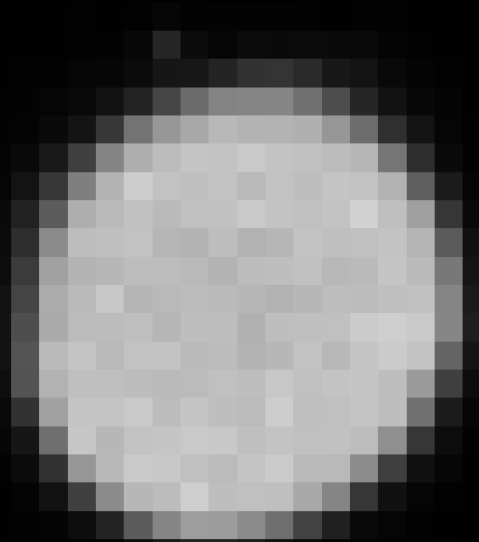


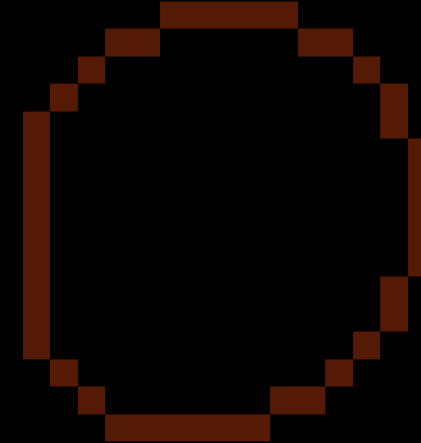
Points & Surfaces

Johannes Müller

Droplet in 2d



Binarized droplet



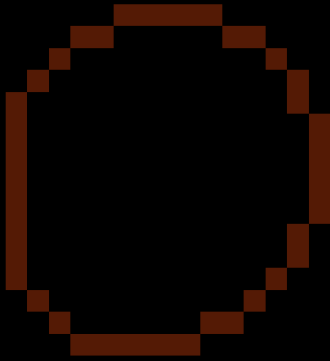
Inherent limitation of image data as arrays:

- Biology does not have edges at pixels
- Measurements get distorted by pixel artifacts – especially for smaller objects

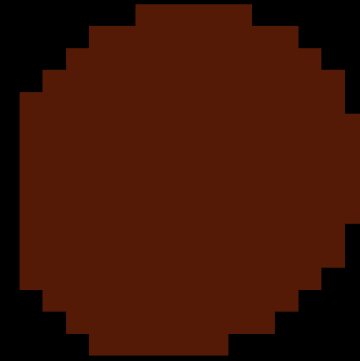
Limitations of image data as arrays:

- **Small objects:** Biology does not have edges at pixels
- **Small objects:** Measurements get distorted by pixel artifacts
- **Large objects:** Storing raw and processed images requires huge storage!
- **Large objects after segmentation:** Most of the pixels/voxels carry no useful information

Useful pixels for shape



Actually stored pixels in label



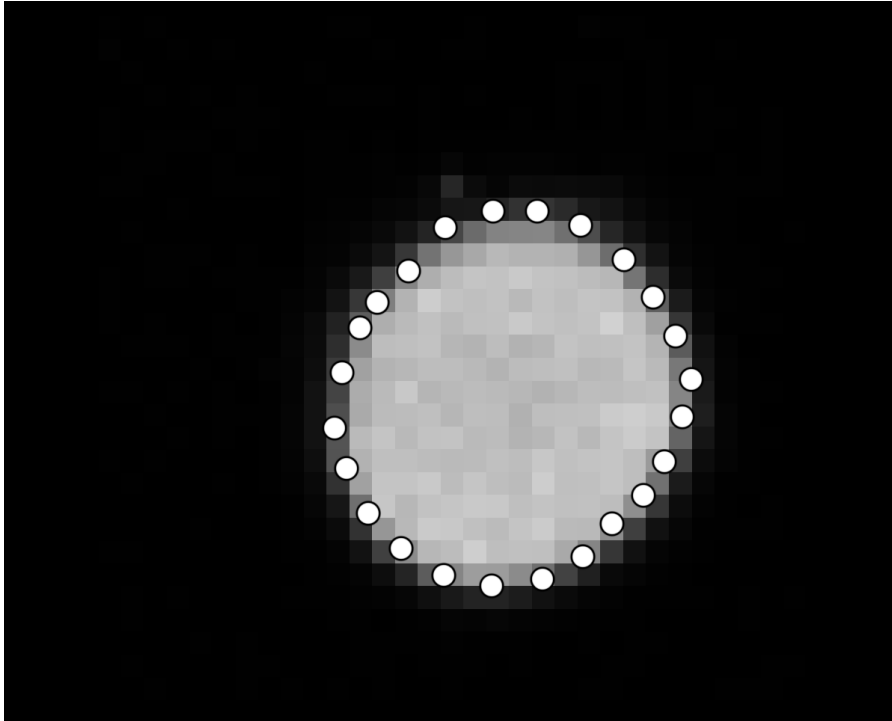
1000x1000x1000
Intensity Image



1000x1000x1000
Binary image



1000x1000x1000
Label image



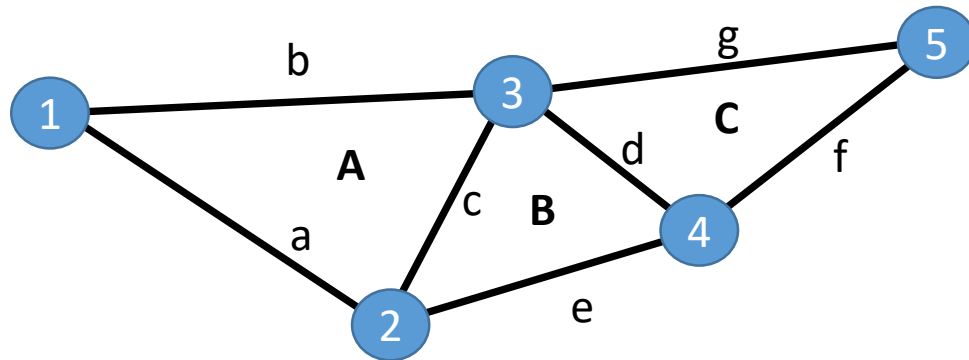
Points:

- Represent shape of objects by points on surface
- Points can be between pixels!
- Coordinates have **physical units**

```
array([[13.1896702, 10.97073578],  
       [15.15516485, 10.18453792],  
       [17.57927492, 9.85695548],  
       [19.34822011, 10.38108738],  
       [21.31371476, 11.33107646],  
       [22.85335224, 12.77243921],  
       [24.03264903, 14.63965913],  
       [24.49126445, 16.73618675],  
       [24.19644025, 18.96374736],  
       [23.21369292, 20.73269255],  
       [21.77233018, 22.01026407],  
       [20.5275169, 23.38611032],  
       [19.05339591, 24.30334116],  
       [17.08790126, 25.08953902],  
       [15.44998905, 25.48263795],  
       [13.55001089, 24.79471482],
```

Surfaces/Meshes:

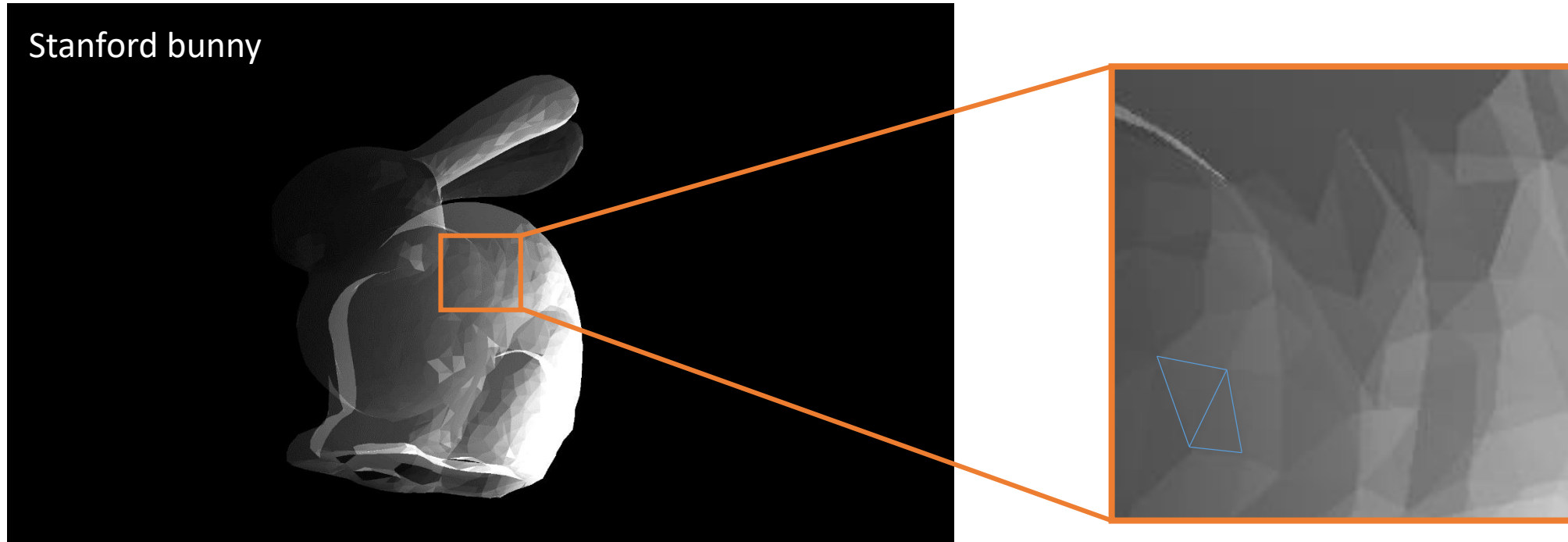
- Connect neighboring points with graph structure



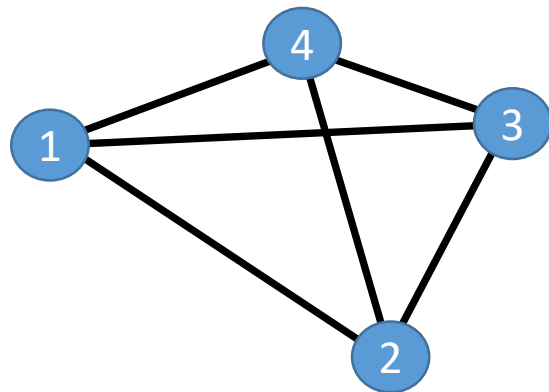
Meshes consist of vertices, edges and faces:

- Vertices (1, 2, 3, 4, 5)
- Edges (a, b, c, d, e, f, g)
- Faces (A, B, C)

Stanford bunny



Mesh data structure:



Point x	Point y	Point z
x_1	y_1	z_1
x_2	y_2	z_2
x_3	y_3	z_3
x_4	y_4	z_4
...

+

Point 1	Point 2	Point 3
1	2	3
1	2	4
2	3	4
1	3	4

For simplicity: Only triangles

How many vertices would a pyramid have as a mesh object?

3

4

5

6

How many edges would a pyramid have as a mesh object?

5

6

7

8

How many faces would a pyramid have as a mesh object?

5

6

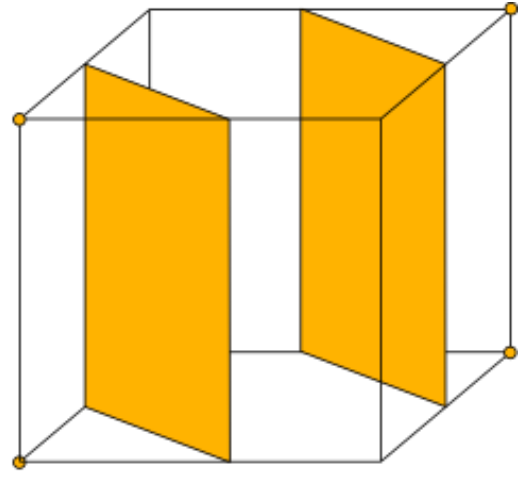
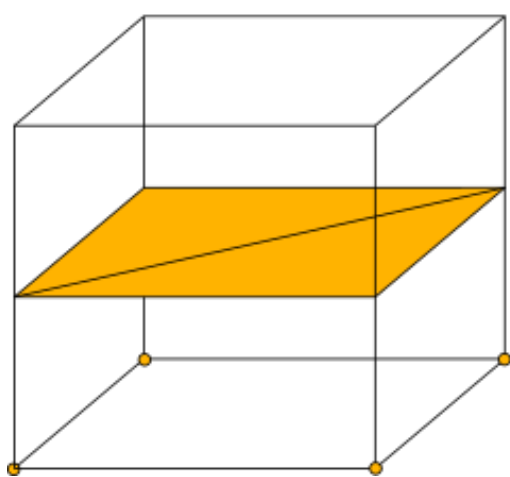
7

8

Marching cubes: Converts label image into surface (Cline & Lorensen, 1985)

→ Slide cube over image (similar to filter concept)

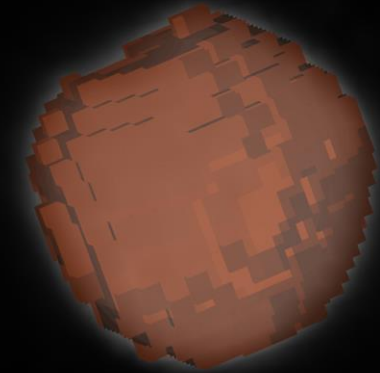
→ Are cube corners inside or outside of the label object?

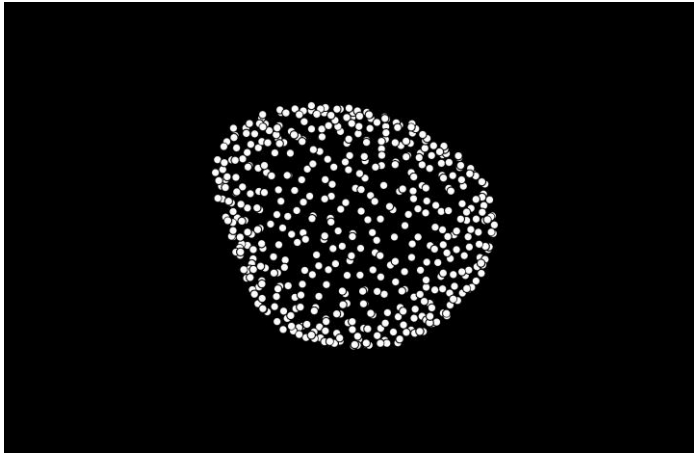


Symmetries: Currently
33 unique topologies
of arranging polygons
inside cube
(Chernyaev, 1995)

→ Finite possibilities of how surface planes can be oriented
with respect to the label surface

Droplet as label and surface

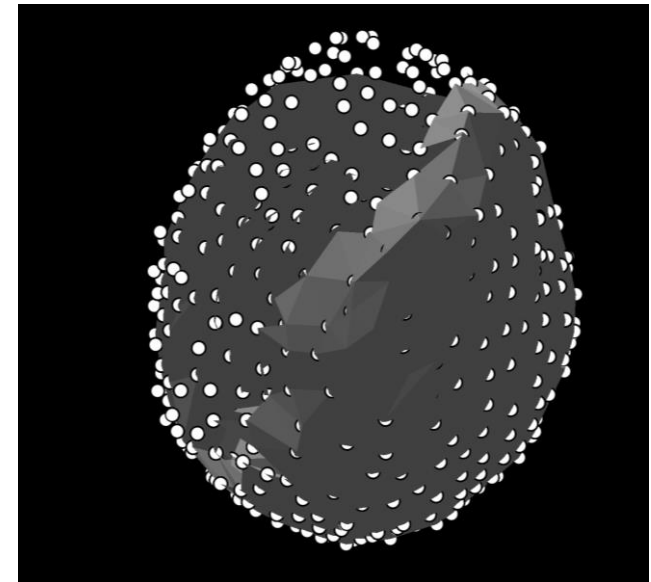
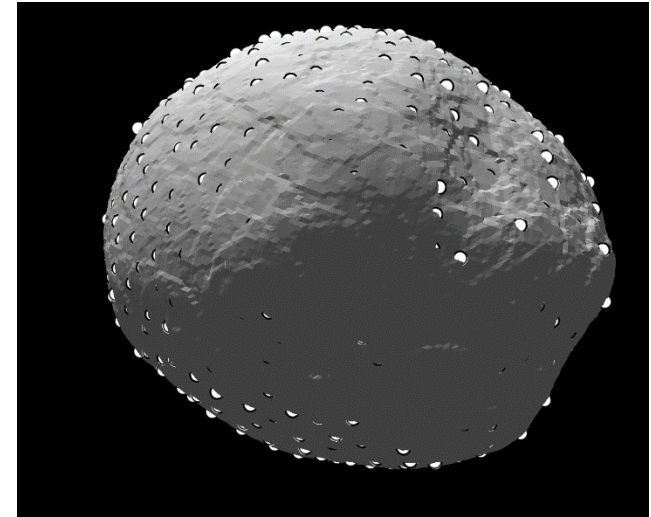
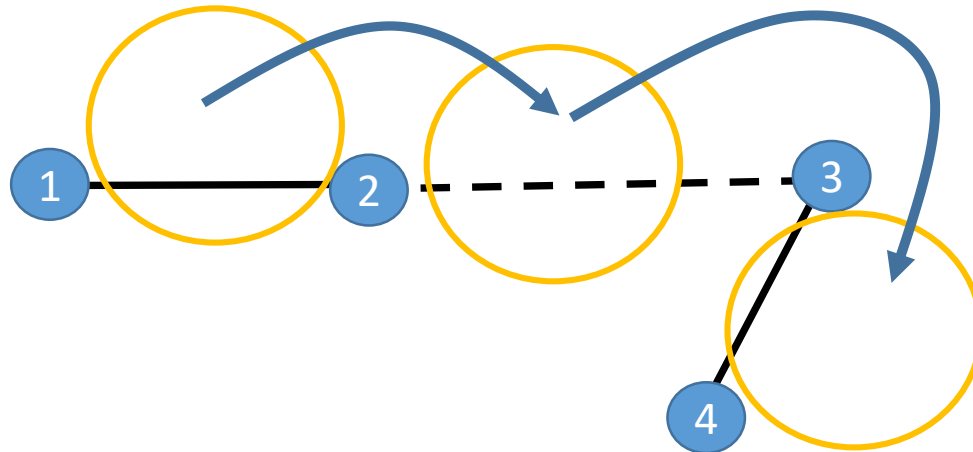


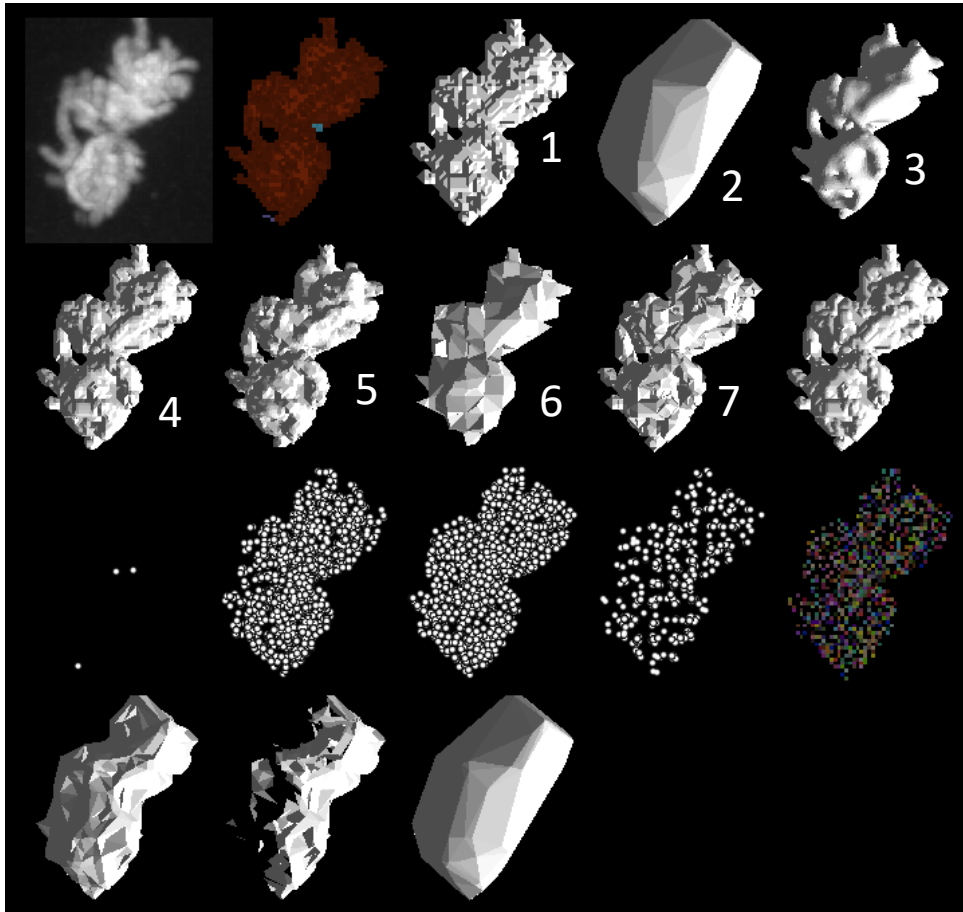


Surface reconstruction algorithms:

- Signed distance + flying edges
 - Calculate distance to nearest point at sample location
 - Fit curve to zero-crossings
 - Surface vertices approximated

- Ball pivoting
 - Rolling ball over points
 - Preserves original points
 - Can lead to holes





Source: <https://github.com/haesleinhuepf/napari-process-points-and-surfaces>

Typical mesh processing algorithms

- Reconstruction (1)
- Convex hull (2)
- Mesh smoothing (simple) (3)
- Mesh smoothing (Laplacian) (4)
- Mesh smoothing (taubin) (5)
- Mesh simplification I (6)
- Mesh simplification II (7)

Further operations:

- Connected components
- Hole filing

Standard shape features

- Volume, surface area (perimeter), elongation

```
print('Surface area:', mesh.area())  
print('Volume:', mesh.volume())
```

Surface area: 5.646863245482599
Volume: 0.7620278501410354

- Asphericity: $\frac{\pi^{\frac{1}{3}}(6V)^{\frac{2}{3}}}{A}$
(V: Volume, A: Surface area)
→ measure of compactness
→ This is actually just

Circularity in 3D

Roundness in 3D

Solidity in 3D

- Face-wise features: Local surface shape descriptors

- EDGE_RATIO, 0
- ASPECT_RATIO, 1
- RADIUS_RATIO, 2
- ASPECT_FROBENIUS, 3
- MED_ASPECT_FROBENIUS, 4
- MAX_ASPECT_FROBENIUS, 5
- MIN_ANGLE, 6
- COLLAPSE_RATIO, 7
- MAX_ANGLE, 8
- CONDITION, 9
- SCALED_JACOBIAN, 10
- SHEAR, 11
- RELATIVE_SIZE_SQUARED, 12
- SHAPE, 13
- SHAPE_AND_SIZE, 14
- DISTORTION, 15
- MAX_EDGE_RATIO, 16
- SKEW, 17
- TAPER, 18
- VOLUME, 19
- STRETCH, 20
- DIAGONAL, 21
- DIMENSION, 22
- ODDY, 23
- SHEAR_AND_SIZE, 24
- JACOBIAN, 25
- WARPAGE, 26
- ASPECT_GAMMA, 27
- AREA, 28
- ASPECT_BETA, 29

Measurable features: fancy

Geodesic distances:

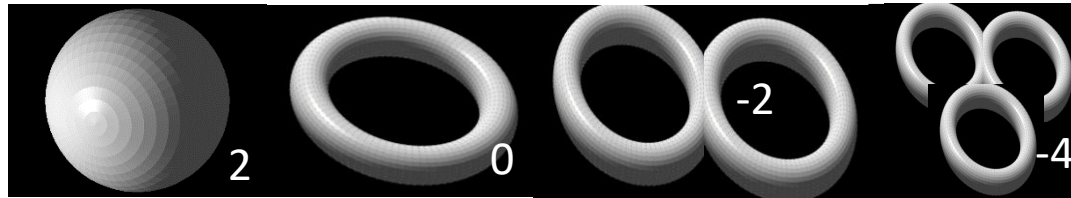
- Shortest distance on surface between two points
- Can be found with Dijkstra algorithm

Surface curvature

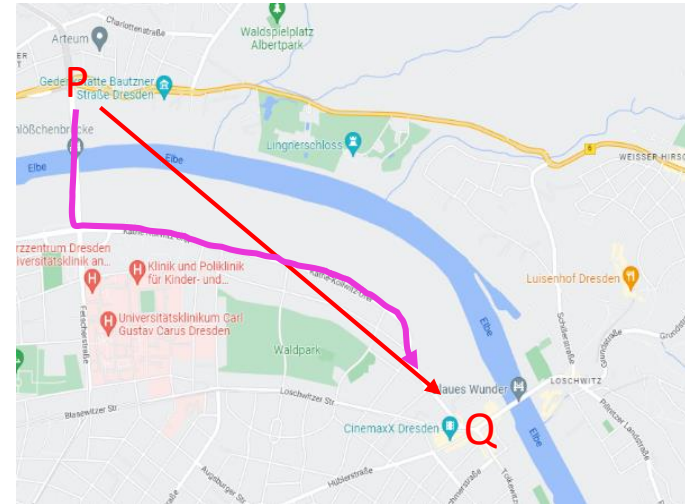
- Mean curvature: $[0, \dots]$
- Gaussian curvature: $[-\dots, \dots]$ (identifies ridge regions)

Euler characteristics:

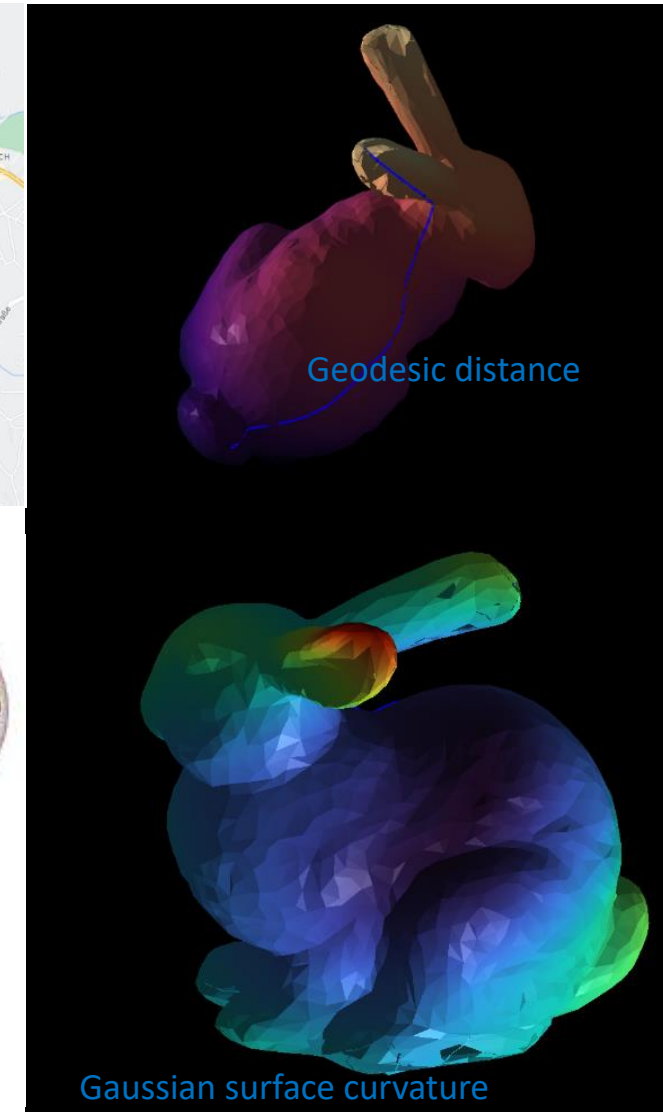
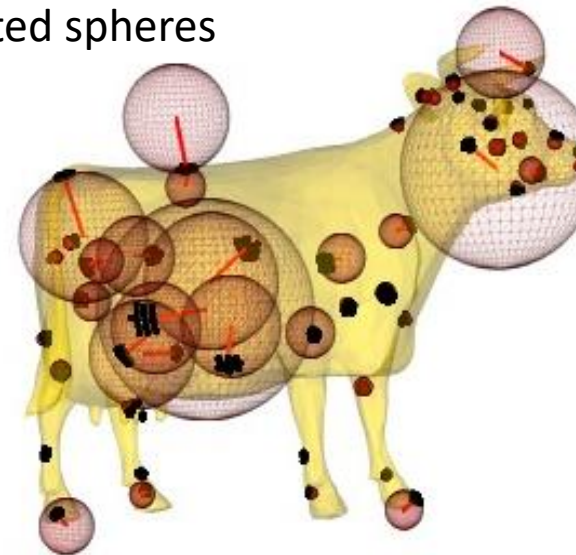
- Distinguish topologies of objects



- Euler characteristic $\chi = n_{\text{points}} - n_{\text{edges}} + n_{\text{faces}}$

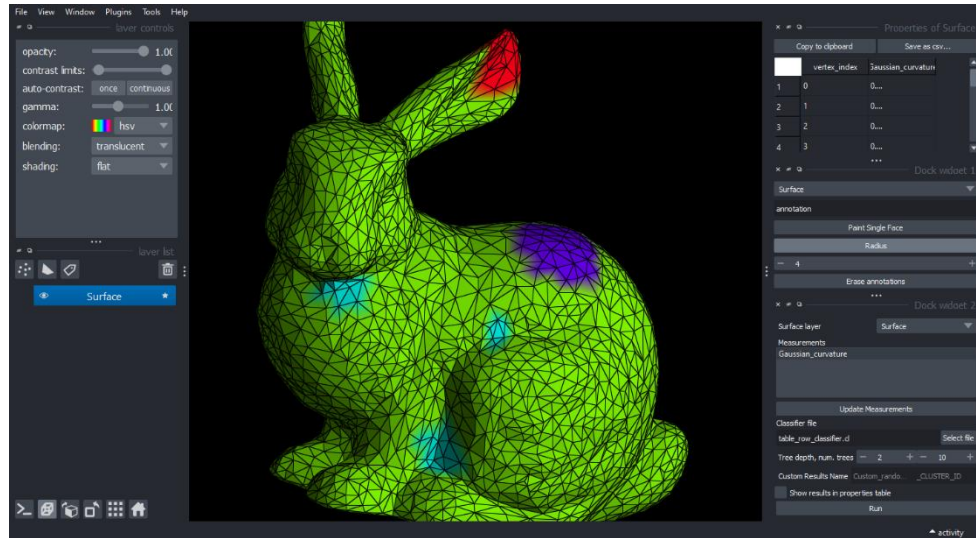


Fitted spheres

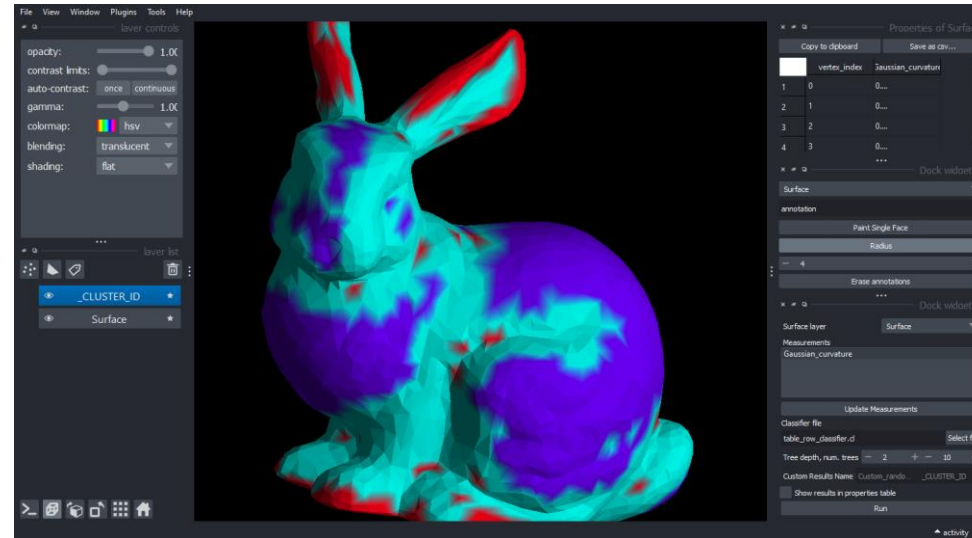
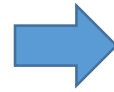


Where there are features...

There is data science!



Annotated surface regions



Prediction on surface

Recent feature: SVeC (surface vertex classification)

- Interactively annotate surface vertices
- Predict class membership of all other vertices



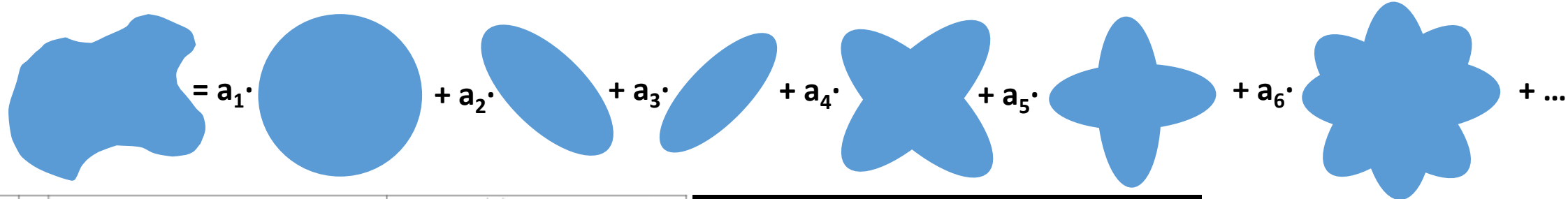
Coming soon:















- Dimensionality reduction on meshes
- Clustering on meshes

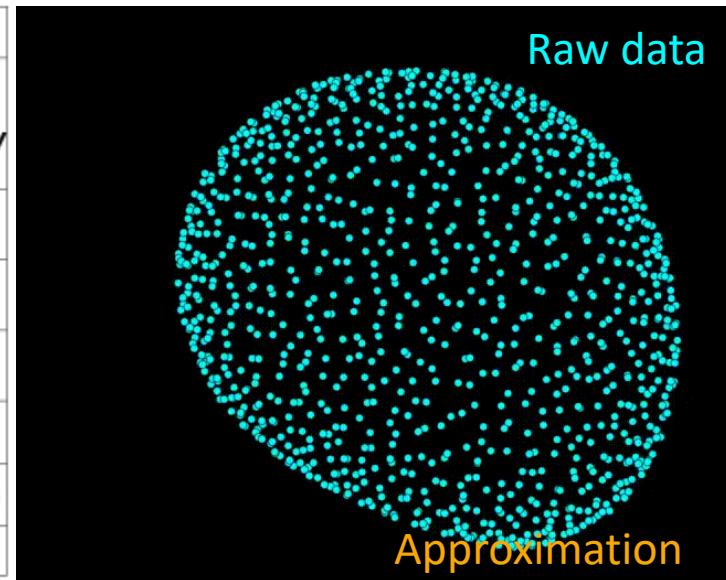
Spherical harmonics

Basic idea:

→ Express object shape as superposition of multiple elemental shapes



l:		$P_l^m(\cos \theta) \cos(m\varphi)$	$P_l^{ m }(\cos \theta) \sin(m \varphi)$
0	s		
1	p		
2	d		
3	f		
4	g		
5	h		
6	i		
m:		6 5 4 3 2 1 0	-1 -2 -3 -4 -5 -6



→ Final object shape described by spherical harmonics coefficients

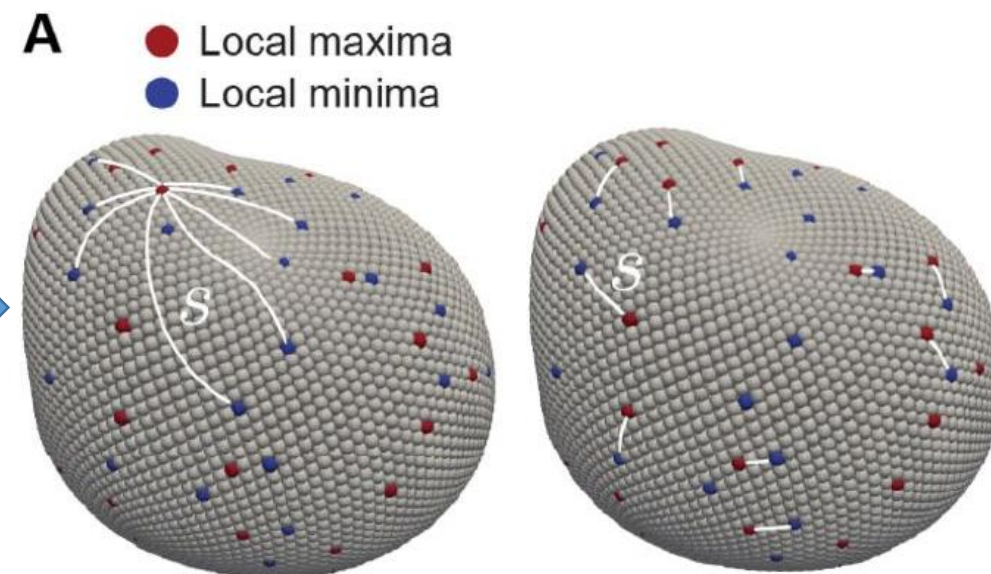
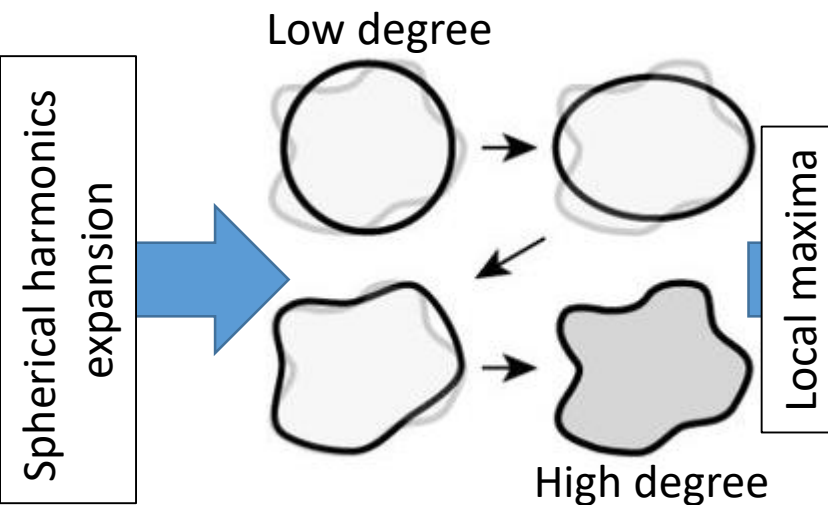
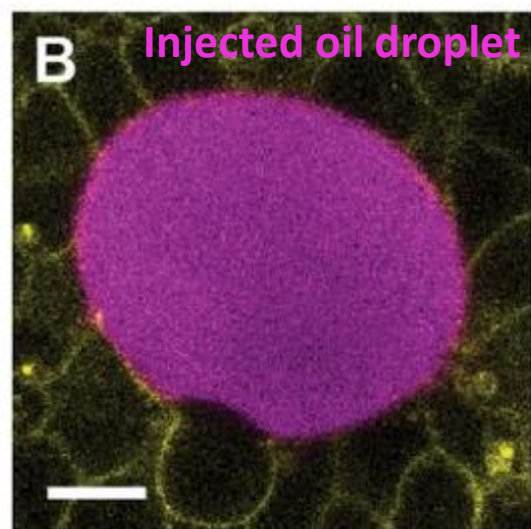
Historical background:

- Describe probability distributions of electron location in nuclear shell
- First described in 1782

<https://commons.wikimedia.org/wiki/File:Sphericalfunctions.svg>

→ Leads to the concept of **shape-spectra** of objects

Application: Measure quantitative tissue stresses



- Inject droplet of known interfacial tension into tissue
- Reconstruct surface
- Spherical harmonics expansion
- Calculate surface curvature

Local maxima

↕

tight junctions of neighboring cells

Source: Gross et al. (2021)

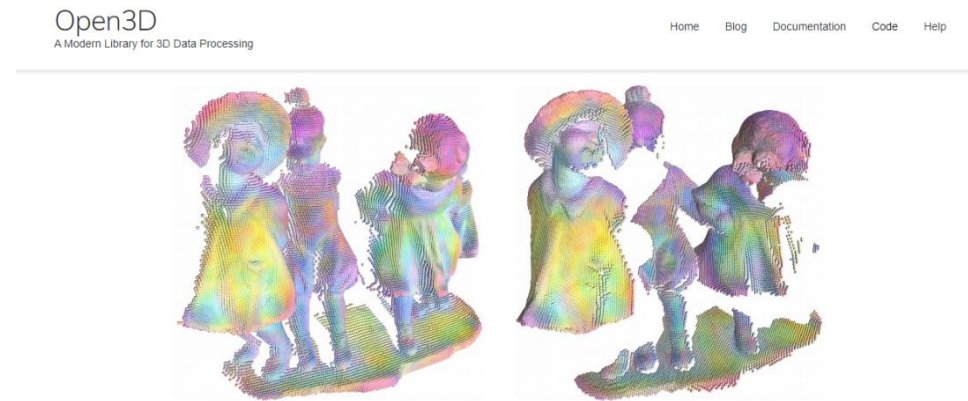
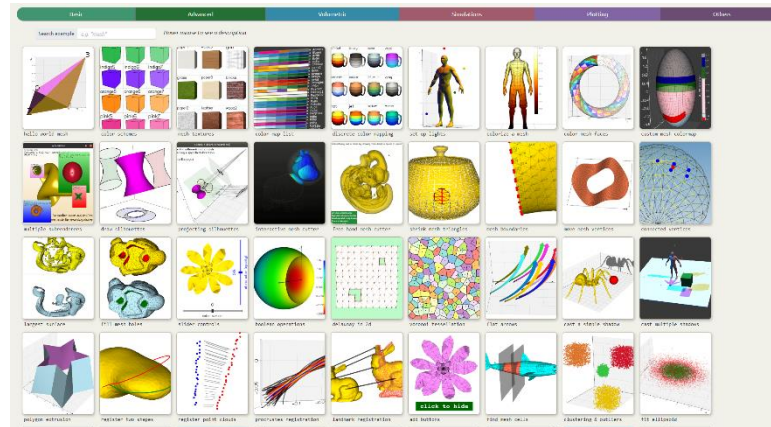
<https://www.biorxiv.org/content/10.1101/2021.03.26.437148v1.full>

- **Lightweight:** Label image of 0.5GB = Surface data of ~7MB
- **Speed:** Mesh structures are fast to render and interact visually
- **Quantitative measurements:** Using Mesh structures capture the shape of objects on a more abstract levels
- **3D-printable:** Meshes are the native format for 3D printers!



3D printed cell nucleus from cryo-EM

Relevant Python packages: Vedo, open3d



- **Downside:** Many python packages use their own “Mesh” data classes

<https://vedo.embl.es/>
<http://www.open3d.org/>

```
viewer.add_surface(bunny)
```

```
viewer.add_points(points_array, size=0.01, face_color='red')
```