PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

GPU-accelerated image processing

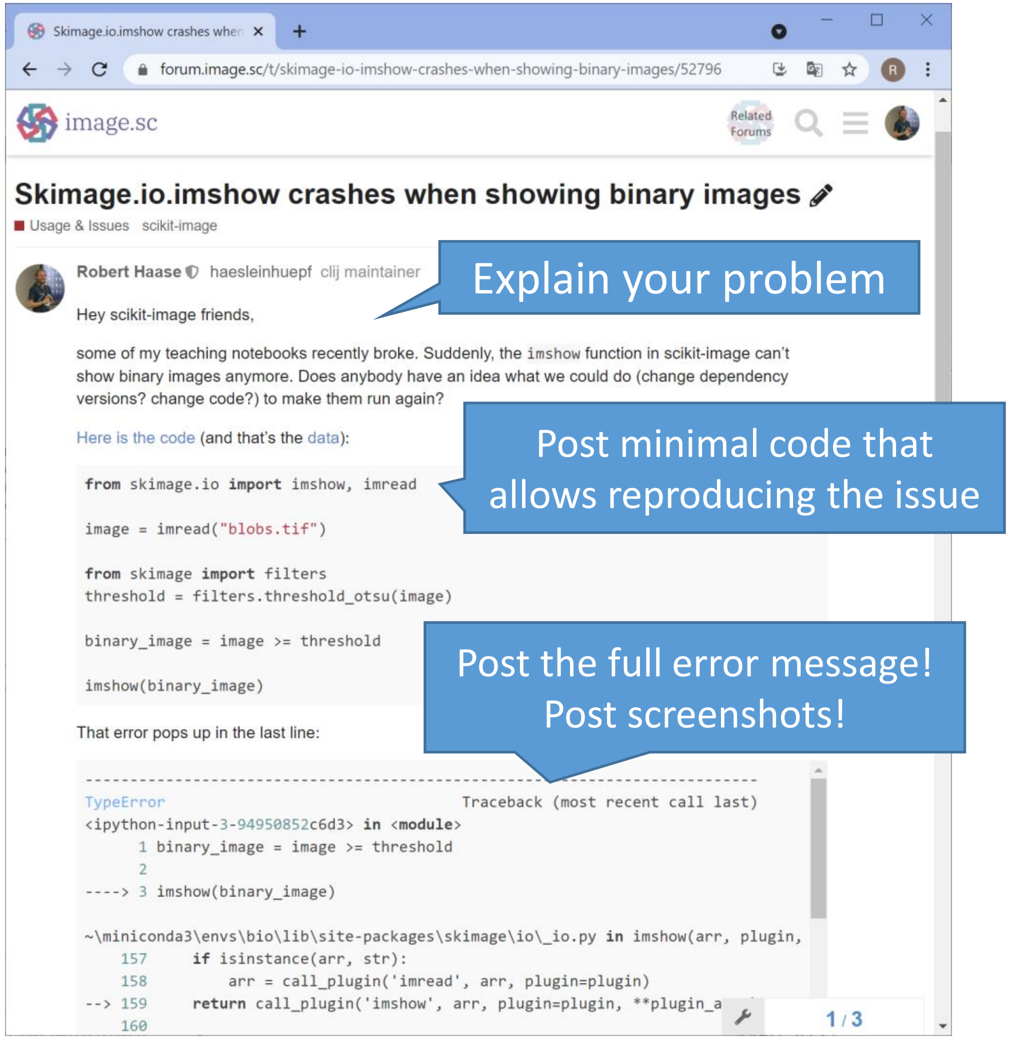
Robert Haase

May 2021

@haesleinhuepf

The image science community: https://image.sc

- In case you run in trouble with image analysis, observe bugs in open-source software or want to know how to analyze your image data: <u>Ask experts!</u>


Skimage.io.imshow crashes when
forum.image.sc/t/skimage-io-imshow-crashes-when-showing-binary-images/52796
image.sc
Related Forums

Skimage.io.imshow crashes when showing binary images

Usage & Issues scikit-image

Robert Haase haesleinhuepf clij maintainer

Explain your problem

Hey scikit-image friends,

some of my teaching notebooks recently broke. Suddenly, the `imshow` function in scikit-image can't show binary images anymore. Does anybody have an idea what we could do (change dependency versions? change code?) to make them run again?

Here is the code (and that's the data):

Post minimal code that allows reproducing the issue

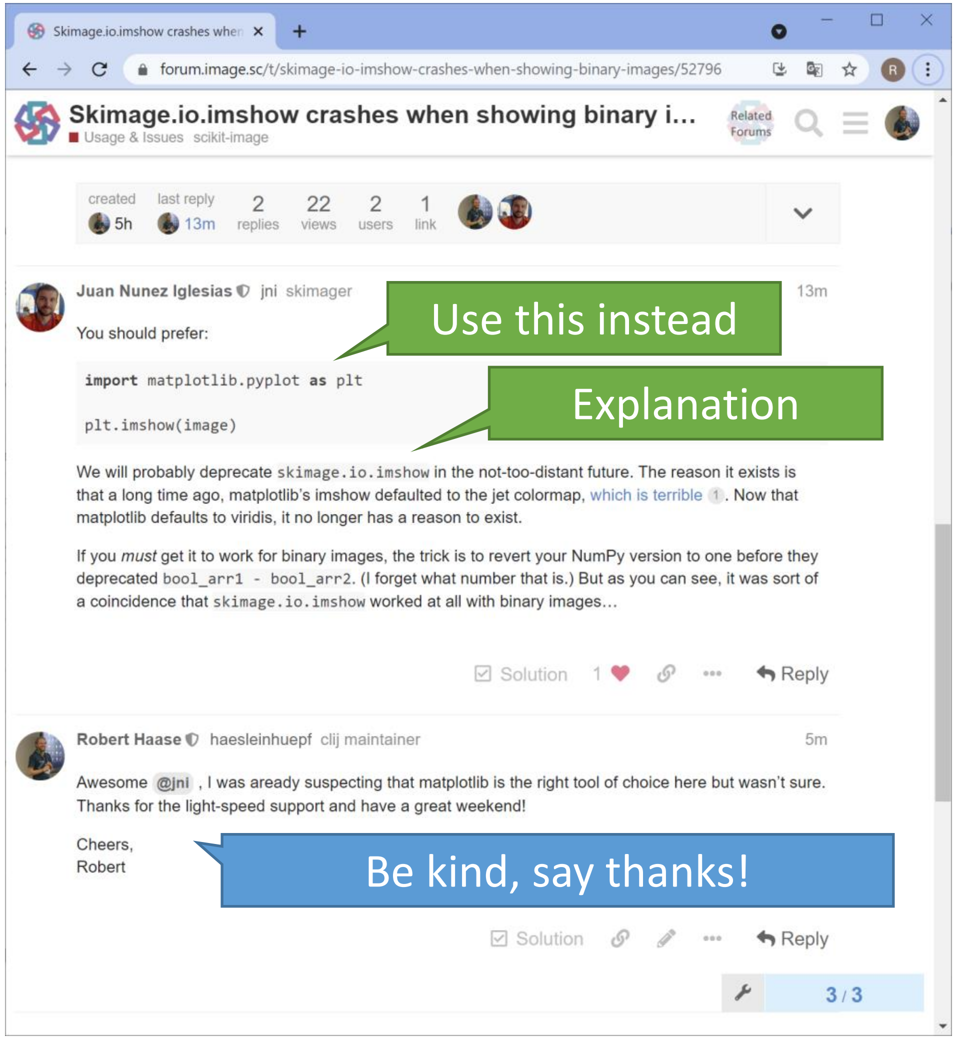
```
from skimage.io import imshow, imread

image = imread("blobs.tif")

from skimage import filters
threshold = filters.threshold_otsu(image)

binary_image = image >= threshold

imshow(binary_image)
```

That error pops up in the last line:

Post the full error message! Post screenshots!

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-94950852c6d3> in <module>
      1 binary_image = image >= threshold
      2 
----> 3 imshow(binary_image)

~\miniconda3\envs\bio\lib\site-packages\skimage\io\_io.py in imshow(arr, plugin,
    157     if isinstance(arr, str):
    158         arr = call_plugin('imread', arr, plugin=plugin)
--> 159     return call_plugin('imshow', arr, plugin=plugin, **plugin_a
    160 
```

1 / 3


Skimage.io.imshow crashes when
forum.image.sc/t/skimage-io-imshow-crashes-when-showing-binary-images/52796
Related Forums

Skimage.io.imshow crashes when showing binary i...

Usage & Issues scikit-image

created	last reply	2	22	2	1
5h	13m	replies	views	users	link

Juan Nunez Iglesias jni skimager 13m

You should prefer:

Use this instead

```
import matplotlib.pyplot as plt

plt.imshow(image)
```

Explanation

We will probably deprecate `skimage.io.imshow` in the not-too-distant future. The reason it exists is that a long time ago, matplotlib's imshow defaulted to the jet colormap, which is terrible 1. Now that matplotlib defaults to viridis, it no longer has a reason to exist.

If you *must* get it to work for binary images, the trick is to revert your NumPy version to one before they deprecated `bool_arr1 - bool_arr2`. (I forget what number that is.) But as you can see, it was sort of a coincidence that `skimage.io.imshow` worked at all with binary images…

☑ Solution 1 ♥ ••• Reply

Robert Haase haesleinhuepf clij maintainer 5m

Awesome @jni , I was aready suspecting that matplotlib is the right tool of choice here but wasn't sure. Thanks for the light-speed support and have a great weekend!

Cheers,
Robert

Be kind, say thanks!

☑ Solution ••• Reply

3 / 3

Juan Nunez-Iglesias
jni

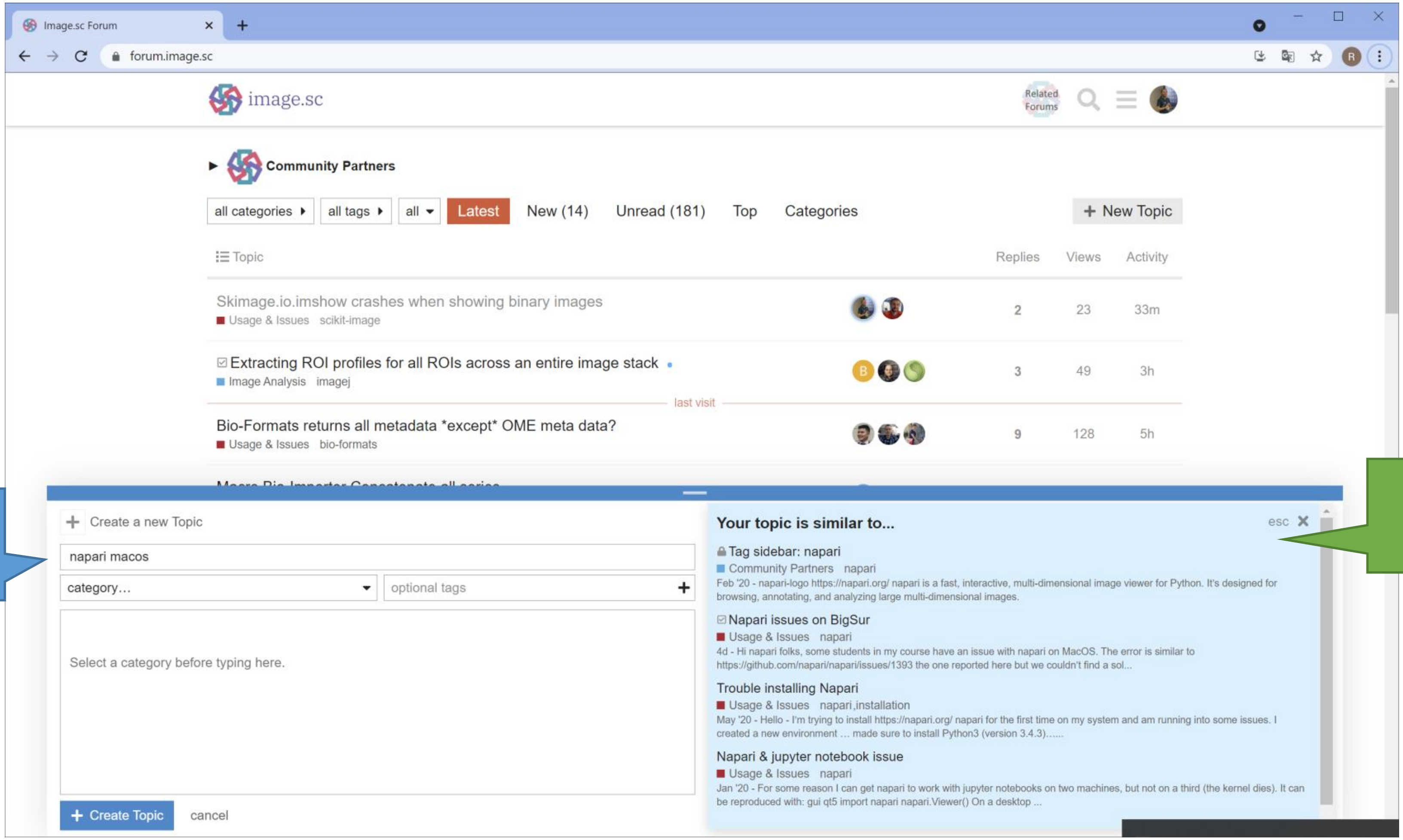
Biomedicine Discovery Institute, Monas...
Melbourne, VIC, Australia
juan.nunez-iglesias@monash.edu
http://ilovesymposia.com
@jnuneziglesias

Tell a friend!

May 2021 https://forum.image.sc/t/skimage-io-imshow-crashes-when-showing-binary-images/52796

The image science community: https://image.sc

- In case you run in trouble with image analysis, observe bugs in open-source software or want to know how to analyze your image data: Ask experts!
- Plus: discover other’s people similar questions



Topic		Replies	Views	Activity
Skimage.io.imshow crashes when showing binary images Usage & Issues scikit-image		2	23	33m
☑ Extracting ROI profiles for all ROIs across an entire image stack Image Analysis imagej		3	49	3h
last visit				
Bio-Formats returns all metadata *except* OME meta data? Usage & Issues bio-formats		9	128	5h

+ Create a new Topic

napari macos

category…

optional tags

Select a category before typing here.

+ Create Topic   cancel

Your topic is similar to... esc ✕

Tag sidebar: napari
Community Partners napari
Feb '20 - napari-logo https://napari.org/ napari is a fast, interactive, multi-dimensional image viewer for Python. It's designed for browsing, annotating, and analyzing large multi-dimensional images.

☑ Napari issues on BigSur
Usage & Issues napari
4d - Hi napari folks, some students in my course have an issue with napari on MacOS. The error is similar to https://github.com/napari/napari/issues/1393 the one reported here but we couldn't find a sol...

Trouble installing Napari
Usage & Issues napari,installation
May '20 - Hello - I'm trying to install https://napari.org/ napari for the first time on my system and am running into some issues. I created a new environment … made sure to install Python3 (version 3.4.3)......

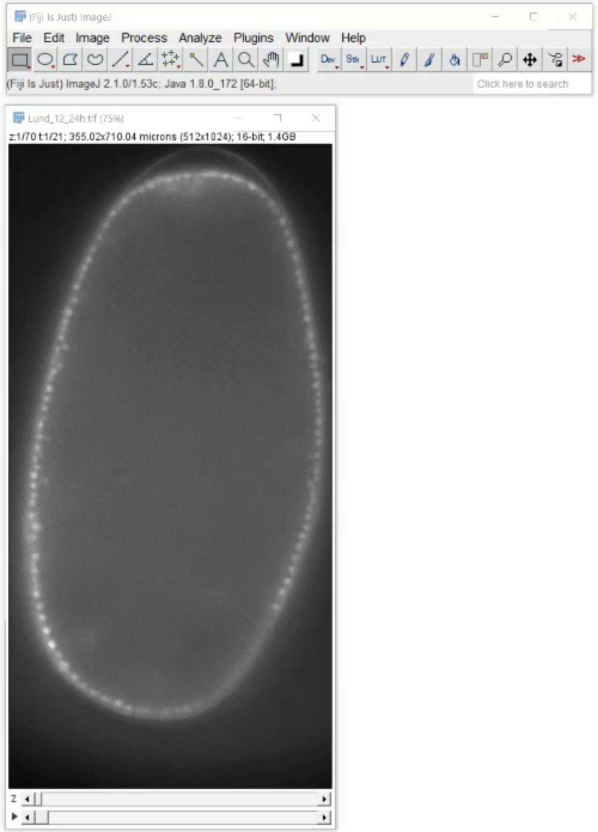
Napari & jupyter notebook issue
Usage & Issues napari
Jan '20 - For some reason I can get napari to work with jupyter notebooks on two machines, but not on a third (the kernel dies). It can be reproduced with: gui qt5 import napari napari.Viewer() On a desktop ...

Name your problem

Discover similar problems

GPU-accelerated image processing

- State-of-the-art software for more than 20 years: ImageJ / Fiji



(Fiji Is Just) ImageJ 2.1.0/1.53c; Java 1.8.0_172 [64-bit];

z:1/70 t:1/21; 355.02x710.04 microns (512x1024); 16-bit; 1.4GB

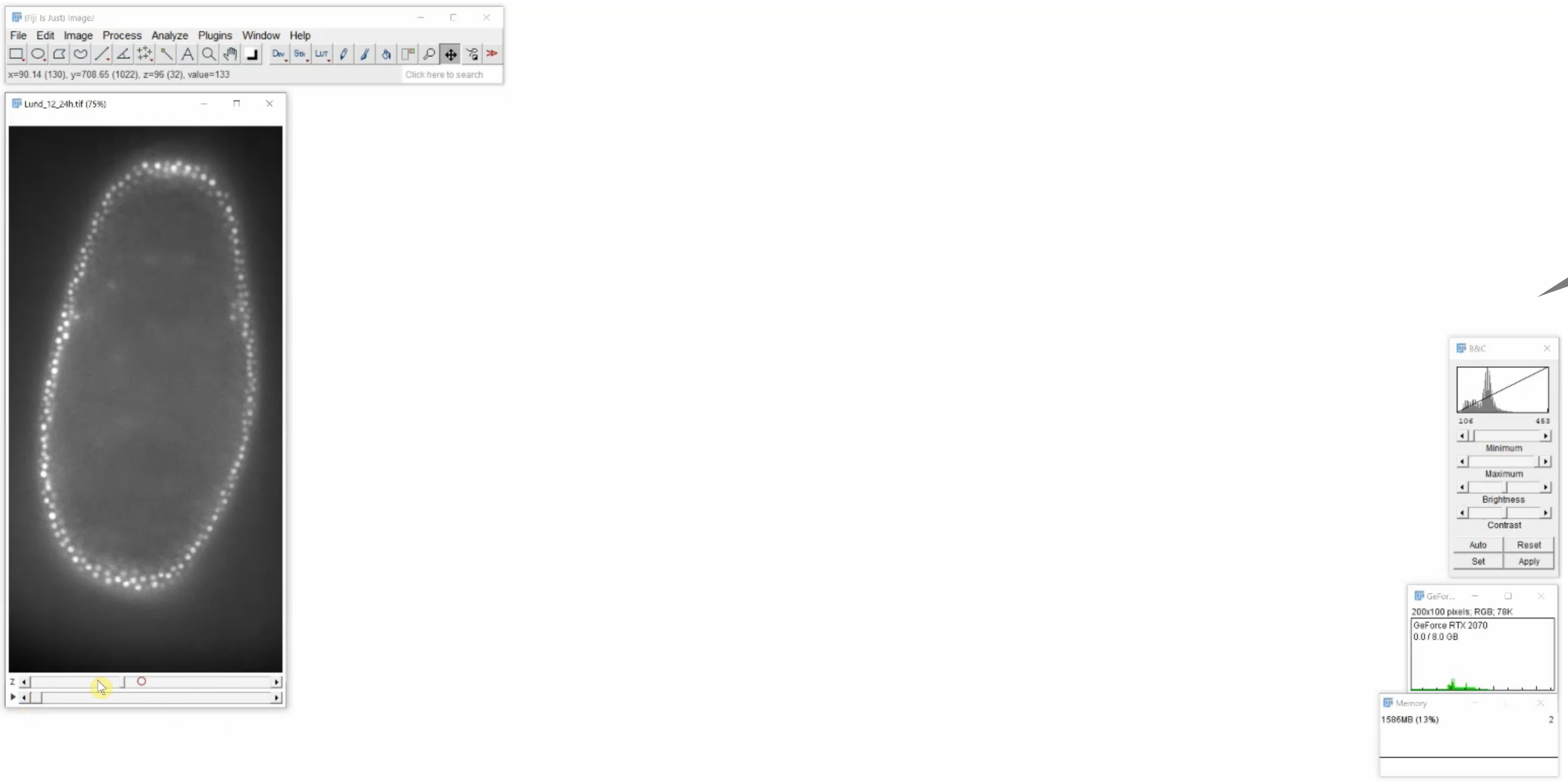
2x

https://imagej.nih.gov/ij/ https://fiji.sc

Image data source: Daniela Vorkel, Myers lab, MPI CBG

GPU-accelerated image processing

- The dawn of graphics processing units changes how we interact with image data



x=90.14 (130), y=708.65 (1022), z=96 (32), value=133

200x100 pixels; RGB; 78K

GeForce RTX 2070
0.0 / 8.0 GB

1586MB (13%) 2

2x

https://clij.github.io/assistant

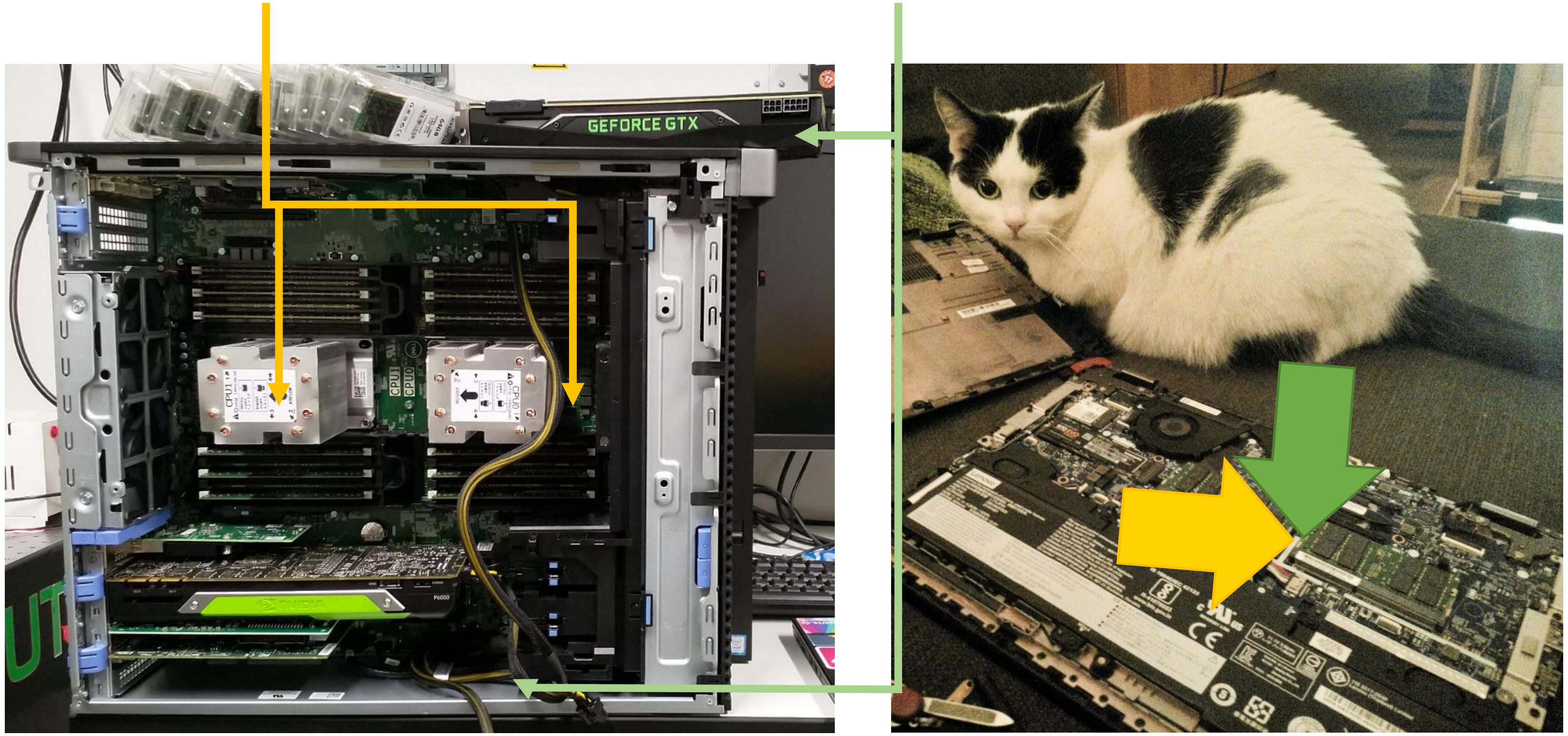
Image data source: Daniela Vorkel, Myers lab, MPI CBG

GPU-accelerated image processing

- Every computer that has a screen also has a GPU.

Central Processing Unit (CPU)

Graphics Processing Unit (GPU)



dedicated GPUs

integrated GPUs

external “eGPUs”

Why GPU-acceleration?

- 1-2 orders of magnitude speedup are possible

vs.

✕ Workstation CPU

2x Intel Xeon Silver 4110

✕ Workstation GPU

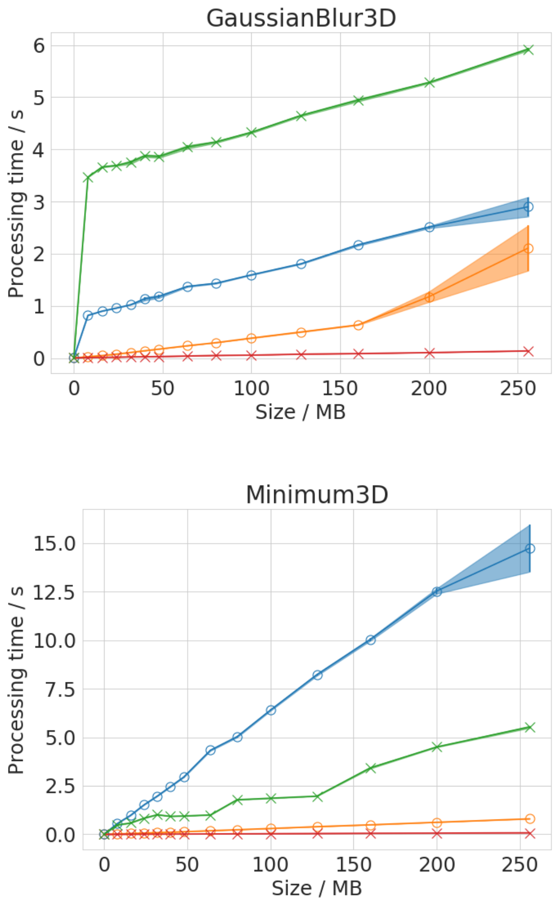
Nvidia Quadro P6000

○ Laptop CPU

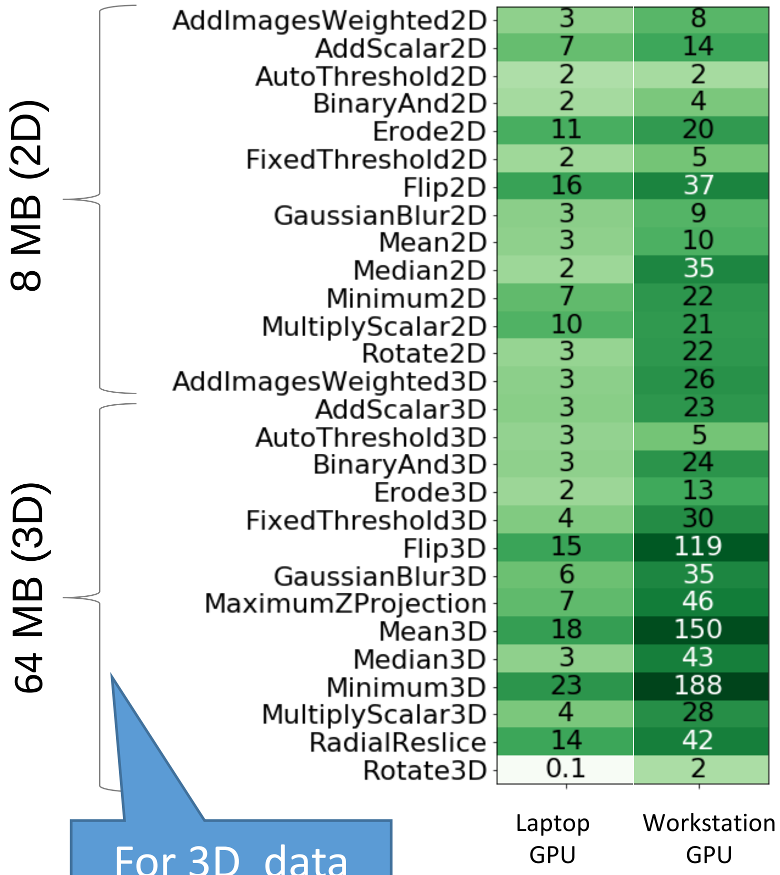
Intel Core i7-8650U

○ Laptop GPU

Intel UHD 620 GPU



Speedup factor compared to Laptop CPU



		Laptop GPU	Workstation GPU
8 MB (2D)	AddImagesWeighted2D	3	8
	AddScalar2D	7	14
	AutoThreshold2D	2	2
	BinaryAnd2D	2	4
	Erode2D	11	20
	FixedThreshold2D	2	5
	Flip2D	16	37
	GaussianBlur2D	3	9
	Mean2D	3	10
	Median2D	2	35
	Minimum2D	7	22
	MultiplyScalar2D	10	21
	Rotate2D	3	22
64 MB (3D)	AddImagesWeighted3D	3	26
	AddScalar3D	3	23
	AutoThreshold3D	3	5
	BinaryAnd3D	3	24
	Erode3D	2	13
	FixedThreshold3D	4	30
	Flip3D	15	119
	GaussianBlur3D	6	35
	MaximumZProjection	7	46
	Mean3D	18	150
	Median3D	3	43
	Minimum3D	23	188
	MultiplyScalar3D	4	28
	RadialReslice	14	42
	Rotate3D	0.1	2

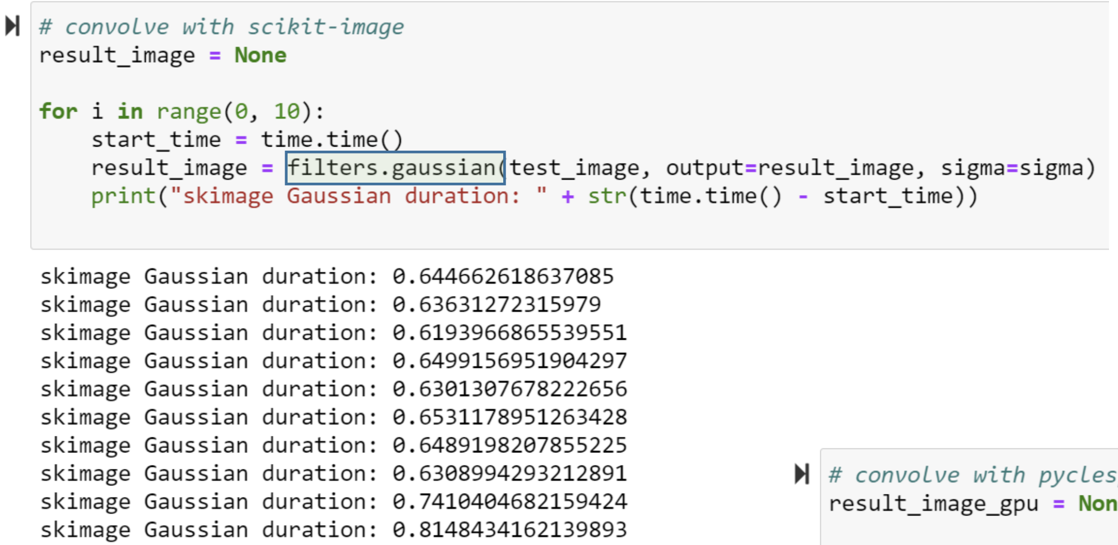
For 3D data worthwhile!

https://www.youtube.com/watch?v=uMj0OS1TiQE&t=873s

Nat Methods 17, 5-6 (2020) doi:10.1038/s41592-019-0650-1

Why GPU-acceleration?

- 1-2 orders of magnitude speedup are possible

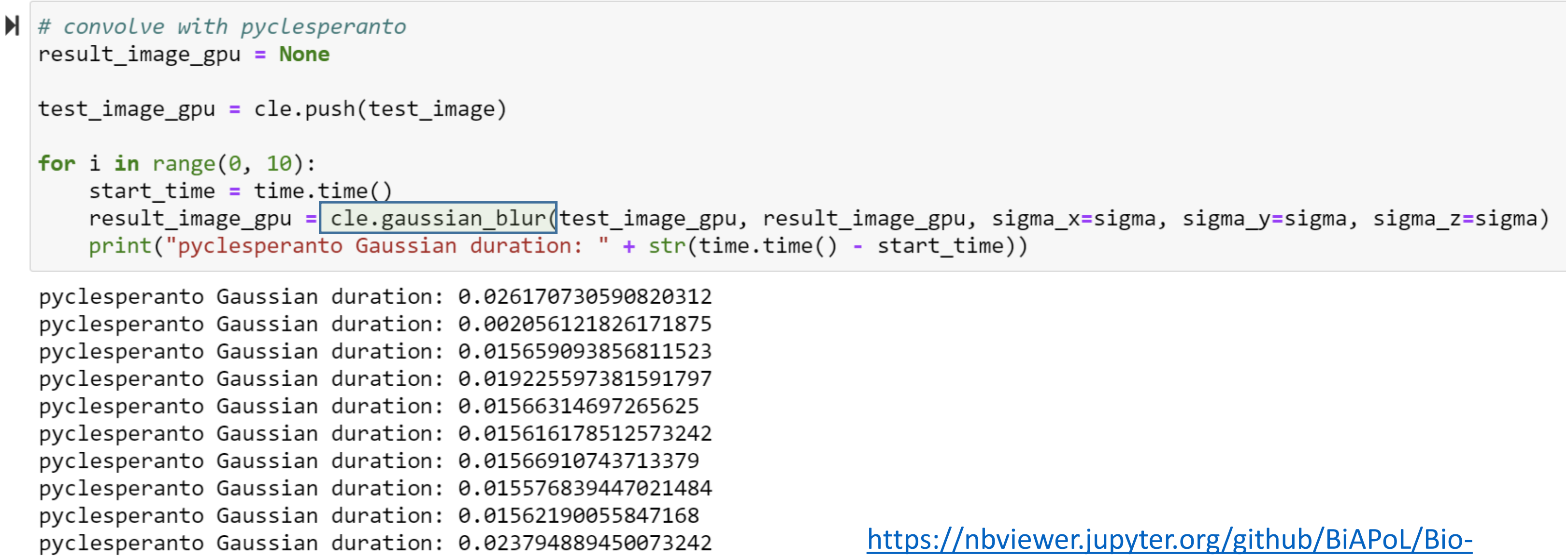


```
# convolve with scikit-image
result_image = None

for i in range(0, 10):
    start_time = time.time()
    result_image = filters.gaussian(test_image, output=result_image, sigma=sigma)
    print("skimage Gaussian duration: " + str(time.time() - start_time))
```

```
skimage Gaussian duration: 0.644662618637085
skimage Gaussian duration: 0.63631272315979
skimage Gaussian duration: 0.6193966865539551
skimage Gaussian duration: 0.6499156951904297
skimage Gaussian duration: 0.6301307678222656
skimage Gaussian duration: 0.6531178951263428
skimage Gaussian duration: 0.6489198207855225
skimage Gaussian duration: 0.6308994293212891
skimage Gaussian duration: 0.7410404682159424
skimage Gaussian duration: 0.8148434162139893
```

scikit-image

clesperanto



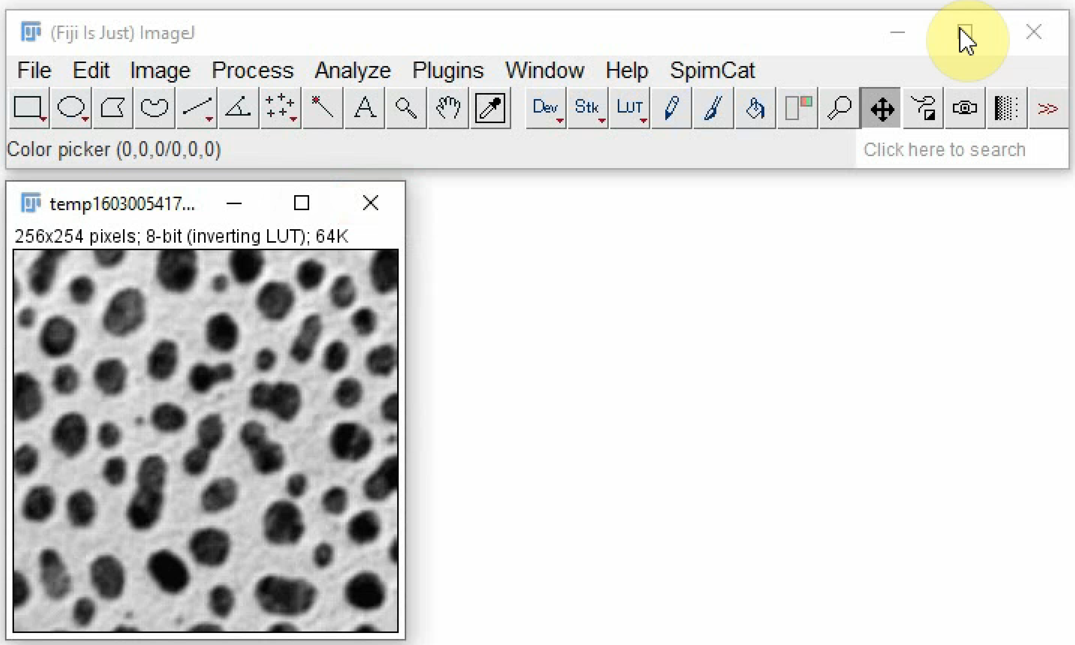
```
# convolve with pyclesperanto
result_image_gpu = None

test_image_gpu = cle.push(test_image)

for i in range(0, 10):
    start_time = time.time()
    result_image_gpu = cle.gaussian_blur(test_image_gpu, result_image_gpu, sigma_x=sigma, sigma_y=sigma, sigma_z=sigma)
    print("pyclesperanto Gaussian duration: " + str(time.time() - start_time))
```

```
pyclesperanto Gaussian duration: 0.026170730590820312
pyclesperanto Gaussian duration: 0.002056121826171875
pyclesperanto Gaussian duration: 0.015659093856811523
pyclesperanto Gaussian duration: 0.019225597381591797
pyclesperanto Gaussian duration: 0.01566314697265625
pyclesperanto Gaussian duration: 0.015616178512573242
pyclesperanto Gaussian duration: 0.01566910743713379
pyclesperanto Gaussian duration: 0.015576839447021484
pyclesperanto Gaussian duration: 0.01562190055847168
pyclesperanto Gaussian duration: 0.023794889450073242
```

https://nbviewer.jupyter.org/github/BiAPoL/Bio-image_Analysis_with_Python/blob/main/gpu_acceleration/03_why_GPU_acceleration.ipynb

How to GPU-accelerate workflows

- In ImageJ / Fiji: CLIJ



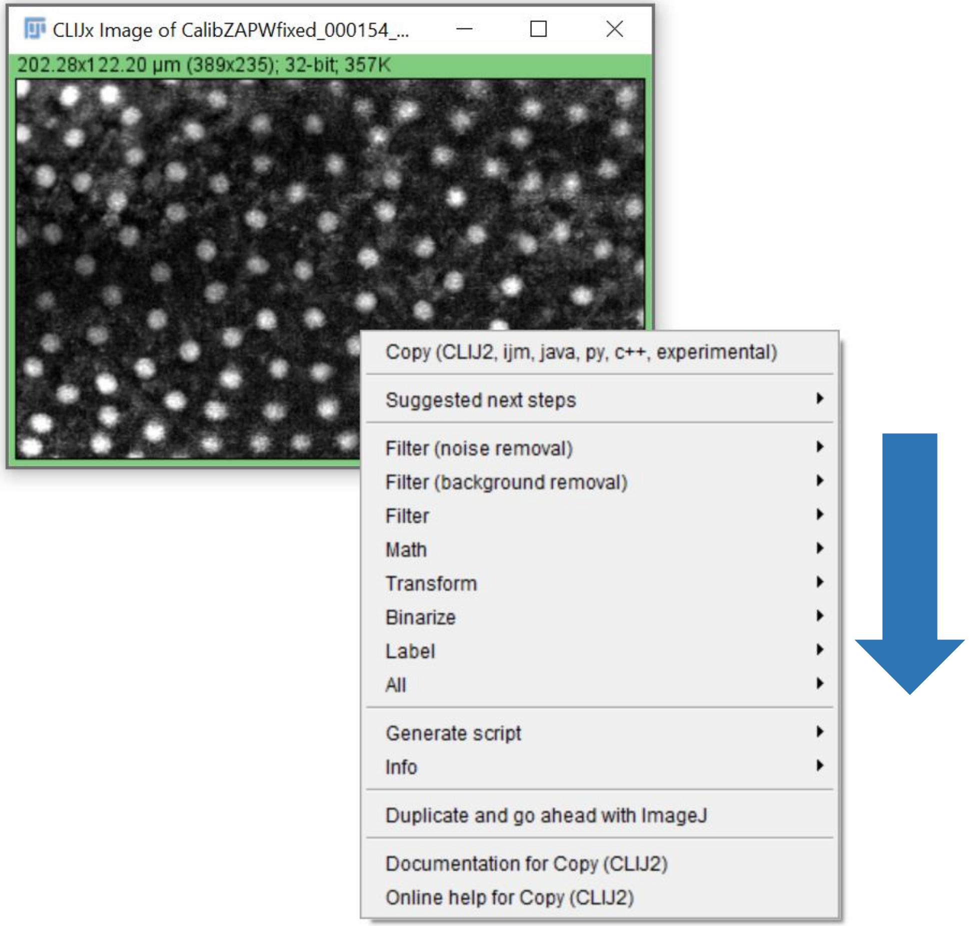
Color picker (0,0,0/0,0,0)

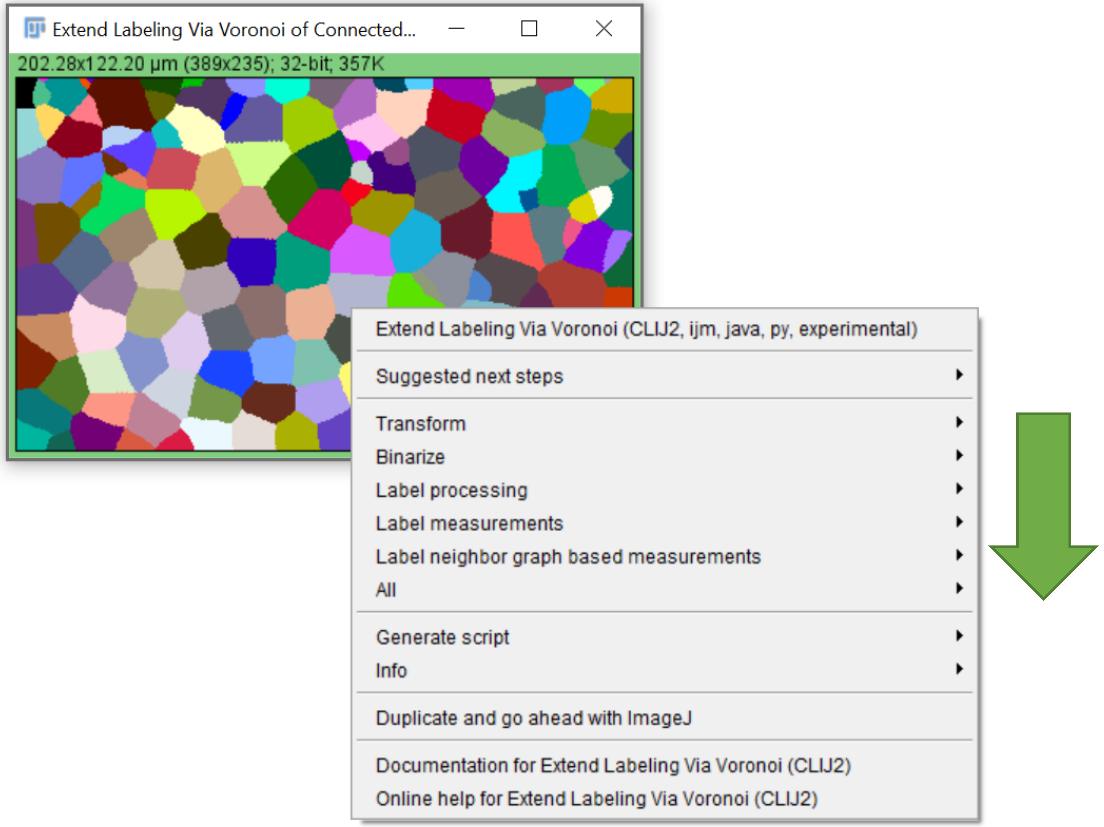
256x254 pixels; 8-bit (inverting LUT); 64K

https://clij.github.io/assistant/getting_started

The right-click menu in CLIJ

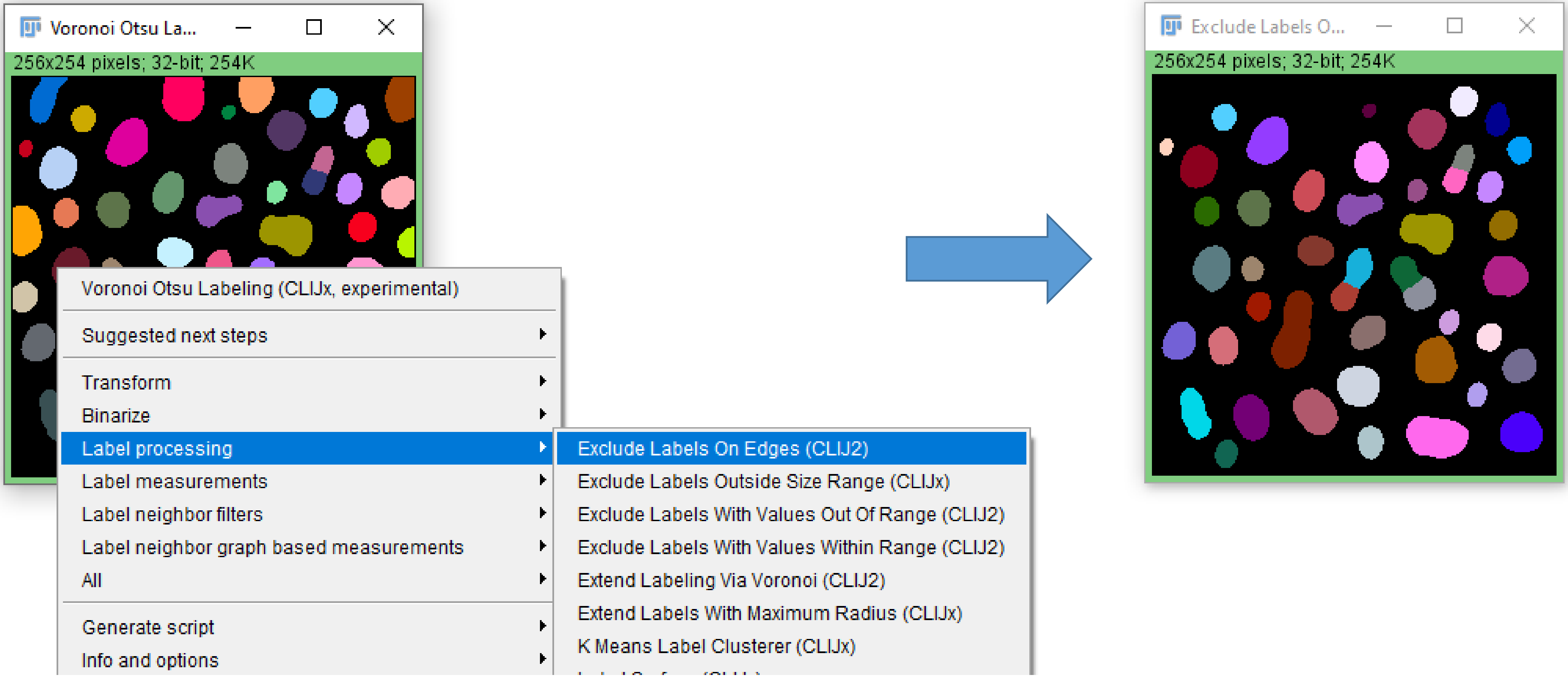
- The menu order is intentional: From preprocessing to analysis


CLIJx Image of CalibZAPWfixed_000154_...
202.28x122.20 µm (389x235); 32-bit; 357K
Copy (CLIJ2, ijm, java, py, c++, experimental)
Suggested next steps
Filter (noise removal)
Filter (background removal)
Filter
Math
Transform
Binarize
Label
All
Generate script
Info
Duplicate and go ahead with ImageJ
Documentation for Copy (CLIJ2)
Online help for Copy (CLIJ2)


Extend Labeling Via Voronoi of Connected...
202.28x122.20 µm (389x235); 32-bit; 357K
Extend Labeling Via Voronoi (CLIJ2, ijm, java, py, experimental)
Suggested next steps
Transform
Binarize
Label processing
Label measurements
Label neighbor graph based measurements
All
Generate script
Info
Duplicate and go ahead with ImageJ
Documentation for Extend Labeling Via Voronoi (CLIJ2)
Online help for Extend Labeling Via Voronoi (CLIJ2)

Post-processing label images: CLIJ

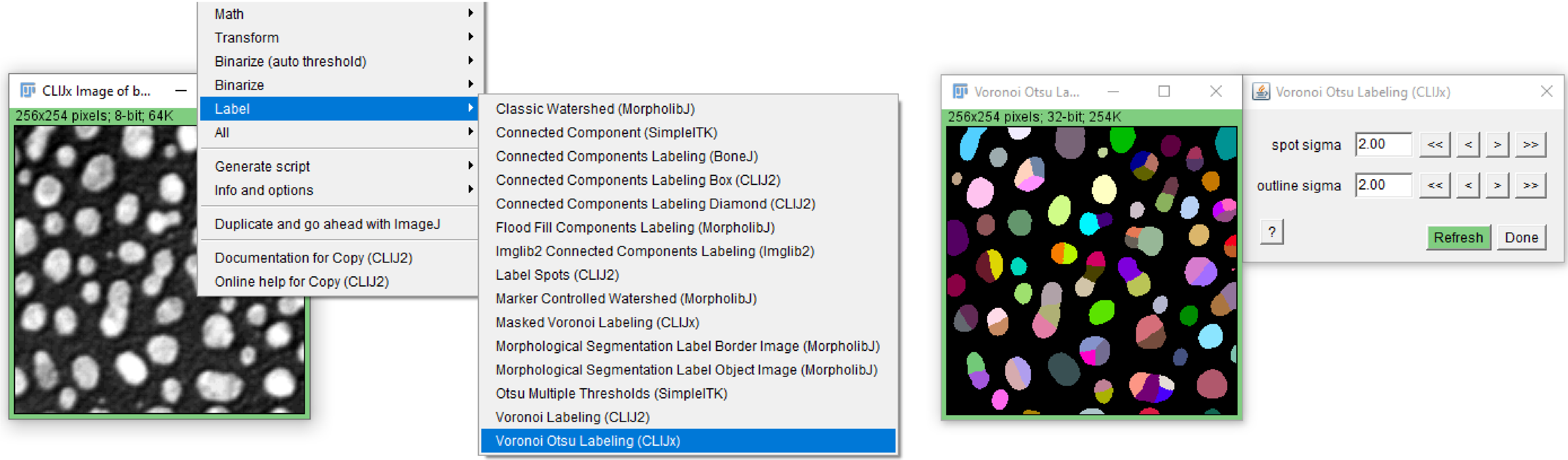
- [Right-click menu]: Label processing


Voronoi Otsu La...
256x254 pixels; 32-bit; 254K
Voronoi Otsu Labeling (CLIJx, experimental)
Suggested next steps
Transform
Binarize
Label processing
Label measurements
Label neighbor filters
Label neighbor graph based measurements
All
Generate script
Info and options
Exclude Labels On Edges (CLIJ2)
Exclude Labels Outside Size Range (CLIJx)
Exclude Labels With Values Out Of Range (CLIJ2)
Exclude Labels With Values Within Range (CLIJ2)
Extend Labeling Via Voronoi (CLIJ2)
Extend Labels With Maximum Radius (CLIJx)
K Means Label Clusterer (CLIJx)
Exclude Labels O...
256x254 pixels; 32-bit; 254K

Short-cuts: Labeling directly from grey-value images: CLIJ

- GPU-accelerated image processing…

… is also just image processing

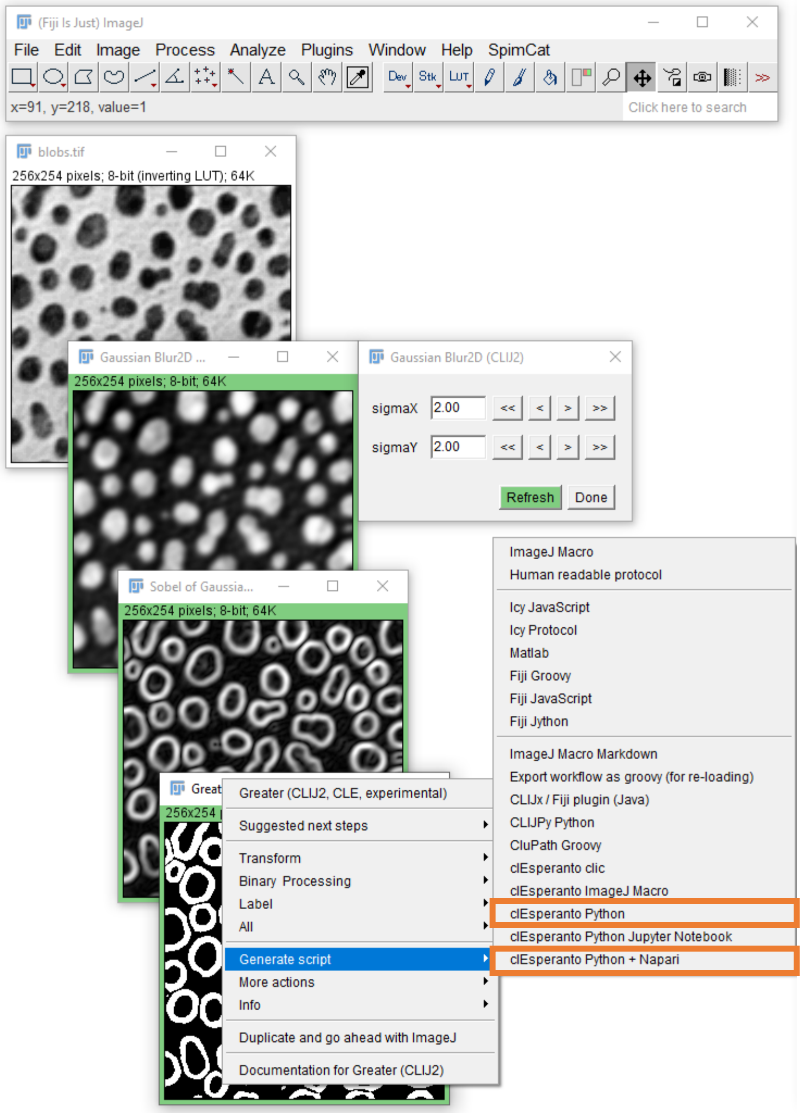

CLIJx Image of b...
256x254 pixels; 8-bit; 64K
Math
Transform
Binarize (auto threshold)
Binarize
Label
All
Generate script
Info and options
Duplicate and go ahead with ImageJ
Documentation for Copy (CLIJ2)
Online help for Copy (CLIJ2)
Classic Watershed (MorpholibJ)
Connected Component (SimpleITK)
Connected Components Labeling (BoneJ)
Connected Components Labeling Box (CLIJ2)
Connected Components Labeling Diamond (CLIJ2)
Flood Fill Components Labeling (MorpholibJ)
Imglib2 Connected Components Labeling (Imglib2)
Label Spots (CLIJ2)
Marker Controlled Watershed (MorpholibJ)
Masked Voronoi Labeling (CLIJx)
Morphological Segmentation Label Border Image (MorpholibJ)
Morphological Segmentation Label Object Image (MorpholibJ)
Otsu Multiple Thresholds (SimpleITK)
Voronoi Labeling (CLIJ2)
Voronoi Otsu Labeling (CLIJx)
Voronoi Otsu La...
256x254 pixels; 32-bit; 254K
Voronoi Otsu Labeling (CLIJx)
<< < > >>
<< < > >>
?
Refresh
Done

spot sigma 2.00

outline sigma 2.00

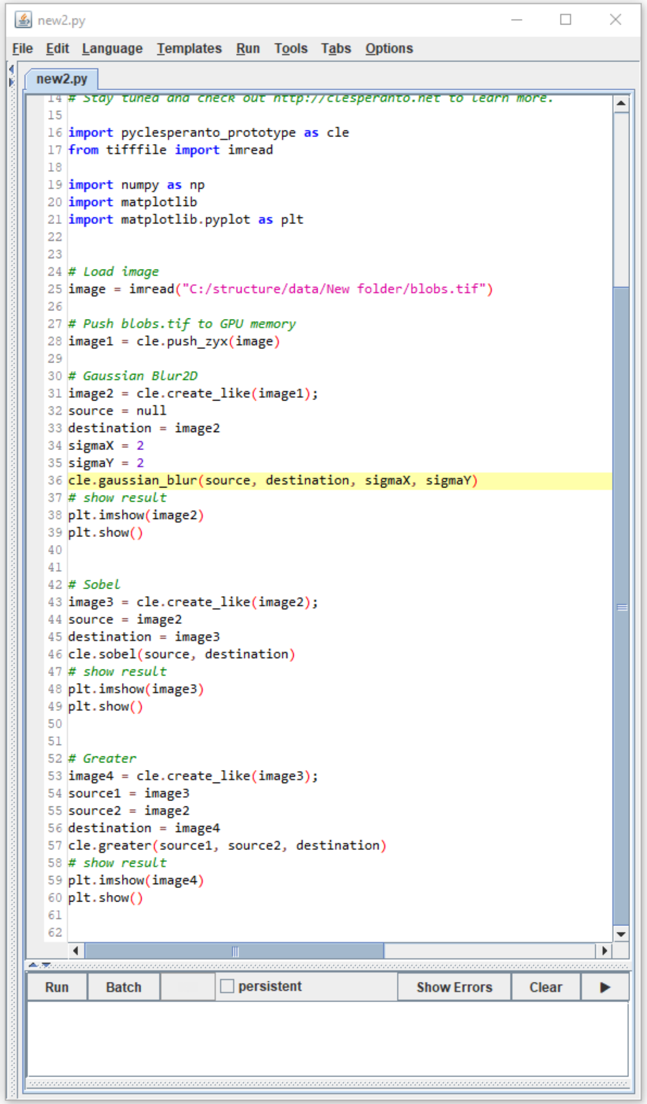
Code generators in CLIJ

- Right-click > Generate Script > clEsperanto Python
- Copy & Paste it to your Jupyter Notebook
- *If* you want to run it from Fiji, follow these instructions: https://clij.github.io/assistant/installation#te_oki



sigmaX 2.00

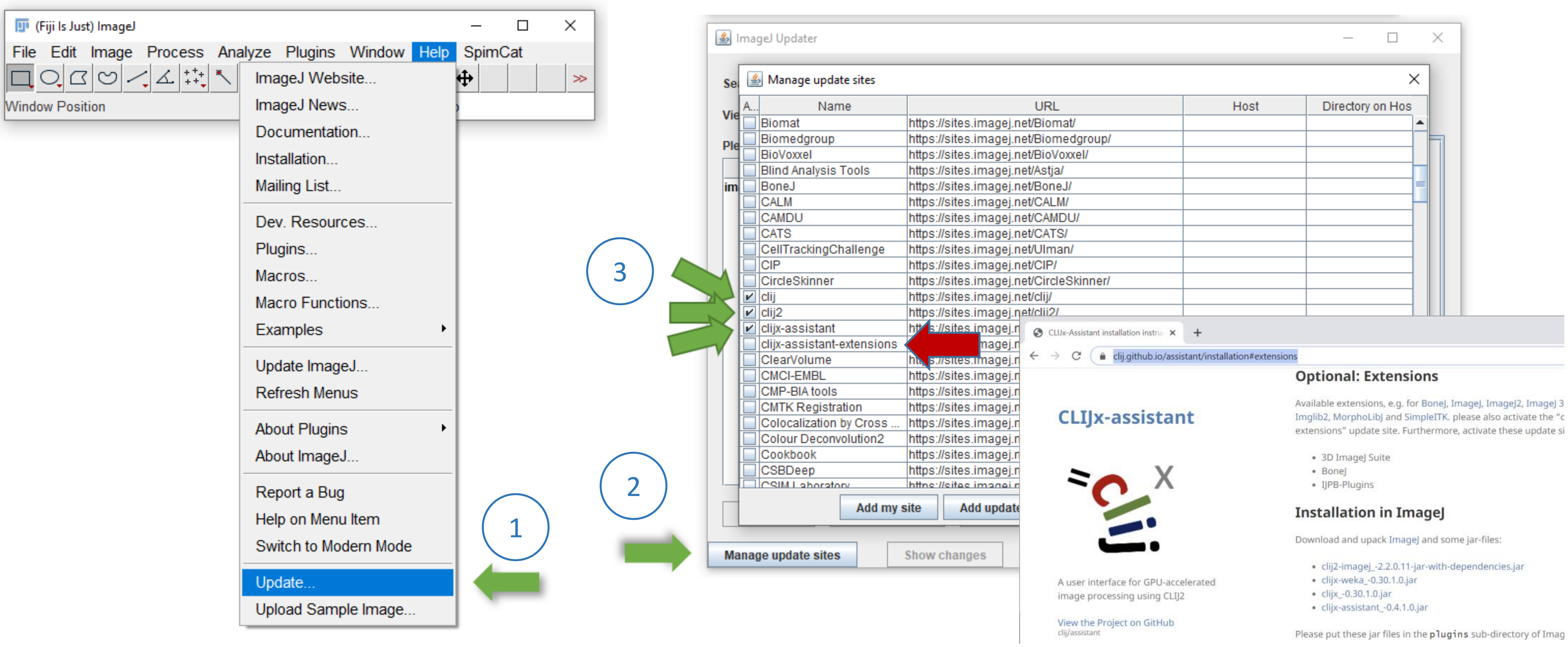
sigmaY 2.00



```
# Stay tuned and check out http://clesperanto.net to learn more.

import pyclesperanto_prototype as cle
from tifffile import imread

import numpy as np
import matplotlib
import matplotlib.pyplot as plt


# Load image
image = imread("C:/structure/data/New folder/blobs.tif")

# Push blobs.tif to GPU memory
image1 = cle.push_zyx(image)

# Gaussian Blur2D
image2 = cle.create_like(image1);
source = null
destination = image2
sigmaX = 2
sigmaY = 2
cle.gaussian_blur(source, destination, sigmaX, sigmaY)
# show result
plt.imshow(image2)
plt.show()


# Sobel
image3 = cle.create_like(image2);
source = image2
destination = image3
cle.sobel(source, destination)
# show result
plt.imshow(image3)
plt.show()


# Greater
image4 = cle.create_like(image3);
source1 = image3
source2 = image2
destination = image4
cle.greater(source1, source2, destination)
# show result
plt.imshow(image4)
plt.show()


```

CLIJ: Installation

- Just activate the CLIJ update sites, in menu Help > Update...
- Linux users: you may need to install OpenCL, e.g. `apt-get install ocl-icd-devel`



Manage update sites

A...	Name	URL	Host	Directory on Hos
□	Biomat	https://sites.imagej.net/Biomat/		
□	Biomedgroup	https://sites.imagej.net/Biomedgroup/		
□	BioVoxxel	https://sites.imagej.net/BioVoxxel/		
□	Blind Analysis Tools	https://sites.imagej.net/Astja/		
□	BoneJ	https://sites.imagej.net/BoneJ/		
□	CALM	https://sites.imagej.net/CALM/		
□	CAMDU	https://sites.imagej.net/CAMDU/		
□	CATS	https://sites.imagej.net/CATS/		
□	CellTrackingChallenge	https://sites.imagej.net/Ulman/		
□	CIP	https://sites.imagej.net/CIP/		
□	CircleSkinner	https://sites.imagej.net/CircleSkinner/		
☑	clij	https://sites.imagej.net/clij/		
☑	clij2	https://sites.imagej.net/clij2/		
☑	clijx-assistant	ht s://sites.imagej.n		
□	clijx-assistant-extensions	magej.n		
□	ClearVolume	ht s://sites.imagej.n		
□	CMCI-EMBL	https://sites.imagej.n		
□	CMP-BIA tools	https://sites.imagej.n		
□	CMTK Registration	https://sites.imagej.n		
□	Colocalization by Cross ...	https://sites.imagej.n		
□	Colour Deconvolution2	https://sites.imagej.n		
□	Cookbook	https://sites.imagej.n		
□	CSBDeep	https://sites.imagej.n		
□	CSIM Laboratory	https://sites.imagej.n		

CLIJx-Assistant installation instru

clij.github.io/assistant/installation#extensions

CLIJx-assistant

A user interface for GPU-accelerated image processing using CLIJ2

View the Project on GitHub
clij/assistant

Optional: Extensions

Available extensions, e.g. for BoneJ, ImageJ, ImageJ2, ImageJ 3 Imglib2, MorphoLibJ and SimpleITK. please also activate the "c extensions" update site. Furthermore, activate these update si

- 3D ImageJ Suite
- BoneJ
- IJPB-Plugins

Installation in ImageJ

Download and upack ImageJ and some jar-files:

- clij2-imagej_-2.2.0.11-jar-with-dependencies.jar
- clijx-weka_-0.30.1.0.jar
- clijx_-0.30.1.0.jar
- clijx-assistant_-0.4.1.0.jar

Please put these jar files in the `plugins` sub-directory of Imag

clEsperanto: GPU-accelerated image processing in Python

- The CLIJ/Fiji counter-part in python/napari

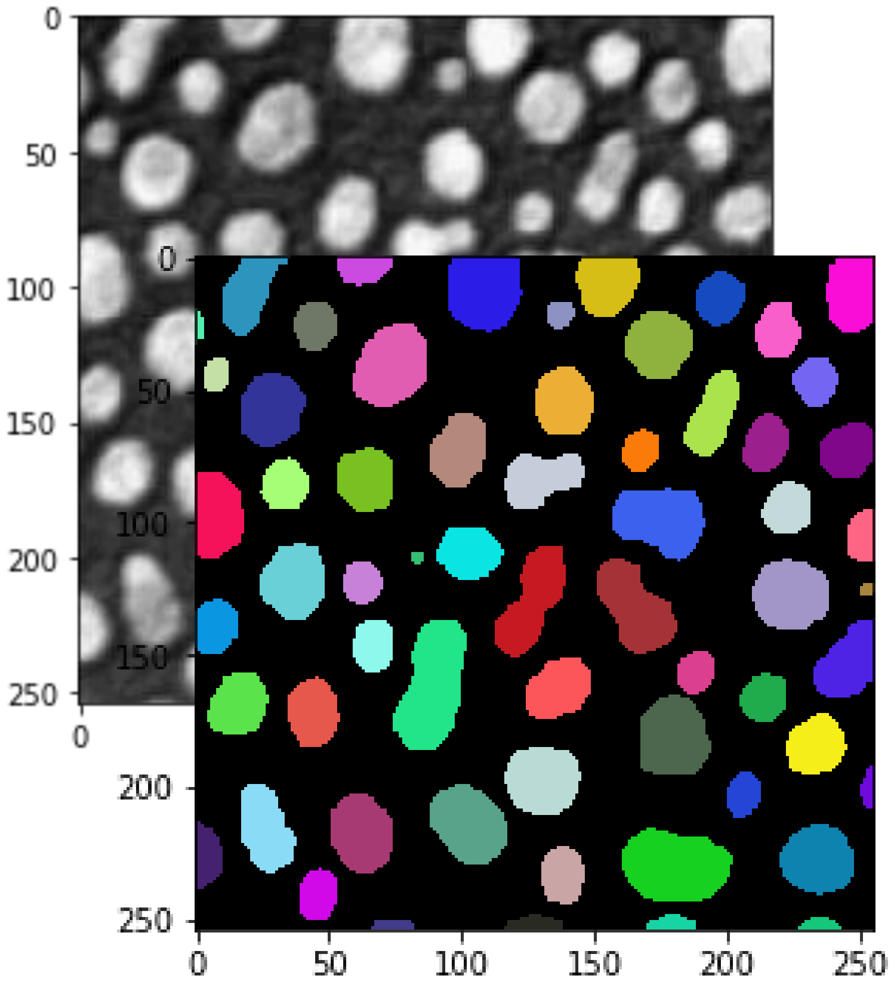
```
conda install -c conda-forge pyopencl
pip install pyclesperanto-prototype
```

- Napari user interface

```
pip install napari-pyclesperanto-assistant
```

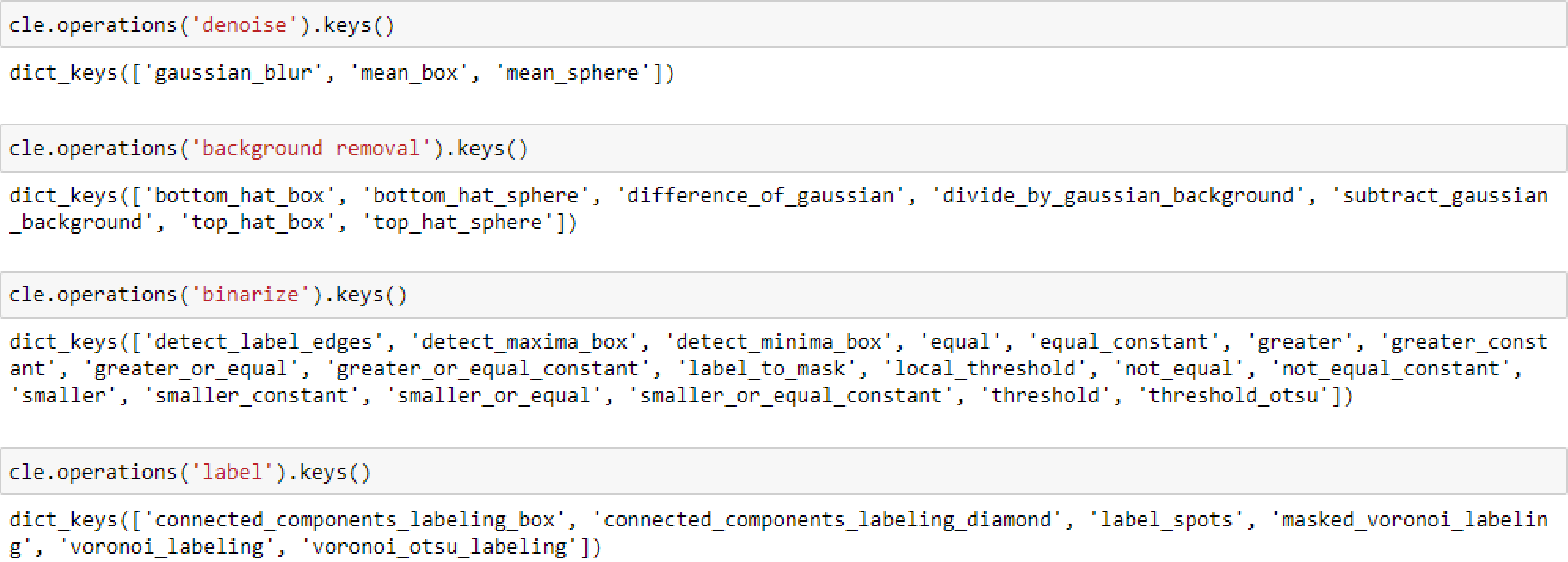
The cle gateway

```
from skimage.io import imread, imshow
image = imread("blobs.tif")
imshow(image)

import pyclesperanto_prototype as cle

# noise removal
blurred = cle.gaussian_blur(image, sigma_x=1, sigma_y=1)
# binarization
binary = cle.threshold_otsu(blurred)
# labeling
labels = cle.connected_components_labeling_box(binary)
# visualize results
cle.imshow(labels, labels=True)
```



The `cle` gateway

- … allows you to explore available functions for given purposes



```
cle.operations('denoise').keys()
```

```
dict_keys(['gaussian_blur', 'mean_box', 'mean_sphere'])
```

```
cle.operations('background removal').keys()
```

```
dict_keys(['bottom_hat_box', 'bottom_hat_sphere', 'difference_of_gaussian', 'divide_by_gaussian_background', 'subtract_gaussian
_background', 'top_hat_box', 'top_hat_sphere'])
```

```
cle.operations('binarize').keys()
```

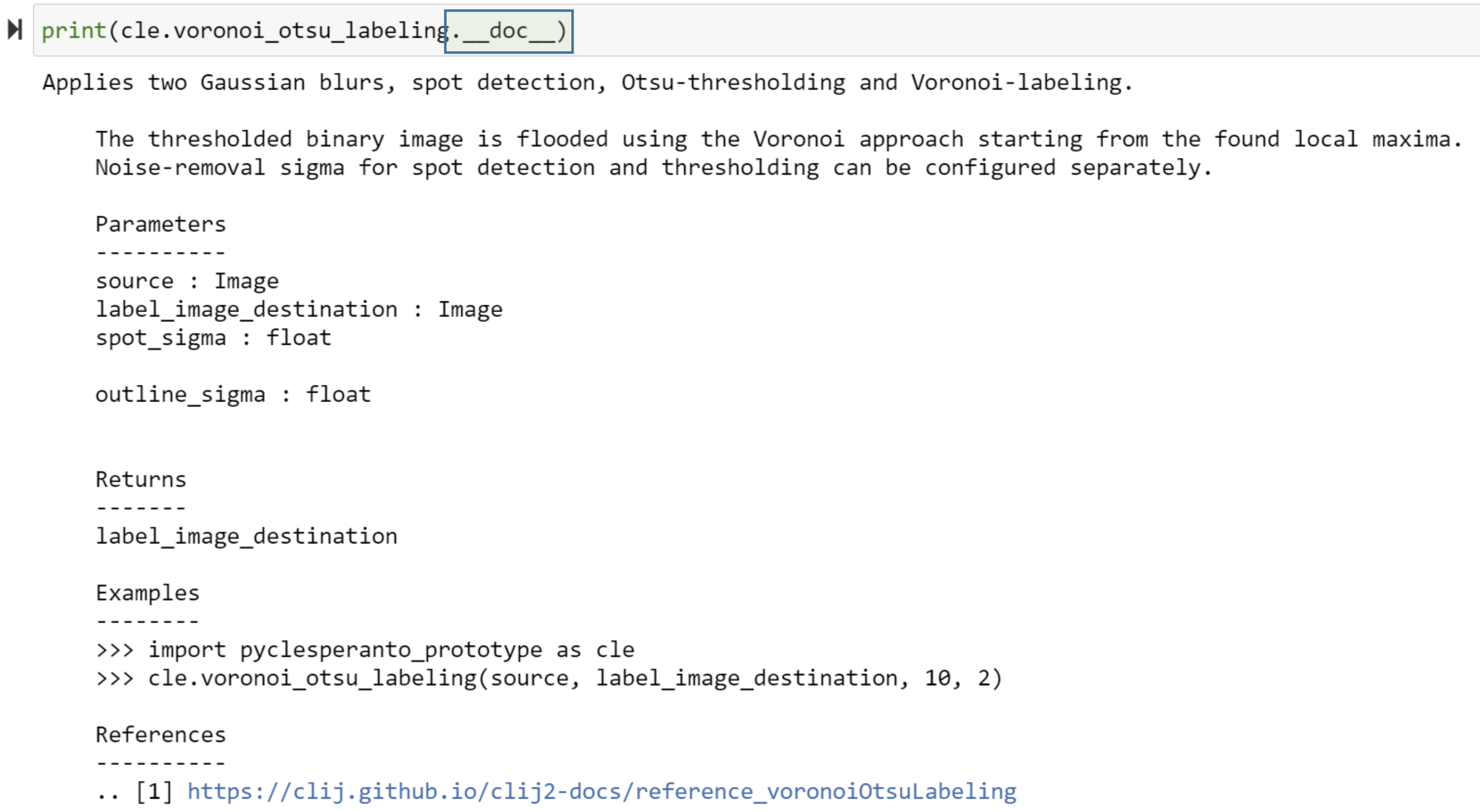
```
dict_keys(['detect_label_edges', 'detect_maxima_box', 'detect_minima_box', 'equal', 'equal_constant', 'greater', 'greater_const
ant', 'greater_or_equal', 'greater_or_equal_constant', 'label_to_mask', 'local_threshold', 'not_equal', 'not_equal_constant',
'smaller', 'smaller_constant', 'smaller_or_equal', 'smaller_or_equal_constant', 'threshold', 'threshold_otsu'])
```

```
cle.operations('label').keys()
```

```
dict_keys(['connected_components_labeling_box', 'connected_components_labeling_diamond', 'label_spots', 'masked_voronoi_labelin
g', 'voronoi_labeling', 'voronoi_otsu_labeling'])
```

Recap: .__doc__

- … allows you to read the documentation

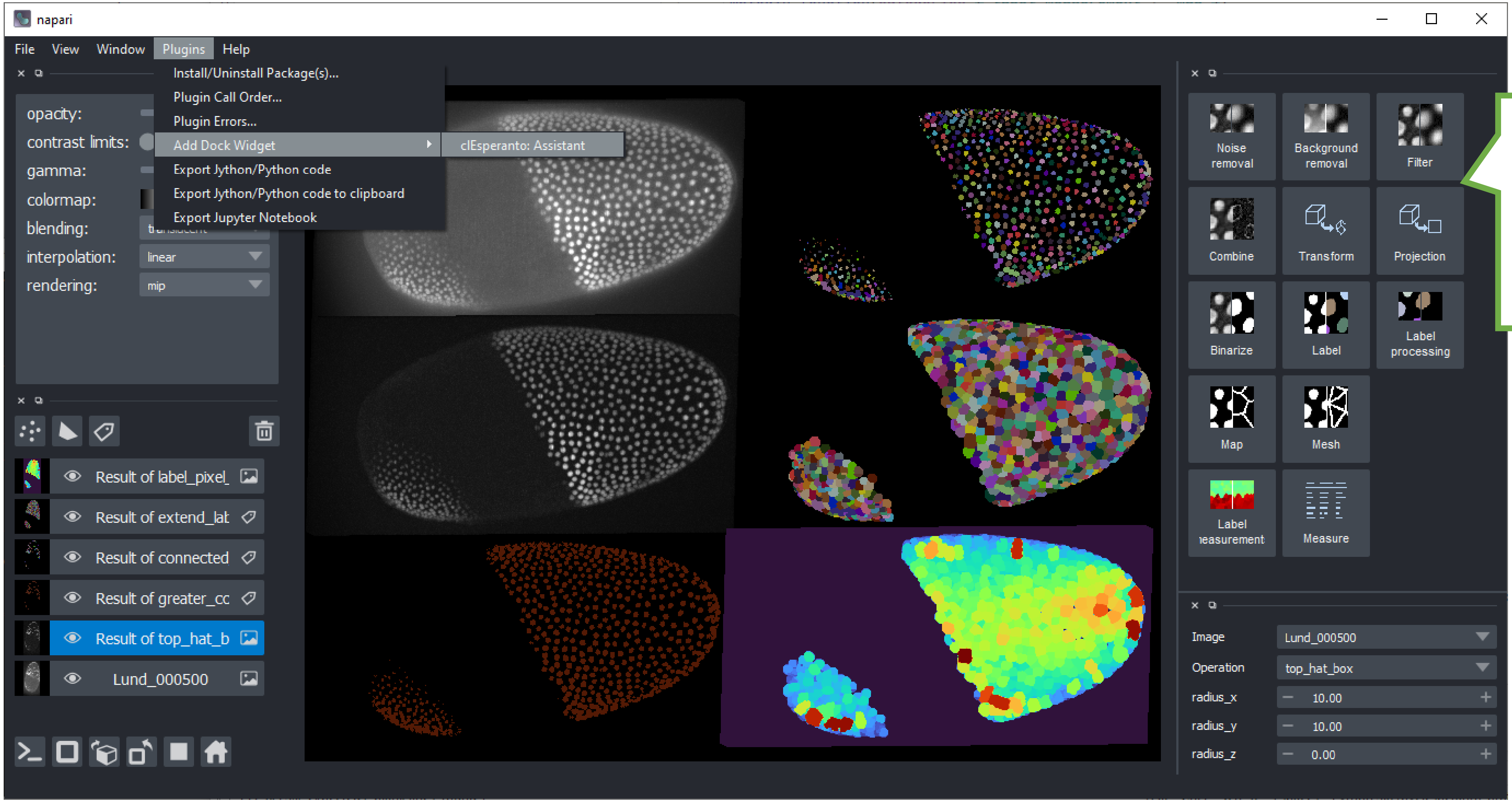


```
print(cle.voronoi_otsu_labeling.__doc__)
```

```
Applies two Gaussian blurs, spot detection, Otsu-thresholding and Voronoi-labeling.

    The thresholded binary image is flooded using the Voronoi approach starting from the found local maxima.
    Noise-removal sigma for spot detection and thresholding can be configured separately.

    Parameters
    ----------
    source : Image
    label_image_destination : Image
    spot_sigma : float

    outline_sigma : float


    Returns
    -------
    label_image_destination

    Examples
    --------
    >>> import pyclesperanto_prototype as cle
    >>> cle.voronoi_otsu_labeling(source, label_image_destination, 10, 2)

    References
    ----------
    .. [1] https://clij.github.io/clij2-docs/reference_voronoiOtsuLabeling
```

Why GPU-acceleration?

- … because it renders projects feasible which were not without GPUs



opacity:
contrast limits:
gamma:
colormap:
blending: translucent
interpolation: linear
rendering: mip

Result of label_pixel_
Result of extend_lat
Result of connected
Result of greater_cc
Result of top_hat_b
Lund_000500

Image: Lund_000500
Operation: top_hat_box
radius_x: 10.00
radius_y: 10.00
radius_z: 0.00

Lab / Bachelor / Master Projects available here. Get in touch!

PoL Physics of Life TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

Quantitative measurements

Robert Haase

With material from

Daniela Vorkel (Myers lab MPI CBG)

Cyrus Jin (Dana-Farber Cancer Institute - Harvard BCMP)

Federico M. Gasparoli (NIC@HMS)

Mauricio Rocha Martins, Norden Lab MPI CBG

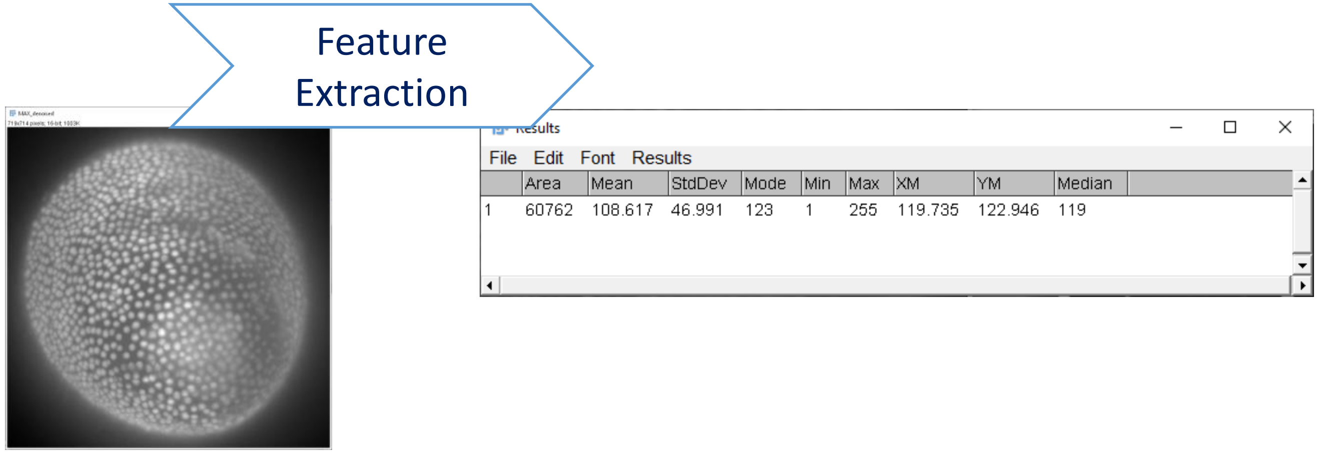
David Legland (INRAE)

May 2021

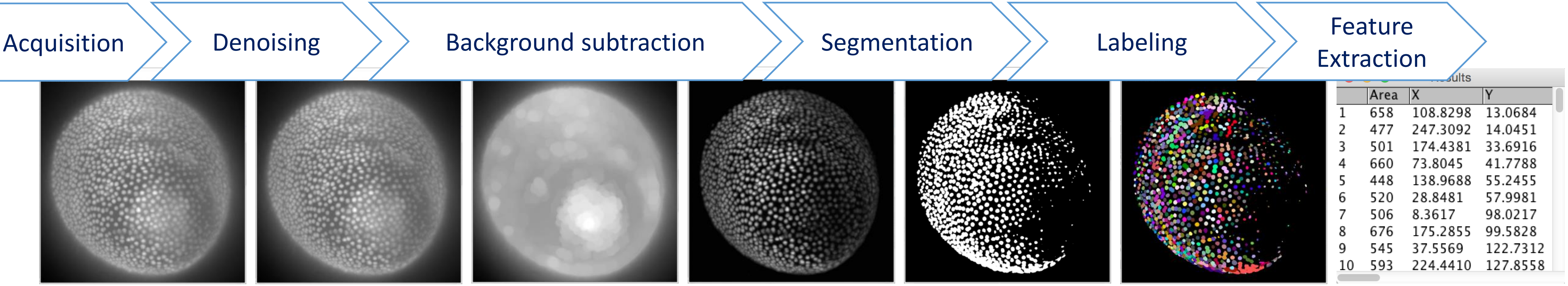
@haesleinhuepf

Feature extraction

- Feature extraction a *late* processing step in image analysis.
- It can be used for images, or segmented/labelled images



	Area	Mean	StdDev	Mode	Min	Max	XM	YM	Median
1	60762	108.617	46.991	123	1	255	119.735	122.946	119



	Area	X	Y
1	658	108.8298	13.0684
2	477	247.3092	14.0451
3	501	174.4381	33.6916
4	660	73.8045	41.7788
5	448	138.9688	55.2455
6	520	28.8481	57.9981
7	506	8.3617	98.0217
8	676	175.2855	99.5828
9	545	37.5569	122.7312
10	593	224.4410	127.8558

Image data source: Daniela Vorkel, Myers lab, MPI CBG

Feature extraction

- A *feature* is a countable or measurable property of an image or object.
- Goal of feature extraction is finding a minimal set of features to describe an object well enough to differentiate it from other objects.

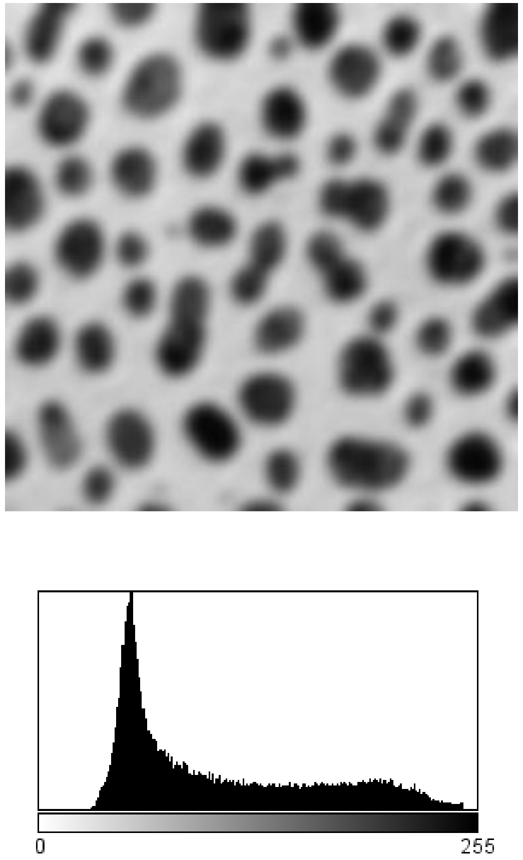
- Intensity based features
  - **Mean intensity**
  - **Standard deviation**
  - **Total intensity**
  - Textures
  - …
- Shape based / spatial features
  - Area / Volume
  - **Roundness**
  - **Solidity**
  - **Circularity /** Sphericity
  - Elongation
  - **Centroid**
  - **Bounding box**
  - …
- Spatio-temporal features
  - Displacement,
  - Speed,
  - Acceleration,
  - …
- Others
  - Overlap
  - Colocalisation
  - Network-analysis
  - …
- Mixed features
  - **Center of mass**
  - Local minima / maxima
  - …

Intensity based features

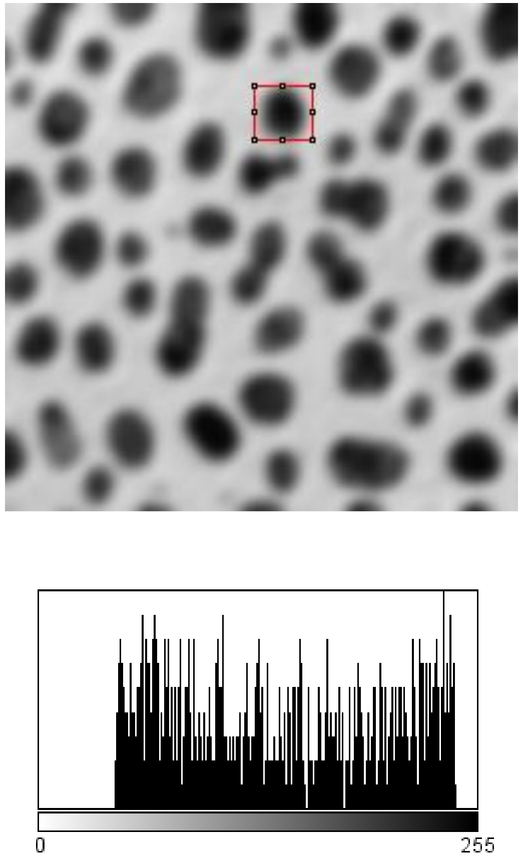
- Min / max
- Median
- Mean
- Mode
- Variance
- Standard deviation

- Can be derived from pixel values
- Don’t take spatial relationship of pixels into account

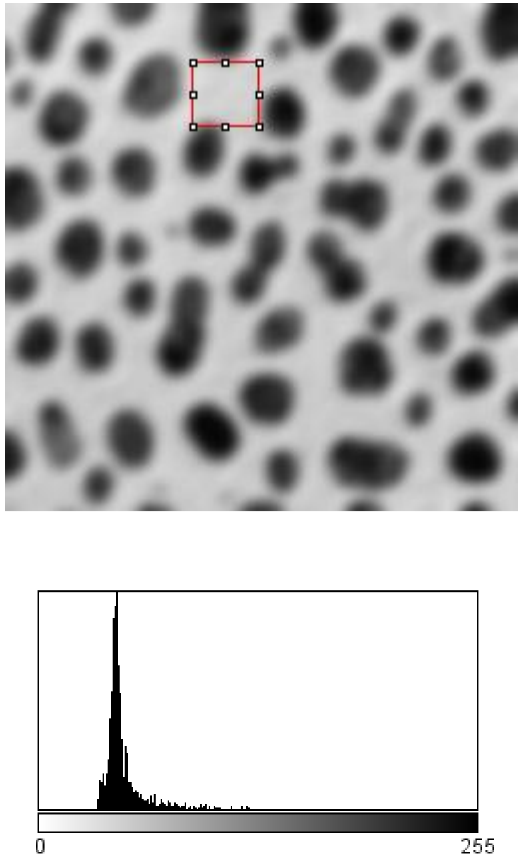
- See also:
  - descriptive statistics
  - histogram



```
Count: 65024      Min: 29
Mean: 103.301     Max: 248
StdDev: 57.991    Mode: 53 (1663)
```



```
Count: 783        Min: 44
Mean: 141.308     Max: 243
StdDev: 61.876    Mode: 236 (9)
```



```
Count: 1056       Min: 34
Mean: 49.016      Max: 122
StdDev: 12.685    Mode: 45 (120)
```

Center of mass

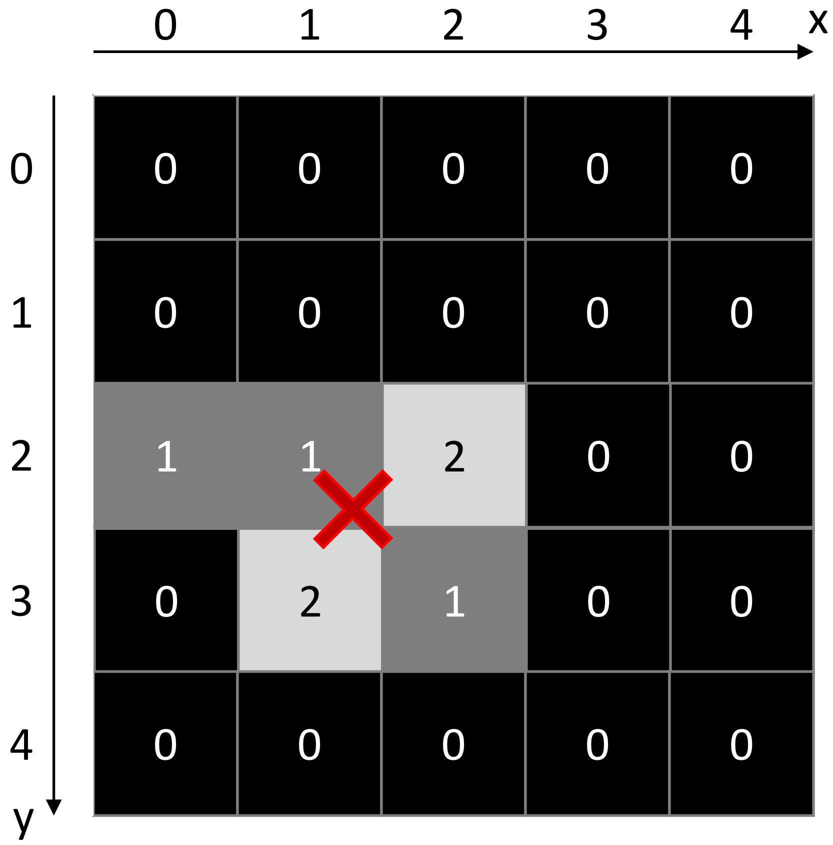
- Relative position in an image weighted by pixel intensities
  - x, y … pixel coordinates
  - w … image width
  - h … image height
  - µ … mean intensity
  - $g_{x,y}$ … pixel grey value
  - x_m, y_m … center of mass coordinates

$$\mu = \frac{1}{wh}\sum_{y=0}^{h-1}\sum_{x=0}^{w-1} g_{x,y}$$

$$x_m = \frac{1}{wh\mu}\sum_{y=0}^{h-1}\sum_{x=0}^{w-1} x\, g_{x,y}$$

$$y_m = \frac{1}{wh\mu}\sum_{y=0}^{h-1}\sum_{x=0}^{w-1} y\, g_{x,y}$$

"sum intensity"
"total intensity"



y \ x	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	1	1	2	0	0
3	0	2	1	0	0
4	0	0	0	0	0

x_m = 1/7 (1·0 + 1·1 + 2·2 + 2·1 + 1·2) = 1.3

y_m = 1/7 (1·2 + 1·2 + 2·3 + 2·2 + 1·3) = 2.4

Center of geometry / centroid

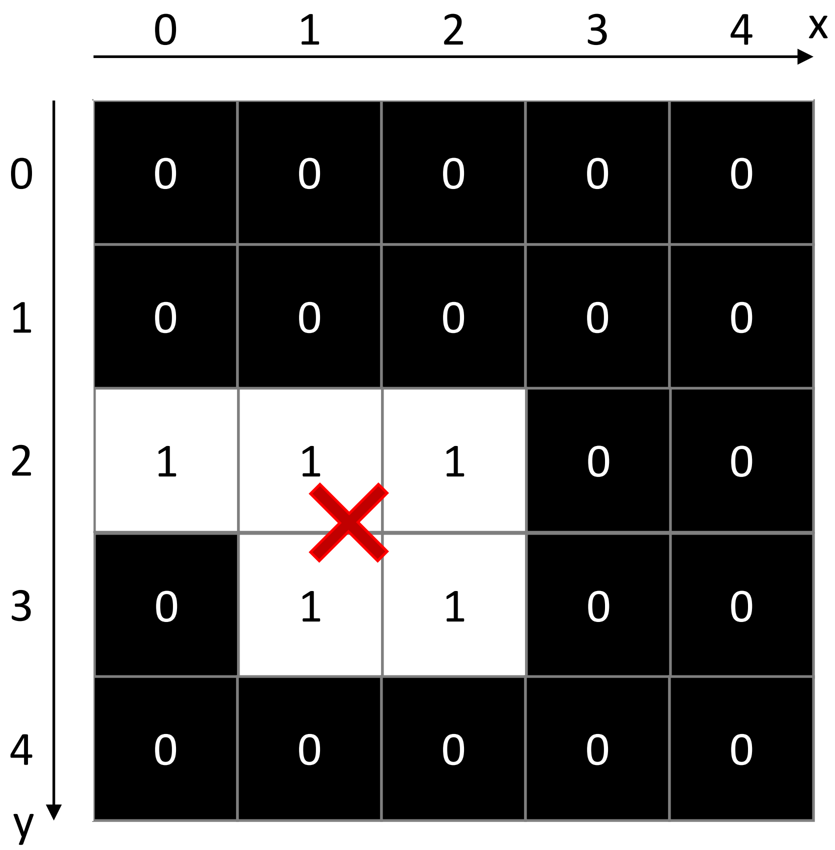
- Relative position in an image weighted by pixel intensities
- Special case of center of mass for binary images
  - x, y … pixel coordinates
  - w … image width
  - h … image height
  - µ … mean intensity
  - $g_{x,y}$ … pixel grey value, integer in range [0;1]
  - x_m, y_m … center of mass coordinates

$$\mu = \frac{1}{wh}\sum_{y=0}^{h-1}\sum_{x=0}^{w-1} g_{x,y}$$

$$x_m = \frac{1}{wh\mu}\sum_{y=0}^{h-1}\sum_{x=0}^{w-1} x\, g_{x,y}$$

$$y_m = \frac{1}{\boxed{wh\mu}}\sum_{y=0}^{h-1}\sum_{x=0}^{w-1} y\, g_{x,y}$$

Number of white pixels



y \ x	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	1	1	1	0	0
3	0	1	1	0	0
4	0	0	0	0	0

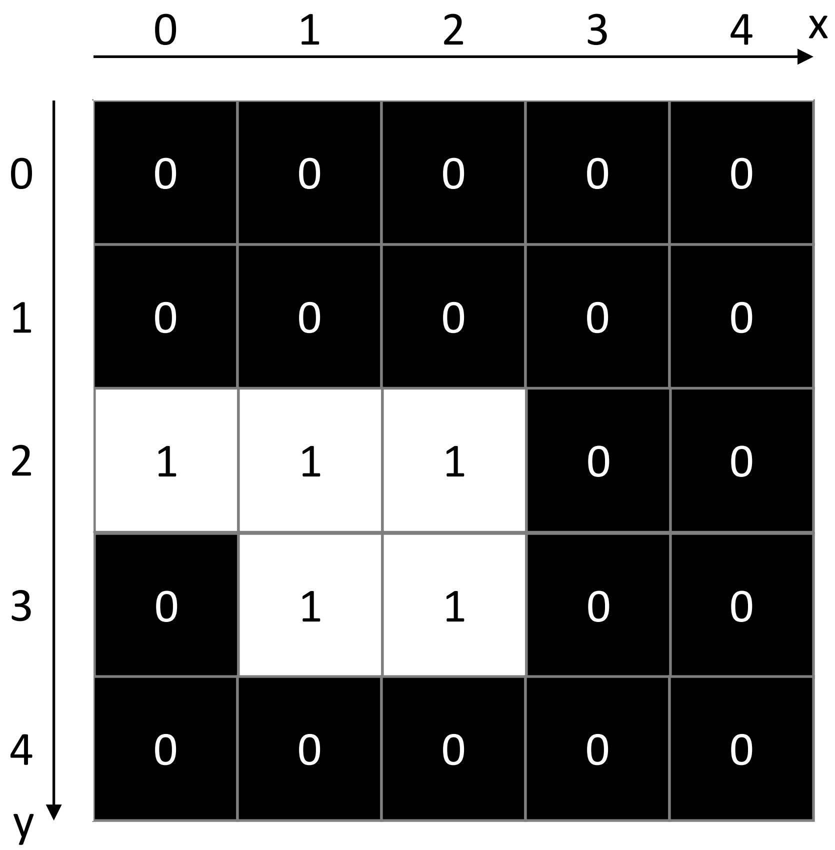
x_m = 1/5 (1·0 + 1·1 + 1·2 + 1·1 + 1·2) = 1.2

y_m = 1/5 (1·2 + 1·2 + 1·3 + 1·2 + 1·3) = 2.4

Bounding rectangle / bounding box

- Position and size of the smallest rectangle containing all pixels of an object
  - x_b, y_b … position of the bounding box
  - w_b … width of the bounding box
  - h_b … height of the bounding box

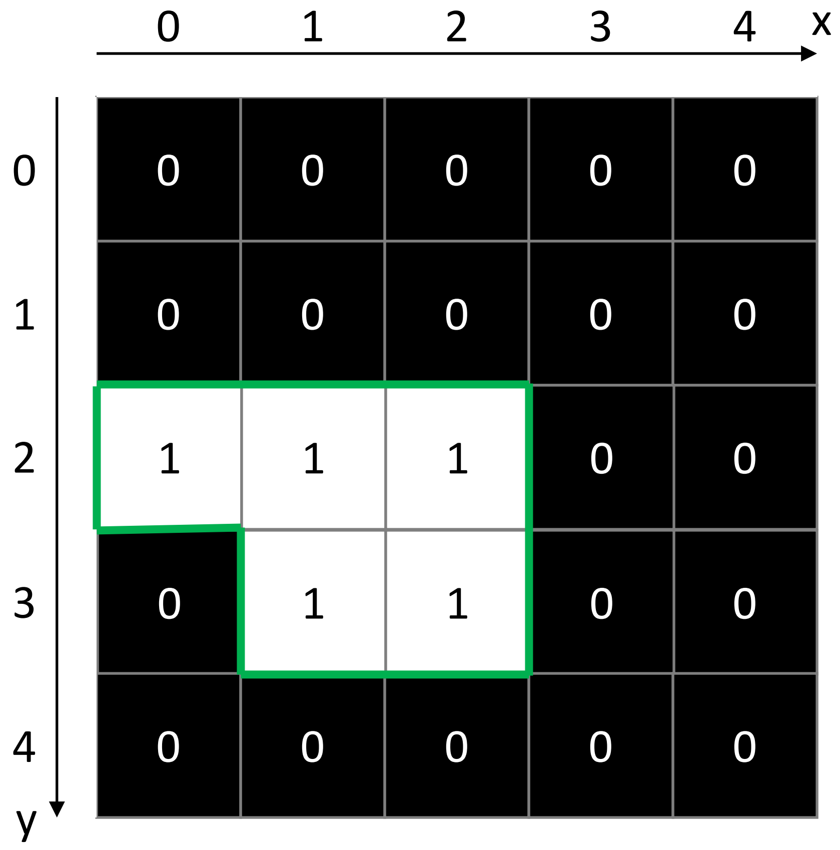
variable	value
x_b	0
y_b	2
w_b	3
h_b	2



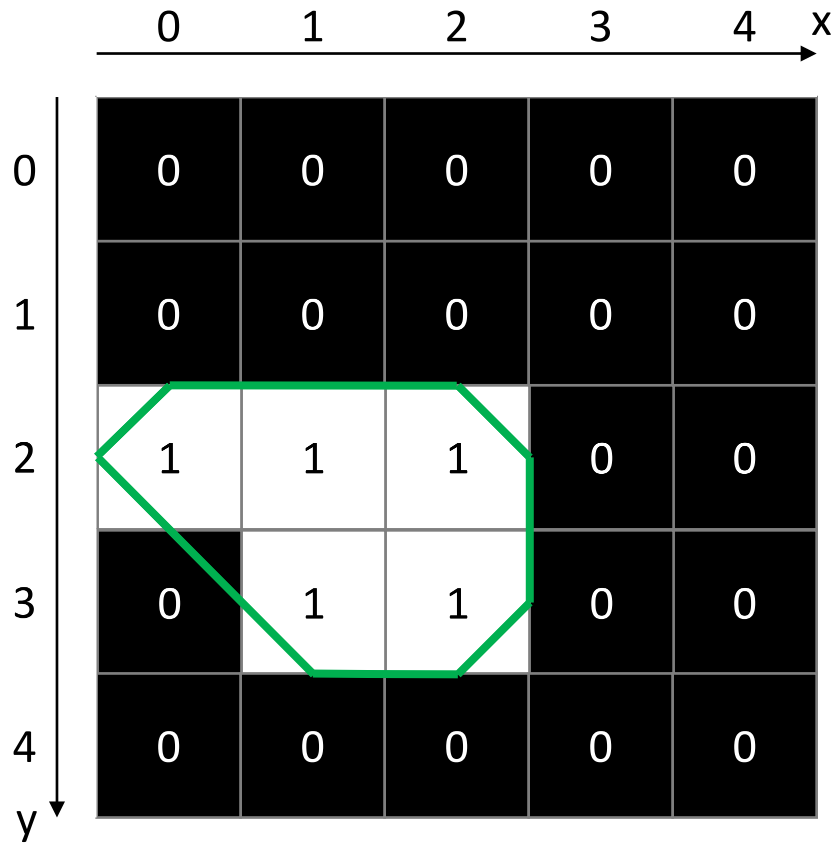
y \ x	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	1	1	1	0	0
3	0	1	1	0	0
4	0	0	0	0	0

Perimeter

- Length of the outline around an object
- Depends on the actual implementation



	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	1	1	1	0	0
3	0	1	1	0	0
4	0	0	0	0	0



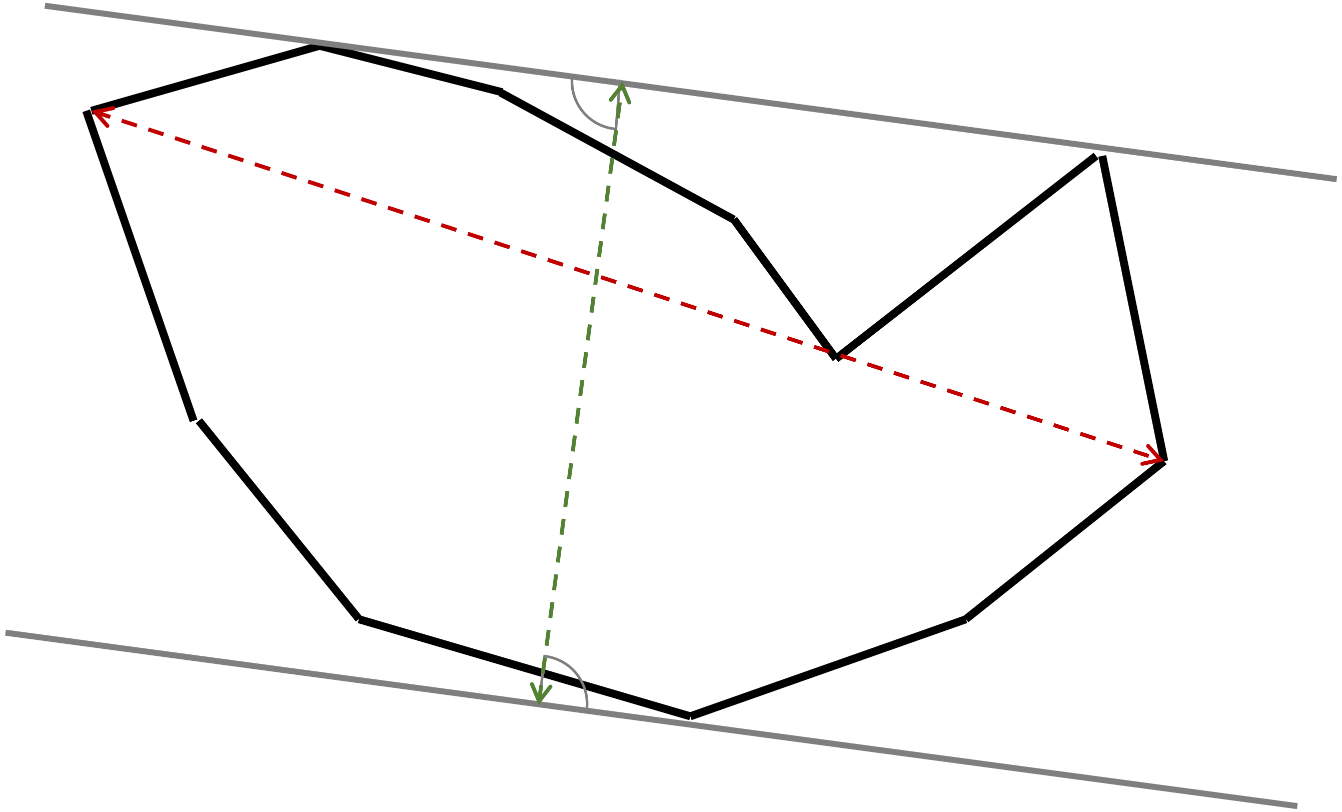
	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	1	1	1	0	0
3	0	1	1	0	0
4	0	0	0	0	0

Feret’s diameter

- Feret’s diameter describes the maximum distance between any two points of an outline.

- The minimum caliper (“Minimum Feret”) describes the shortest distance, the object would fit through.

- Feret and Minimum Feret do not need to be perpendicular to each other!

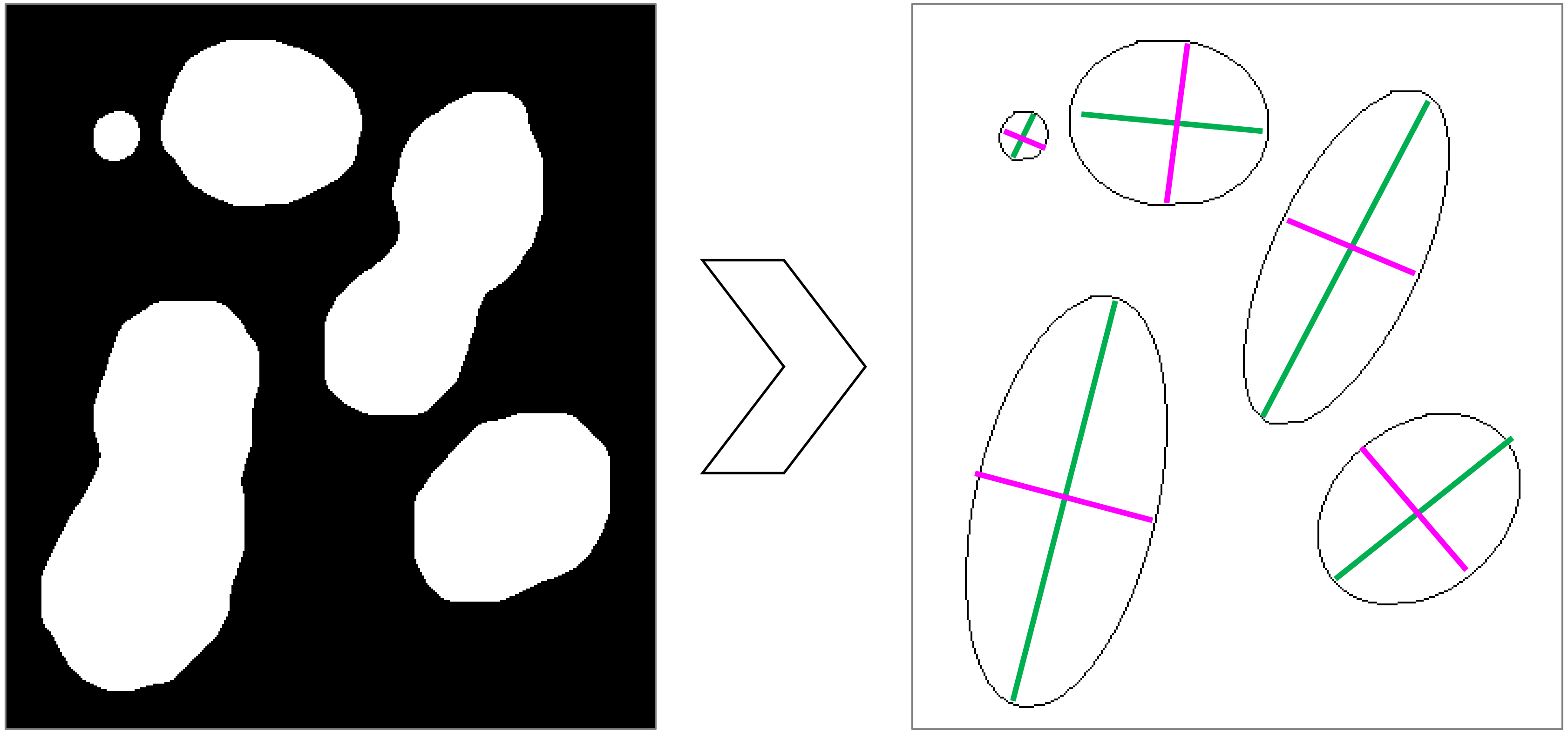


Fit ellipse

- For every object, find the optimal ellipse simplifying the object.

- Major axis … long diameter
- Minor axis … short diameter

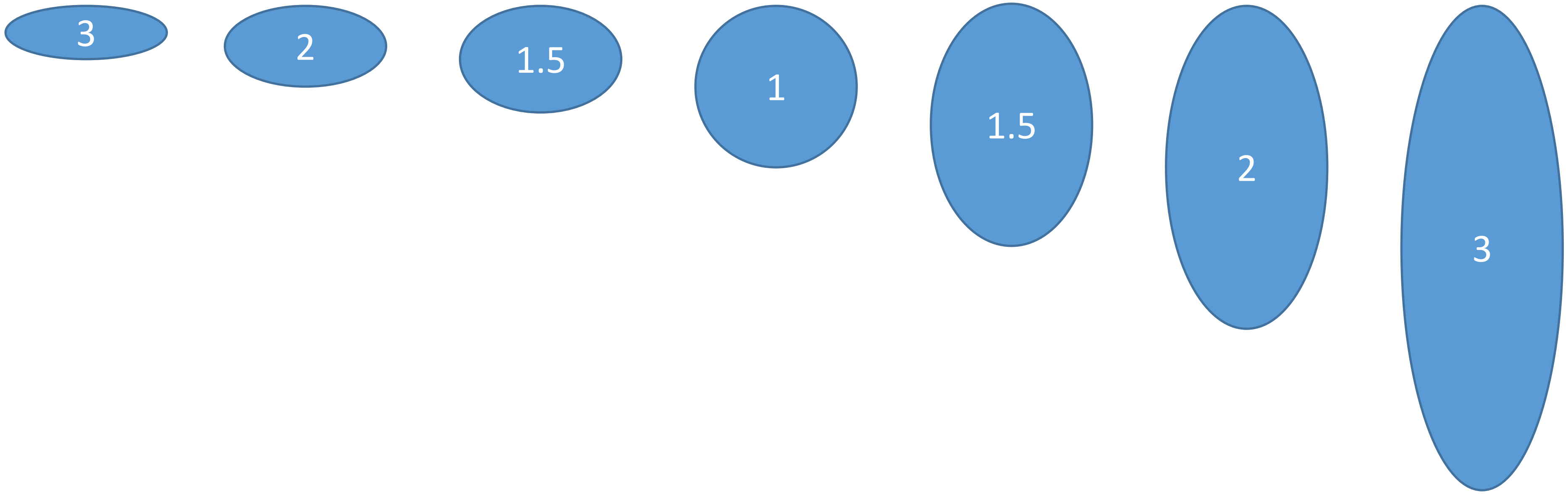
- Major and minor axis are perpendicular to each other



Aspect ratio

- The aspect ratio describes the elongation of an object.

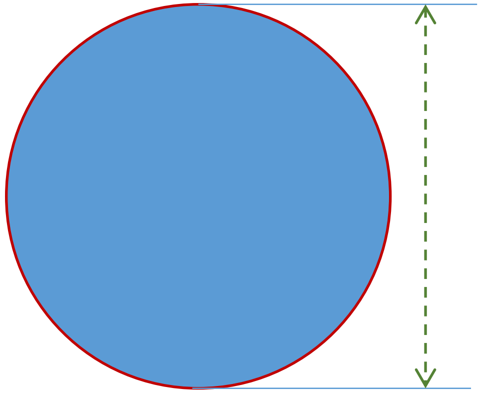
AR = major / minor



Roundness and circularity

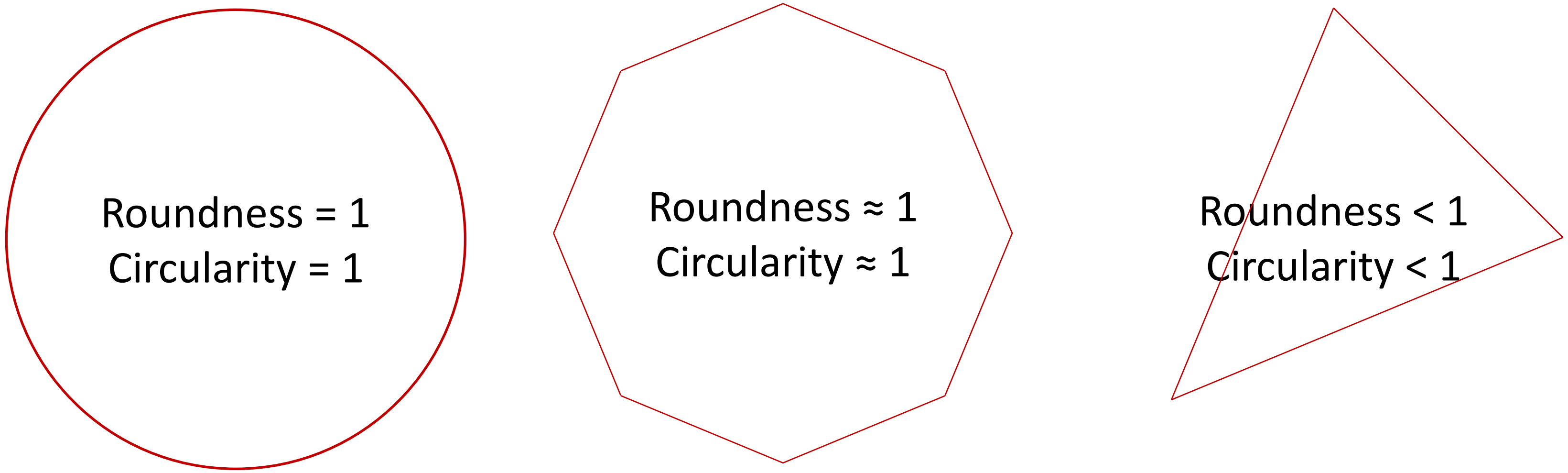
- The definition of a circle leads us to measurements of circularity and roundness.
- In case you use these measures, define them correctly. They are not standardized!

Diameter	d
Circumference	$C = \pi d$
Area	$A = \frac{\pi d^2}{4}$



$$roundness = \frac{4 \ * A}{\pi \, major^2}$$

$$circularity = \frac{4\pi \ * A}{perimeter^2}$$



Roundness = 1
Circularity = 1

Roundness ≈ 1
Circularity ≈ 1

Roundness < 1
Circularity < 1

- In 3D: ~~Circularity~~ -> Sphericity

$$sphericity = 36\pi \frac{V^2}{S^3}$$

Volume

Surface area

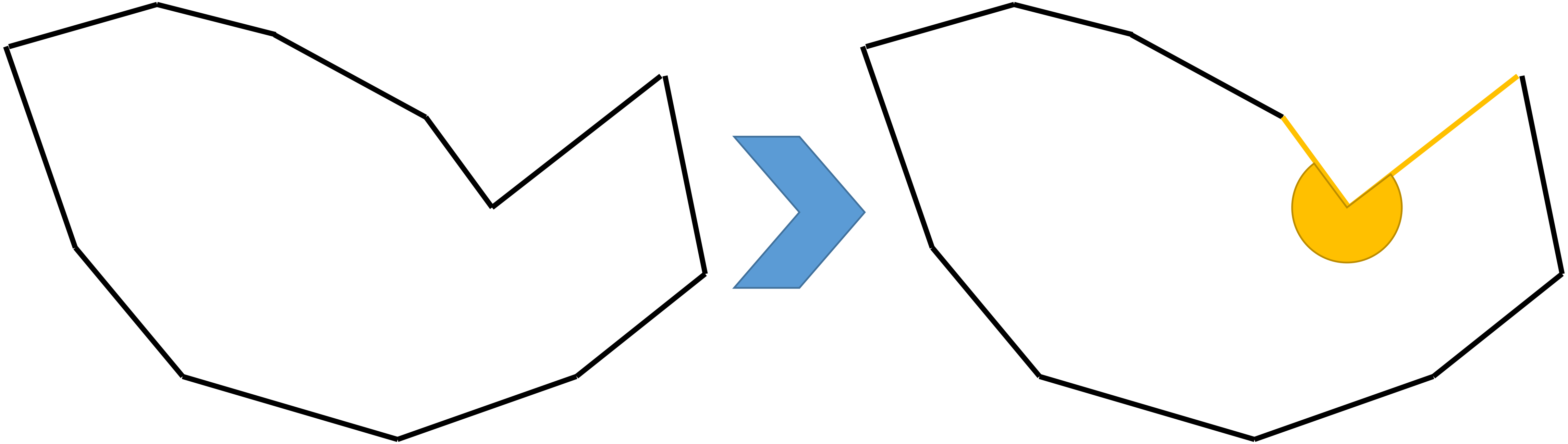
What’s the difference between circularity and roundness?

$$roundness = \frac{4\ * A}{\pi\ major^2}$$

$$circularity = \frac{4\pi\ * A}{perimeter^2}$$

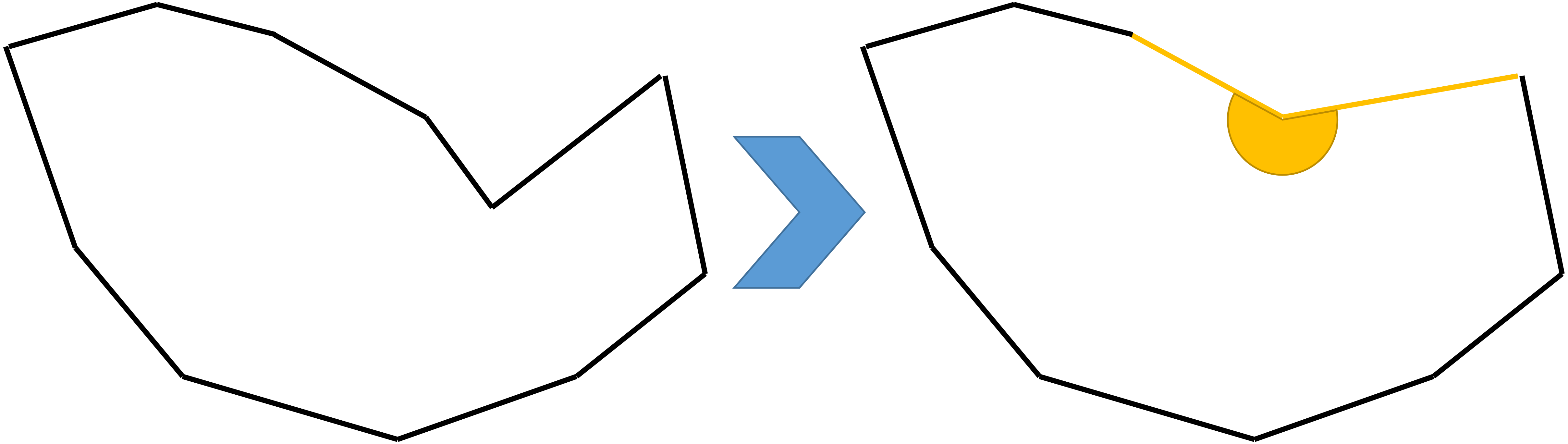
Convex hull

- By removing all concave corners of an object, we retrieve its convex hull.



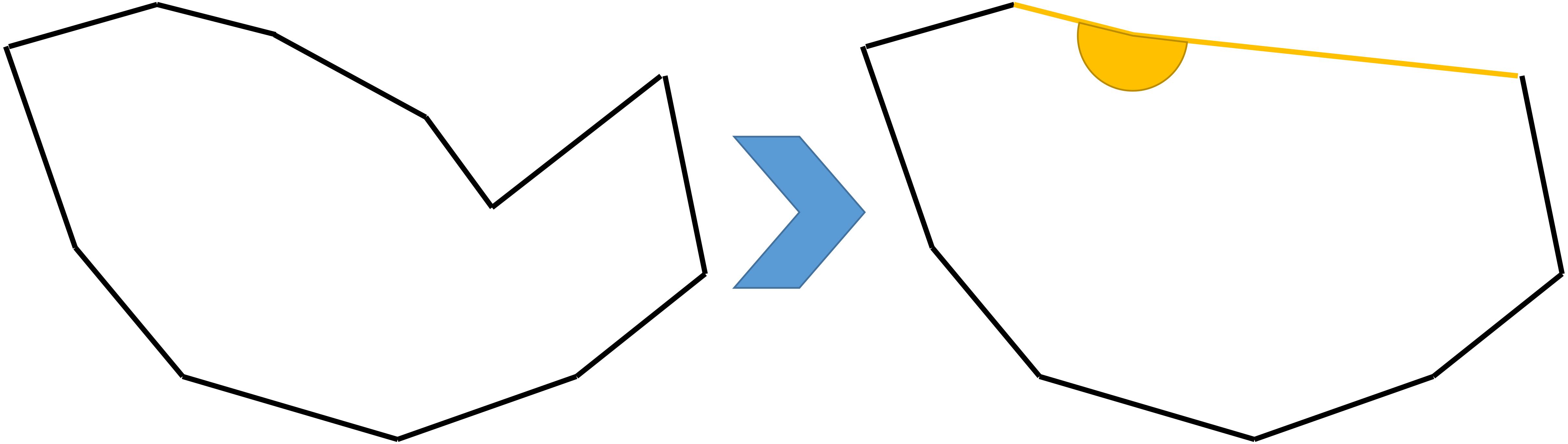
Convex hull

- By removing all concave corners of an object, we retrieve its convex hull.



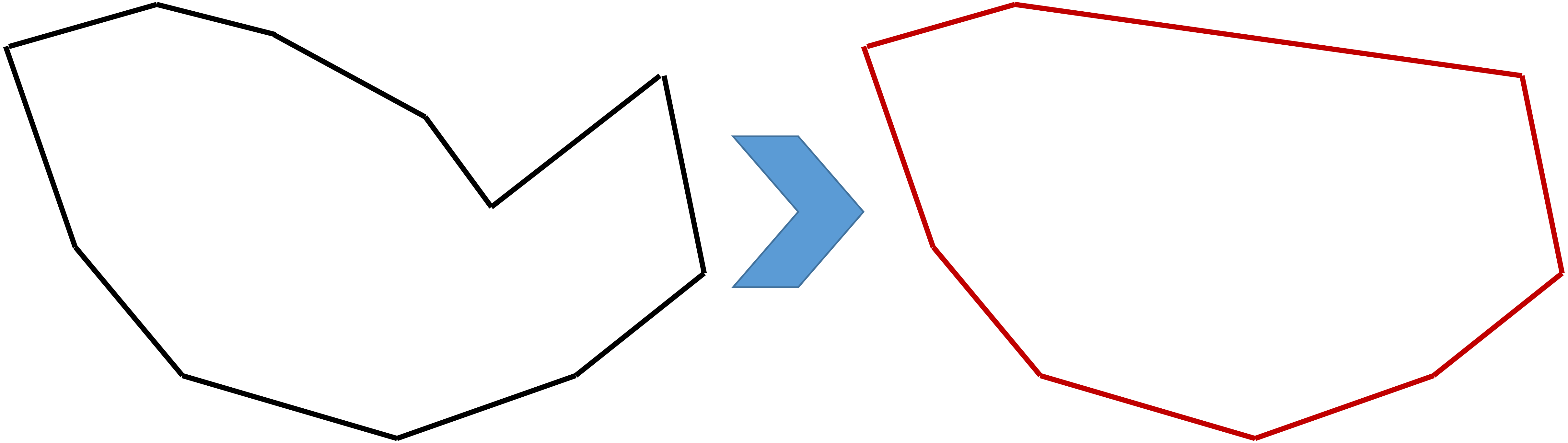
Convex hull

- By removing all concave corners of an object, we retrieve its convex hull.



Convex hull

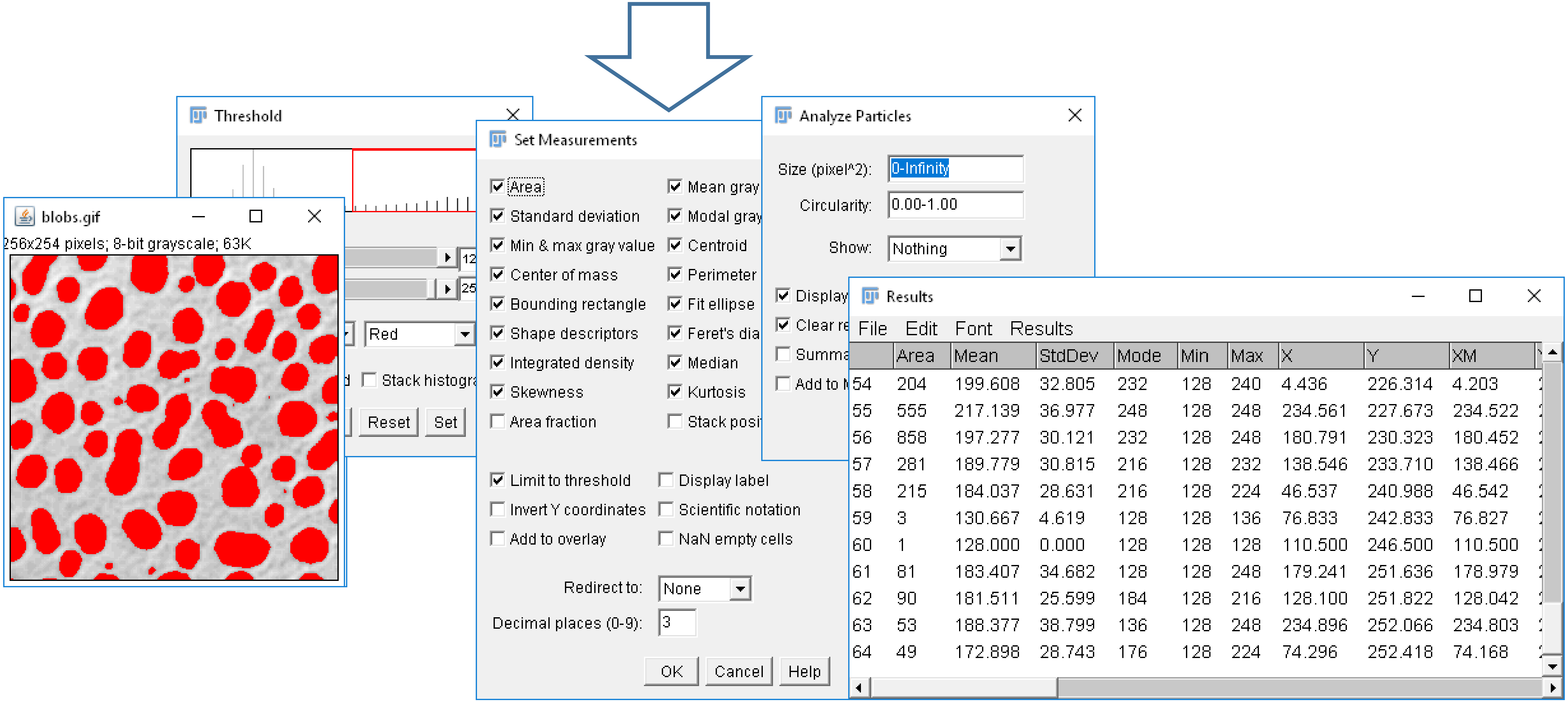
- By removing all concave corners of an object, we retrieve its convex hull.



$$solidity = \frac{A}{A_{convexHull}}$$

Measurements in 2D: The Particle Analyzer

- Tell Fiji what to measure with *Analyze > Set Measurements*



Set Measurements

☑ Area
☑ Mean gray
☑ Standard deviation
☑ Modal gray
☑ Min & max gray value
☑ Centroid
☑ Center of mass
☑ Perimeter
☑ Bounding rectangle
☑ Fit ellipse
☑ Shape descriptors
☑ Feret's dia
☑ Integrated density
☑ Median
☑ Skewness
☑ Kurtosis
□ Area fraction
□ Stack posi

☑ Limit to threshold
□ Display label
□ Invert Y coordinates
□ Scientific notation
□ Add to overlay
□ NaN empty cells

Redirect to: None
Decimal places (0-9): 3

OK Cancel Help

Analyze Particles

Size (pixel^2): 0-Infinity
Circularity: 0.00-1.00
Show: Nothing

☑ Display
☑ Clear re
□ Summa
□ Add to M

Results

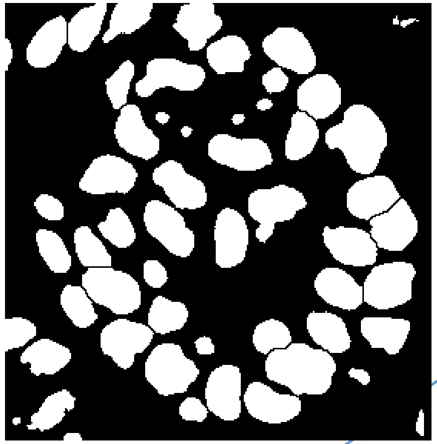
File Edit Font Results

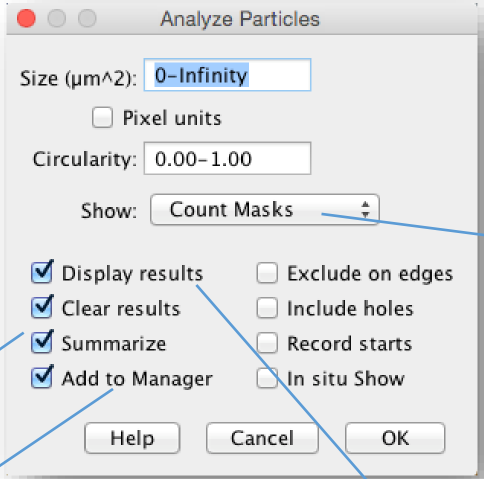
	Area	Mean	StdDev	Mode	Min	Max	X	Y	XM
54	204	199.608	32.805	232	128	240	4.436	226.314	4.203
55	555	217.139	36.977	248	128	248	234.561	227.673	234.522
56	858	197.277	30.121	232	128	248	180.791	230.323	180.452
57	281	189.779	30.815	216	128	232	138.546	233.710	138.466
58	215	184.037	28.631	216	128	224	46.537	240.988	46.542
59	3	130.667	4.619	128	128	136	76.833	242.833	76.827
60	1	128.000	0.000	128	128	128	110.500	246.500	110.500
61	81	183.407	34.682	128	128	248	179.241	251.636	178.979
62	90	181.511	25.599	184	128	216	128.100	251.822	128.042
63	53	188.377	38.799	136	128	248	234.896	252.066	234.803
64	49	172.898	28.743	176	128	224	74.296	252.418	74.168

Measurements in 2D: The Particle Analyzer

- The particle analyser
  - performs connected components labelling,
  - can make selections and
  - Measure properties of particles (feature extraction)

- Analyze > Analyse particles…





Analyze Particles

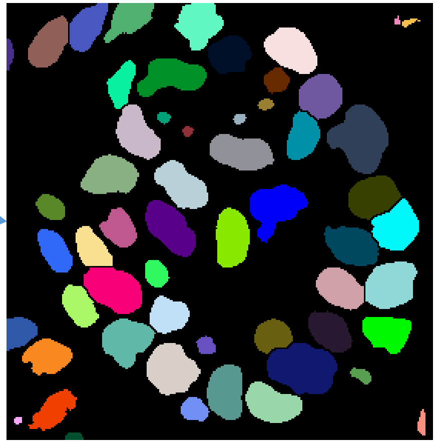
Size (µm^2): 0-Infinity

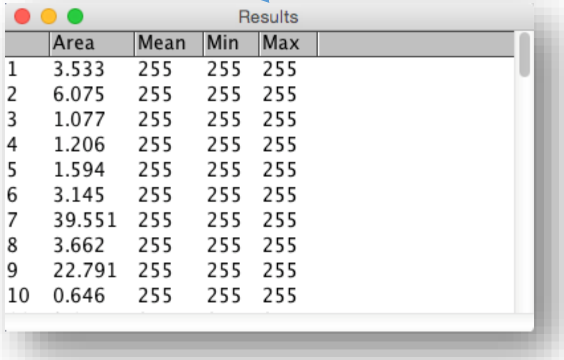
□ Pixel units

Circularity: 0.00-1.00

Show: Count Masks

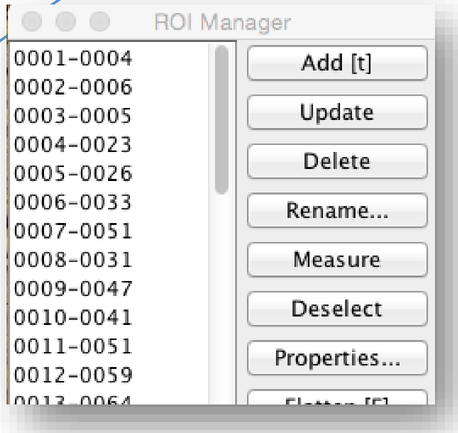
☑ Display results
☑ Clear results
☑ Summarize
☑ Add to Manager
□ Exclude on edges
□ Include holes
□ Record starts
□ In situ Show





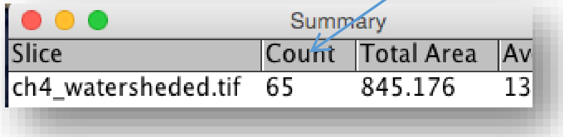
Results

	Area	Mean	Min	Max
1	3.533	255	255	255
2	6.075	255	255	255
3	1.077	255	255	255
4	1.206	255	255	255
5	1.594	255	255	255
6	3.145	255	255	255
7	39.551	255	255	255
8	3.662	255	255	255
9	22.791	255	255	255
10	0.646	255	255	255



ROI Manager

0001-0004
0002-0006
0003-0005
0004-0023
0005-0026
0006-0033
0007-0051
0008-0031
0009-0047
0010-0041
0011-0051
0012-0059
0013-0064



Summary

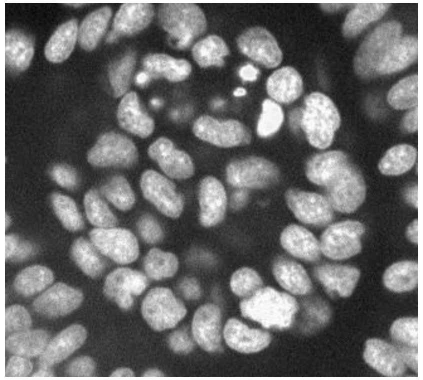
Slice	Count	Total Area	Av
ch4_watersheded.tif	65	845.176	13

Slide adapted from Haase, Lombardot, Scientific Computing Facility, MPI CBG, licensed CC BY-NC 4.0

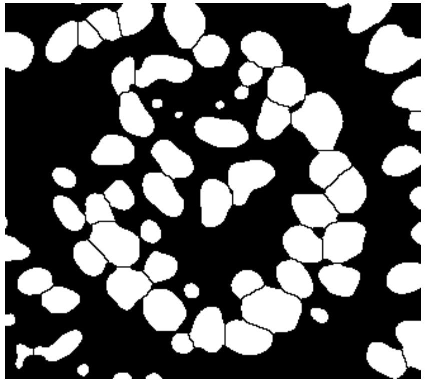
Image data source: Mauricio Rocha Martins, Norden Lab MPI CBG

Measurements in 2D: The Particle Analyzer

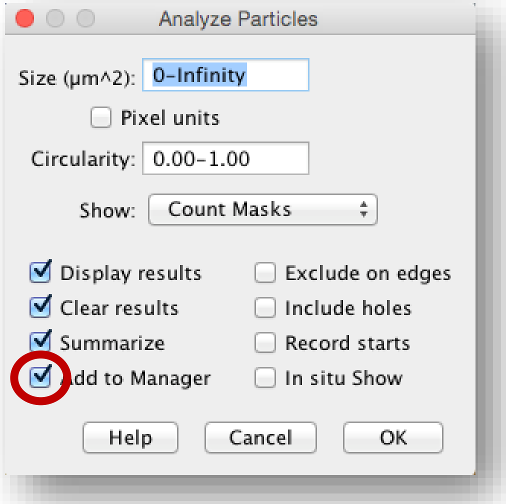
- Suggested workflow



Threshold, refine masks



Particle analysis (CCA)



Analyze Particles

Size (µm^2): 0-Infinity

□ Pixel units

Circularity: 0.00-1.00

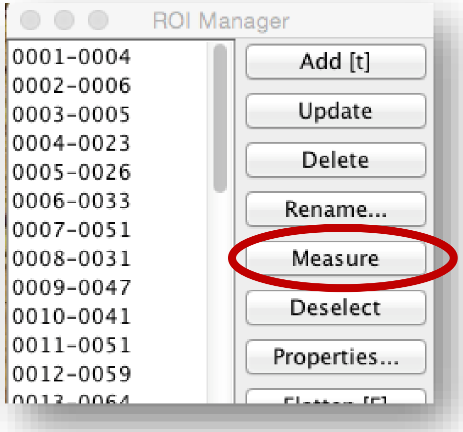
Show: Count Masks

☑ Display results
☑ Clear results
☑ Summarize
☑ Add to Manager

□ Exclude on edges
□ Include holes
□ Record starts
□ In situ Show

Add to manager

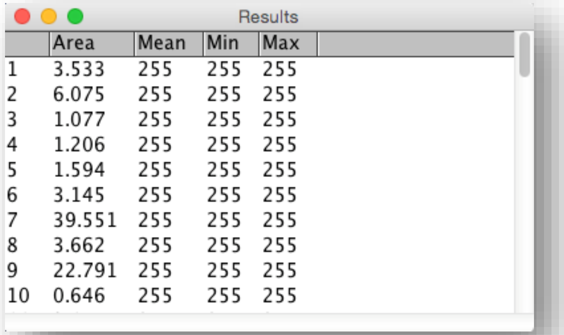
Select original image



ROI Manager

0001-0004
0002-0006
0003-0005
0004-0023
0005-0026
0006-0033
0007-0051
0008-0031
0009-0047
0010-0041
0011-0051
0012-0059
0013-0064

Measure



Results

	Area	Mean	Min	Max
1	3.533	255	255	255
2	6.075	255	255	255
3	1.077	255	255	255
4	1.206	255	255	255
5	1.594	255	255	255
6	3.145	255	255	255
7	39.551	255	255	255
8	3.662	255	255	255
9	22.791	255	255	255
10	0.646	255	255	255

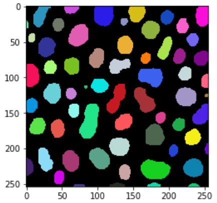
Image data source: Mauricio Rocha Martins, Norden Lab MPI CBG

Measurements with scikit-image: regionprops

- Regionprops to collect measurements

```
from skimage import measure


# analyse objects
properties = measure.regionprops(label_image, intensity_image=image)
```

We start with a label image



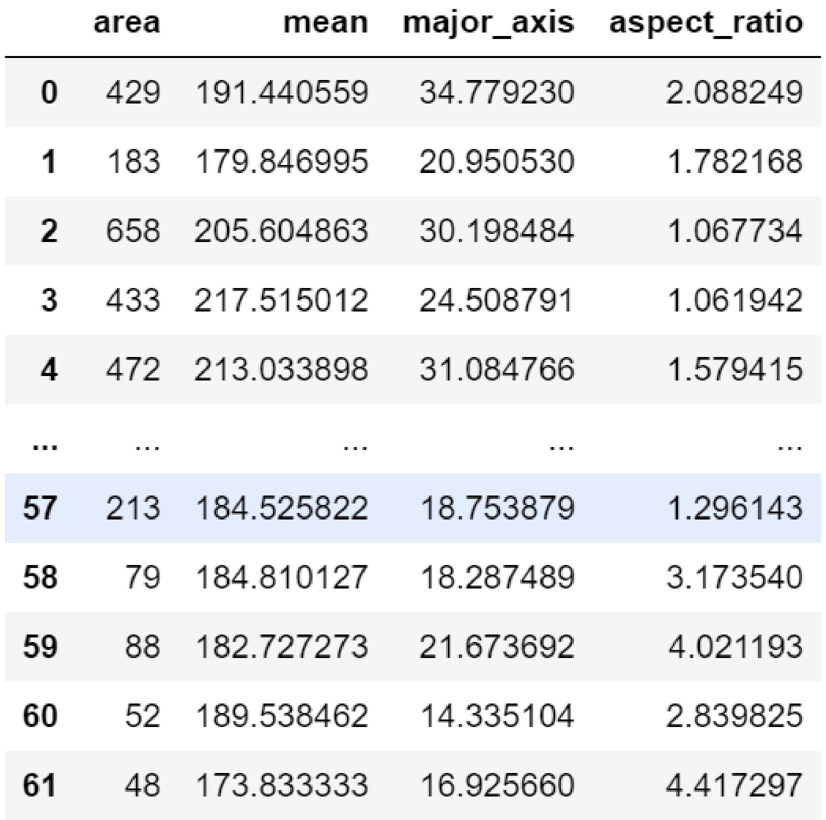
- Reorganize in a dictionary of lists

```
statistics = {
    'area':       [p.area                for p in properties],
    'mean':       [p.mean_intensity      for p in properties],
    'major_axis': [p.major_axis_length   for p in properties]
}
```

- Visualize as table, a.k.a. pandas DataFrame

```
import pandas as pd
dataframe = pd.DataFrame(statistics)
```

https://pandas.pydata.org/



	area	mean	major_axis	aspect_ratio
0	429	191.440559	34.779230	2.088249
1	183	179.846995	20.950530	1.782168
2	658	205.604863	30.198484	1.067734
3	433	217.515012	24.508791	1.061942
4	472	213.033898	31.084766	1.579415
...	...	...	...	...
57	213	184.525822	18.753879	1.296143
58	79	184.810127	18.287489	3.173540
59	88	182.727273	21.673692	4.021193
60	52	189.538462	14.335104	2.839825
61	48	173.833333	16.925660	4.417297

- Save to disk

```
dataframe.to_csv("blobs_analysis.csv")
```

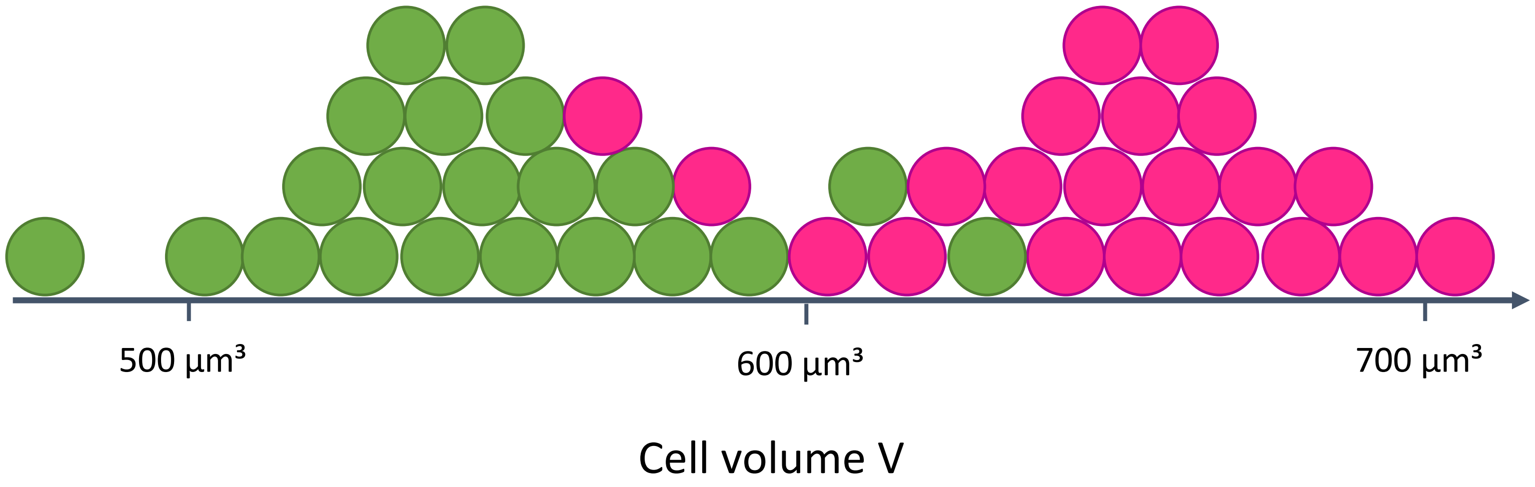
Summary

Today, you learned

- GPU-acceleration
- Quantitative measurements

Coming up next:

- Introduction to Biostatistics with Anna Poetsch


500 µm³
600 µm³
700 µm³
Cell volume V