both point  $\vec{x}$  and  $\vec{N}$
- More generally we can define an angle  $\theta$  about which a single plane containing both can be freely rotated
- We can then define two principal curvatures by taking the maximum and minimum of this curve

$$\kappa_1 = \max(\kappa(\theta))$$

$$\kappa_2 = \min(\kappa(\theta))$$

## Mean Curvature

The mean of the two principal curvatures

$$H = \frac{1}{2} (\kappa_1 + \kappa_2)$$

## Gaussian Curvature

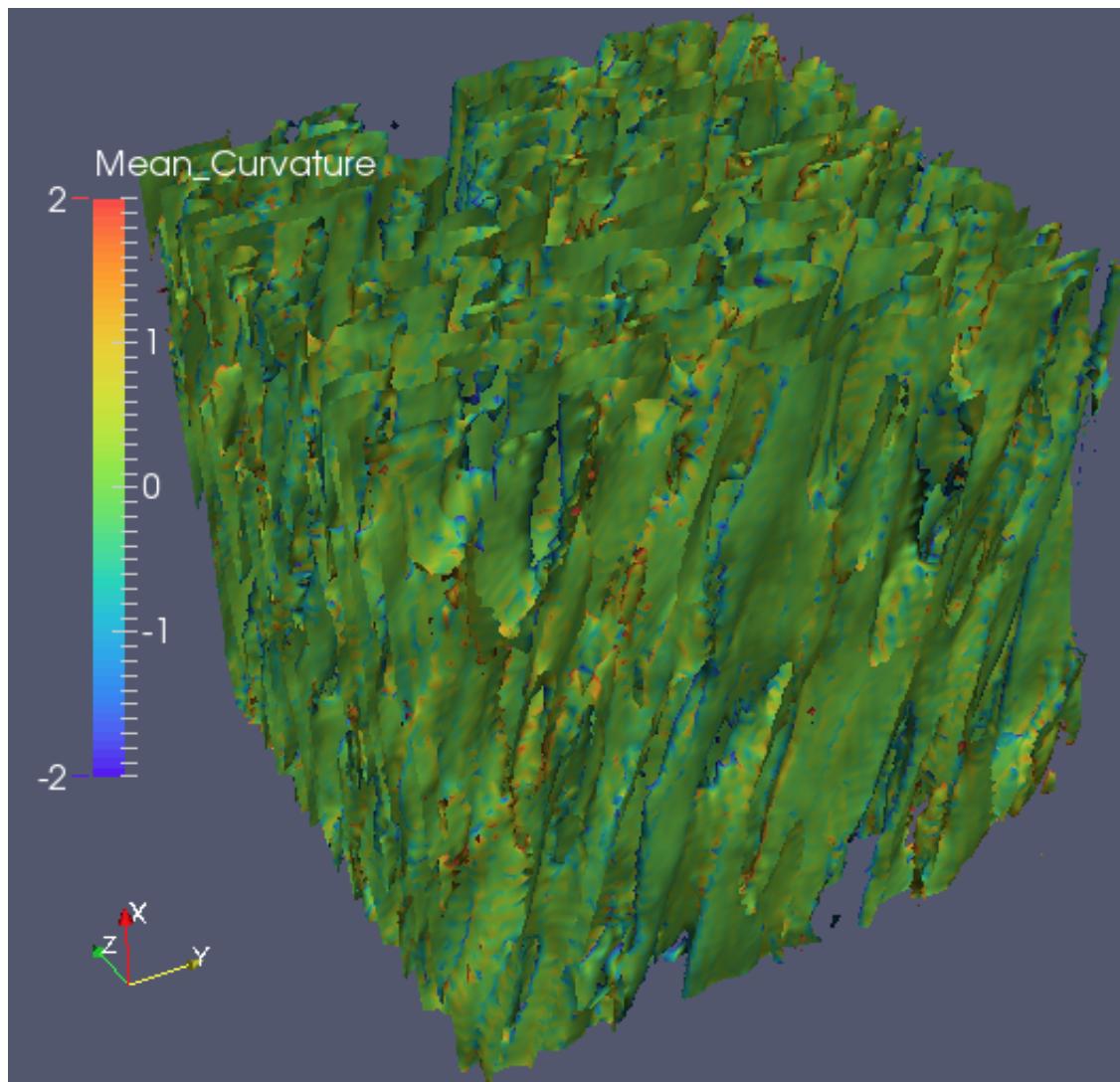
The mean of the two principal curvatures

$$K = \kappa_1 \kappa_2$$

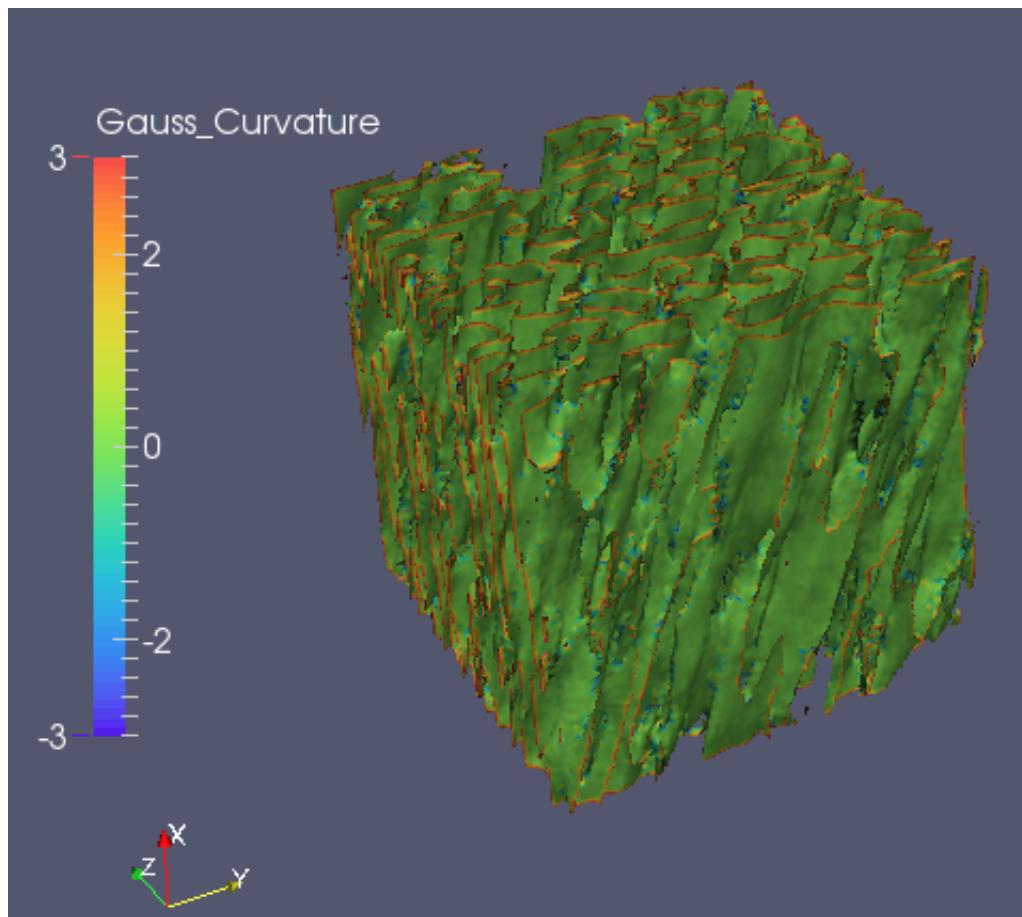
- positive for spheres (or spherical inclusions)
  - curvatures agree in sign
- negative for saddles (hyperboloid surfaces)
  - curvatures disagree in sign
- 0 for planes

## Curvature: 3D Examples

Examining a complex structure with no meaningful ellipsoidal or watershed model. The images themselves show the type of substructures and shapes which are present in the sample.



- gaussian curvature: the comparative amount of surface at, above, and below 0
  - from spherical particles into annealed mesh of balls



## Characteristic Shape

Characteristic shape can be calculated by measuring principal curvatures and normalizing them by scaling to the structure size. A distribution of these curvatures then provides shape information on a structure independent of the size.

For example a structure transitioning from a collection of perfectly spherical particles to a annealed solid will go from having many round spherical faces with positive gaussian curvature to many saddles and more complicated structures with 0 or negative curvature.

## Curvature: Take Home Message

It provides another metric for characterizing complex shapes

- Particularly useful for examining interfaces
  - Folds, saddles, and many other types of points are not characterized well by ellipsoids or thickness maps
- Provides a scale-free metric for assessing structures
- Can provide visual indications of structural changes

# Other Techniques

There are hundreds of other techniques which can be applied to these complicated structures, but they go beyond the scope of this course. Many of them are model-based which means they work well but only for particular types of samples or images. Of the more general techniques several which are easily testable inside of FIJI are

- Directional Analysis = Looking at the orientation of different components using Fourier analysis (*Analyze → Directionality*)
  - Tubeness / Surfaceness (*Plugins → Analyze →*) characterize binary images and the shape at each point similar to curvature but with a different underlying model
- 
- Fractal Dimensionality = A metric for assessing the structure as you scale up and down by examining various spatial relationships
    - Ma, D., Stoica, A. D., & Wang, X.-L. (2009). Power-law scaling and fractal nature of medium-range order in metallic glasses. *Nature Materials*, 8(1), 30–4. doi:10.1038/nmat2340
  - Two (or more) point correlation functions = Used in theoretical material science and physics to describe random materials and can be used to characterize distances, orientations, and organization in complex samples
    - Jiao, Y., Stillinger, F., & Torquato, S. (2007). Modeling heterogeneous materials via two-point correlation functions: Basic principles. *Physical Review E*, 76(3). doi:10.1103/PhysRevE.76.031110
    - Andrey, P., Kiêu, K., Kress, C., Lehmann, G., Tirichine, L., Liu, Z., ... Debey, P. (2010). Statistical analysis of 3D images detects regular spatial distributions of centromeres and chromocenters in animal and plant nuclei. *PLoS Computational Biology*, 6(7), e1000853. doi:10.1371/journal.pcbi.1000853
    - Haghpanahi, M., & Miramini, S. (2008). Extraction of morphological parameters of tissue engineering scaffolds using two-point correlation function, 463–466. Retrieved from <http://portal.acm.org/citation.cfm?id=1713360.1713456> (<http://portal.acm.org/citation.cfm?id=1713360.1713456>)