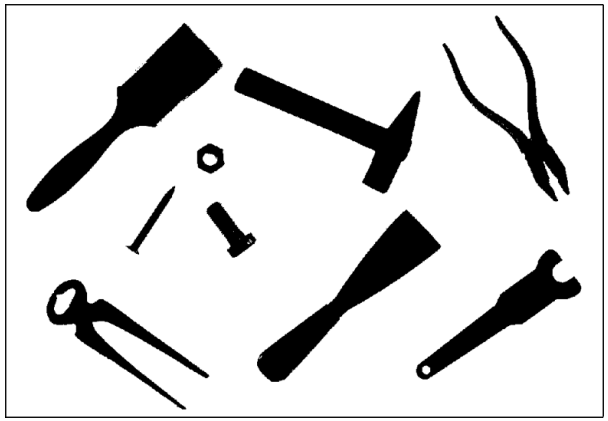
INF250

Regions and objects

Foreground, Background

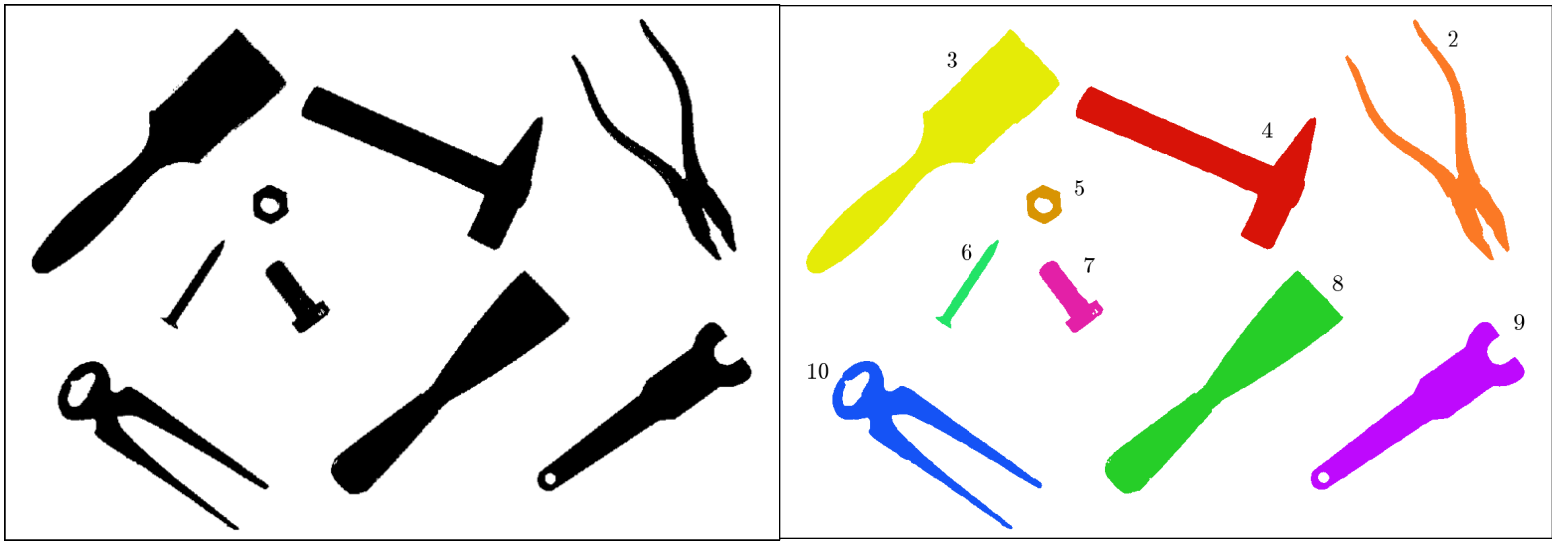
- In a binary image:
  - Pixels with values 1(255) belongs to the foreground and pixels with value 0 belongs to the background
  - Objects are part of neighboring structures or sample

- Our goal is to define objects and their shape
- Objects are defined by pixels that touch each other in coupled binary regions



Contours

1. Find object
2. Define contours of the objects



How to find regions ?

- An important task is to find which pixels that defines a region and how many regions there are in the image

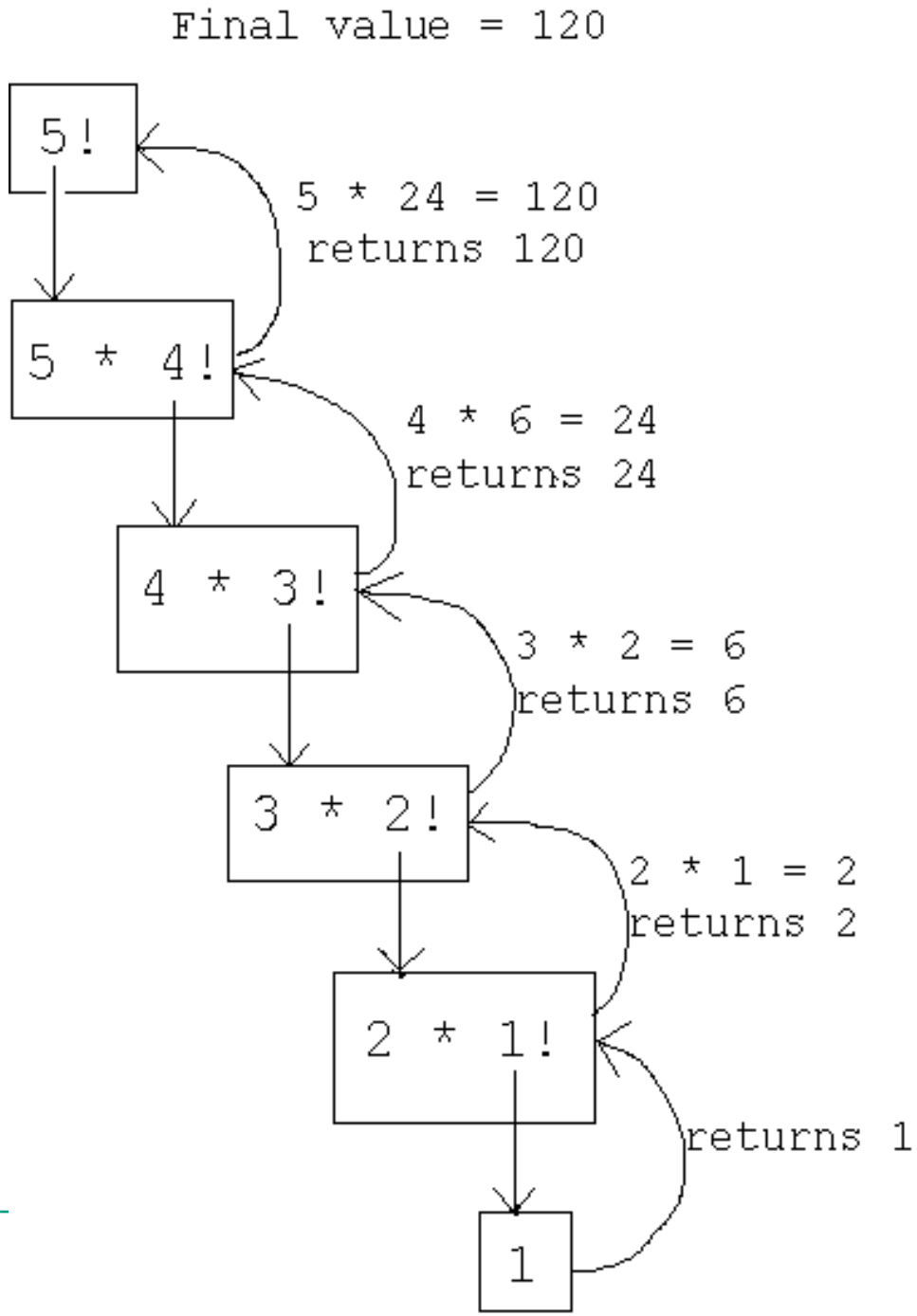
- Two methods are described in the book
  - «Flood Filling»
  - «Sequential region marking»

Regionlabeling

- The underlying algorithm «Flood Fill»:
  - Search for all non marked foreground pixels
  - Fill (mark) the rest of all the neighboring pixels in the region
  - Search for the next pixel outside the earlier found neighborhood
  - Analogy: Water flowing into all the neighboring pixels
  - Different algorithms exist for how the flow is defined
    - Depth-First
    - Breadth-First

Recursive function

- Recursion is programming algorithm where the function calls itself one or several times
- Usually a value is returned from a function call
- A recursive function need a stop criteria (a base) in order to avoid an endless loop
- A common example is the factorial function
  - N!
  - n! = n * (n-1)!, if n > 1 and f(1) = 1


Final value = 120
5!
5 * 24 = 120
returns 120
5 * 4!
4 * 6 = 24
returns 24
4 * 3!
3 * 2 = 6
returns 6
3 * 2!
2 * 1 = 2
returns 2
2 * 1!
returns 1
1

```python
def factorial(n):
    print("factorial has been called with n = " + str(n))
    if n == 1:
        return 1
    else:
        res = n * factorial(n-1)
        print("intermediate result for ", n, " * factorial(" ,n-1, "): ",res)
        return res

print(factorial(5))
```

Python

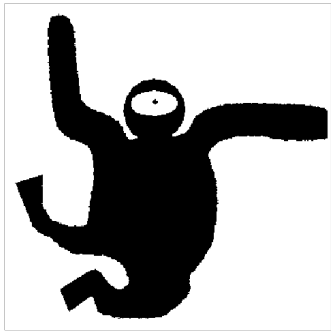
```
In [5]: %run /Users/kkvaal/CloudStation/ImageryPy/pyAMT/faktor.py
factorial has been called with n = 5
factorial has been called with n = 4
factorial has been called with n = 3
factorial has been called with n = 2
factorial has been called with n = 1
('intermediate result for ', 2, ' * factorial(', 1, '): ', 2)
('intermediate result for ', 3, ' * factorial(', 2, '): ', 6)
('intermediate result for ', 4, ' * factorial(', 3, '): ', 24)
('intermediate result for ', 5, ' * factorial(', 4, '): ', 120)
120
```

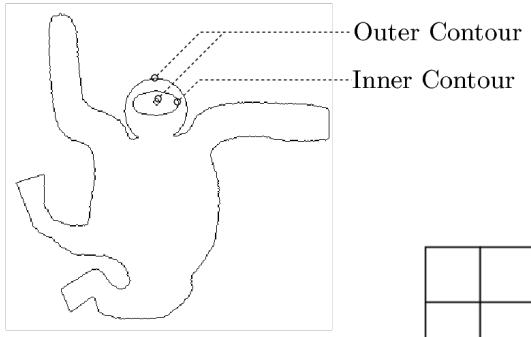


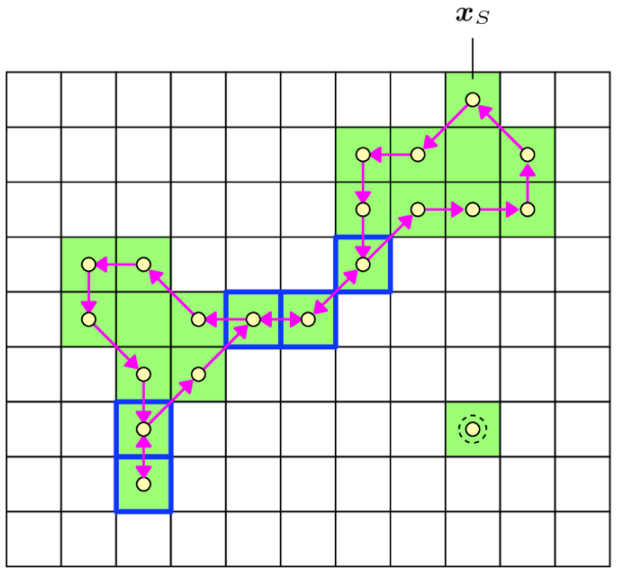
```
In [6]:
```

Region contours

- Inner and outer contour







Representation of areas

- Run Length Encoding
- Enkel og oversiktlig
- Benyttet i datakompresjon

$$Run_i = \langle \text{row}_i, \text{column}_i, \text{length}_i \rangle$$

Bitmap

	0	1	2	3	4	5	6	7	8
0									
1			×	×	×	×	×	×	
2									
3					×	×	×	×	
4		×	×	×		×	×	×	
5	×	×	×	×	×	×	×	×	×
6									

$\longrightarrow$

RLE

$\langle \text{row}, \text{column}, \text{length} \rangle$

$\langle 1, 2, 6 \rangle$
$\langle 3, 4, 4 \rangle$
$\langle 4, 1, 3 \rangle$
$\langle 4, 5, 3 \rangle$
$\langle 5, 0, 9 \rangle$

Chain Code

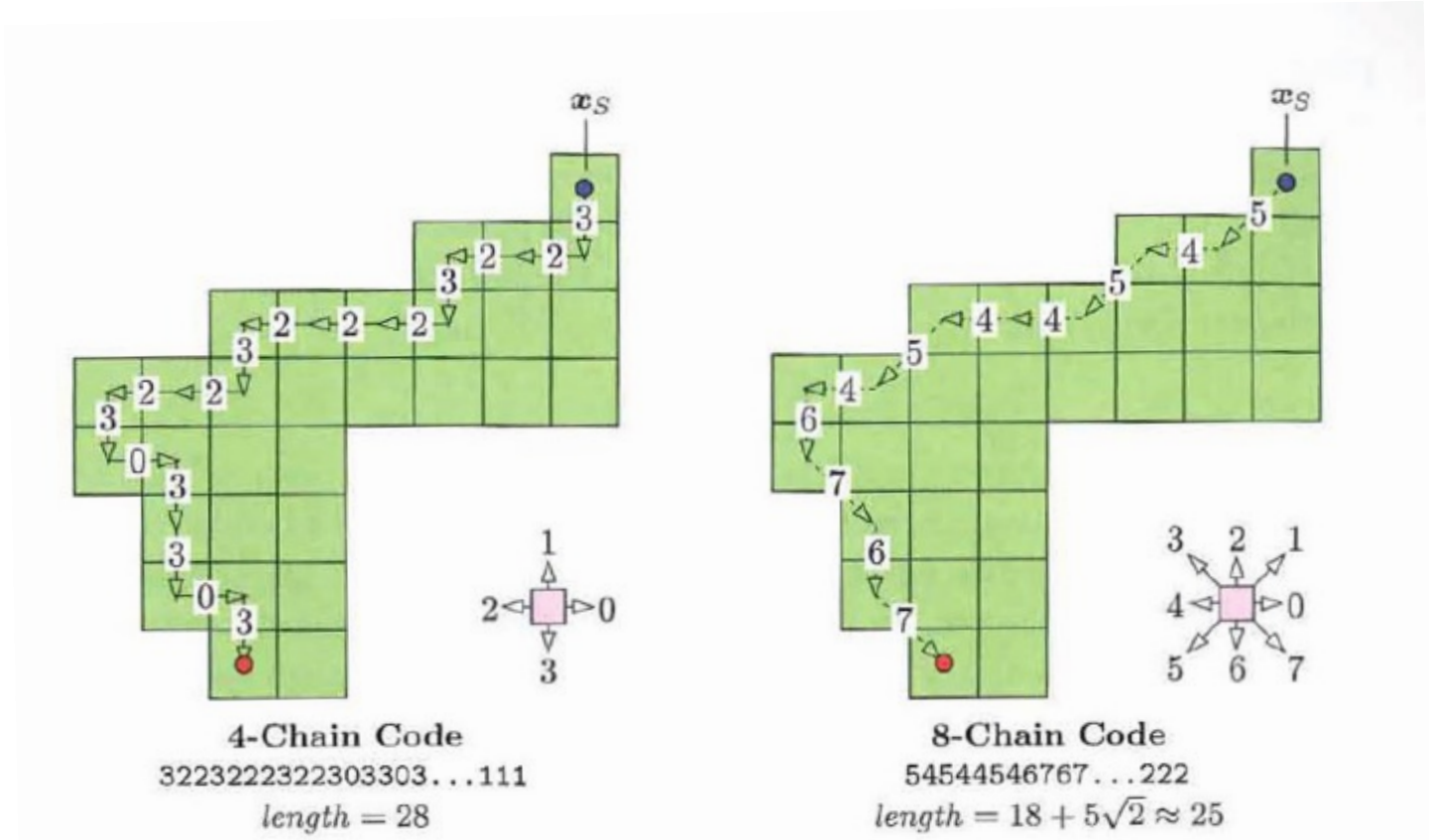
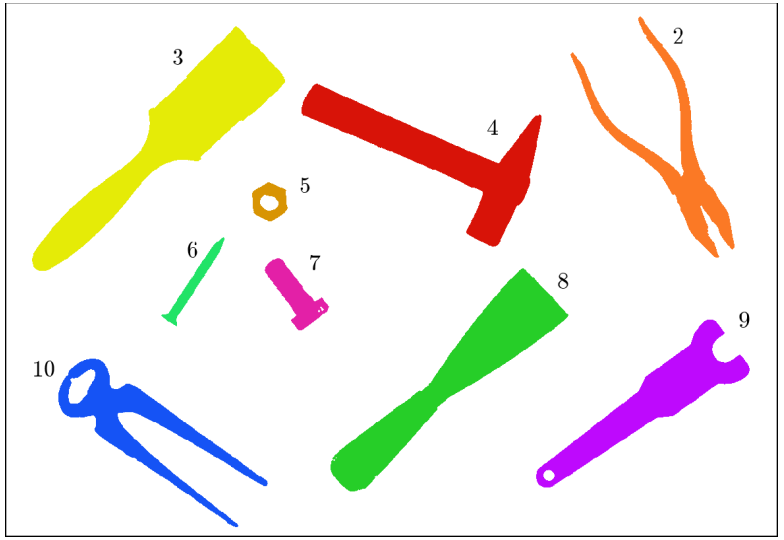


Figure 2.14 Chain codes with 4- and 8-connected neighborhoods. To compute a chain code, begin traversing the contour from a given starting point x_S. Encode the relative position between adjacent contour points using the directional code for either 4-connected (left) or 8-connected (right) neighborhoods. The length of the resulting path, calculated as the sum of the individual segments, can be used to approximate the true length of the contour.

Contours and statistics

- After all regions and contours are found the statistics of the regions can be computed

Label	Area (*pixels*)	Bounding Box (*left, top, right, bottom*)	Center (x_c, y_c)
2	14978	(887, 21, 1144, 399)	(1049.7, 242.8)
3	36156	(40, 37, 438, 419)	(261.9, 209.5)
4	25904	(464, 126, 841, 382)	(680.6, 240.6)
5	2024	(387, 281, 442, 341)	(414.2, 310.6)
6	2293	(244, 367, 342, 506)	(294.4, 439.0)
7	4394	(406, 400, 507, 512)	(454.1, 457.3)
8	29777	(510, 416, 883, 765)	(704.9, 583.9)
9	20724	(833, 497, 1168, 759)	(1016.0, 624.1)
10	16566	(82, 558, 411, 821)	(208.7, 661.6)



Geometrical features

- Perimeter: $P(R) = \sum_{i=0}^{M-1} length(c_i')$

  = 1 when c=0,2,4,6
  = $\sqrt{2}$ when c=1,3,5,7

- Area: A(*R*) = N (number of pixels)

- Circularity $C(R) = 4\pi \cdot \frac{A(R)}{P^2(R)}$

Circularity(sirkularitet)

Figure 2.15 Circularity values for different shapes. Shown are the corresponding estimates for Circularity($\mathcal{R}$) as defined in Eqn. (2.12).

Circularity vs Roundness

There is a difference between the terms «Circularity» and «Roundness» (not described in the book)

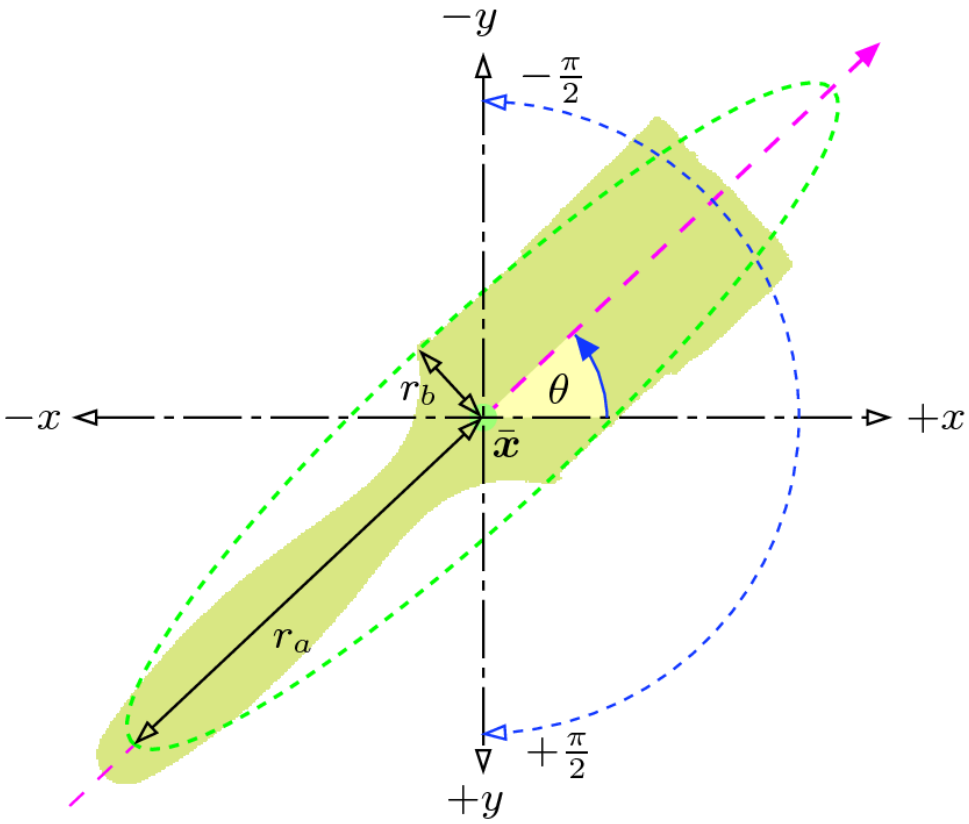
«Circularity» is computed from the perimeter whereas «Roundness» is computed from the major axis

$$Circularity = 4\pi \cdot \frac{Aarea}{Perimeter^2}$$

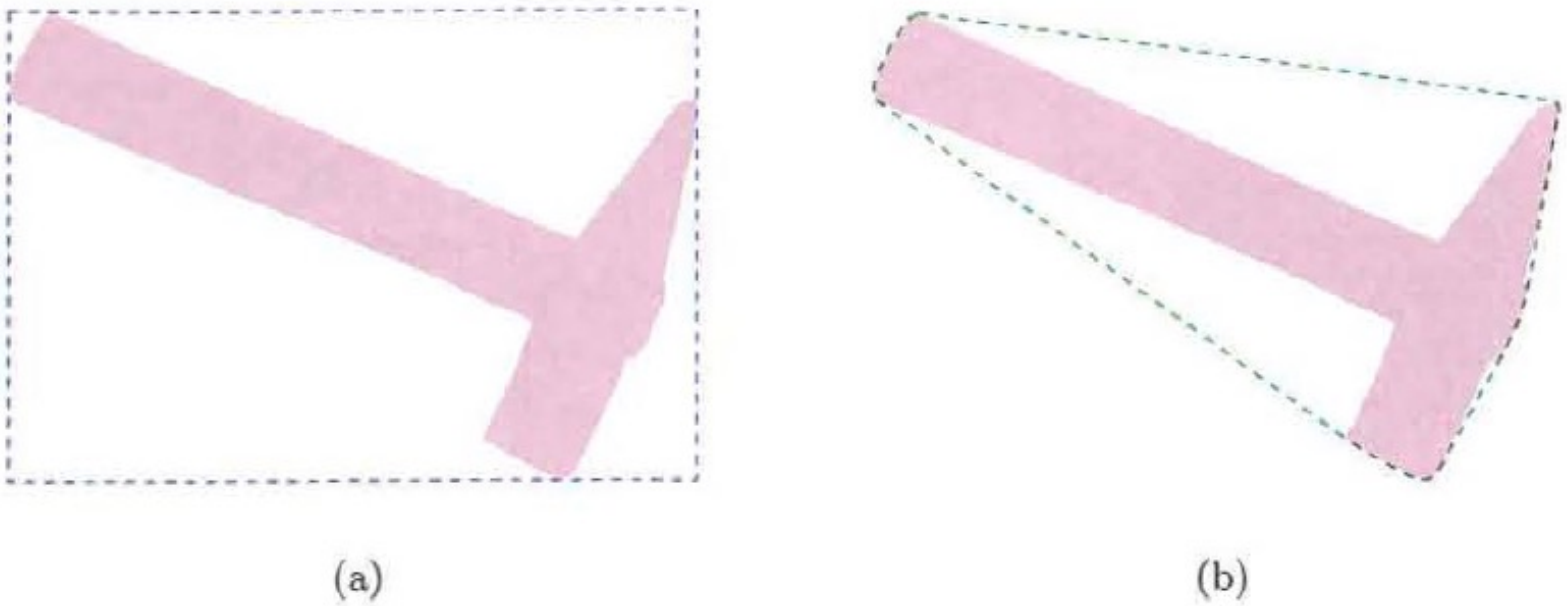
$$Roundness = 4 \cdot \frac{Area}{\pi * \left(Major\_axis\right)^2}$$

Other statistical shape descriptors

- Centroid
- Moments
- Central moments
- Directions
- .....



Bounding box og Convex Hull



(a) (b)

Figure 2.16 Example bounding box (a) and convex hull (b) of a binary image region.

Projections

Figure 2.20 Example of the horizontal projection $P_{\text{hor}}(v)$ (right) and vertical projection $P_{\text{ver}}(u)$ (bottom) of a binary image.

Image moments

Moments summarize a shape given image $I(x, y)$:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$

Central moments are translation invariant:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

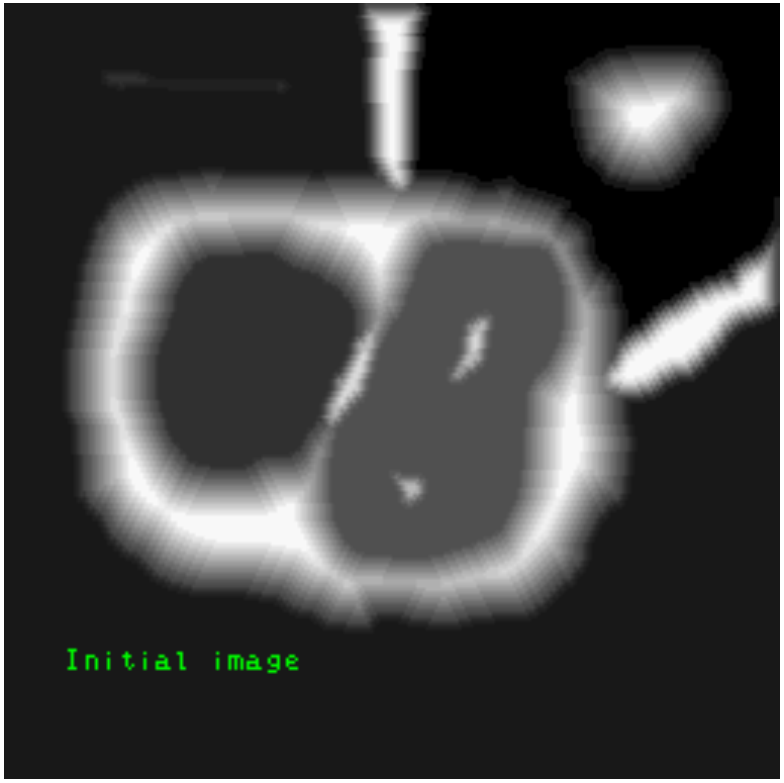
$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \bar{y} = \frac{M_{01}}{M_{00}}$$

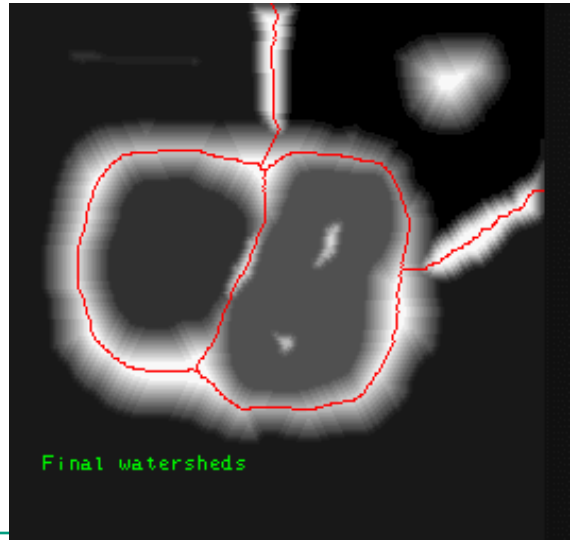
https://www.youtube.com/watch?v=AAbUfZD_09s

https://www.youtube.com/watch?v=uEVrJrJfa0s

Watershed - segmentation

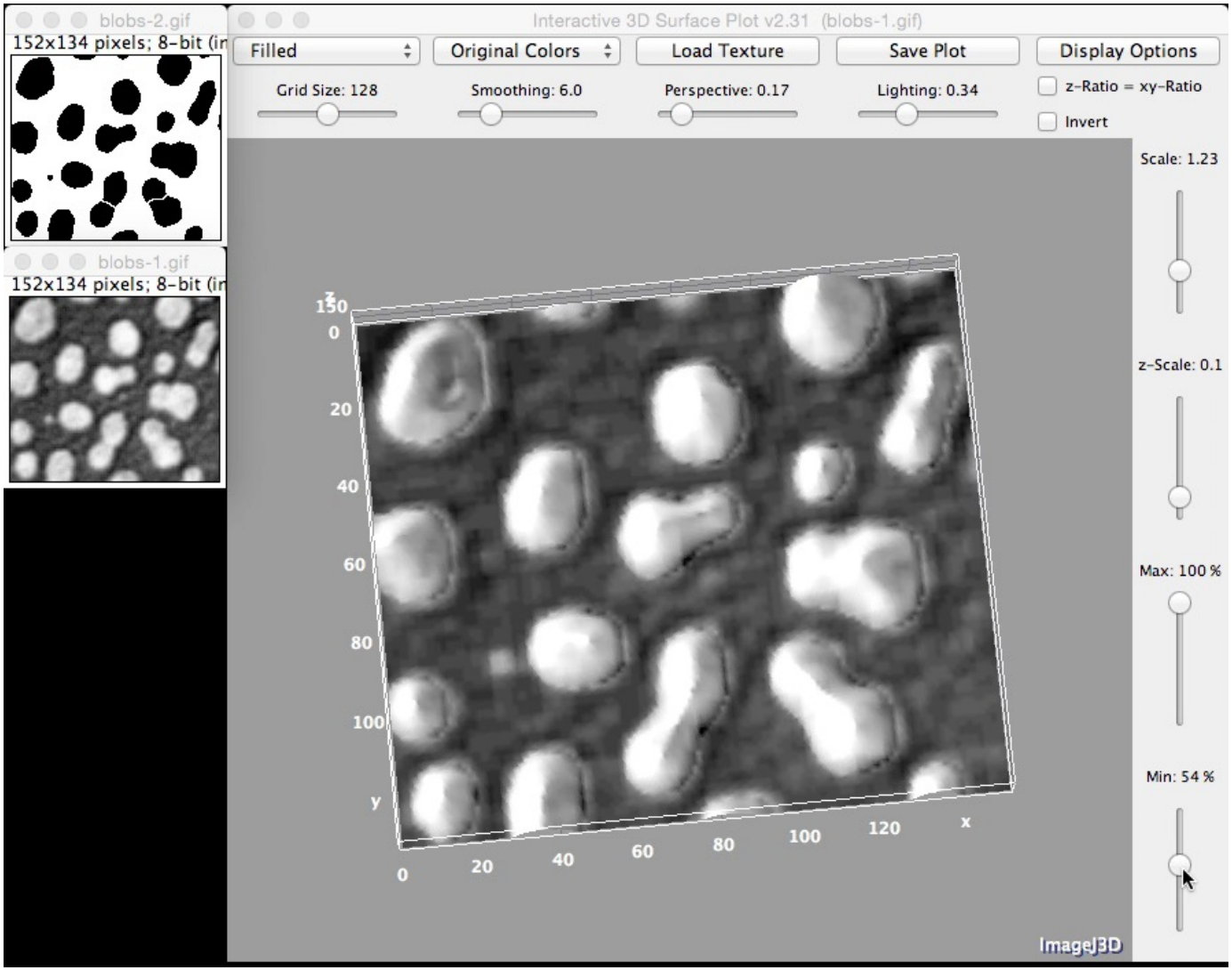
- Grey level images can be viewed as a topologic map
- Binary images can be transformed to a topologic map by using the distance from the perimeter to the center as a weight
- Morphologic operations
  - Fill water from a minimum point and build a shed so that the water doesn't flood inn from other areas
- This gives us two dataset:
  - Water basins
  - Floodlines



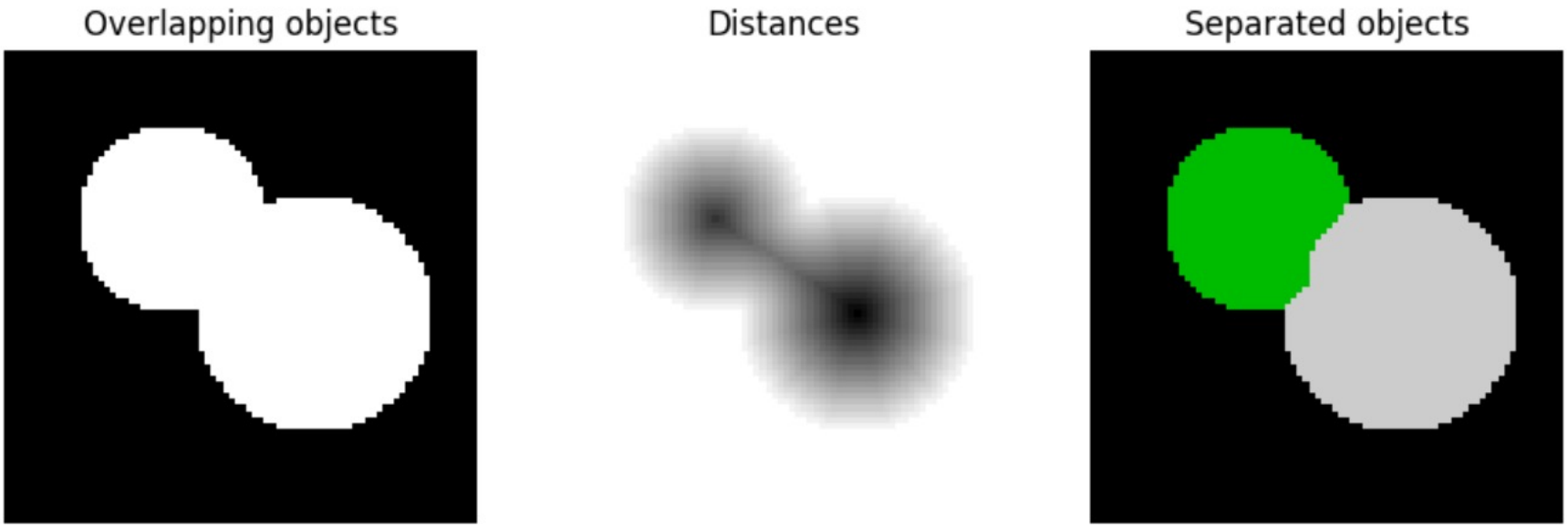


http://cmm.ensmp.fr/~beucher/wtshed.html

Watershed animation

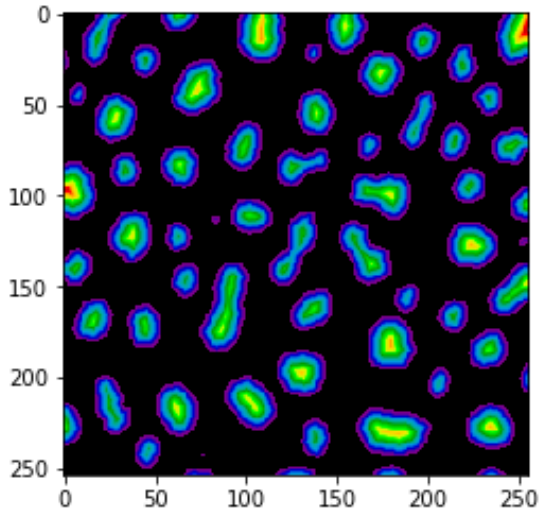


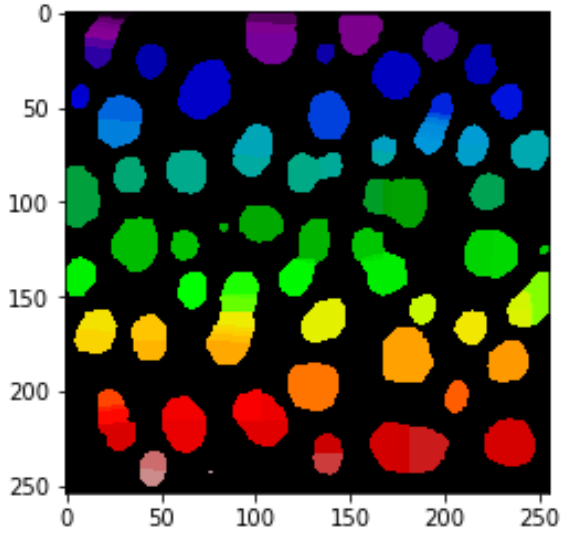
Watershed - Python



http://www.scipy-lectures.org/advanced/image_processing/auto_examples/plot_watershed_segmentation.html

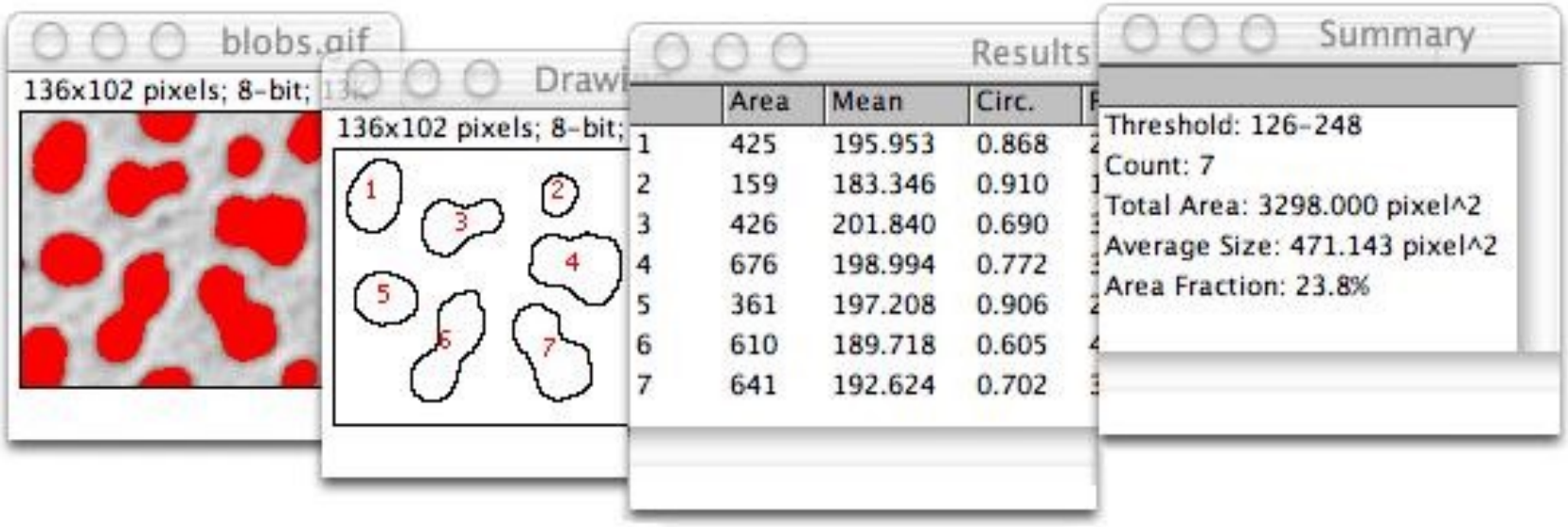
```
from skimage.morphology import watershed
import matplotlib.pyplot as plt
import numpy as np
from skimage.feature import peak_local_max
from scipy import ndimage as ndi
from skimage import io


filename = 'blobs.tif'

blob = io.imread(filename)
plt.imshow(blob,'gray')

distance = ndi.distance_transform_edt(blob)
local_maxi = peak_local_max(distance, indices=False, footprint=np.ones((3, 3)),
                            labels=blob)
markers = ndi.label(local_maxi)[0]
labels = watershed(-distance, markers, mask=blob)

plt.imshow(labels,'spectral')
```

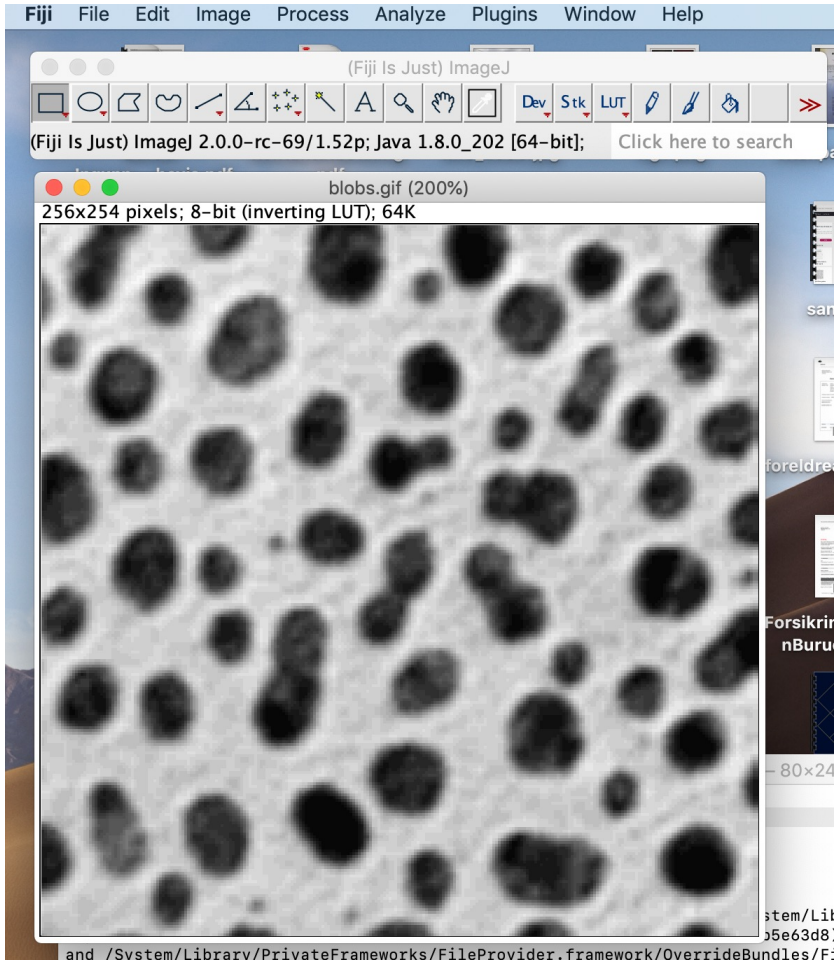


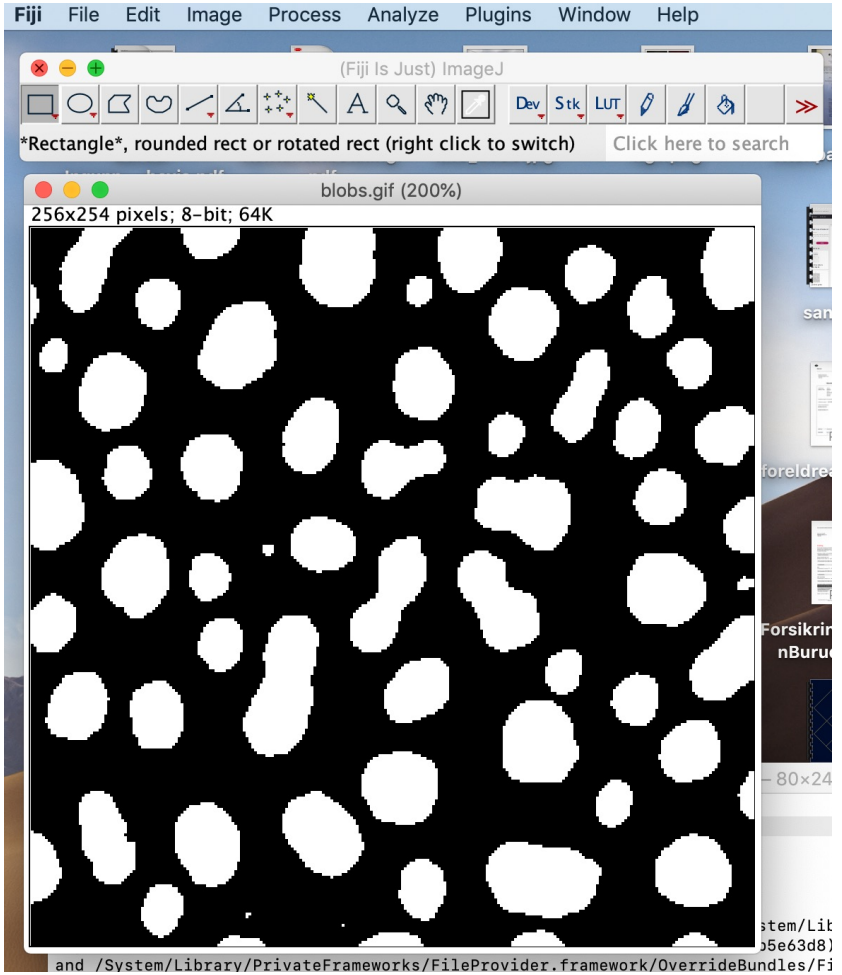


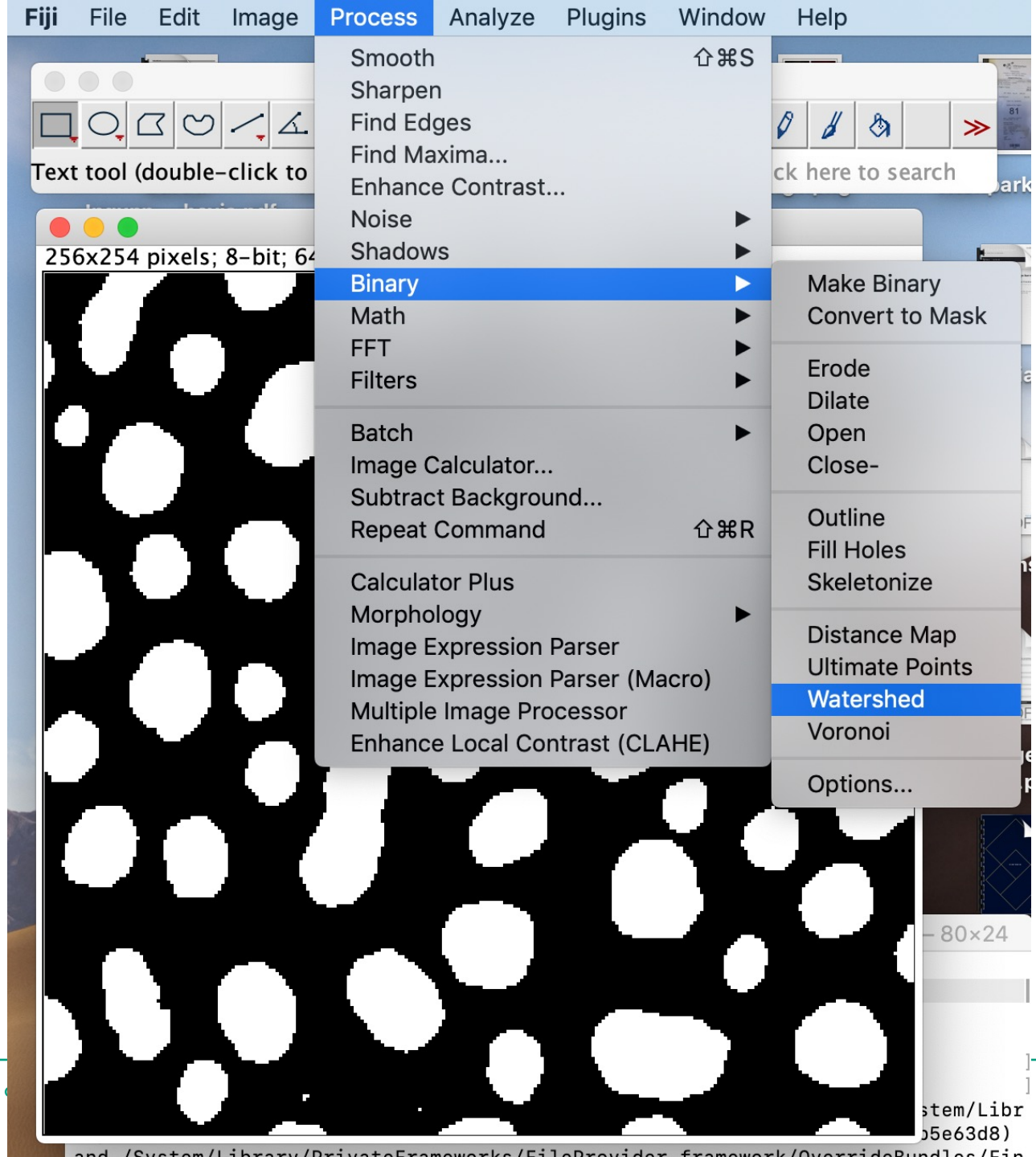
Analyzis of particles

http://imagejdocu.tudor.lu/doku.php?id=gui:analyze:analyze_particles




Fiji File Edit Image Process Analyze Plugins Window Help
(Fiji Is Just) ImageJ
(Fiji Is Just) ImageJ 2.0.0-rc-69/1.52p; Java 1.8.0_202 [64-bit];
Click here to search
blobs.gif (200%)
256x254 pixels; 8-bit (inverting LUT); 64K


Fiji File Edit Image Process Analyze Plugins Window Help
(Fiji Is Just) ImageJ
Rectangle, rounded rect or rotated rect (right click to switch)
Click here to search
blobs.gif (200%)
256x254 pixels; 8-bit; 64K


Fiji File Edit Image Process Analyze Plugins Window Help
Smooth ⇧⌘S
Sharpen
Find Edges
Find Maxima...
Enhance Contrast...
Noise
Shadows
Binary
Math
FFT
Filters
Batch
Image Calculator...
Subtract Background...
Repeat Command ⇧⌘R
Calculator Plus
Morphology
Image Expression Parser
Image Expression Parser (Macro)
Multiple Image Processor
Enhance Local Contrast (CLAHE)
Make Binary
Convert to Mask
Erode
Dilate
Open
Close-
Outline
Fill Holes
Skeletonize
Distance Map
Ultimate Points
Watershed
Voronoi
Options...
Text tool (double-click to
256x254 pixels; 8-bit; 64
80×24

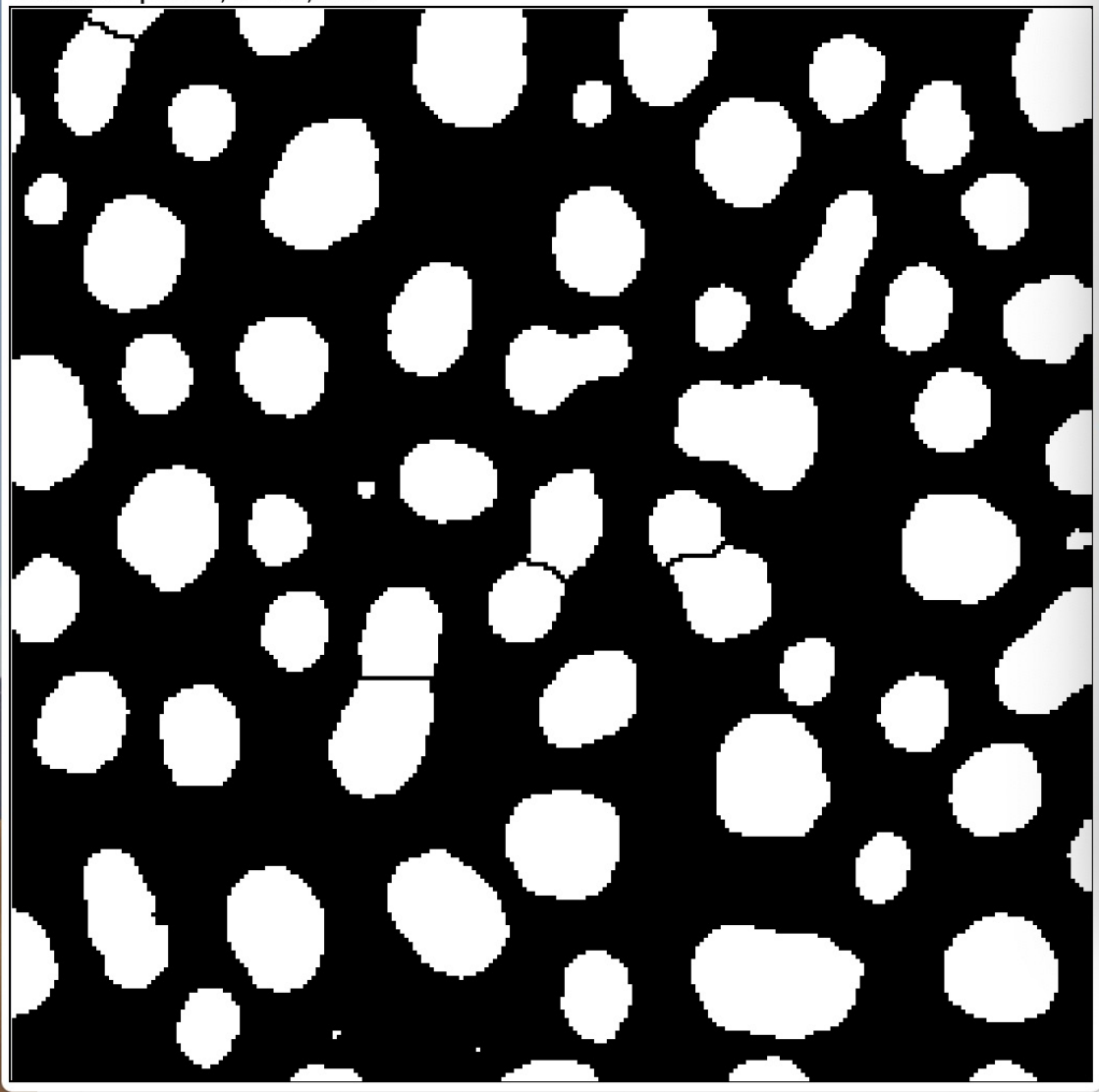
(Fiji Is Just) ImageJ

Text tool (double-click to configure)

Click here to search

blobs.gif (200%)

256x254 pixels; 8-bit; 64K



Set Measurements

- [x] Area
- [ ] Standard deviation
- [x] Min & max gray value
- [ ] Center of mass
- [ ] Bounding rectangle
- [x] Shape descriptors
- [ ] Integrated density
- [ ] Skewness
- [ ] Area fraction
- [x] Mean gray value
- [ ] Modal gray value
- [ ] Centroid
- [ ] Perimeter
- [ ] Fit ellipse
- [ ] Feret's diameter
- [ ] Median
- [ ] Kurtosis
- [ ] Stack position

- [ ] Limit to threshold
- [ ] Invert Y coordinates
- [ ] Add to overlay
- [x] Display label
- [ ] Scientific notation
- [ ] NaN empty cells

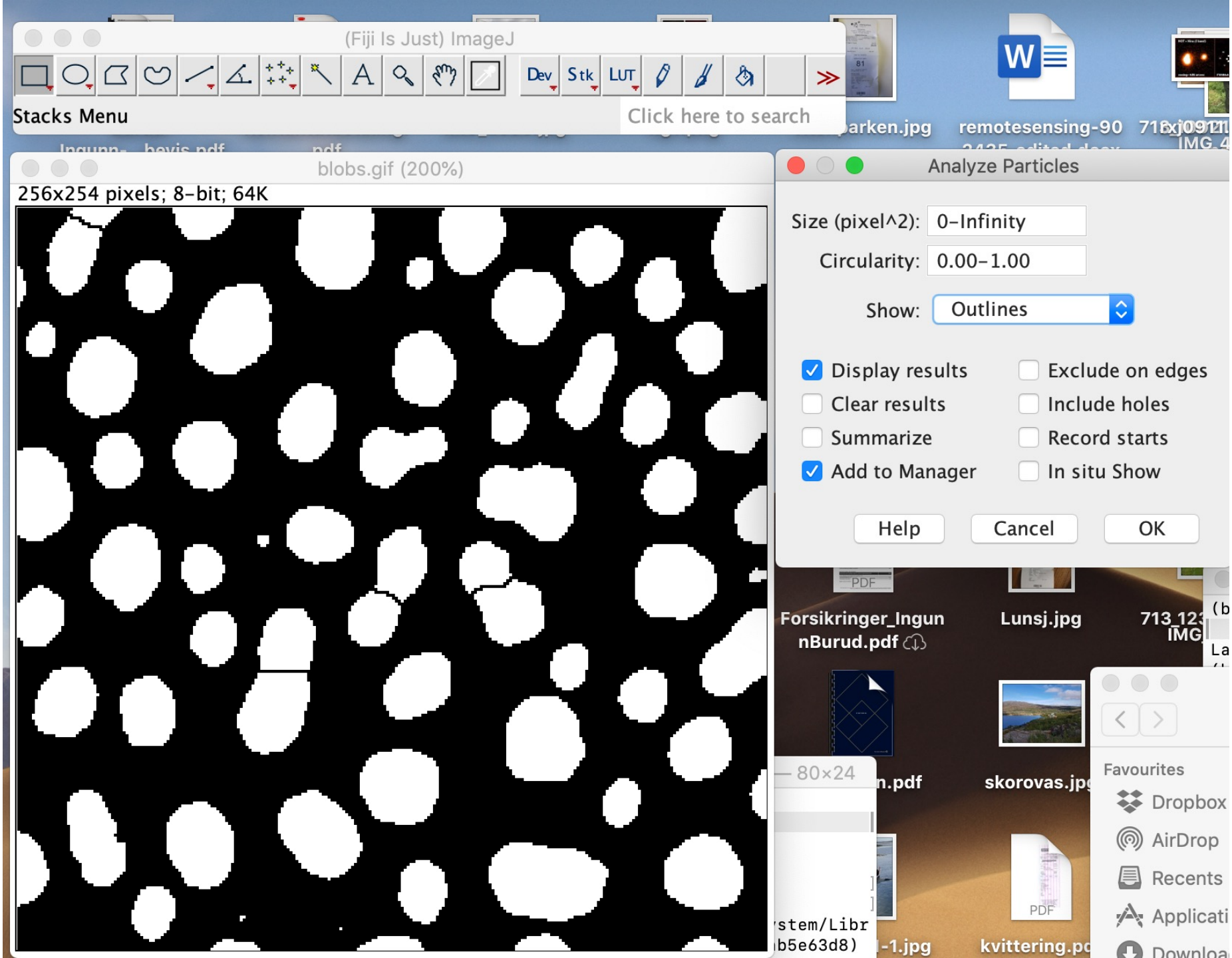
Redirect to: None

Decimal places (0-9): 3

Help Cancel OK

AirDrop
Recents
Applications
Downloads

and /System/Library/PrivateFrameworks/FileProvider.framework/OverrideBundles/Fin


(Fiji Is Just) ImageJ
Dev
Stk
LUT
Stacks Menu
Click here to search
blobs.gif (200%)
256x254 pixels; 8-bit; 64K
Analyze Particles
Size (pixel^2): 0-Infinity
Circularity: 0.00-1.00
Show: Outlines
Display results
Clear results
Summarize
Add to Manager
Exclude on edges
Include holes
Record starts
In situ Show
Help
Cancel
OK
remotesesning-90
Forsikringer_IngunnBurud.pdf
Lunsj.jpg
skorovas.jpg
kvittering.pdf
Favourites
Dropbox
AirDrop
Recents

(Fiji Is Just) ImageJ

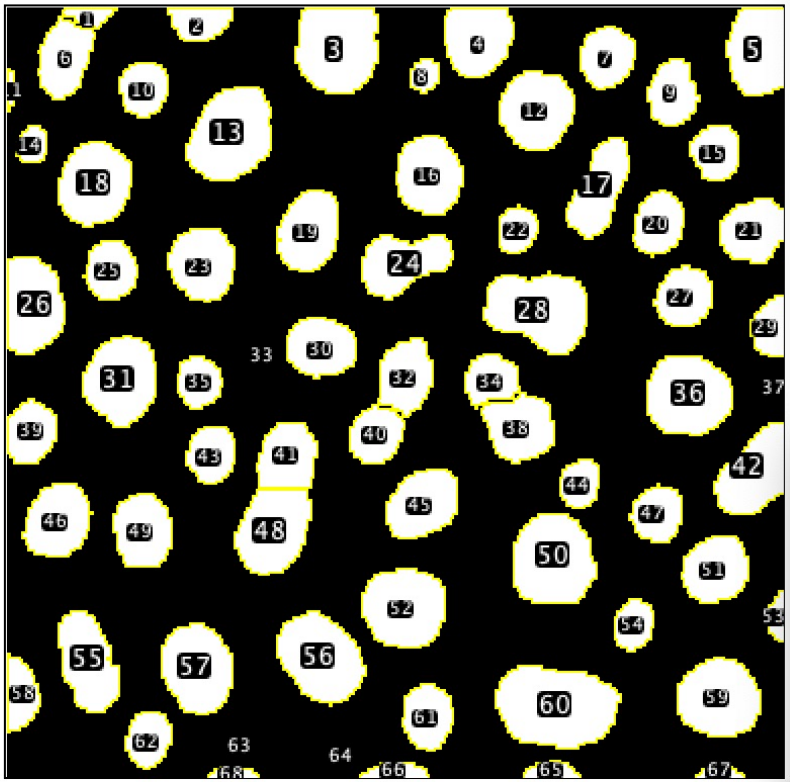
Dev Stk LUT

(Fiji Is Just) ImageJ 2.0.0-rc-69/1.52p; Java 1.8.0_202 [64-bit];

Click here to search

blobs.gif (150%)

256x254 pixels; 8-bit; 64K

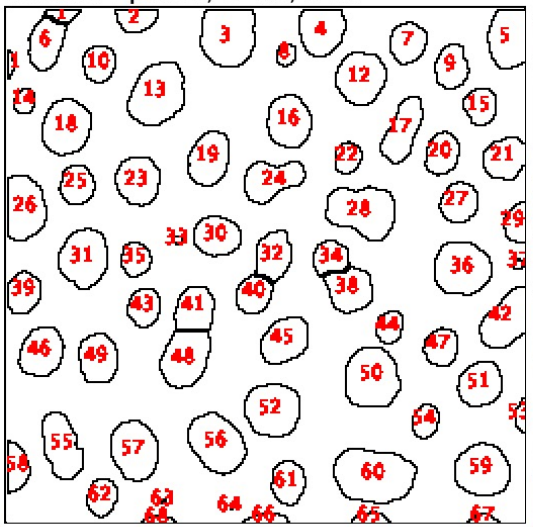


Results

	Area	Mean	Min	Max	Circ.	AR	Round	Solidity
1	76	255	255	255	0.562	2.522	0.396	0.884
2	185	255	255	255	0.763	1.778	0.562	0.951
3	658	255	255	255	0.872	1.068	0.936	0.967
4	434	255	255	255	0.884	1.064	0.940	0.959
5	477	255	255	255	0.821	1.570	0.637	0.968
6	341	255	255	255	0.757	1.603	0.624	0.905
7	285	255	255	255	0.926	1.153	0.867	0.938
8	81	255	255	255	0.971	1.204	0.830	0.926
9	278	255	255	255	0.854	1.389	0.720	0.919
10	231	255	255	255	0.936	1.141	0.877	0.941
11	30	255	255	255	0.406	4.338	0.230	0.870

Drawing of blobs.gif

256x254 pixels; 8-bit; 64K



ROI Manager

0001-0003
0002-0005
0003-0014
0004-0011
0005-0014
0006-0016
0007-0016
0008-0022
0009-0028
0010-0027
0011-0027
0012-0034
0013-0041
0014-0045
0015-0048
0016-0055
0017-0059
0018-0058

Add [t]
Update
Delete
Rename...
Measure
Deselect
Properties...
Flatten [F]
More »
Show All
Labels

timeliste_signed.pdf

iCloud Drive

Python - measure

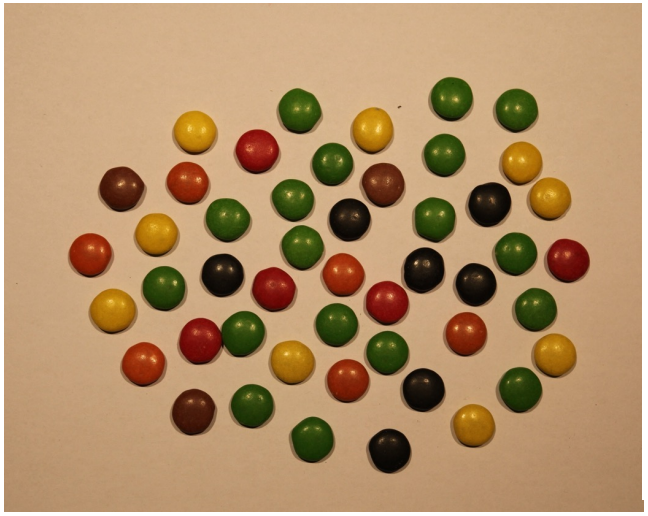
```
from skimage import measure
properties = measure.regionprops(labels)
for prop in properties:
    print(prop.perimeter, prop.area, prop.centroid, prop.eccentricity)
```

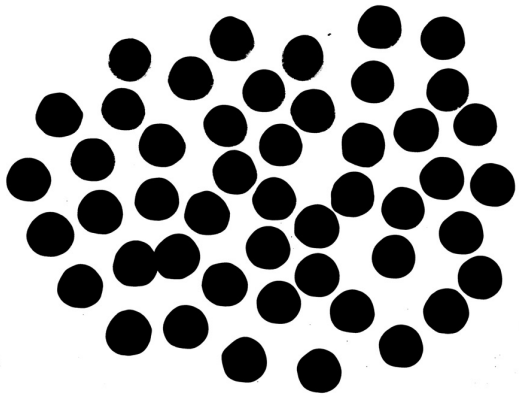
http://scikit-image.org/docs/dev/api/skimage.measure.html#skimage.measure.regionprops

Watershed - Imagej

- Watershed on gray level images
  - http://bigwww.epfl.ch/sage/soft/watershed/

Masks





Making a binary
(holes=0, background=1)

Masking original image by
combining original OR mask