Digital image processing and analysis

11. Object properties: counting, measuring and localisation

Professor Ela Claridge
School of Computer Science

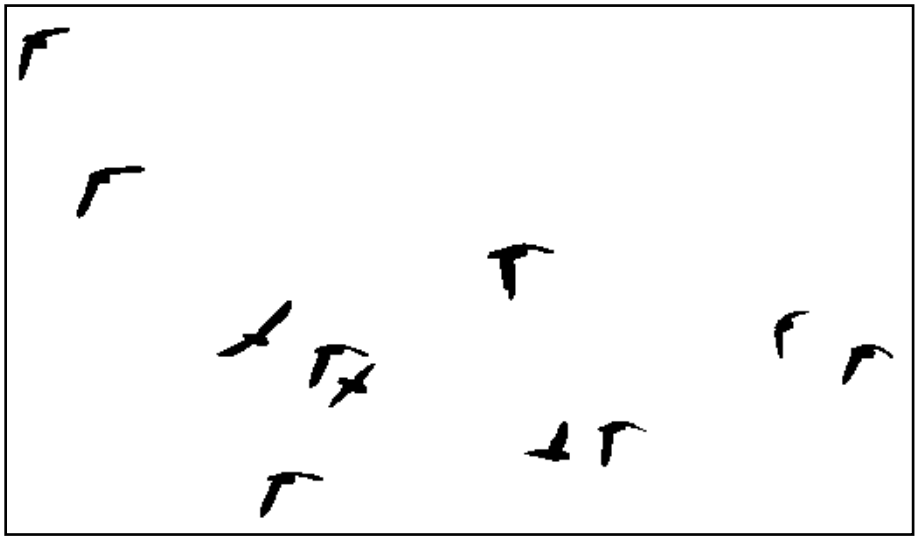
Previous lecture:

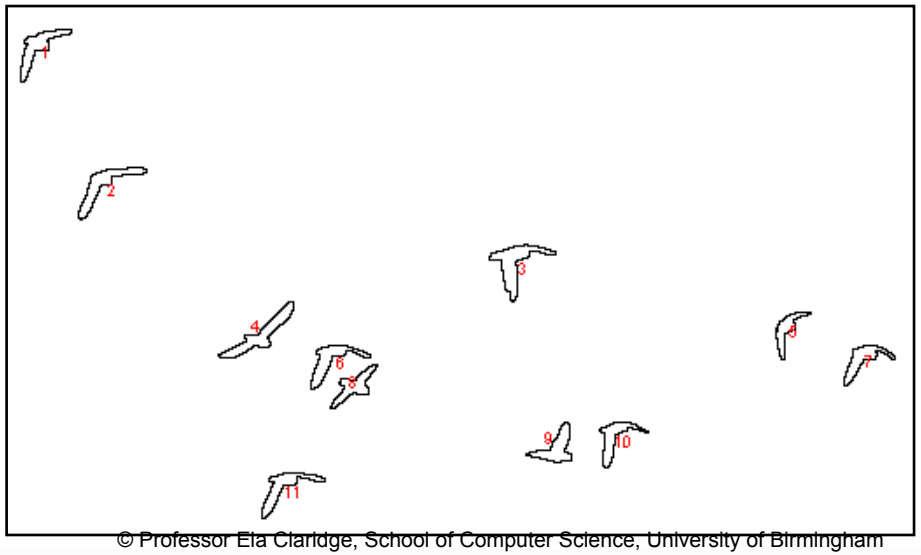
- Post-processing methods
  - Mathematical morphology – basic operations
- Feature extraction
  - Mathematical morphology – combined operations

In this lecture we shall find out about:

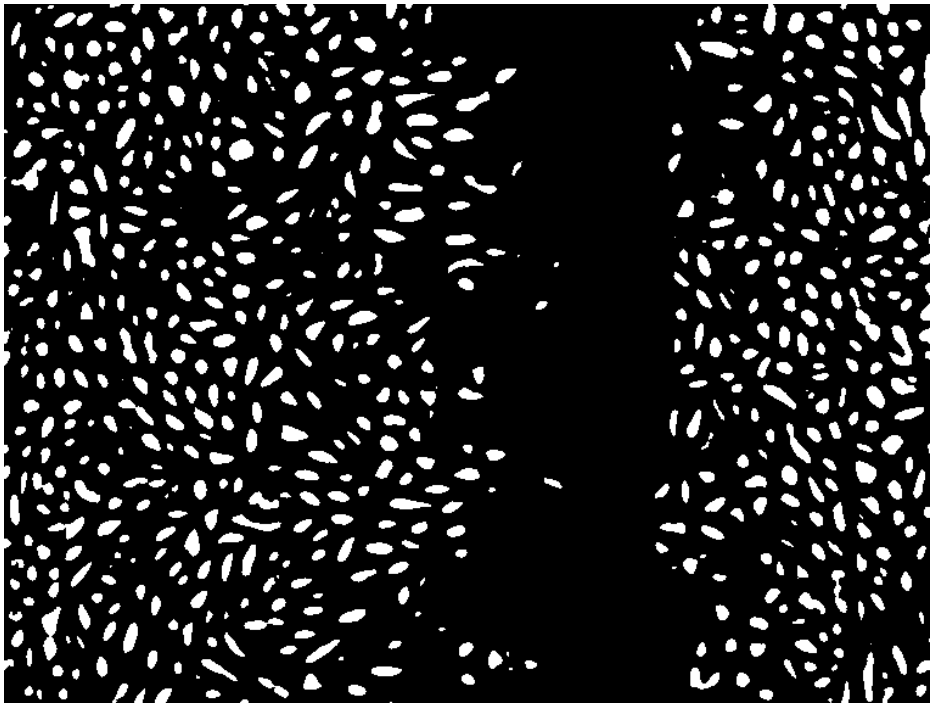
- How to get coordinates of the object boundaries
  - Hand-on-the-wall walk around the object
- How to count objects in a segmented image
- How to measure objects
  - Area
  - Boundary / Perimeter
- How to measure object locations
  - Bounding box
  - Centroid
- How to measure some object properties
  - Fitted ellipse
  - Compactness
  - Concavity

Segmentation
A starting point to object measurement

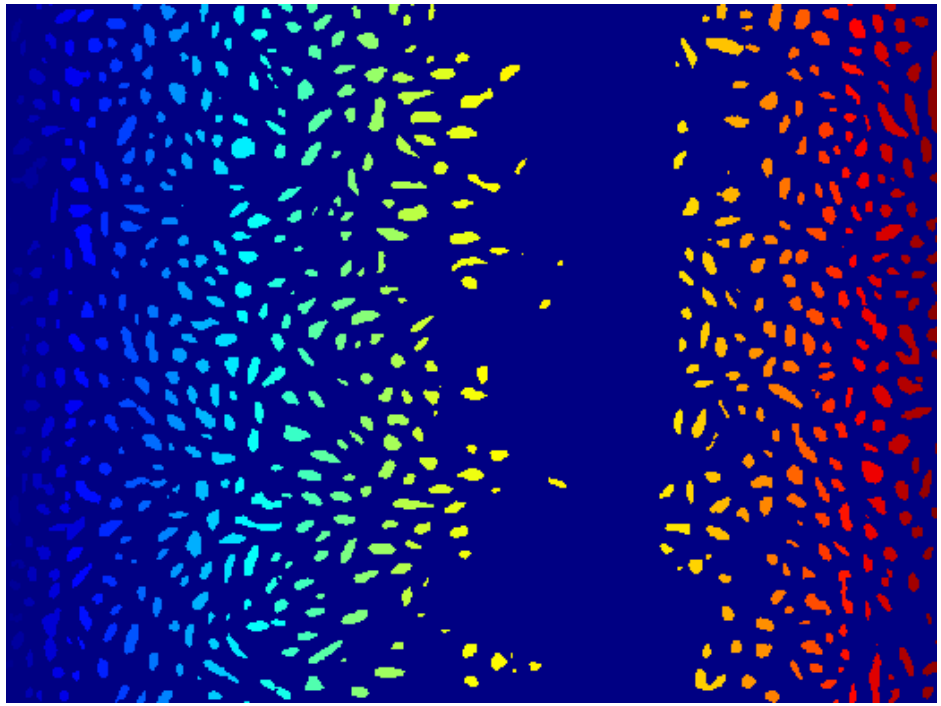




Segmentation
A starting point to object measurement

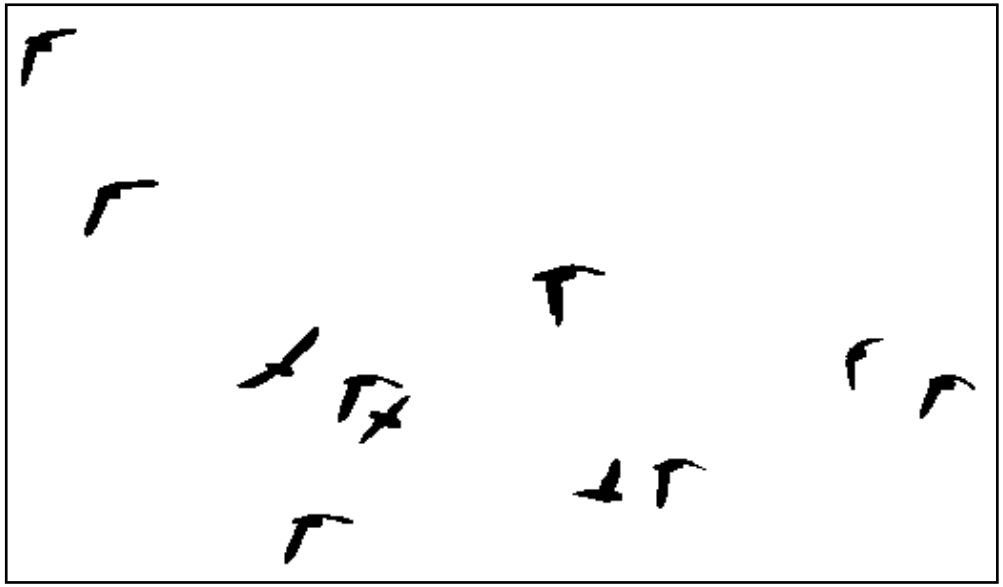


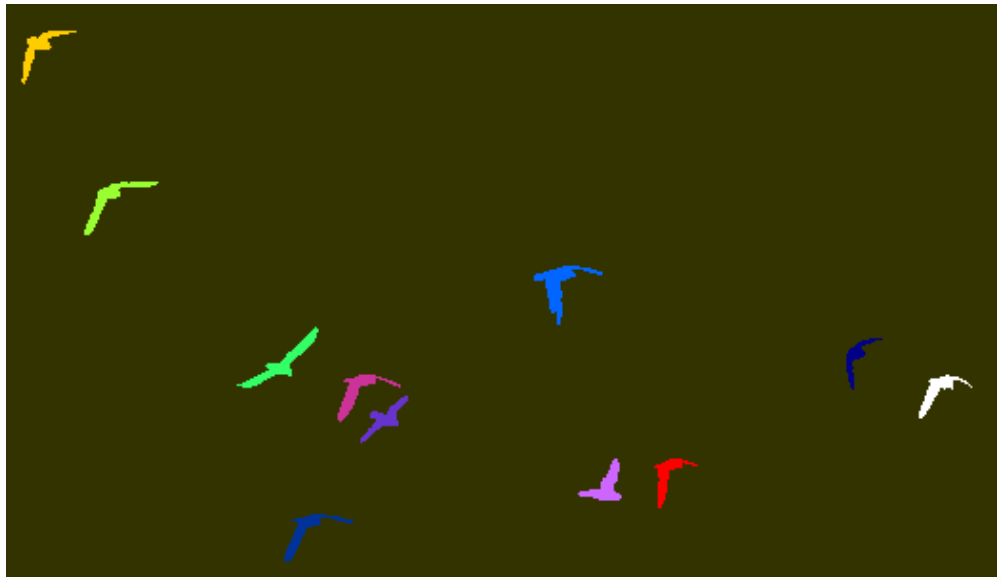
Segmented image



Labeled image
+ cell count

Segmentation
A starting point to object measurement



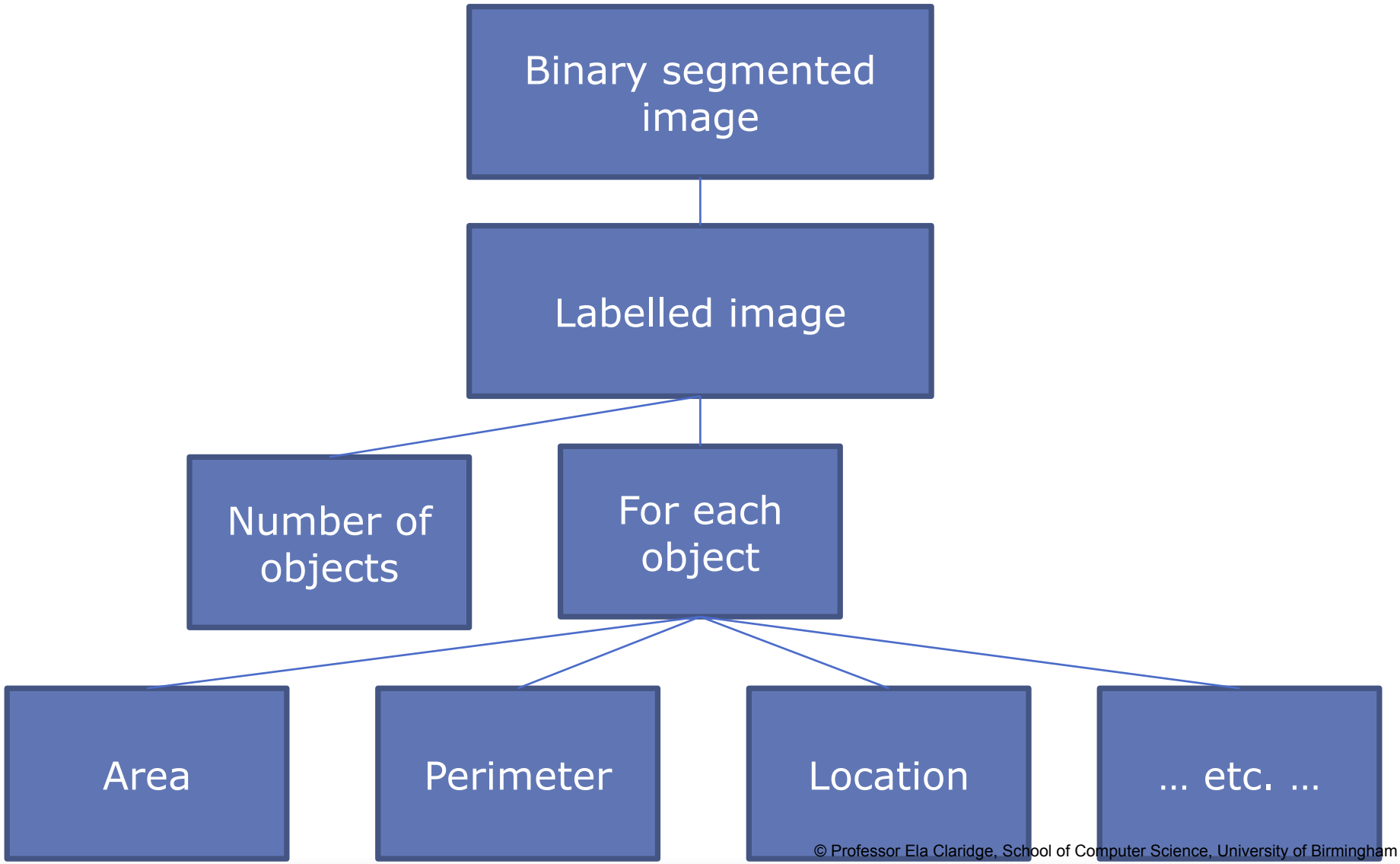


Representations for a segmented image

Binary image:
pixel value = 0 for the background
pixel value = 1 for (any) object

Label image:
pixel value = 0 for the background
pixel value = object number
(each colour represents different number)

Segmentation
A starting point to object measurement



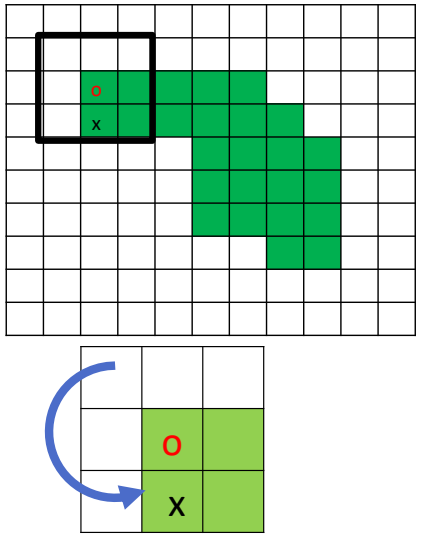
Object boundary extraction
Region outlining

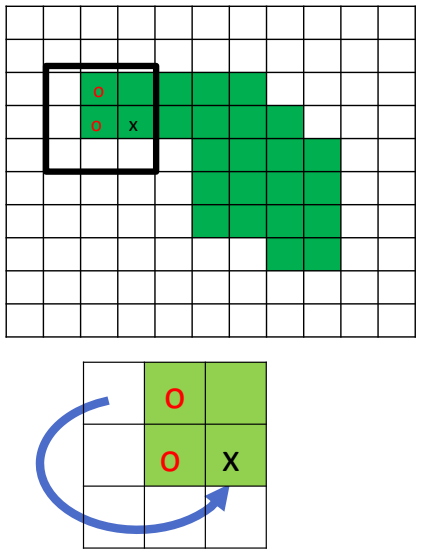
- Used to obtain information about the edge of a region extracted by one of the region segmentation techniques (e.g. thresholding)
- No *a priori* knowledge is required about a shape.
- Useful as a pre-processing step for shape description.

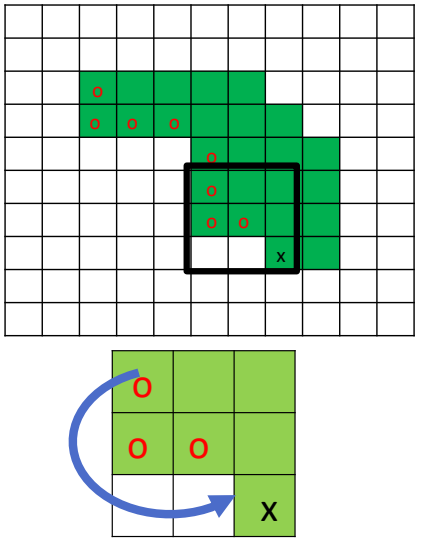
Object boundary extraction
Region outlining

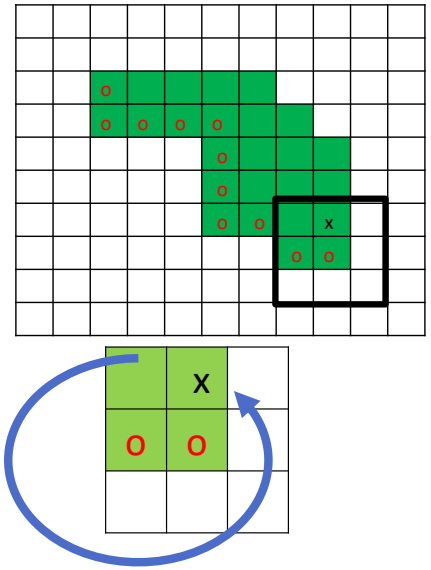
- One of the simplest techniques is the "hand on the wall" approach
- Produces not only an image with the pixels showing the outline but also a list of (x,y) coordinates of the outline.

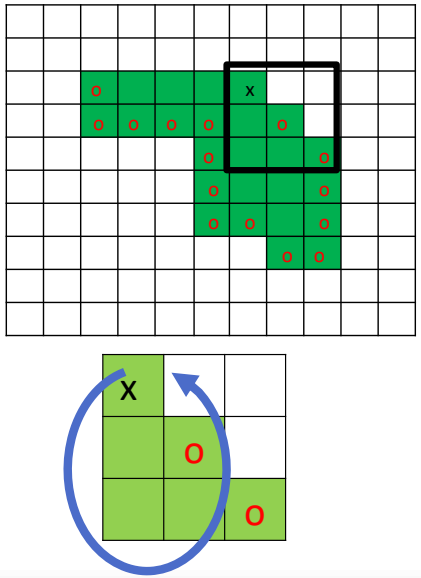
Object boundary extraction
Region outlining

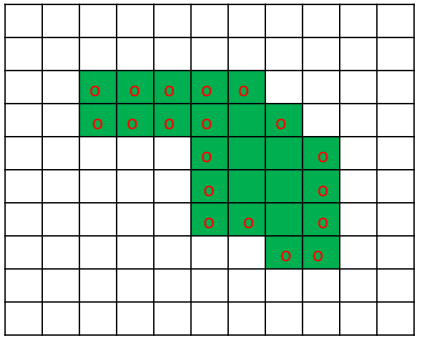


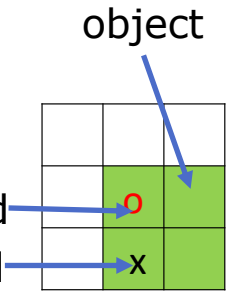










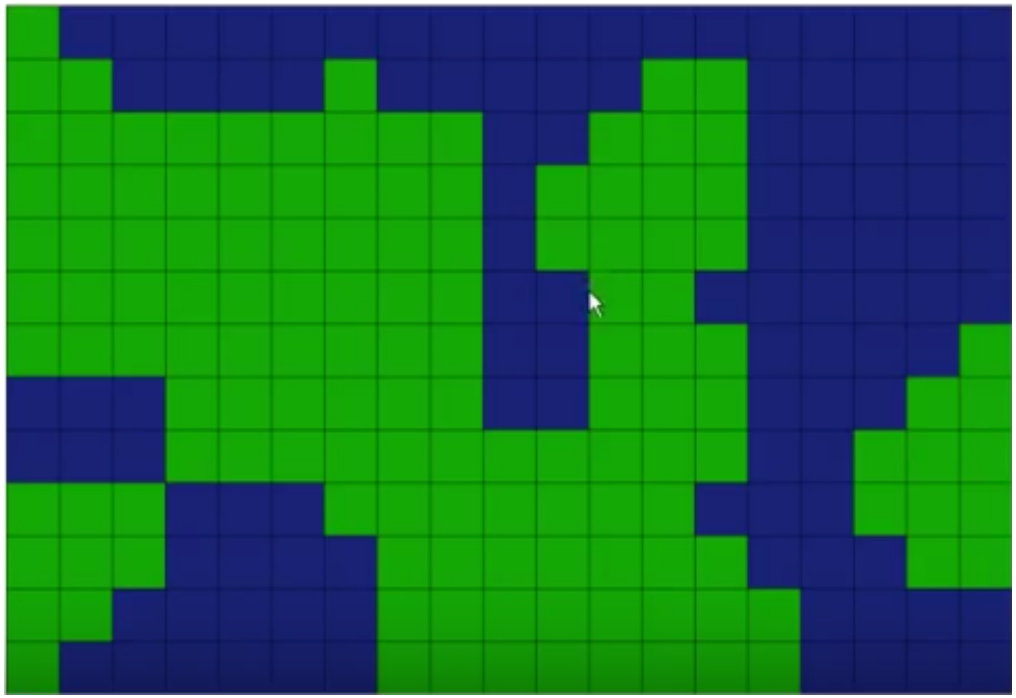


Object boundary extraction
Region outlining

Algorithm (recipe) for the Hand-on-the-wall region outlining

1. Scan image pixels from left to right / top down until the first object pixel is found (seed).
2. Add its coordinates to the list.
3. Consider a 3x3 region of interest (ROI) centred on the seed pixel
4. Starting from the top-left corner of the ROI traverse pixels around the seed pixel in anti-clock-wise direction until the first transition from background pixel to object pixel is found.
5. Make it a new seed.
6. Repeat steps from 2 to 5 until the new seed coordinate equals the first seed coordinate. The outline of this object is now complete.
7. Search for the new object, repeat steps 1-7 until the bottom right corner of the image is reached.

Row-by-row labelling

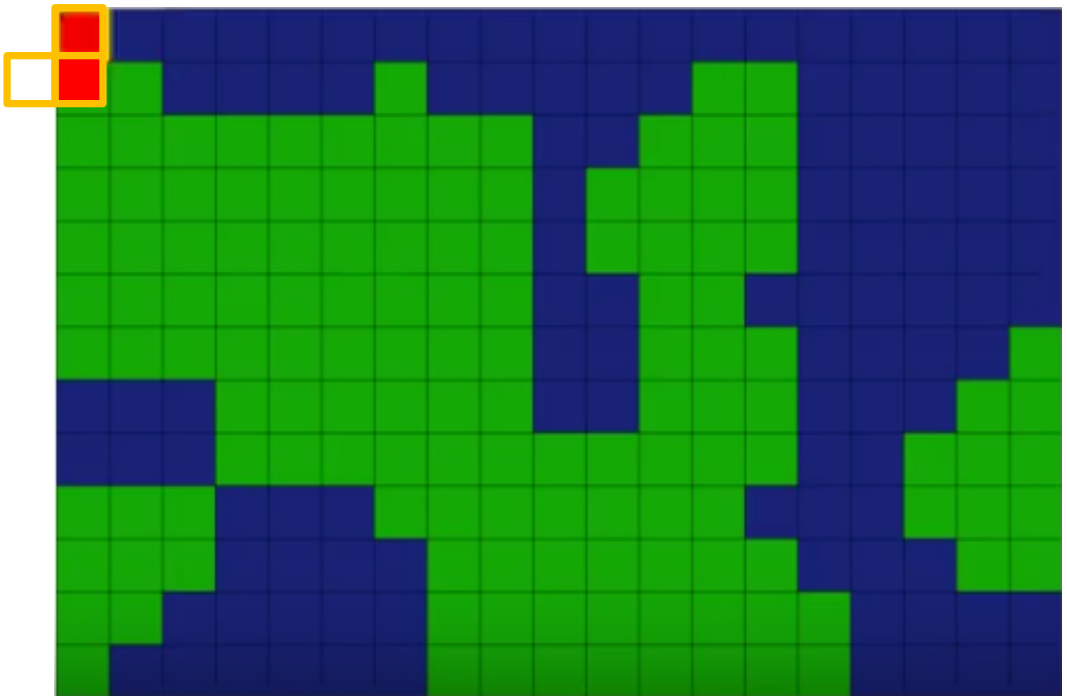


U-L connector
(”up-left)

Segmented image
Green: Object
Blue: background

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

Row-by-row labelling

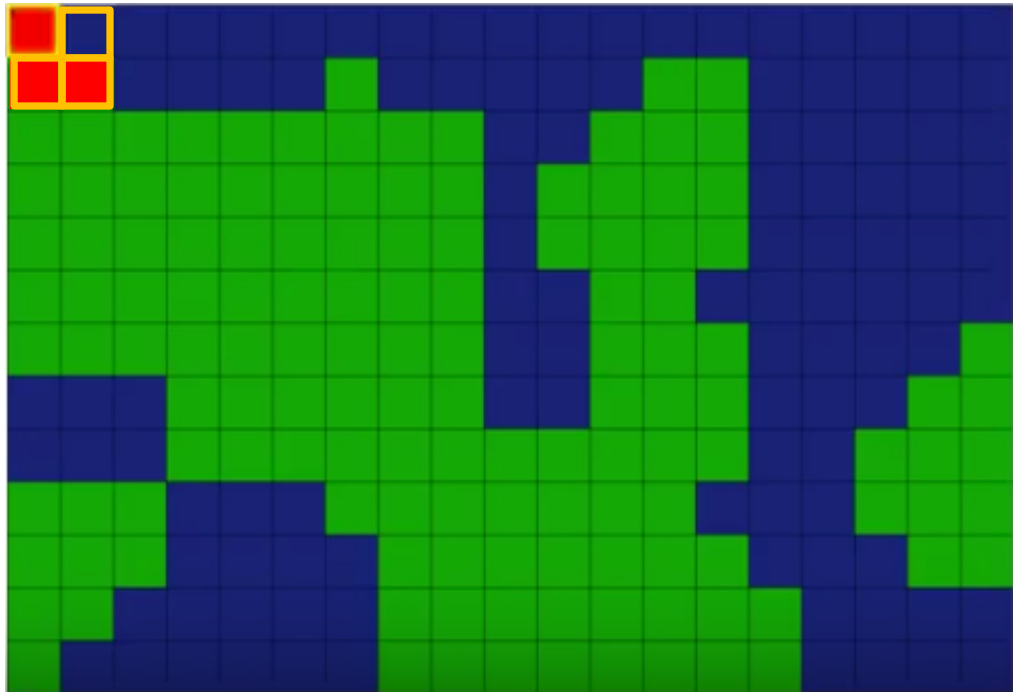


Region ID	Connections
1	

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number ✓
  (the first one will be 1)
- Update connections

Row-by-row labelling

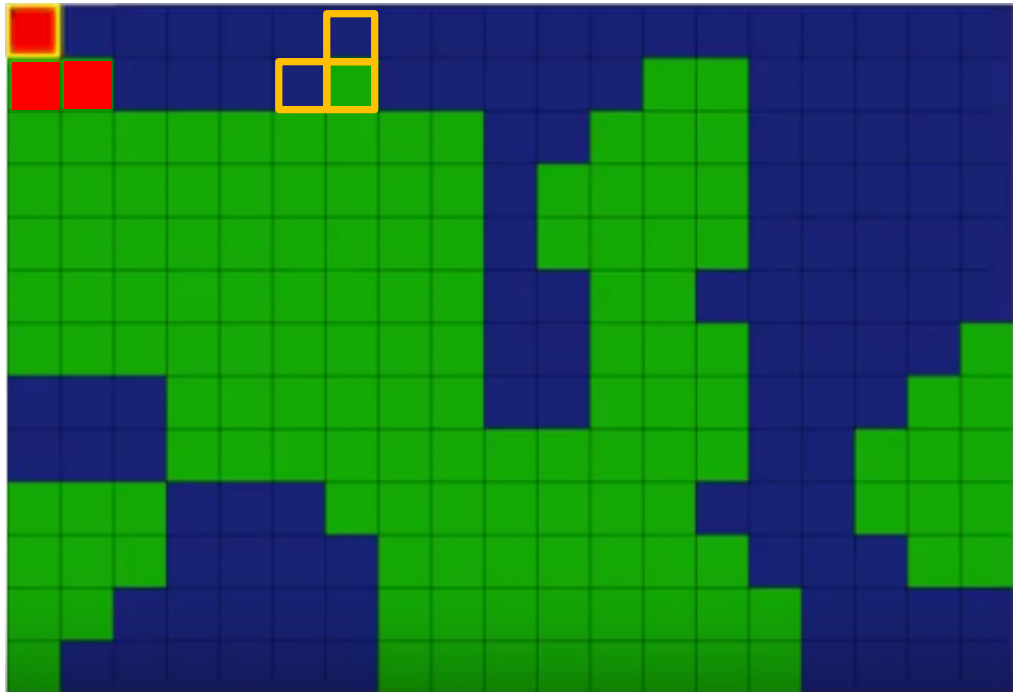


Region ID	Connections
1	1

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number
- If connected U-L, assign the smaller label chosen out of the labels covered by the U-L connector ✓

Row-by-row labelling

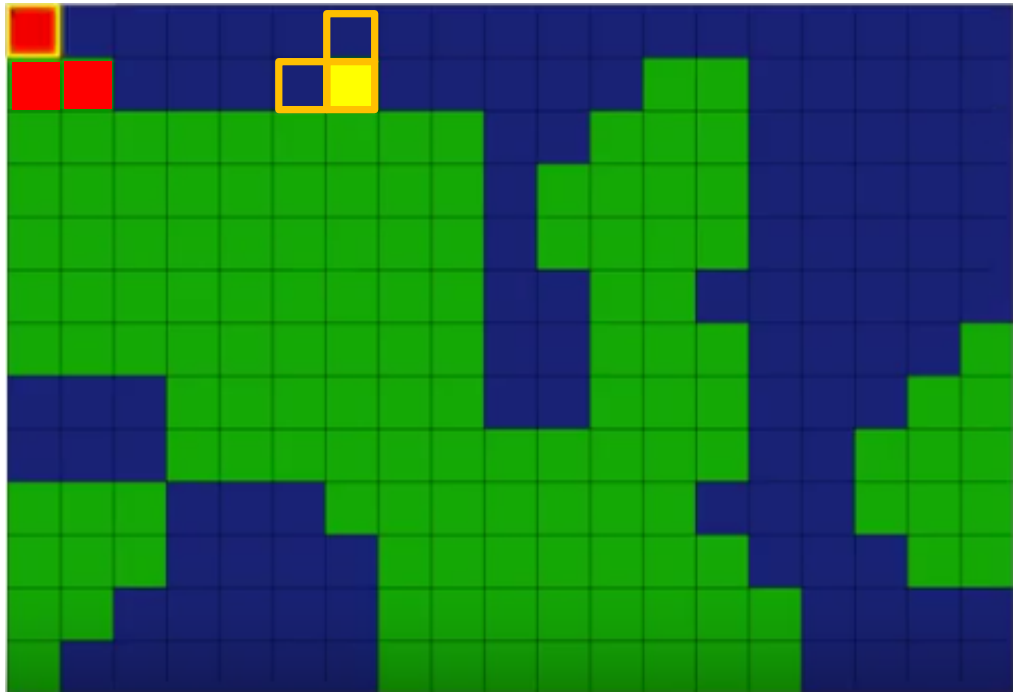


Region ID	Connections
1	1

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number ✓
- If connected U-L, assign the smaller label
- Update connections

Row-by-row labelling

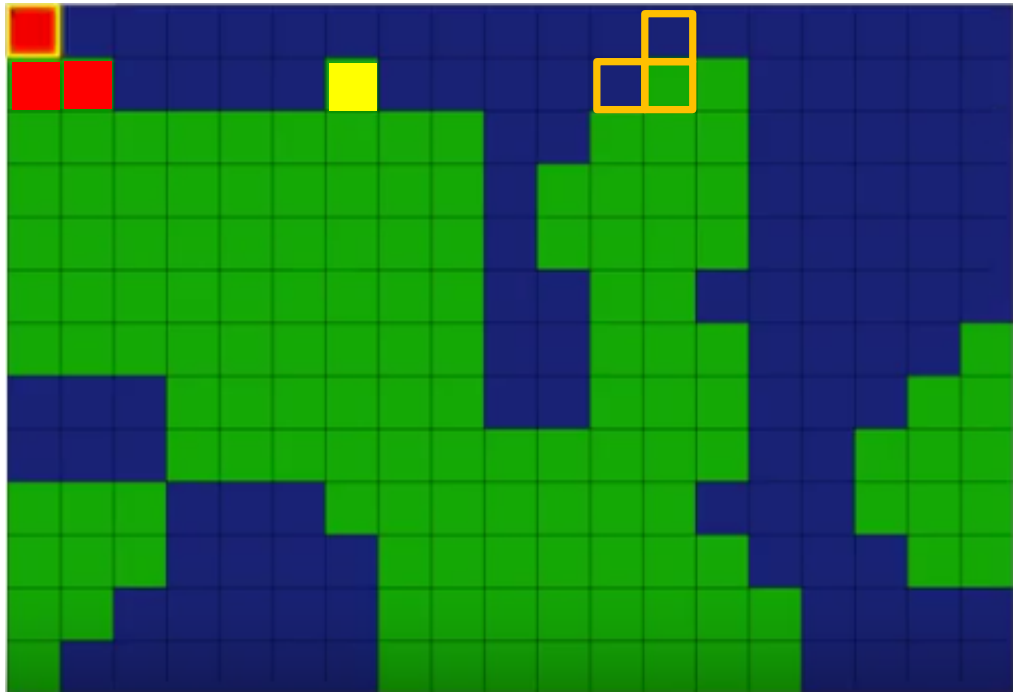


Region ID	Connections
1	1
2	2

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number ✓
- If connected U-L, assign the smaller label
- Update connections

Row-by-row labelling

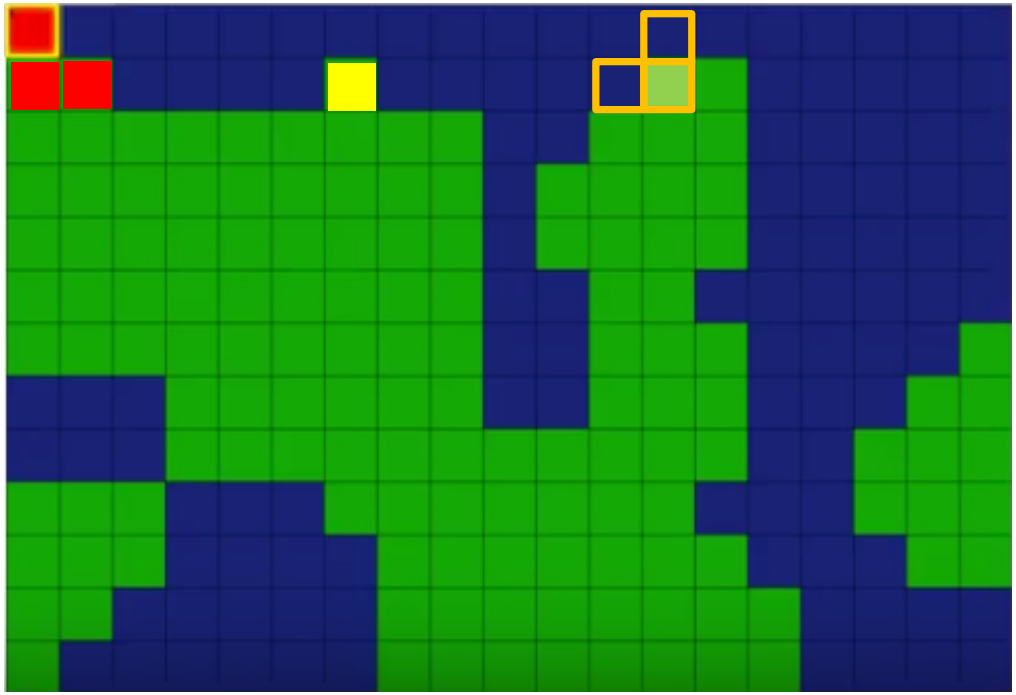


Region ID	Connections
1	1
2	2

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number ✓
- If connected U-L, assign the smaller label
- Update connections

Row-by-row labelling

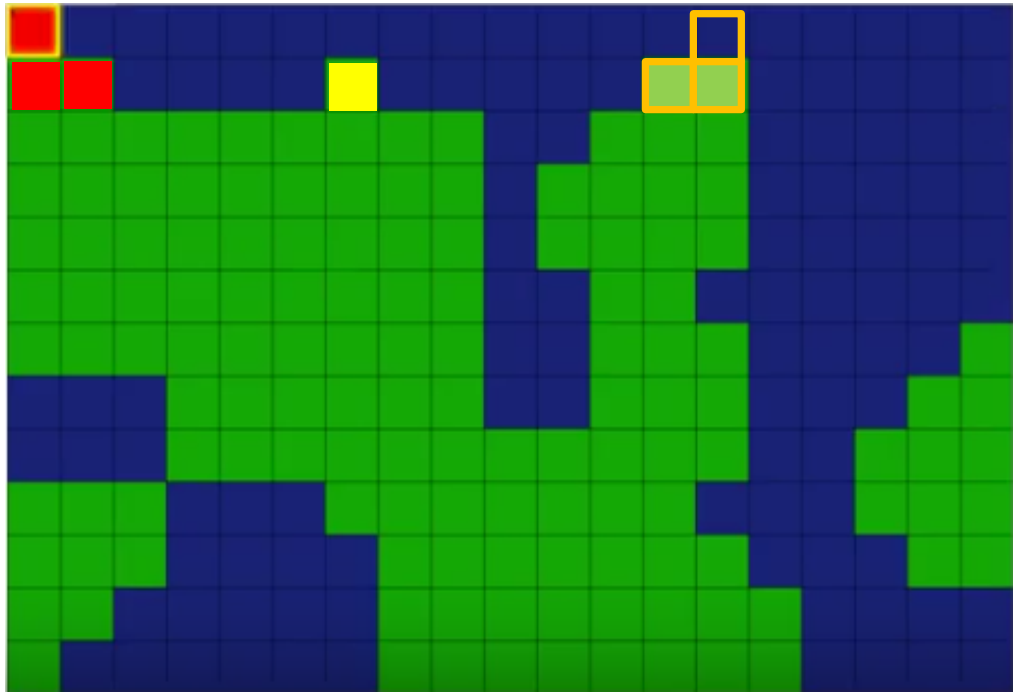


Region ID	Connections
1	1
2	2
3	3

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number ✓
- If connected U-L, assign the smaller label
- Update connections

Row-by-row labelling

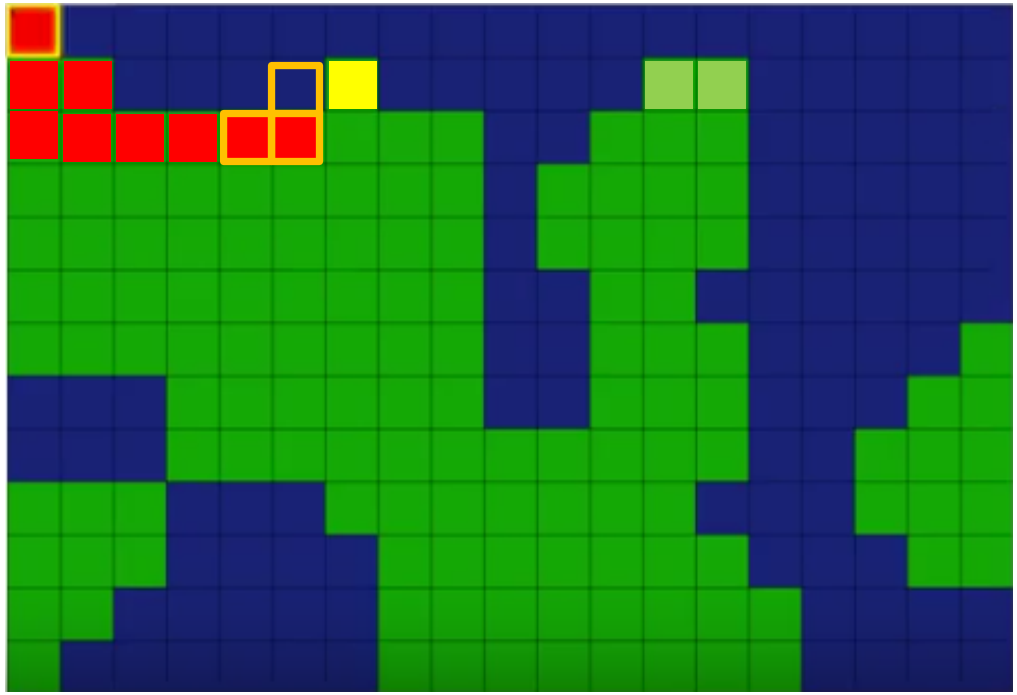


Region ID	Connections
1	1
2	2
3	3

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number ✓
- If connected U-L, assign the smaller label
- Update connections

Row-by-row labelling

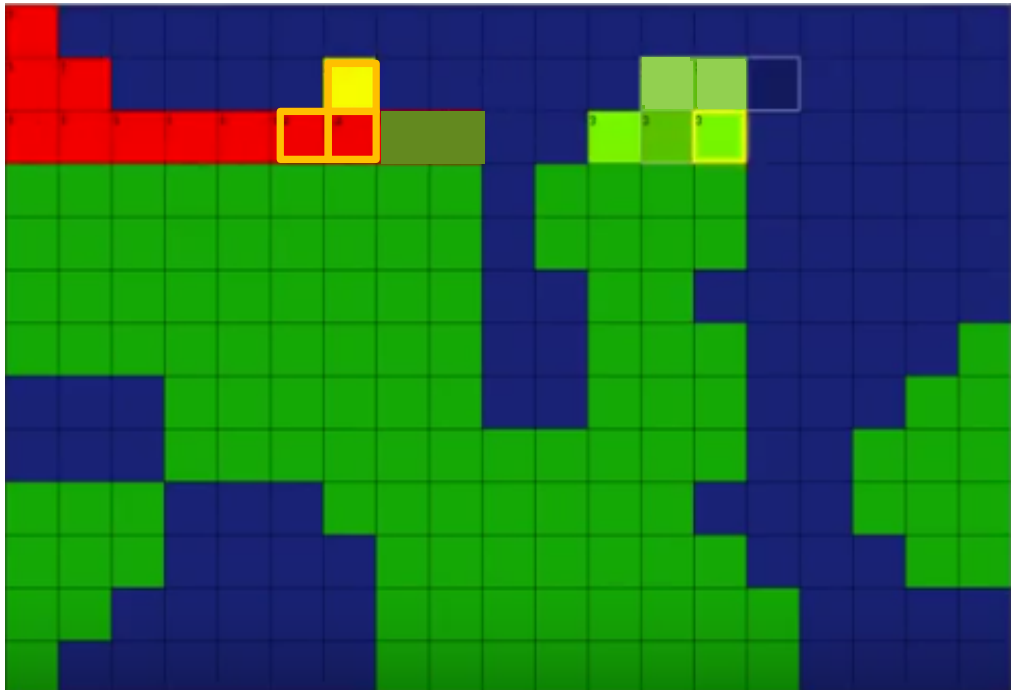


Region ID	Connections
1	1
2	2
3	3

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number
- If connected U-L, assign the smaller label ✓
- Update connections

Row-by-row labelling

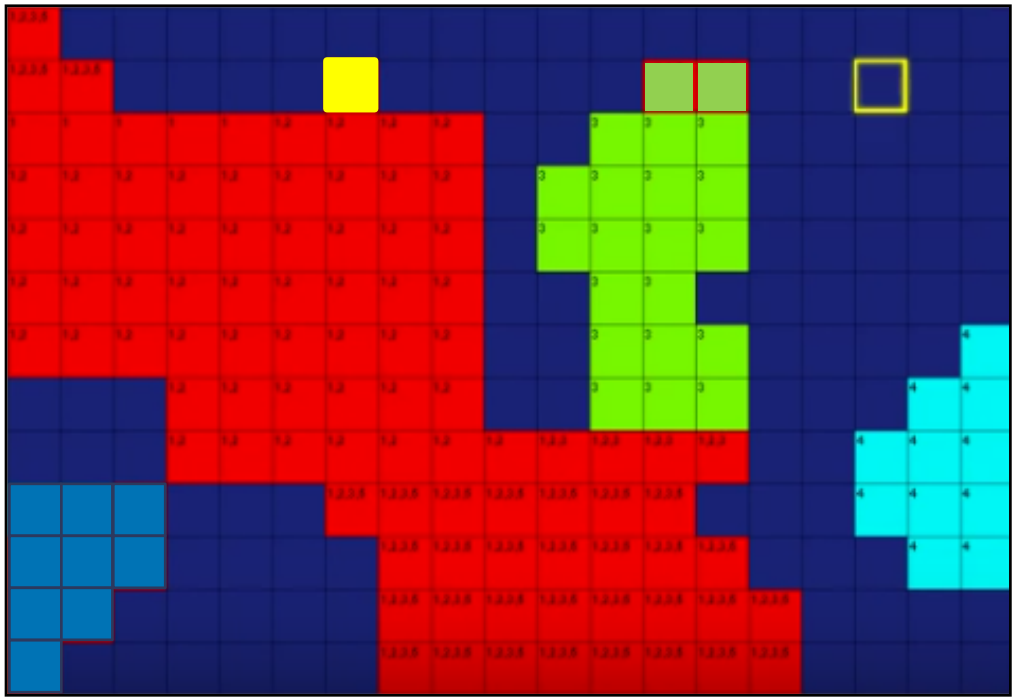


Region ID	Connections
1	1,2
2	1,2
3	3

Step 1 (first pass)

- Scan left-to-right then top-to bottom
- Stop when an unlabelled pixel found
- If unconnected U-L, increment label number
- If connected U-L, assign the smaller label ✓
- Update connections

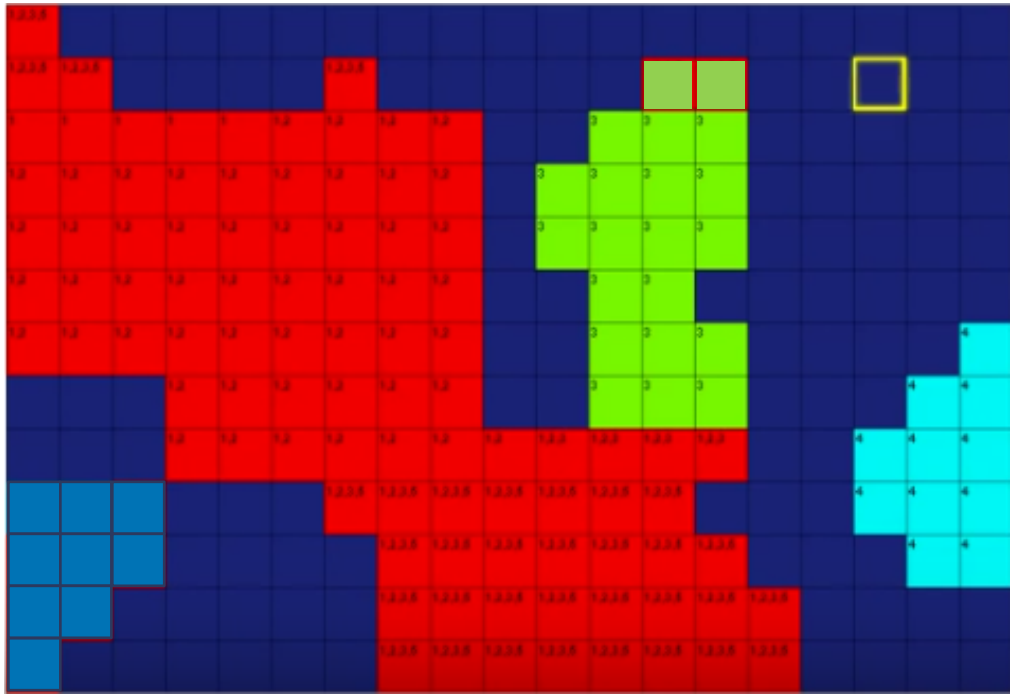
Row-by-row labelling



Region ID	Connections
1	1,2,3
2	1,2
3	1,3
4	4
5	5

Step 2 (second pass):
If a region has connections other than itself, change it to the lowest number

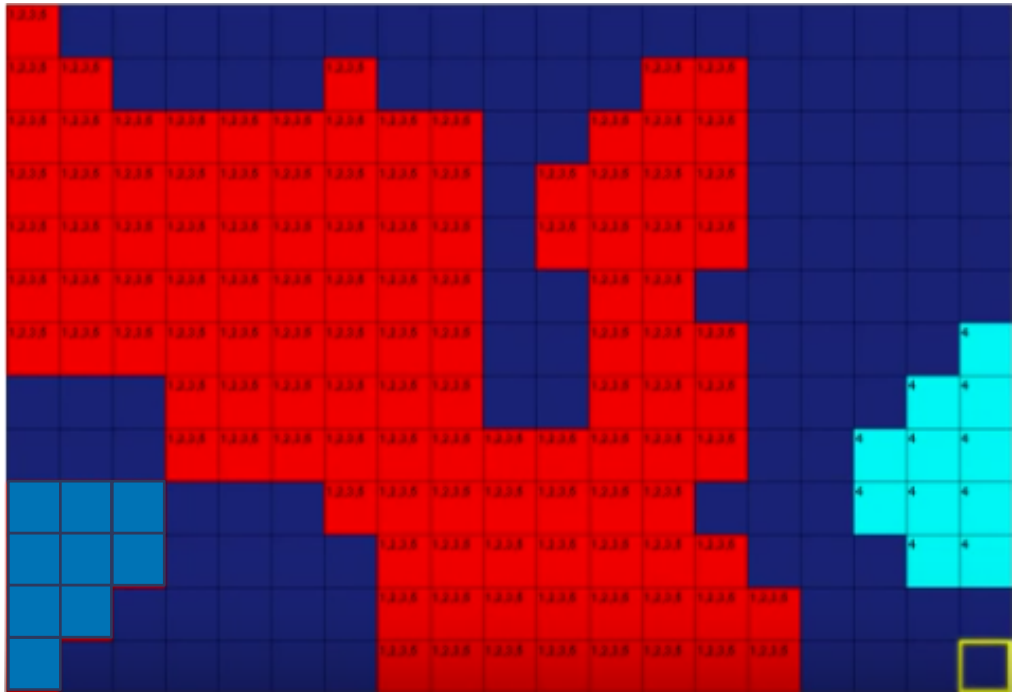
Row-by-row labelling



Region ID	Connections
1	1,~~2,3~~
2	1,~~2~~
3	1,~~3~~
4	4
5	5

Step 2 (second pass):
If a region has connections other than itself, change it to the lowest number

Row-by-row labelling



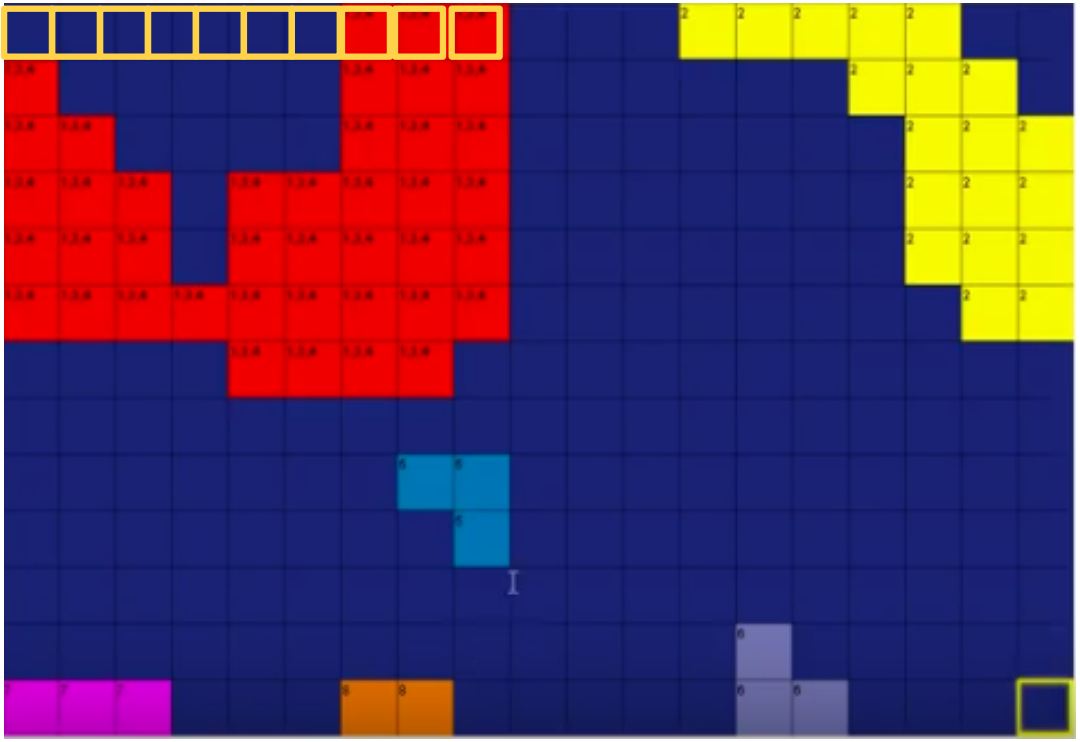
Region ID	Connections
1	1
2	1
3	1
4	4
5	5

Final labels

Labelling

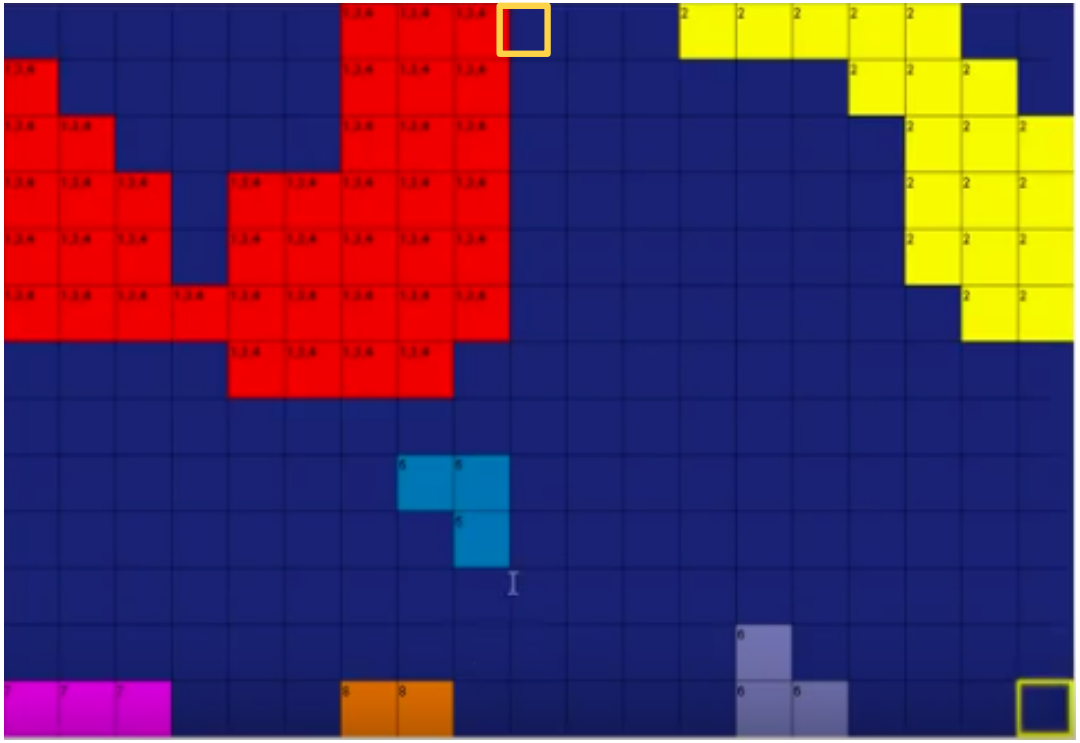
- Row-by-row labelling is one of many labelling methods.
- Other methods include
  - Connected component labelling
  - Recursive labelling
  - Union-find structure
  - Run-Length encoding

Object area



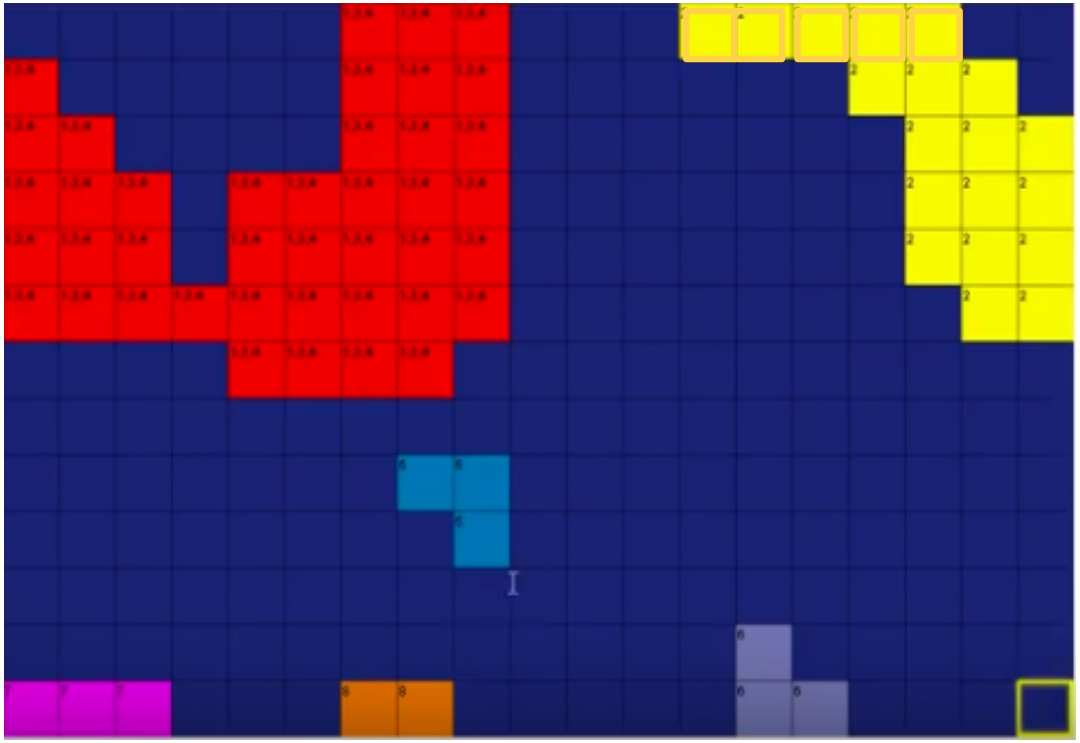
Region ID	Count
1	
2	
3	
4	
5	
6	

Object area



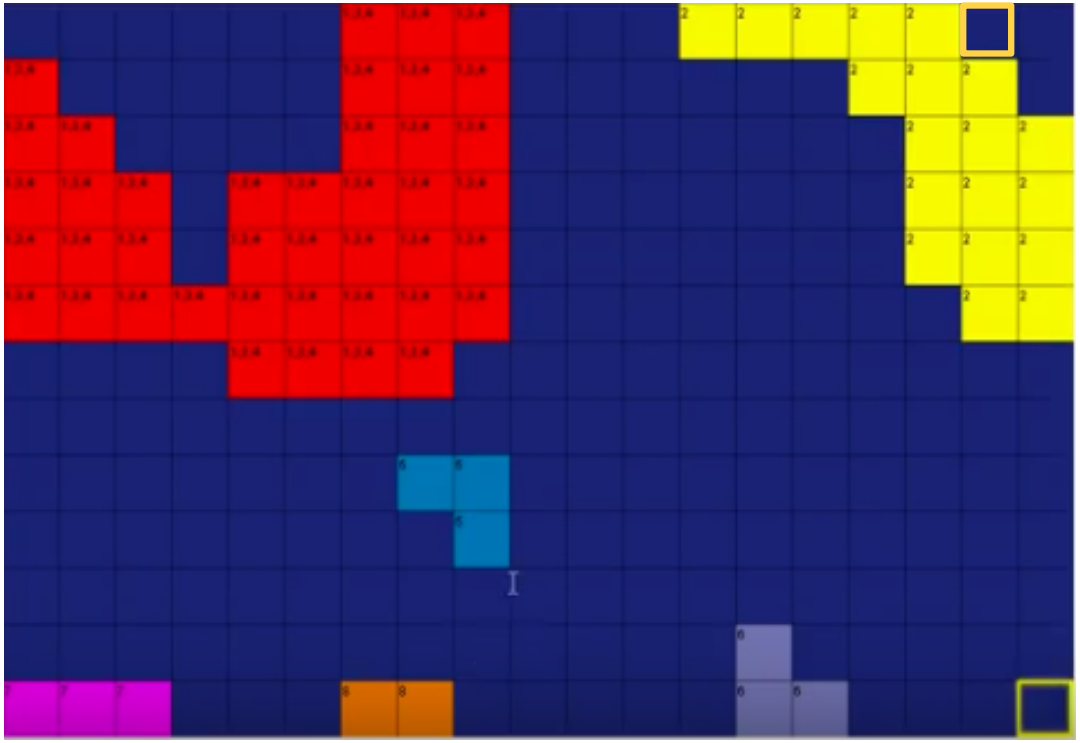
Region ID	Count
1	3
2	
3	
4	
5	
6	

Object area



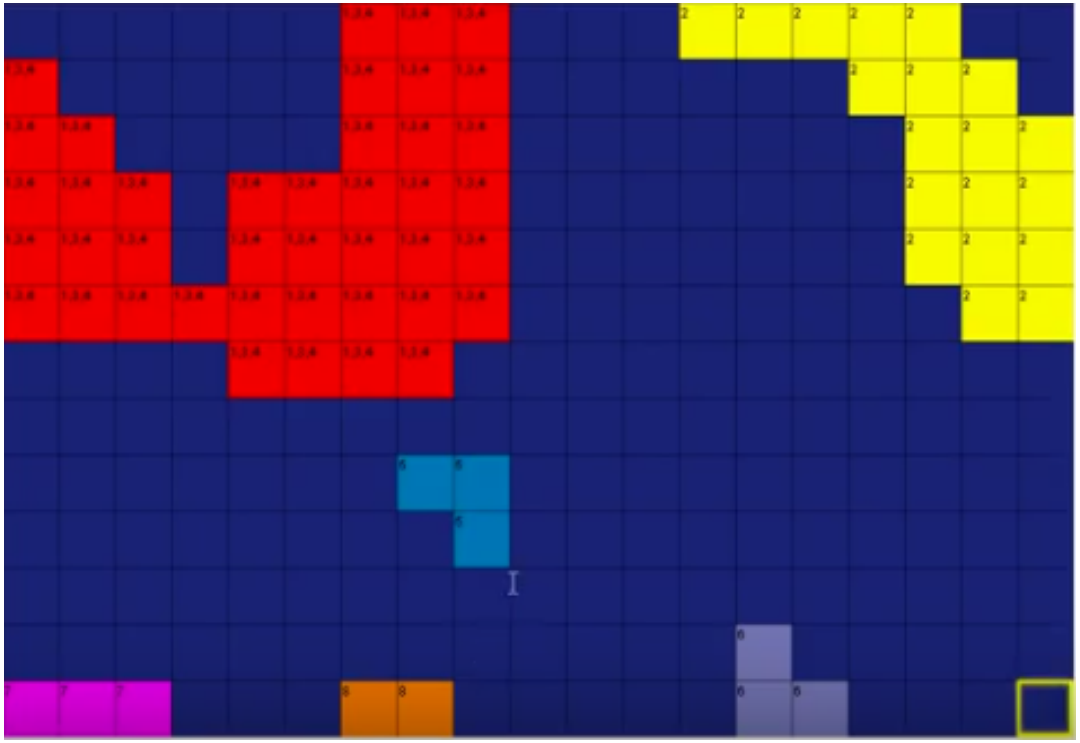
Region ID	Count
1	3
2	
3	
4	
5	
6	

Object area



Region ID	Count
1	3
2	5
3	
4	
5	
6	

Object area

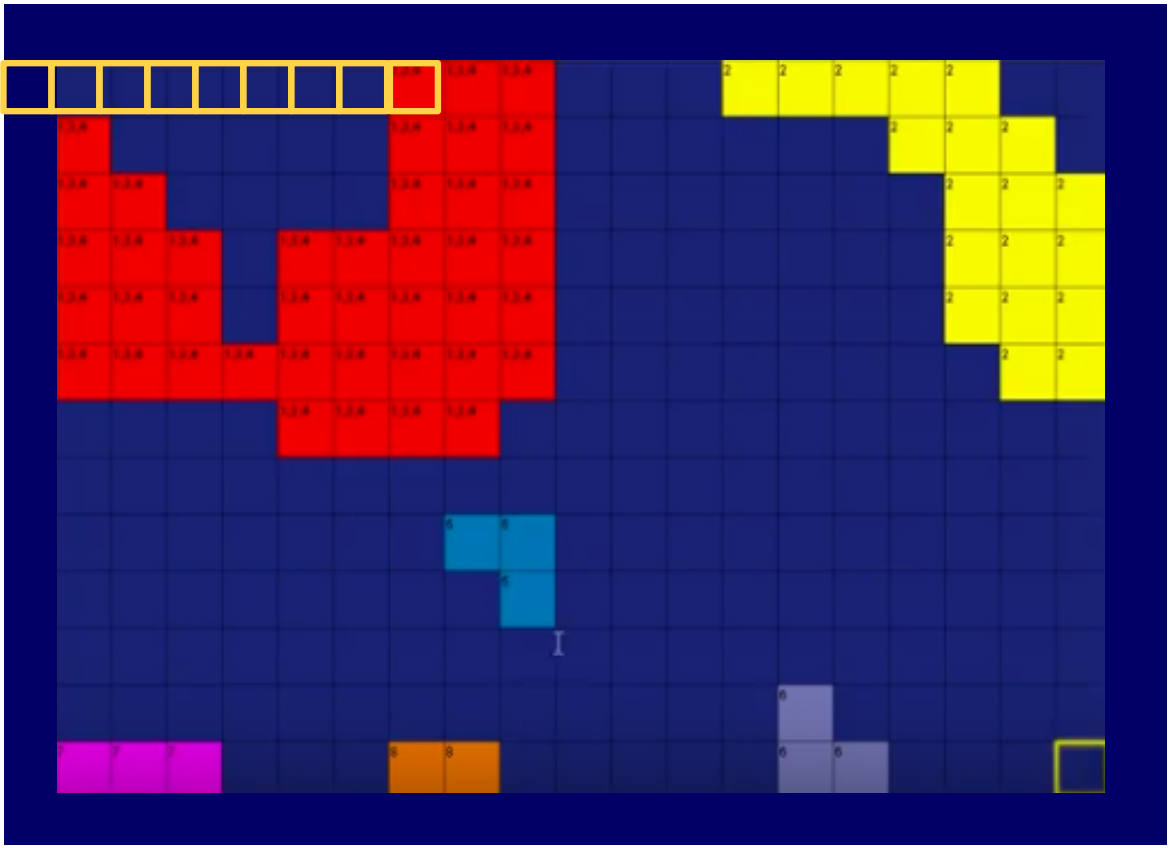


Region ID	Count
1	41
2	19
3	3
4	3
5	2
6	3

Object area

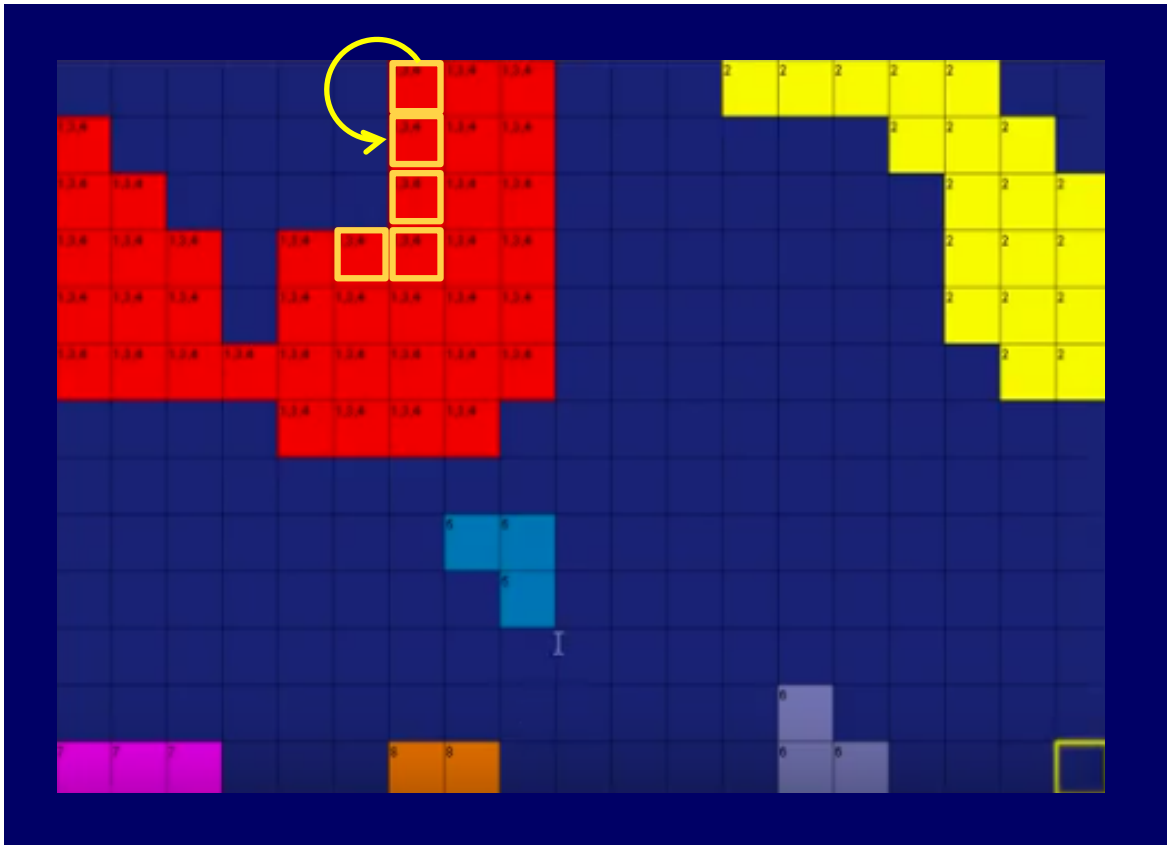
- Algorithm
  - Set up a count table for a given number of objects
  - For each object set the count to zero
  - Scan pixels left-to-right then top-to bottom
  - Get pixel label
  - Increment the pixel count for this label

Object perimeter



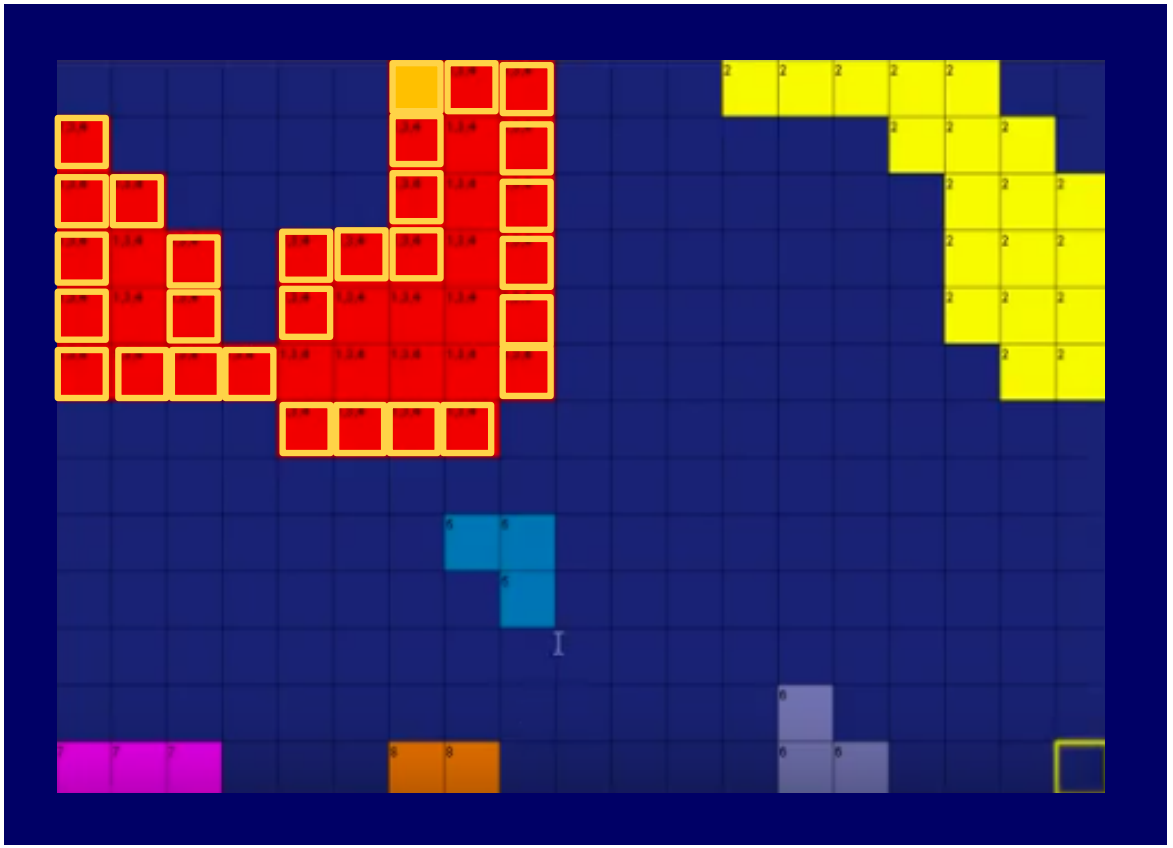
Scan left-to-right then top-to bottom
Stop when a labelled pixel found
Save the pixel's coordinate

Object perimeter



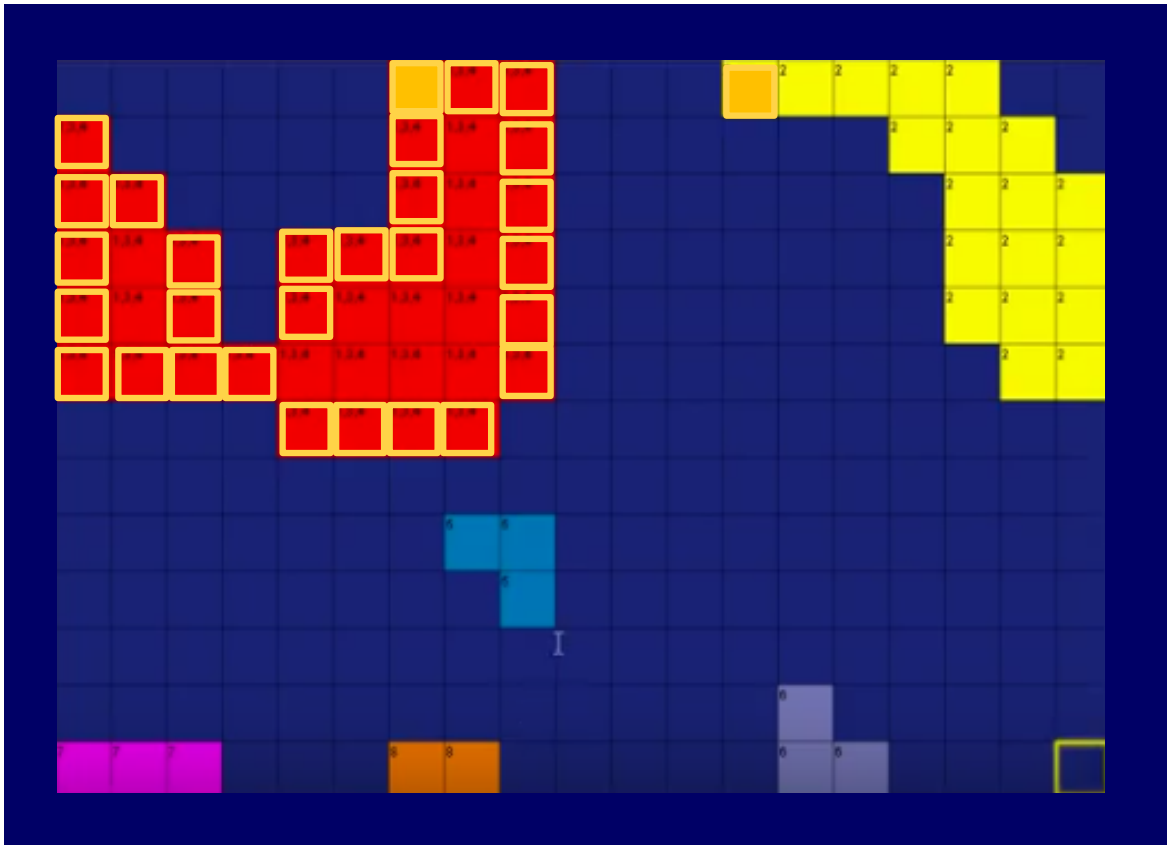
Follow the contour using the “hand-on-the-wall” algorithm

Object perimeter



Continue following the contour until the current pixel has the same coordinates as the first pixel.
Keep saving the coordinates of each new pixel on the contour.

Object perimeter

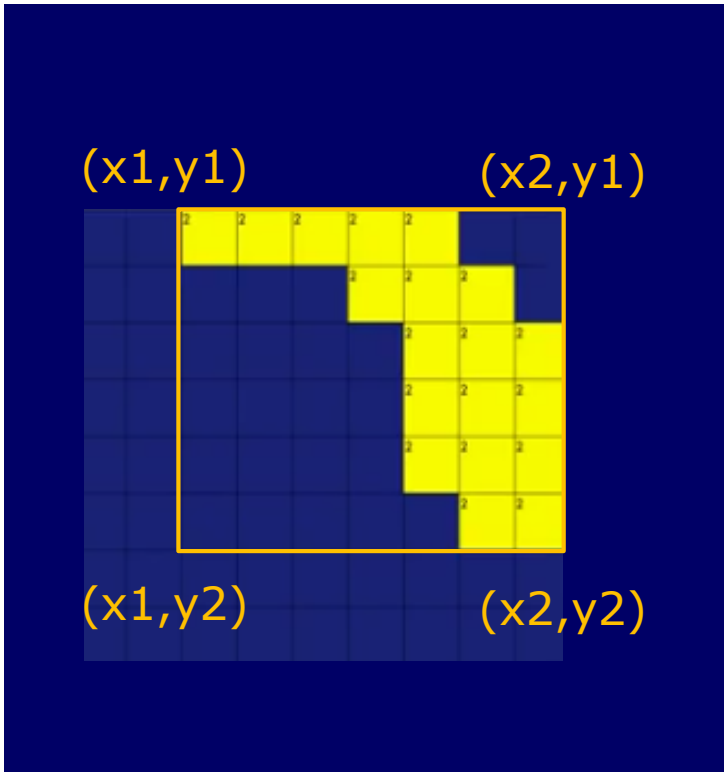


Continue to the next label and repeat the outlining.
Stop when the contours for all the labelled regions are found.

Object locations
Bounding box

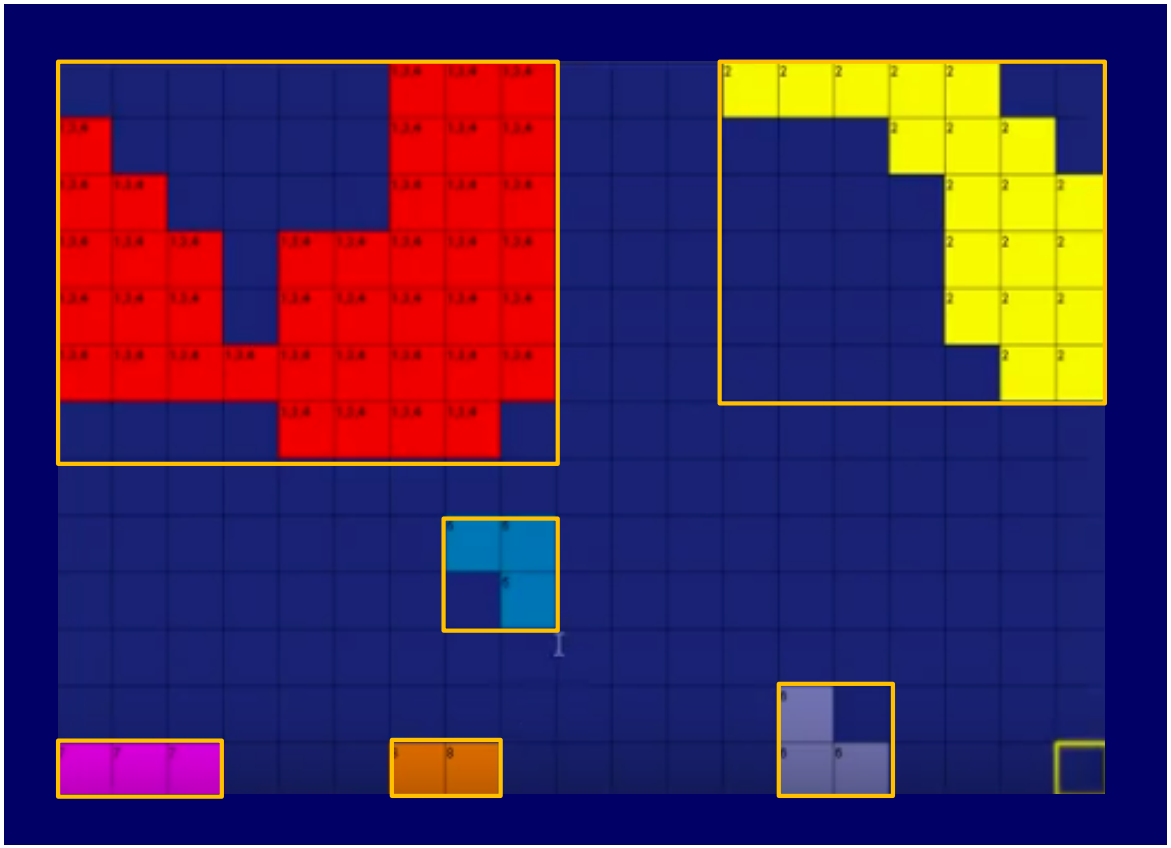
- The bounding box (minimum bounding rectangle, or envelope), is a set of coordinates defining the minimum and maximum x- and y- coordinates of an object.

- **Computation**:
- In the list of object coordinates find:
  - Minimum x-coordinate (x1)
  - Maximum x-coordinate (x2)
  - Minimum y-coordinate (y1)
  - Maximum y-coordinate (y2)
  - Width: x2 – x1 +1
  - Height: y2 – y1 +1
- The coordinates of the bounding box are as shown on the right.



Object locations
Bounding box

Bounding boxes for a sample labelled image



Object locations
Object centre

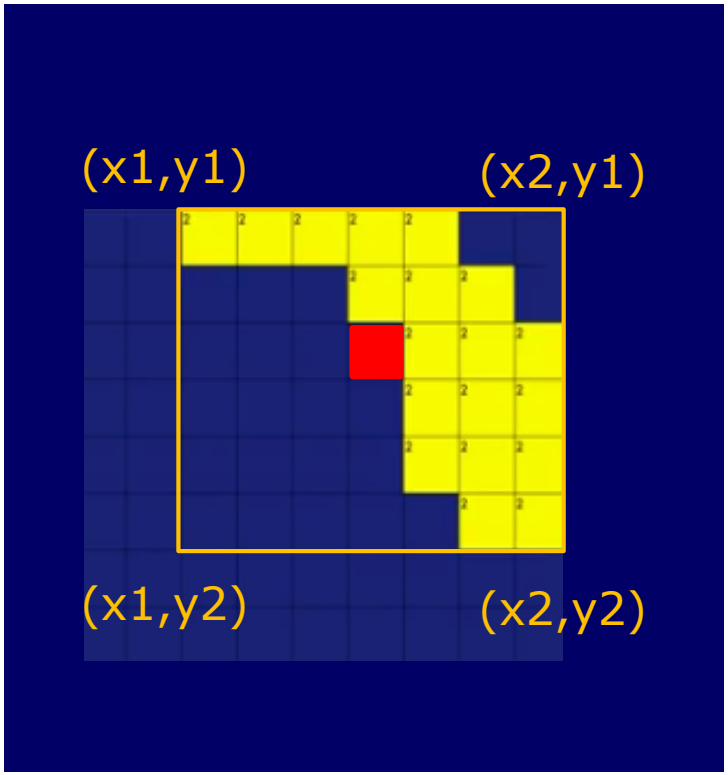
- Object centre can be defined in a number of ways.
- The most common are
  - Centre of the bounding box
  - Centre of mass

Object locations

Object centre: centre of the bounding box

$$Cx = (x2 - x1) / 2$$

$$Cy = (y2 - y1) / 2$$



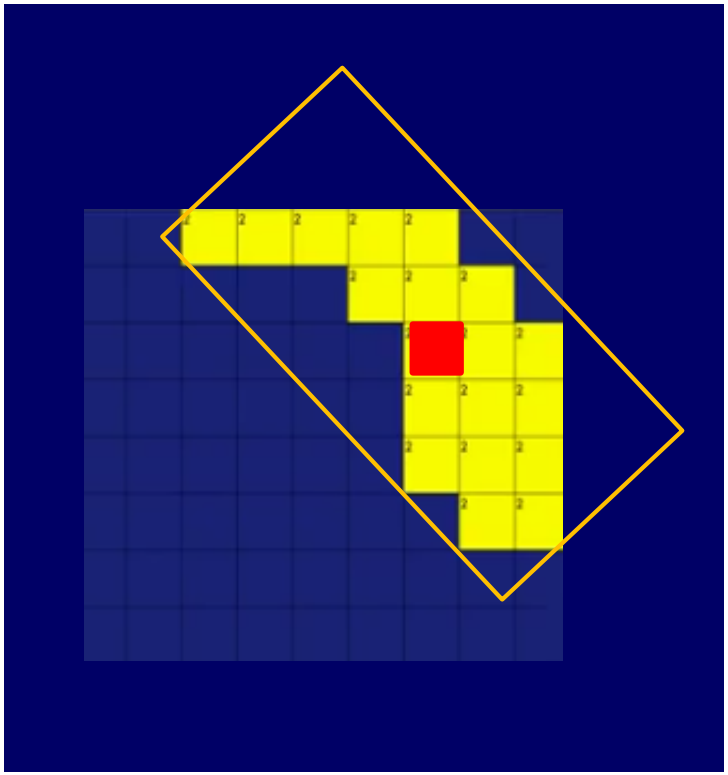
Note:

- The centre can lie outside the object pixels
- The coordinates can be fractional

Object locations
Object centre: centre of the bounding box

Is this a better bounding box?

Fitted ellipse may also be an answer (see later)

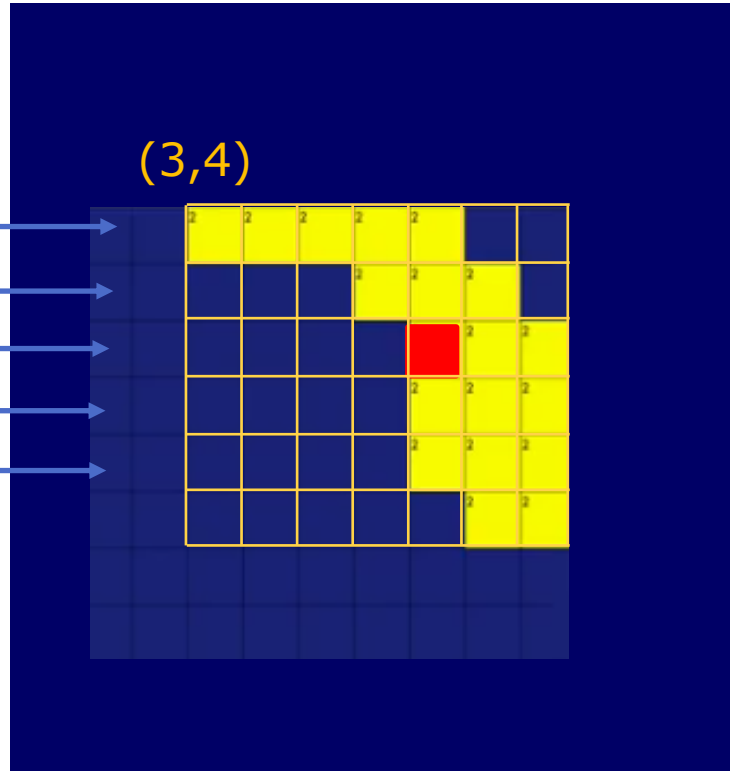


Object locations
Object centre: centre of mass

Sx = sum of all the object *x* *coordinates*
Sy = sum of all the object y coordinates
N = number of pixel in the object

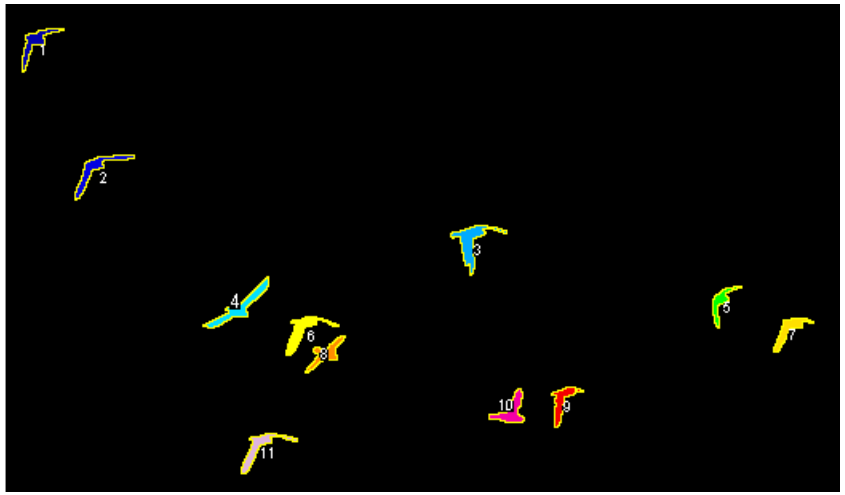
Cx = Sx / N
Cy = Xy / N

Sx=3+4+5+6+7+
6+7+8+
7+8+9+
7+8+9+
7+8+9+
8+9 = 135
Sy = 4+4+ … + 8+9 = 116
N = 19



Cx = 7.1
Cy = 6.1

Geometric measurements
FIJI examples

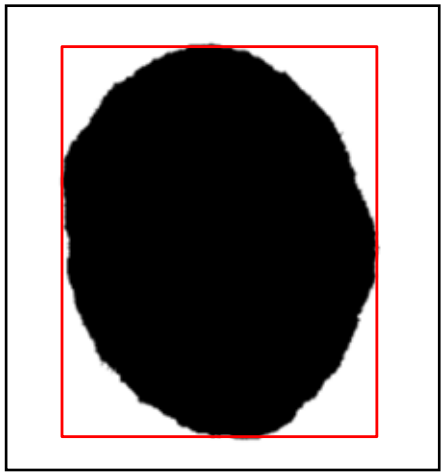


Centre of mass | Bounding box

	Label	Area	XM	YM	BX	BY	Width	Height
1	birds.jpg	169	17.003	23.115	9	14	26	27
2	birds.jpg	204	54.868	102.549	42	93	37	28
3	birds.jpg	264	289.973	147.542	277	137	35	31
4	birds.jpg	244	145.320	187.623	122	169	41	32
5	birds.jpg	117	446.064	185.141	441	175	18	26
6	birds.jpg	190	183.932	203.268	174	194	33	24
7	birds.jpg	150	488.480	202.613	479	195	25	21
8	birds.jpg	158	198.975	217.006	186	206	25	23
9	birds.jpg	153	346.775	247.389	340	238	20	25
10	birds.jpg	187	314.976	252.570	301	239	22	21
11	birds.jpg	188	157.426	276.340	146	267	35	25

Geometric measurements
FIJI examples

Object 1

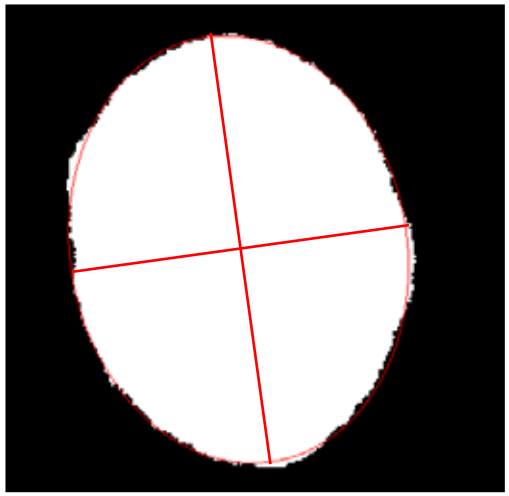


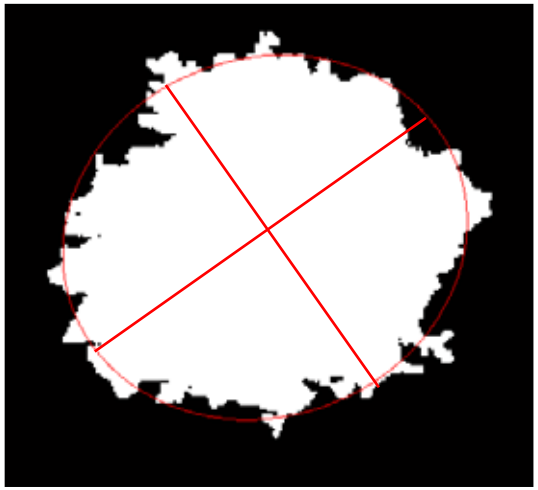
Object 2

Geometric centre | Centre of mass | Bounding box

Object	Area	X	Y	XM	YM	Perim.	BX	BY	Width	Height
1	20930	99.71	104.13	99.71	104.12	844	0	0	214	208
2	17570	104.19	93.07	104.2	93.11	808	0	0	212	192

Geometric measurements
FIJI: Fitted ellipse parameters

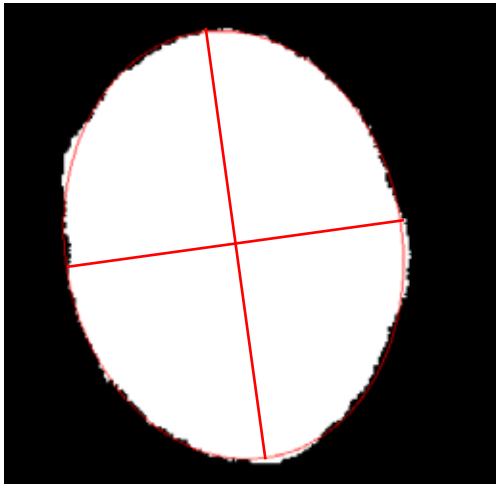


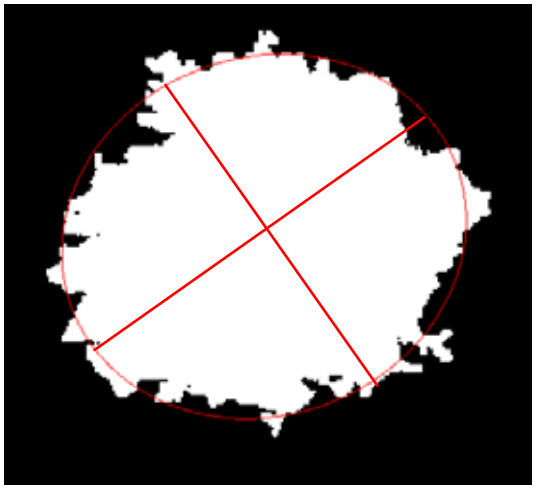


Fitted ellipse axes | Circularity | Aspect ratio

	Major	Minor	Angle	Circ.	%Area	AR	Round	Solidity
1	184.43	144.49	99.59	0.37	47.02	1.28	0.78	1
2	160.1	139.73	20.8	0.34	43.17	1.15	0.87	1

Geometric measurements
FIJI: Fitted ellipse parameters





	Circ.	AR	Round
1	0.37	1.28	0.78
2	0.34	1.15	0.87

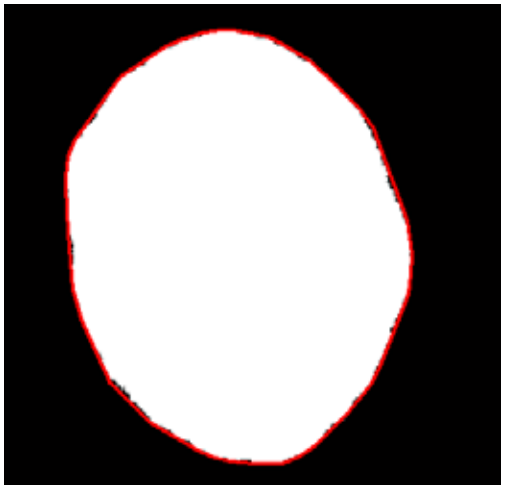
$$\text{Circularity} = \frac{4\pi \cdot Area}{Perimeter^2}$$

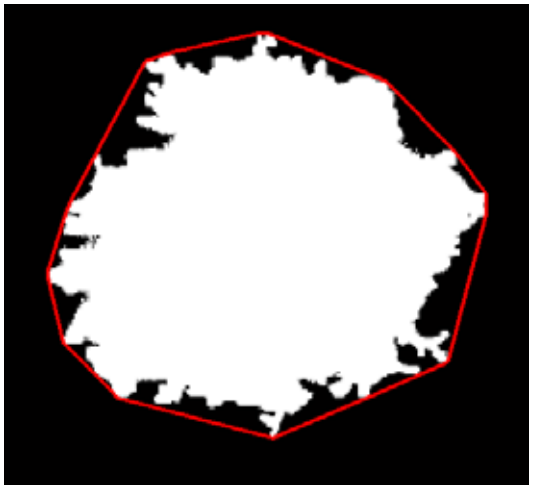
$$\text{Aspect Ratio (Elongation)} = \frac{Major\ axis}{Minor\ axis}$$

Perfect circle = 1
Decreases towards 0.0 for an increasingly elongated shape.

Roundness = inverse of Circularity

Geometric measurements
Other parameters





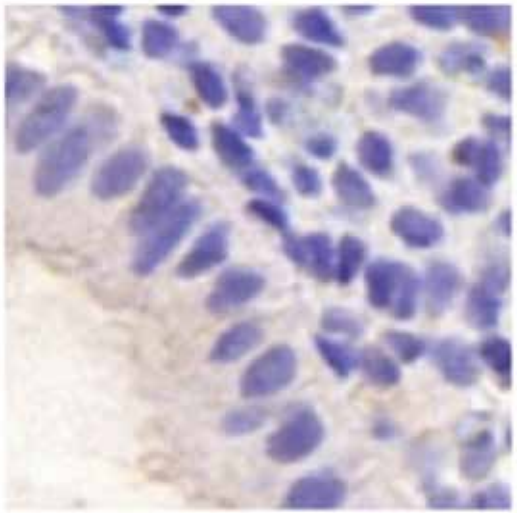
	Comp.	Conc.
1	34.03	1.19
2	37.16	17.46

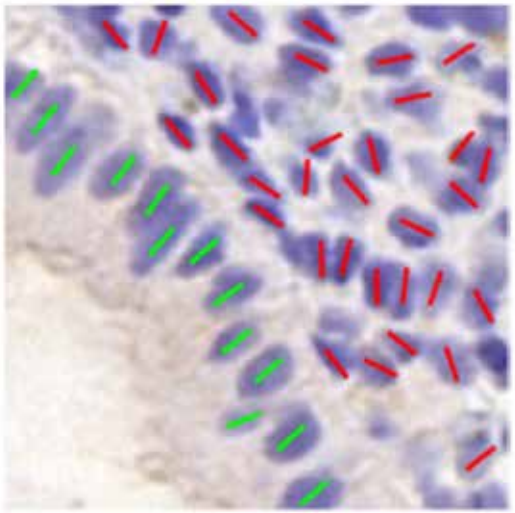
Compactness= $\frac{Perimeter^2}{Area}$

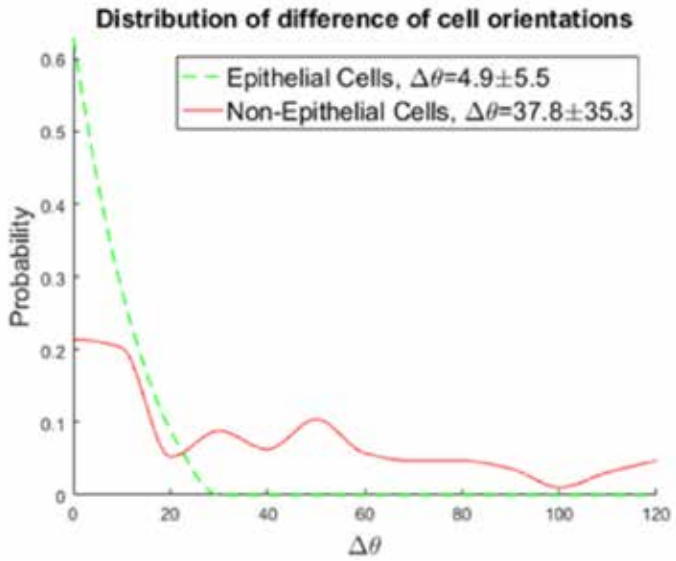
Concavity= 100 $\frac{Convex\ hull\ area - Object\ area}{Object\ area}$

Convex hull of a binary object - the smallest convex polygon that contains the object.

Geometric measurements
Fitted ellipse parameters: application

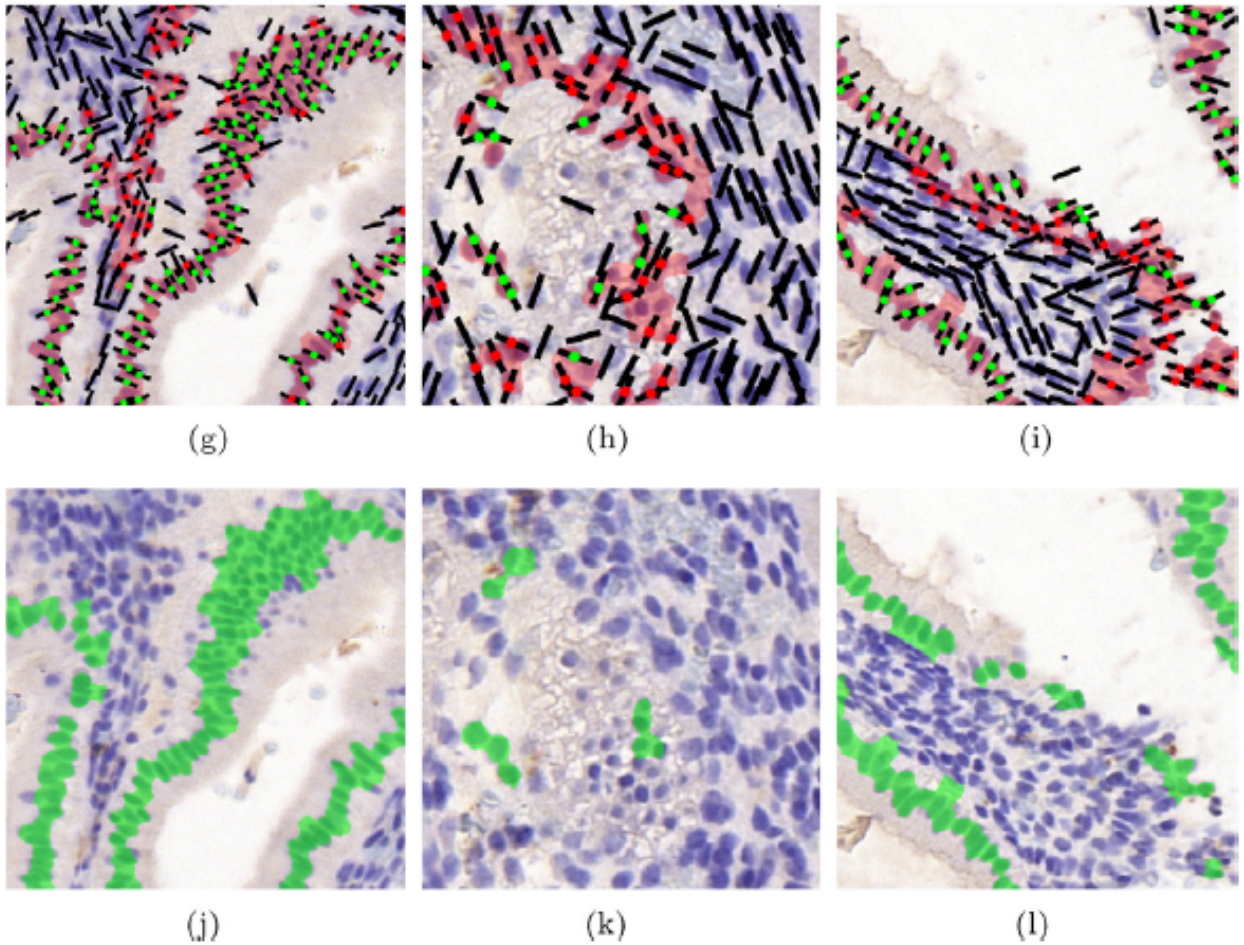






G. Li et al. / Medical Image Analysis 37 (2017) 91–100

Geometric measurements
Fitted ellipse parameters: application



Histological sample segmentation based on the orientation of cells. Correctly segmented into different tissue types.

In this lecture we have covered:

- How to get coordinates of the object boundaries
- How to count objects in a segmented image
- How to measure objects
- How to measure object locations
- How to measure some object properties

Next lecture:

- Further object properties – descriptors and methods
- Shape
  - Outlines
  - Geometric features (e.g. curvature)
  - Fractal dimension
- Texture
  - Grey level statistics
  - Co-occurrence matrices
- From segmented images to segmented object properties

Further reading and experimentation

- **Book chapters**:
- Gonzalez, R.C. & Woods, R.E. Digital Image Processing, Addison-Wesley (various editions), 7.4.2, 8.1.1.
- Sonka, M. Hlavac, V. Boyle, R. (various editions) Image Processing, Analysis and Machine Vision, Chapman & Hall Computing, 6.2.1.
- Umbaugh, S.E. Computer vision and image processing : a practical approach using CVIPtools , Prentice Hall International (various editions), 2.4.2, 2.4.3.

- **Boundary tracing**
- http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/alg.html

- **Binary object properties**
- *https://courses.cs.washington.edu/courses/cse576/book/ch3.pdf*

- *HIPR2 resources*
- **Pixel labelling**
- http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm