GRAPHICS GROUP

Intelligent Vision Systems New Zealand

Computer Graphics and Image Processing

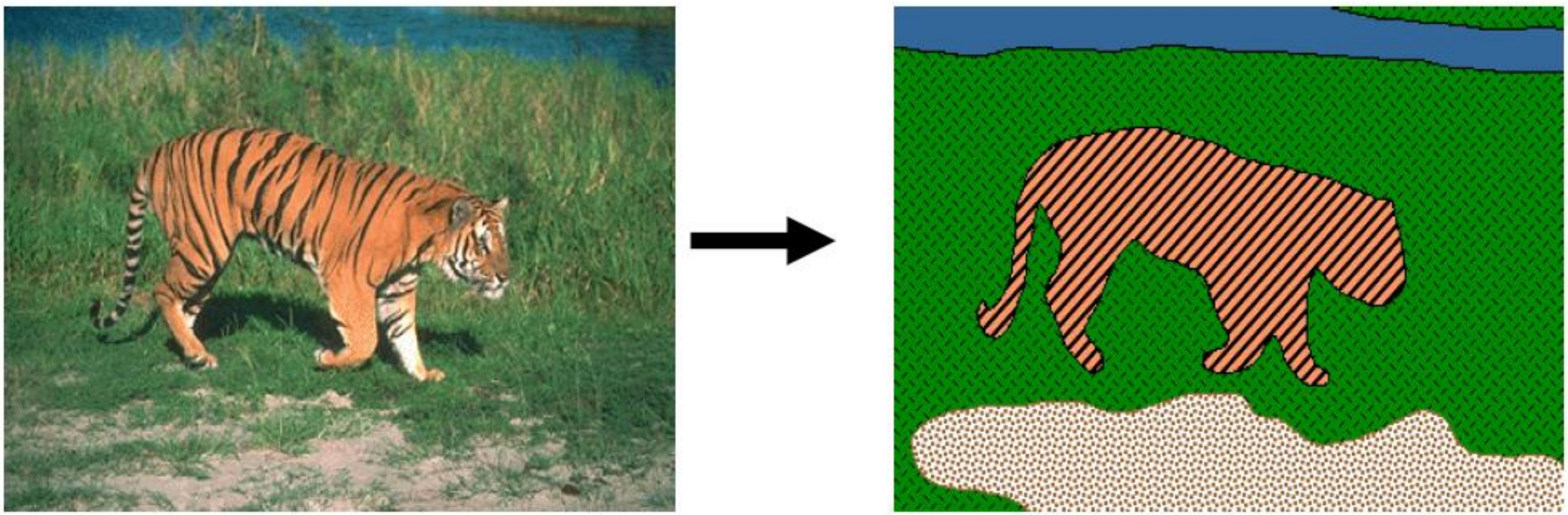
Part 3: Image Processing

8 – Segmentation Part II

Martin Urschler, PhD

Image segmentation definition

- Process of partitioning the image domain R (all pixels) into n subregions $R_1, R_2, \ldots, R_n$



- For $n = 2$: Binary segmentation (or foreground/background segmentation)

Basic segmentation approaches

- Non-contextual thresholding:
  - Grouping pixels with no account of locations in the lattice
- Contextual segmentation:
  - Can be more successful in separating individual objects
  - Accounts for close locations of pixels belonging to an object (prior knowledge)
  - Exploit two signal properties: **discontinuity** or **similarity**
- Discontinuity-based segmentation (finding boundaries):
  - Goal: to build complete boundaries of uniform regions
  - Assumption: abrupt signal changes across each boundary
- Similarity-based segmentation (finding uniform regions):
  - Goal: to group connected pixels that satisfy similarity criteria
- Latter two approaches mirror each other, in the sense that a complete boundary splits one region into two

Pixel Neighborhood Rectangular Lattice

Normal sampling: a digital image on a finite arithmetic lattice:

$$\{(x, y) : \; x = 0, 1, \ldots, X - 1; \; y = 0, 1, \ldots, Y - 1\}$$

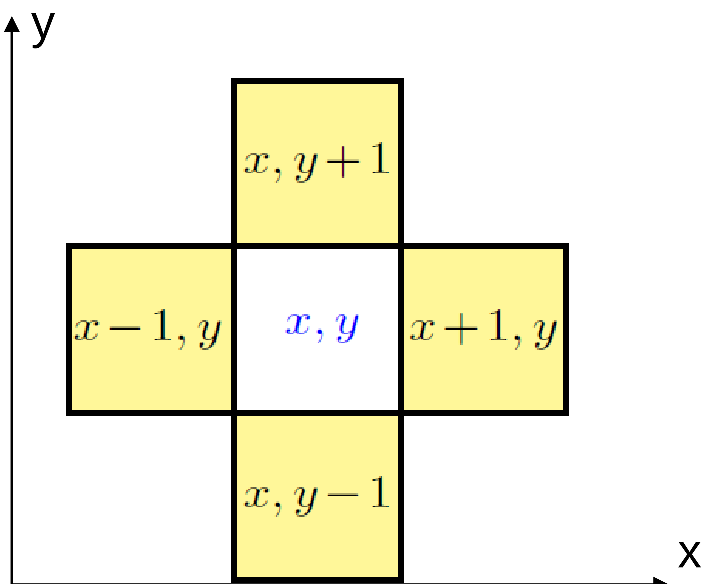
Two types of the nearest neighbourhood of a pixel (x, y):

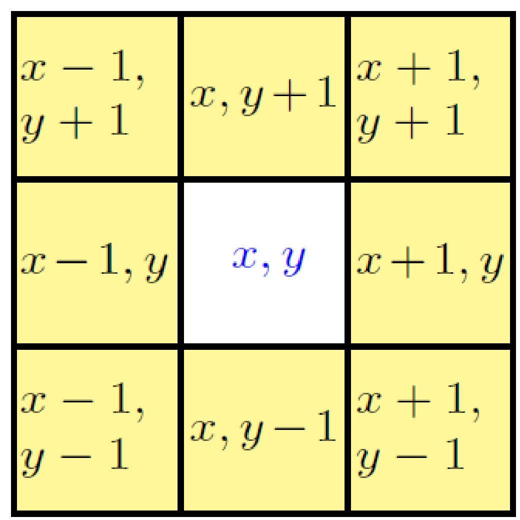
4-neigbourhood:

$$\{(x, y \pm 1), \; (x \pm 1, y)\}$$

8-neighbourhood:

$$\{(x - 1, y \pm 1), \; (x, \pm 1), \; (x + 1, y \pm 1), \; (x \pm 1, y)\}$$





Pixel Connectivity

A 4- or 8-connected path from pixel p_1 to another pixel p_n

is a sequence of pixels $\{p_1, p_2, \ldots, p_n\}$, such that p_{i+1} is a 4- or 8-neighbour, respectively, of p_i for all $i = 1, \ldots, n-1$.

						p_{11}
		p_4	p_5	p_6		p_{10}
p_1	p_2	p_3		p_7	p_8	p_9

4-connected path

						p_7
		p_3	p_4		p_6	
p_1	p_2			p_5		

8-connected path

- A set of pixels is a **4-connected region** if there exists at least one 4-connected path between any pair of pixels from that set.
- The **8-connected region** has at least one 8-connected path between any pair of pixels from that set.

Region similarity

- Uniformity / non-uniformity of pixels in a connected region is represented by a uniformity predicate Q
  - Logical statement, or condition being true if pixels in the regions are similar with respect to some property
  - Pixel properties: colour, grey level, edge strength, local texture pattern, etc.
- Common simple local predicate $Q(R)$
  - Restricted signal variations over a pixel neighbourhood in a connected region R:

$$Q(R) = \begin{cases} \boldsymbol{TRUE} & \text{if } \ |g(x,y) - g(x+\xi, y+\eta)| \leq \delta \\ \boldsymbol{FALSE} & \text{otherwise} \end{cases}$$

    where (x, y) and $(x+\xi, y+\eta)$ are the lattice coordinates of all neighbouring pixels in R

Region similarity

- The simple local predicate in previous slide does not restrict the variation of grey levels within an entire region
  - Small changes in the neighbouring signal values can accumulate over the region
- Intra-region signal variations can be restricted with a similar, but non-local predicate:

$$Q(R) = \begin{cases} \boldsymbol{TRUE} & \text{if } |g(x,y) - \mu_R| \le \varepsilon \\ \boldsymbol{FALSE} & \text{otherwise} \end{cases}$$

where

  - ε is a fixed signal similarity threshold
  - (x, y) are the lattice coordinates of a pixel from the region R
  - μ_R is the mean value of signals $g(x, y)$ over the entire region R

Region Growing: Bottom-up algorithm

- **Initialisation**: a set of seed pixels defined by the user.
- **Region growth**: sequentially add a pixel to a region under the following conditions:
  1. The pixel has not been assigned to any other region.
  2. The pixel is a neighbour of that region.
  3. Addition of the pixel does not impact the uniformity of the growing region.

Region growing is simple but unstable:

- It is very sensitive to a chosen uniformity predicate: small changes of the uniformity threshold may result in large changes of the regions found.
- Very different segmentation maps under different routes of image scanning, modes of exhausting neighbours of each growing region, seeds, and types of pixel connectivity.

Region Growing Algorithm

Uniformity predicate Q:

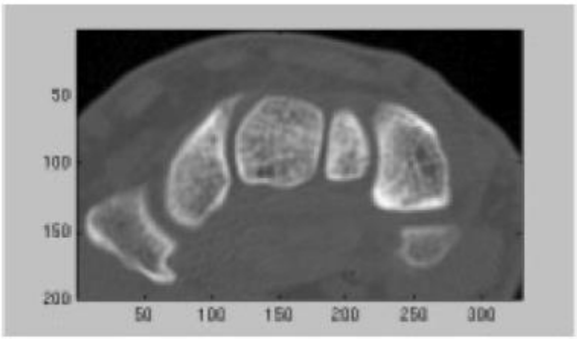
Pixel intensity of neighbour within +/- 2 of current pixel (4 connectivity)

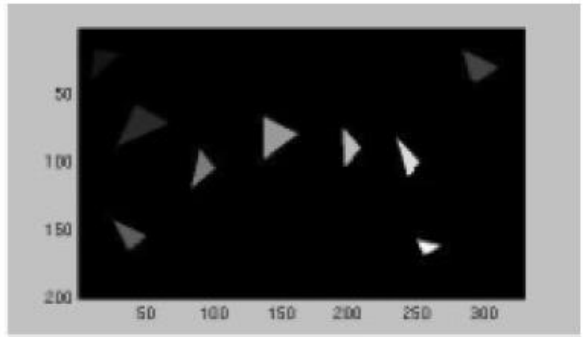
Implementation based on a Queue datastructure!

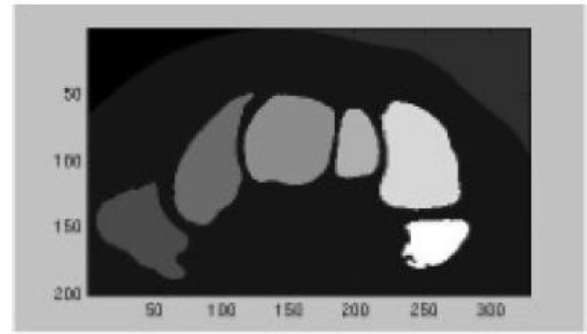
1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

Seed pixel

Region Growing: Examples



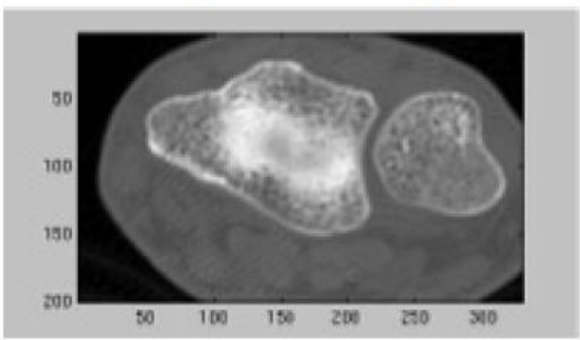


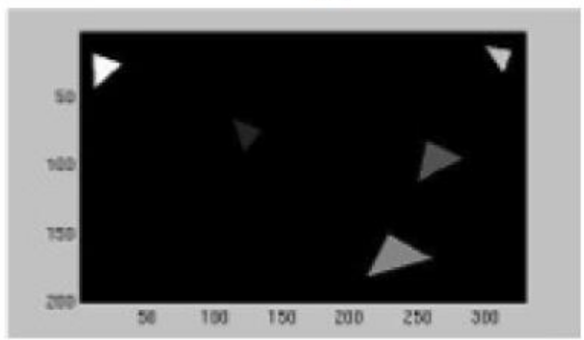


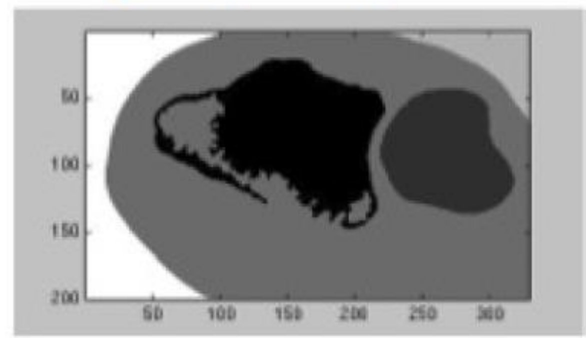
Greyscale image

Seed regions

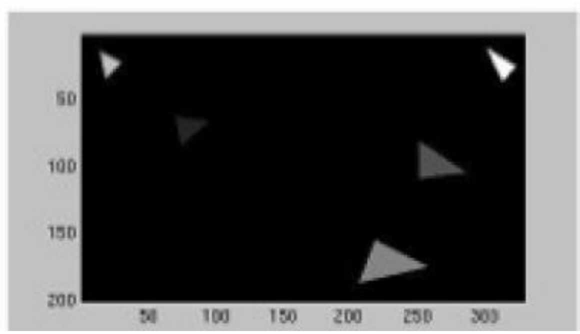
Region growing results

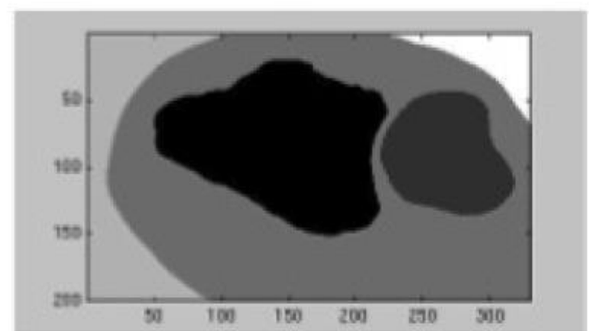






Region growing from two variants of seed regions.





Region Growing: Examples

4-conn

8-conn

Basic Segmentation Criteria

1. All the pixels have to be assigned to regions
2. Each pixel can belong to a single region only
3. Each region is a connected set of pixels
4. Each region is uniform w.r.t. a given uniformity predicate
5. Merging two adjacent regions gives a non-uniform region

- Region growing:
  - Criteria 1 and 2 are not satisfied: In general, the number of seeds may not be sufficient to create a region for every pixel
  - Criteria 3 and 4 are satisfied: Each region is connected and uniform
  - Criterion 5 may hold: Regions grown from two nearby seeds within a potentially uniform part of the image are always regarded as distinct

Split-and-Merge Segmentation: Top-Down

- Initialization: The entire image is a single region
- Iterative segmentation:
  - Splitting stage: Split each region into sub-regions
  - Merging stage: Merge adjacent regions if the resulting larger region remains uniform
- Stopping rule:
  - All regions become uniform OR
  - The desired number of regions have been established

Split-and-Merge Example

Split-and-Merge Segmentation: Top-Down

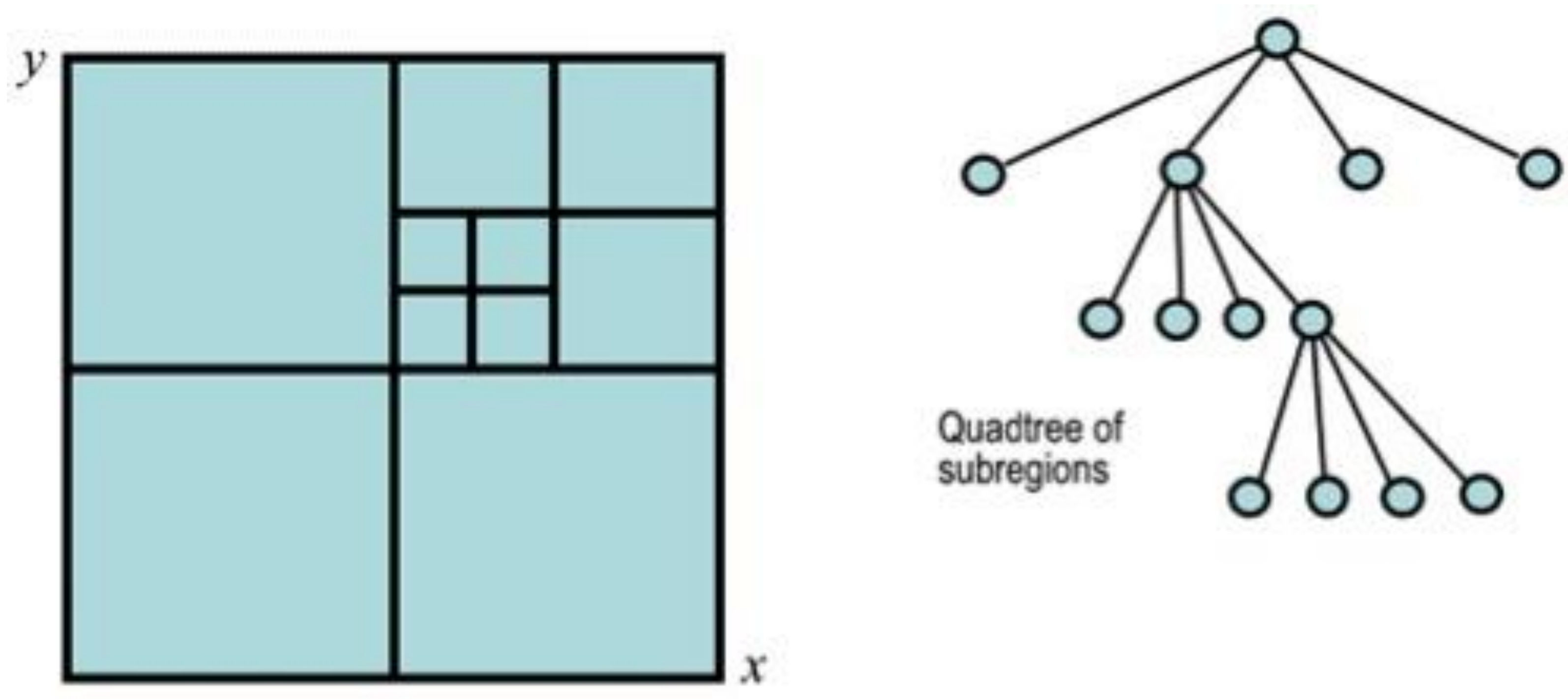
- Common splitting strategy for a square image:
  - Divide it recursively into smaller and smaller quadrants until, for any region R, the uniformity predicate $Q(R)$ is TRUE

- The strategy builds a top-down quadrant tree (*quadtree*):
  - If $Q(image)$ is FALSE, divide the image into 4 quadrants
  - If $Q(quadrant)$ is FALSE, divide the quadrant into 4 sub-quadrants
  - Etc.

  - Terminate if minimum quadregion size is reached

Split-and-Merge Segmentation: Top-Down

- The splitting stage alternates with the merging stage



The merging stage: two adjacent regions R_i and R_j are combined into a new, larger region if the uniformity predicate for the union of these two regions, $Q(R_i \cup R_j)$ is TRUE

The Split-and-Merge Algorithm

1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

Sample image

1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

First split

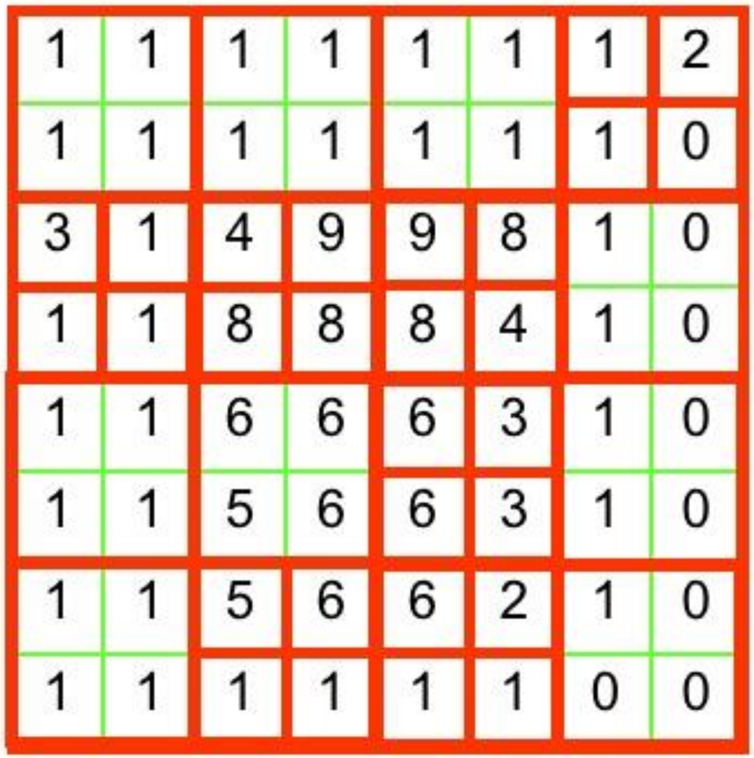
Uniformity predicate Q: Variance of intensities in quadregion below 0.5

1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

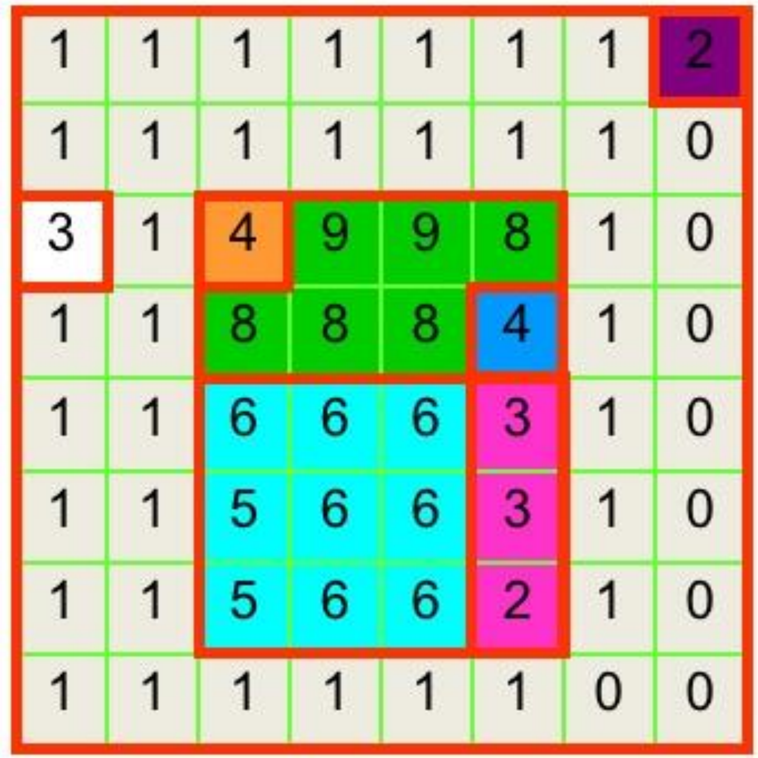
Second split

1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

Third split

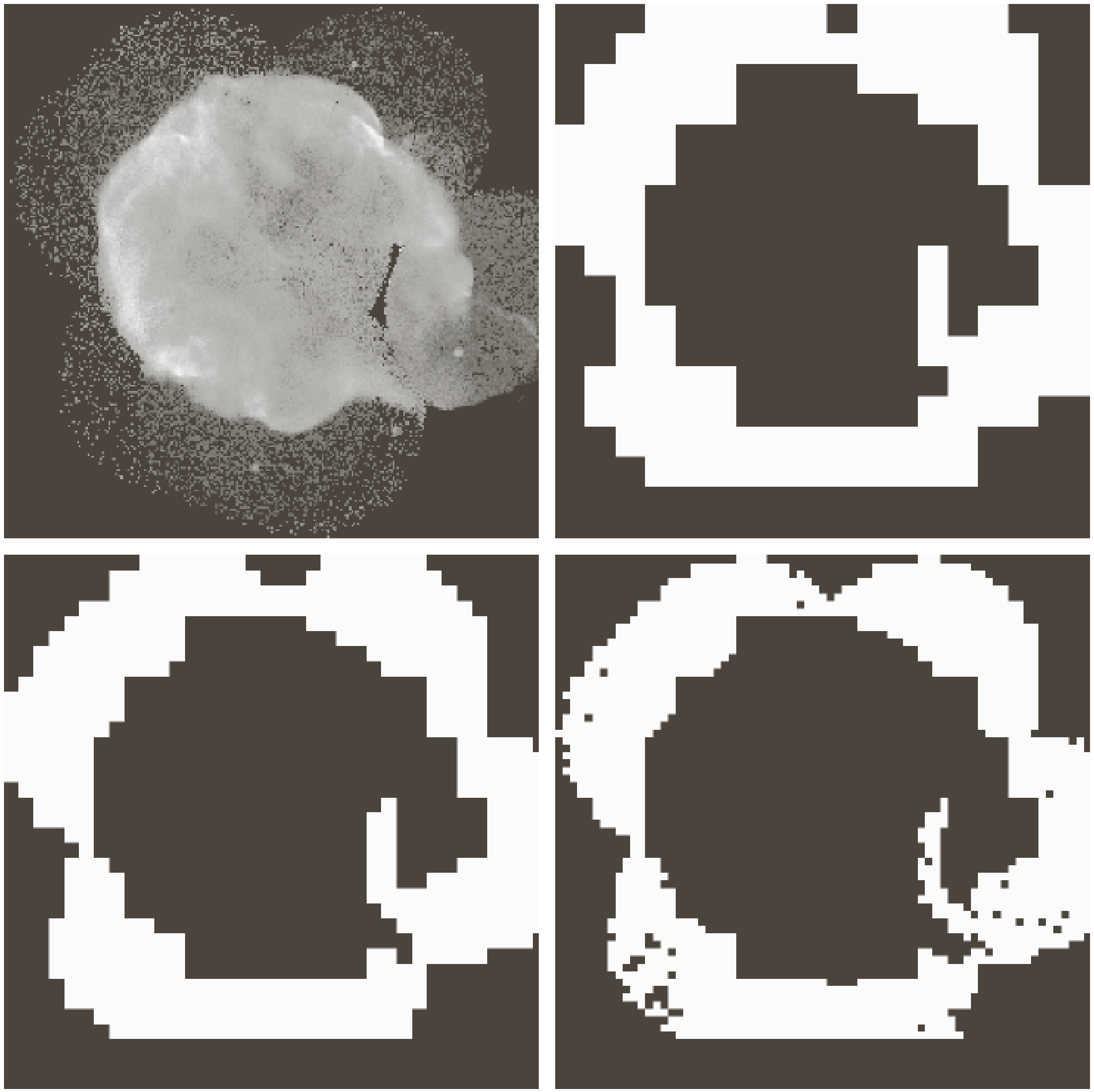


Merge



Final result

Split-and-Merge Example



a b
c d

FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

Aim: Segment parts with high variation

$$Q(R) = \begin{cases} \boldsymbol{TRUE} & \text{if} \quad \sigma > a \; AND \; 0 < \mu < b \\ \boldsymbol{FALSE} & \text{otherwise} \end{cases}$$

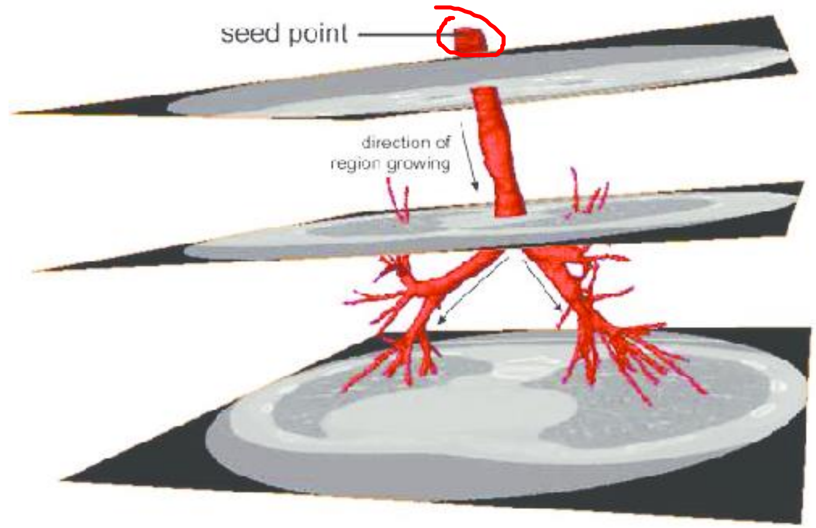
μ, σ: *mean and stddev of quadregion*

$a = 10, b = 125$

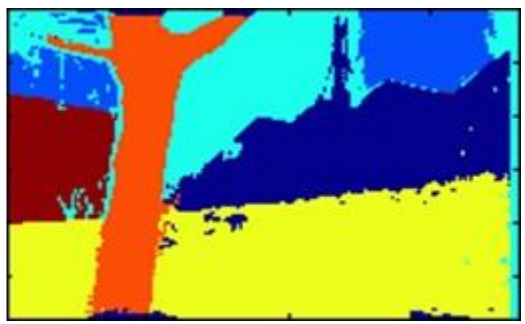
Images from: Gonzalez & Woods, Digital Image Processing, 3rd ed.

Segmentation recap

- 2 kinds of region similarity based approaches
  - Bottom up: Region Growing
  - Top down: Split-and-Merge



Human airway tree (CT data)



RGB color segmentation

Connected component labelling

- In binary segmentation results we often have the need for separating/identifying different objects
- Object is defined w.r.t. a connectedness scheme

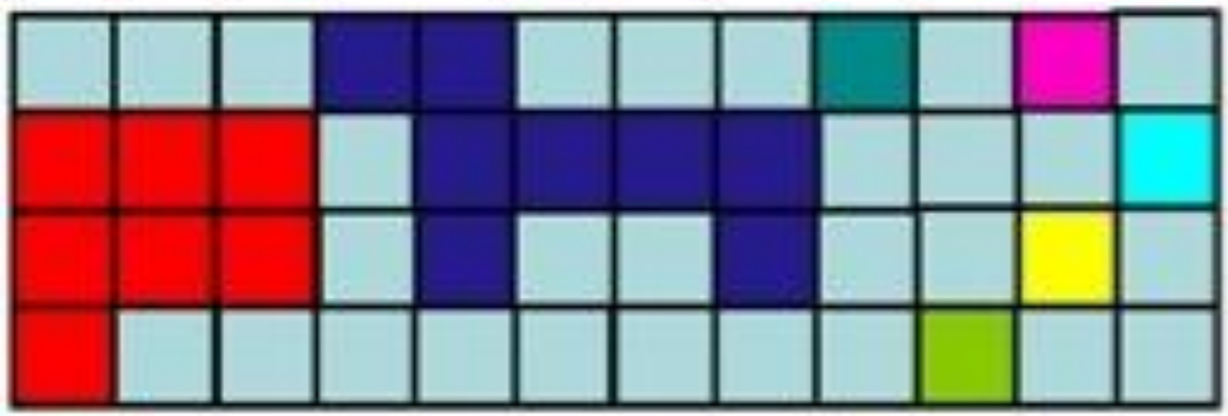
Binary image 64x64: white objects on black background

86 4-connected object regions (color coded)

10 8-connected object regions (color coded)

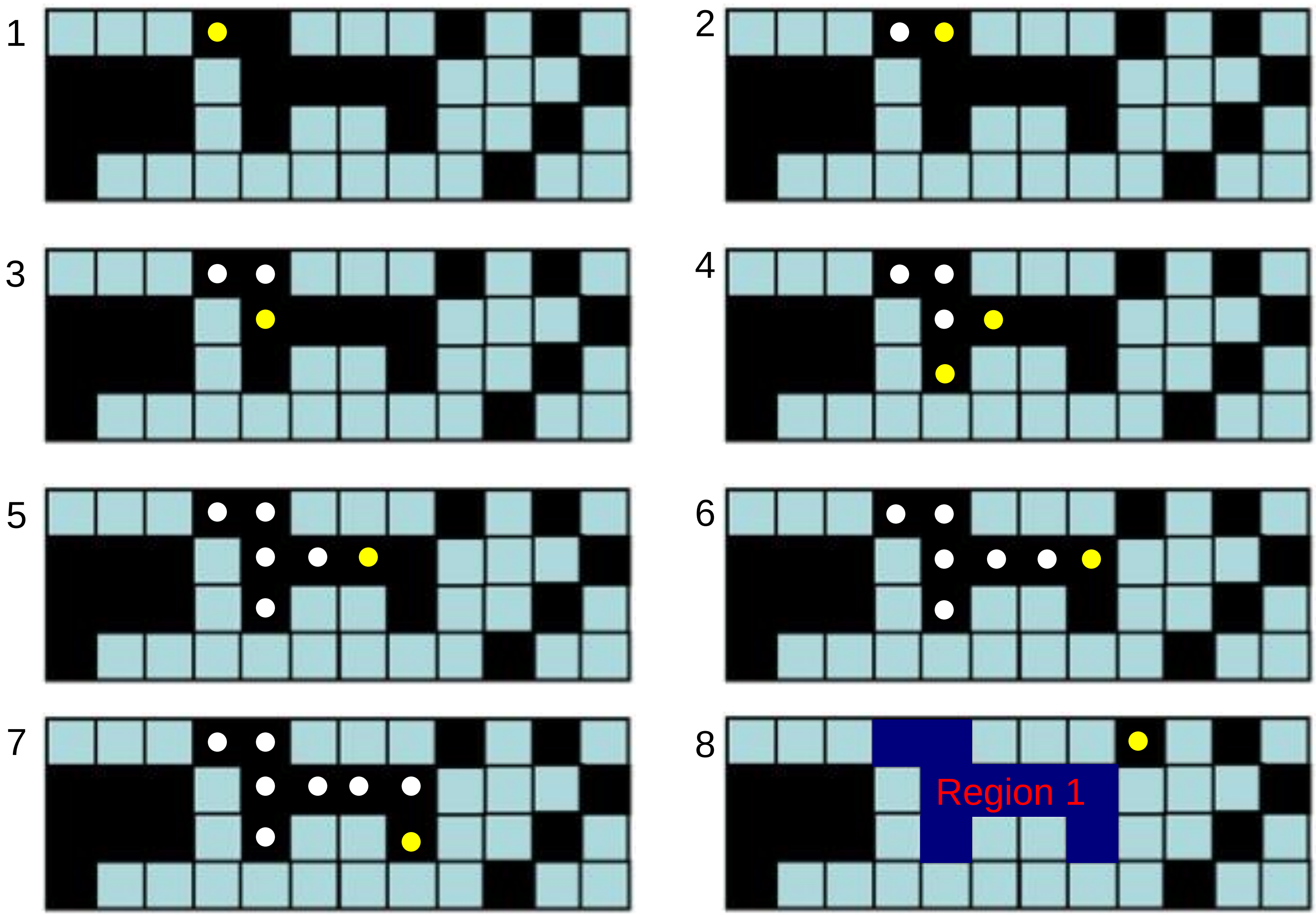
Connected component labelling

- Example



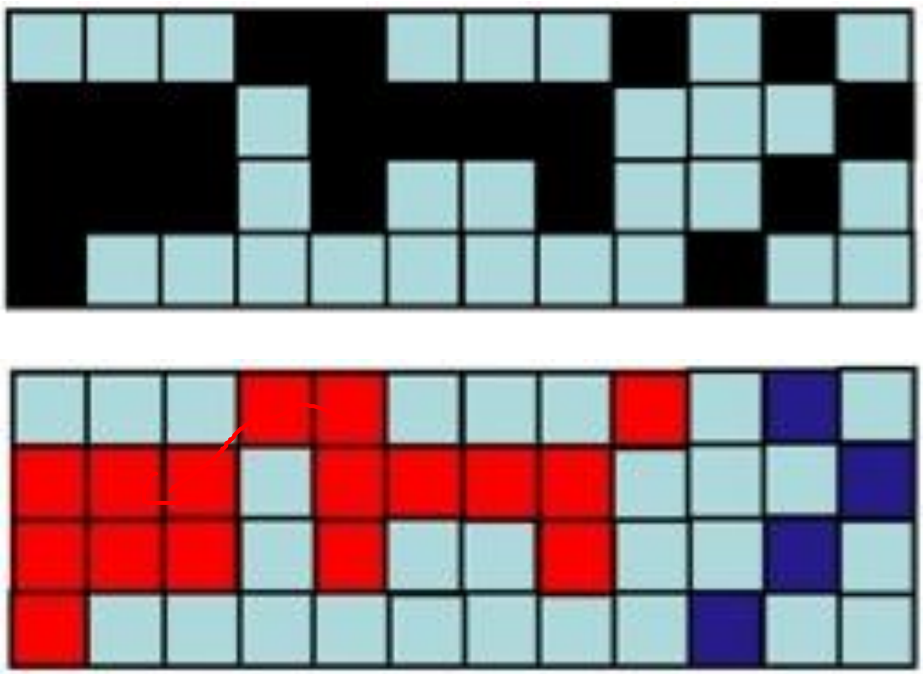
Binary image
0: objects
1: background

Connected component labelling

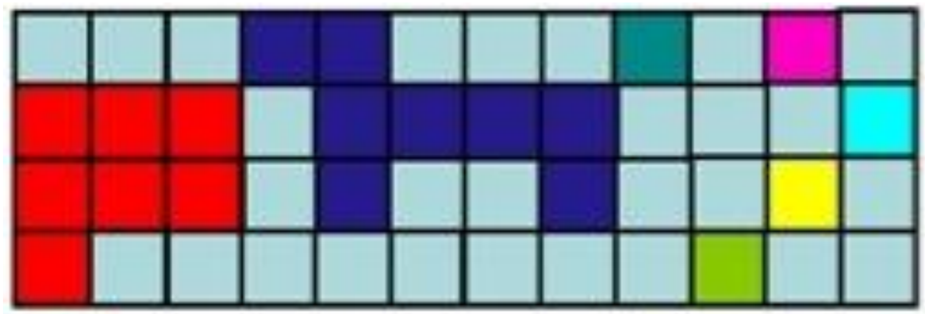


4-conn
“grassfire”

Connected component labelling



8-connected objects
8-connected background

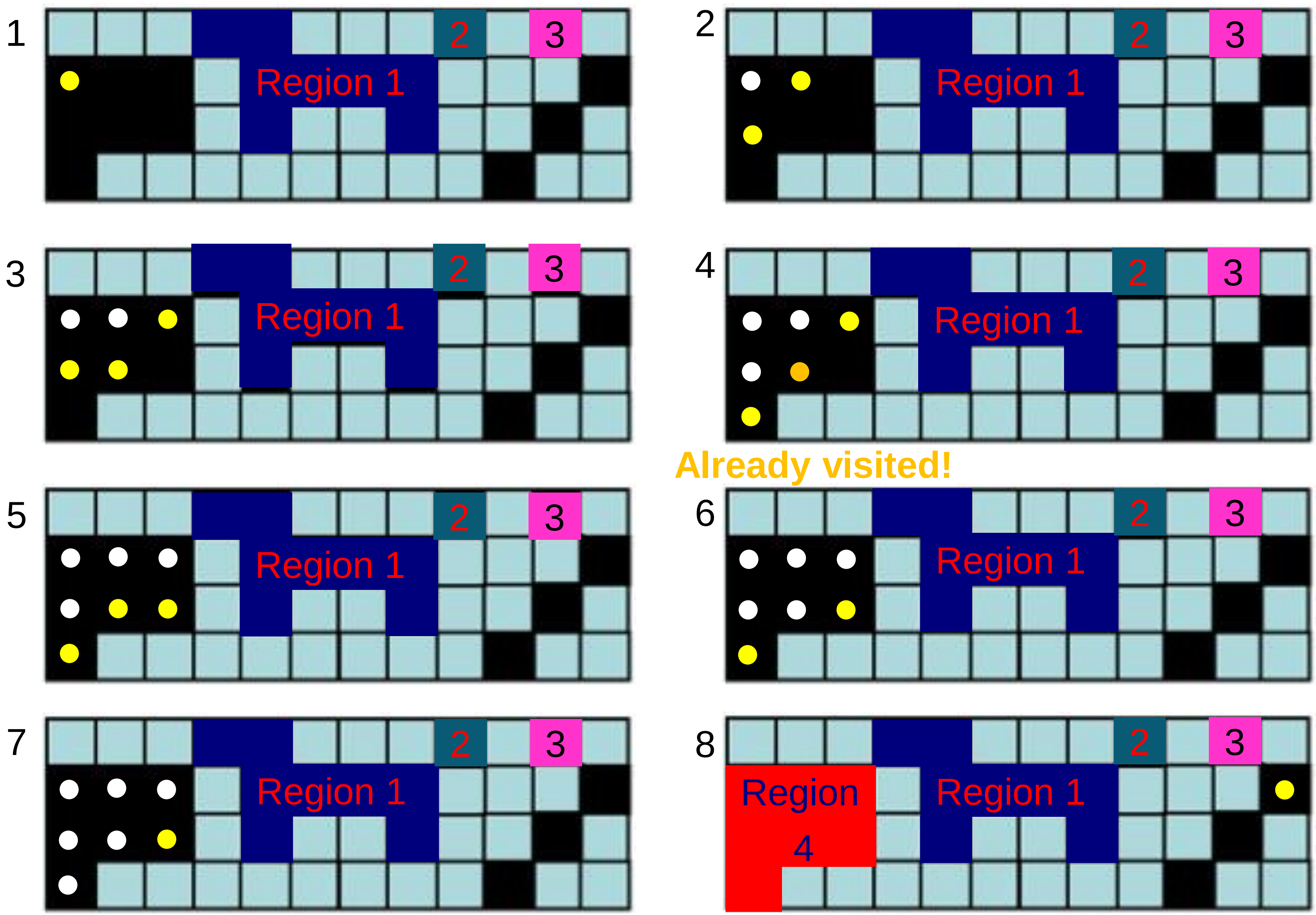


4-connected objects
8-connected background

Queue-based CCL (“grass-fire”)

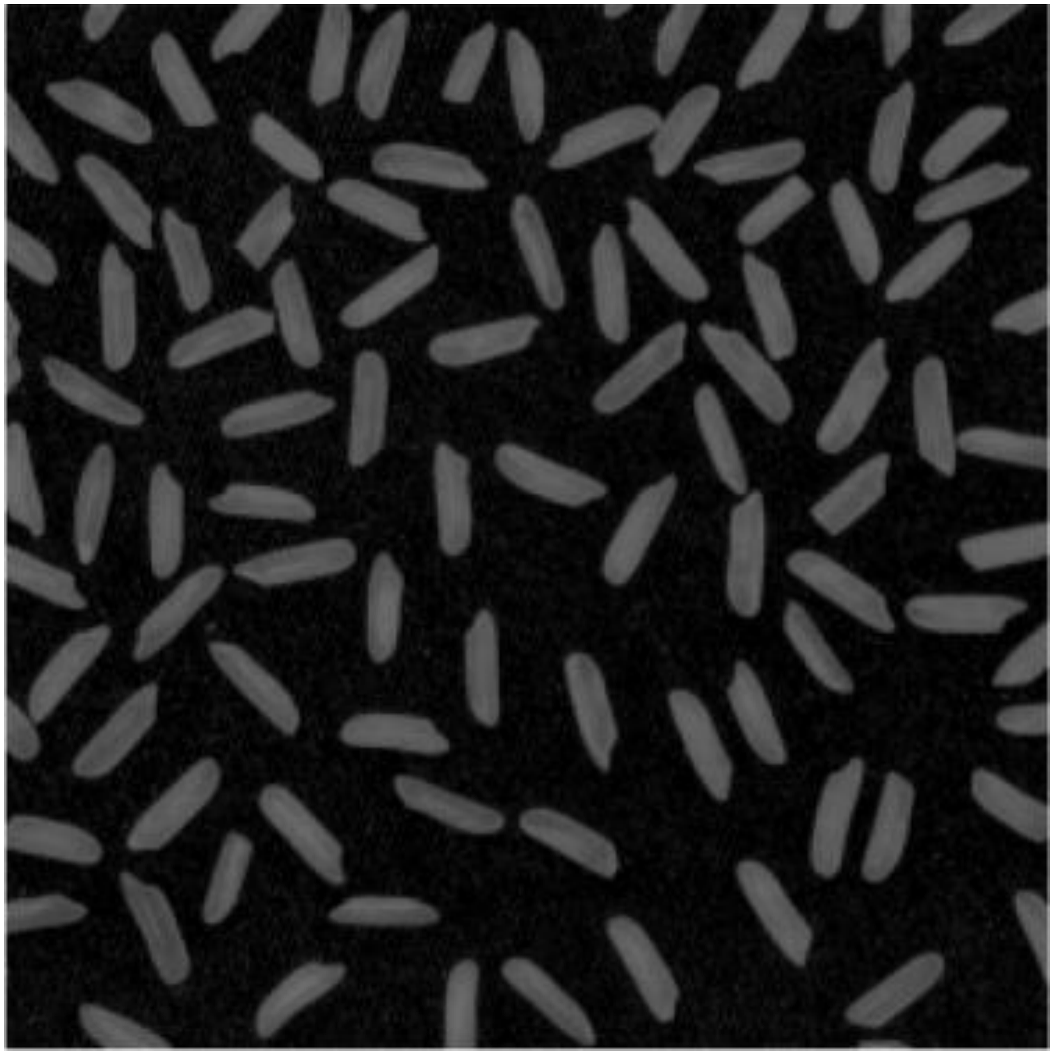
- Current label = 1, raster-scan all pixels (x, y) of f:
  - If $\mathrm{f}(x, y)$ is object and (x, y) not yet visited:
    - Initialize new empty queue q
    - Enqueue pixel (x, y)
    - While q is not empty:
      - Dequeue pixel (x', y')
      - Set labelling result at (x', y') to current label
      - If left neighbor pixel inside image, left pixel is object and not yet visited:
        - Enqueue left pixel
      - If right neighbor pixel inside image, right pixel is object and not yet visited:
        - Enqueue right pixel
      - If upper neighbor pixel inside image, upper pixel is object and not yet visited:
        - Enqueue upper pixel
      - If lower neighbor pixel inside image, lower pixel is object and not yet visited:
        - Enqueue lower pixel
    - Increase current label

Connected component labelling



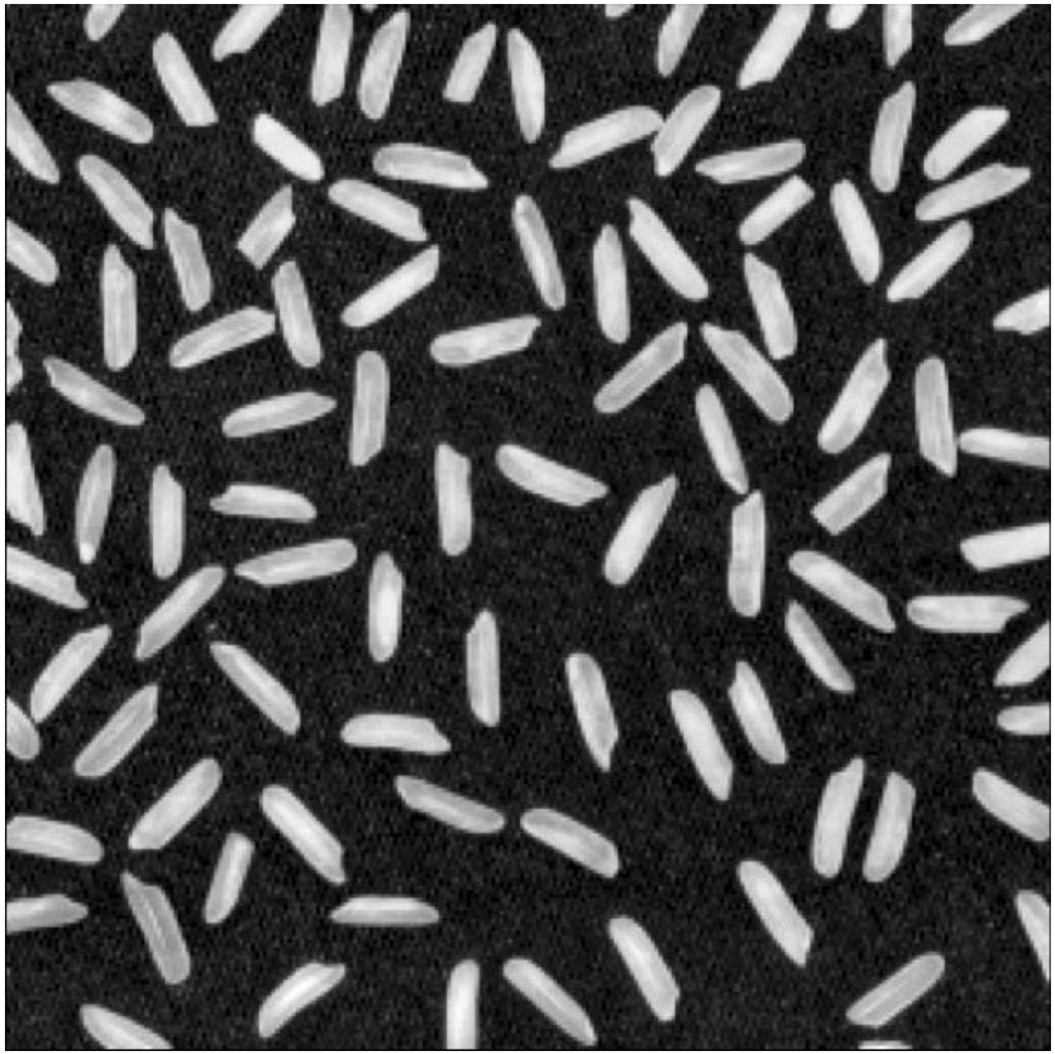
4-conn
“grassfire”

Segmentation example



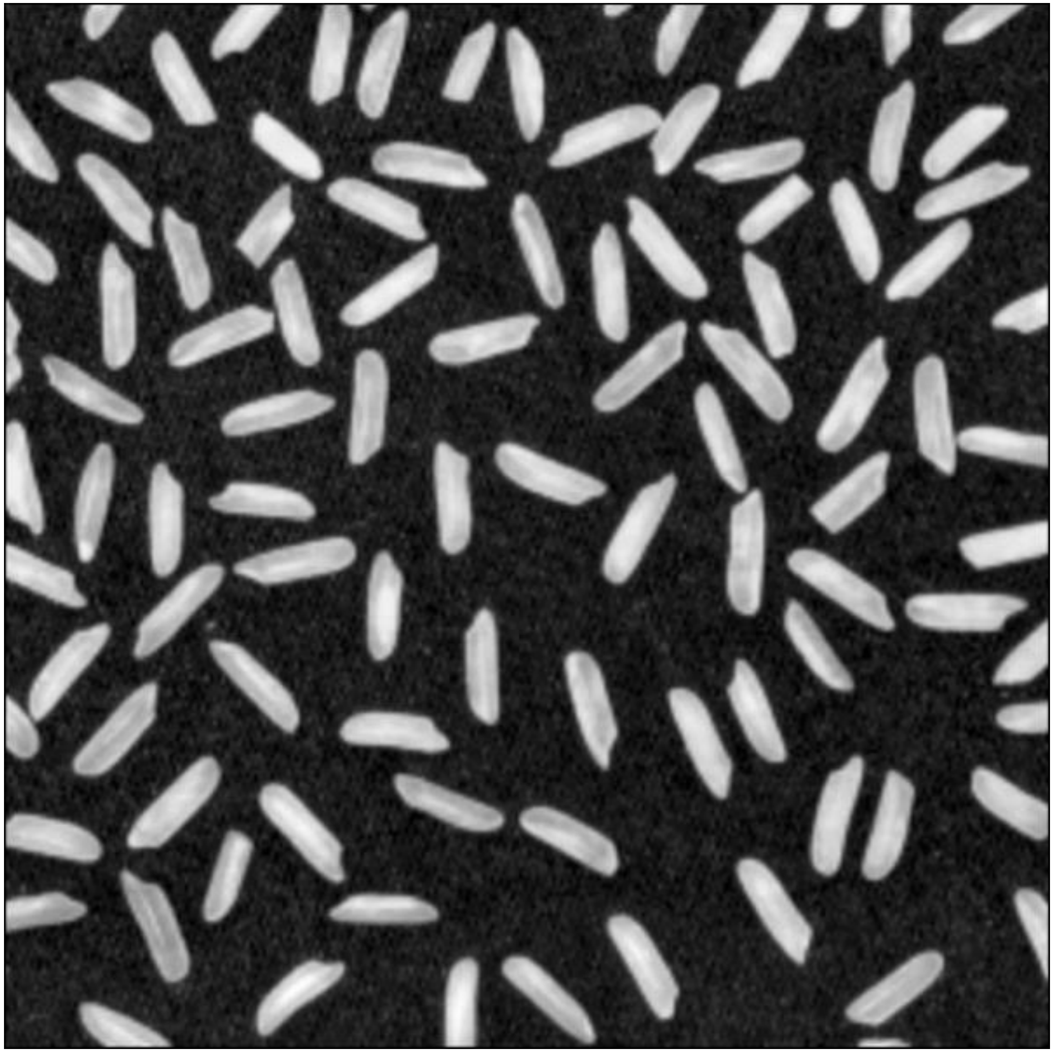
Goal: Count number of objects

Segmentation example



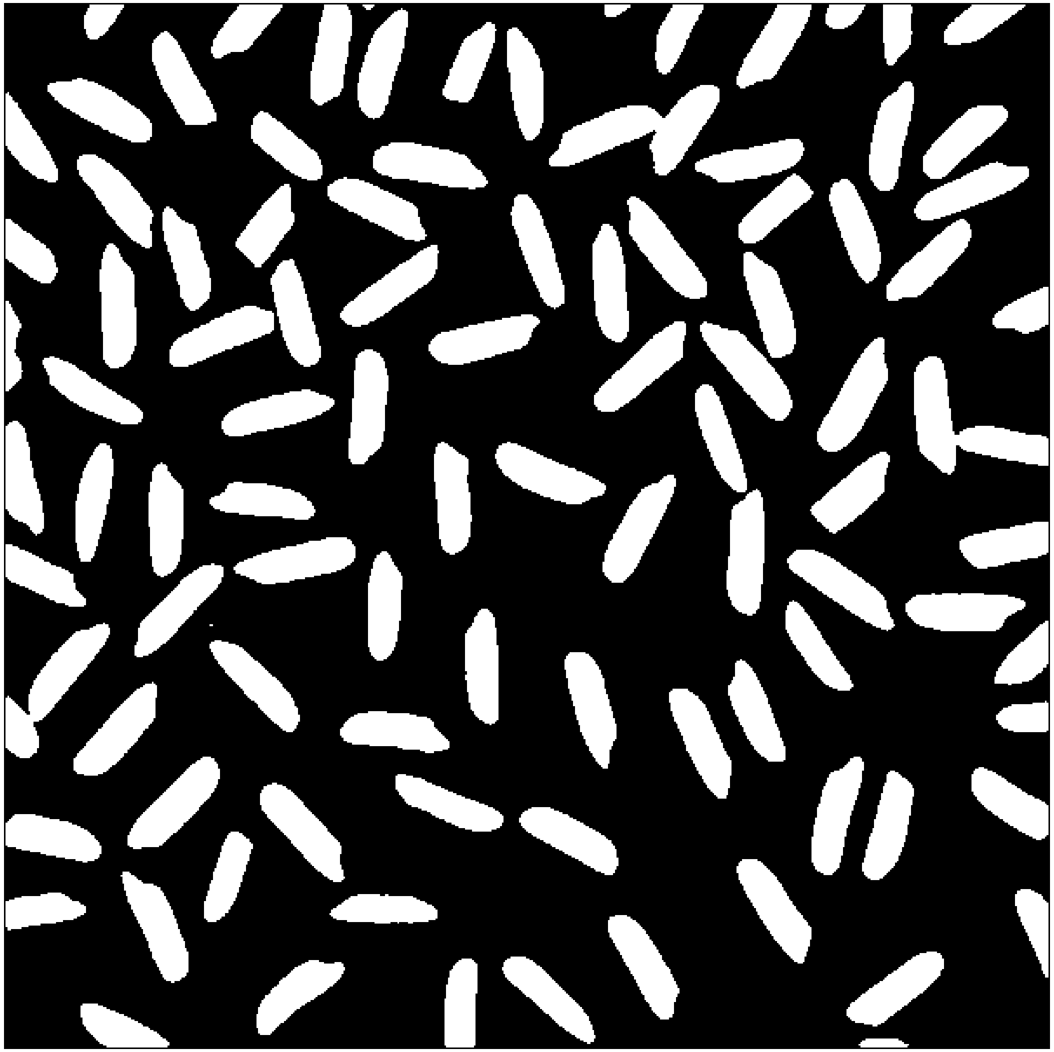
1. Contrast Stretching

Segmentation example



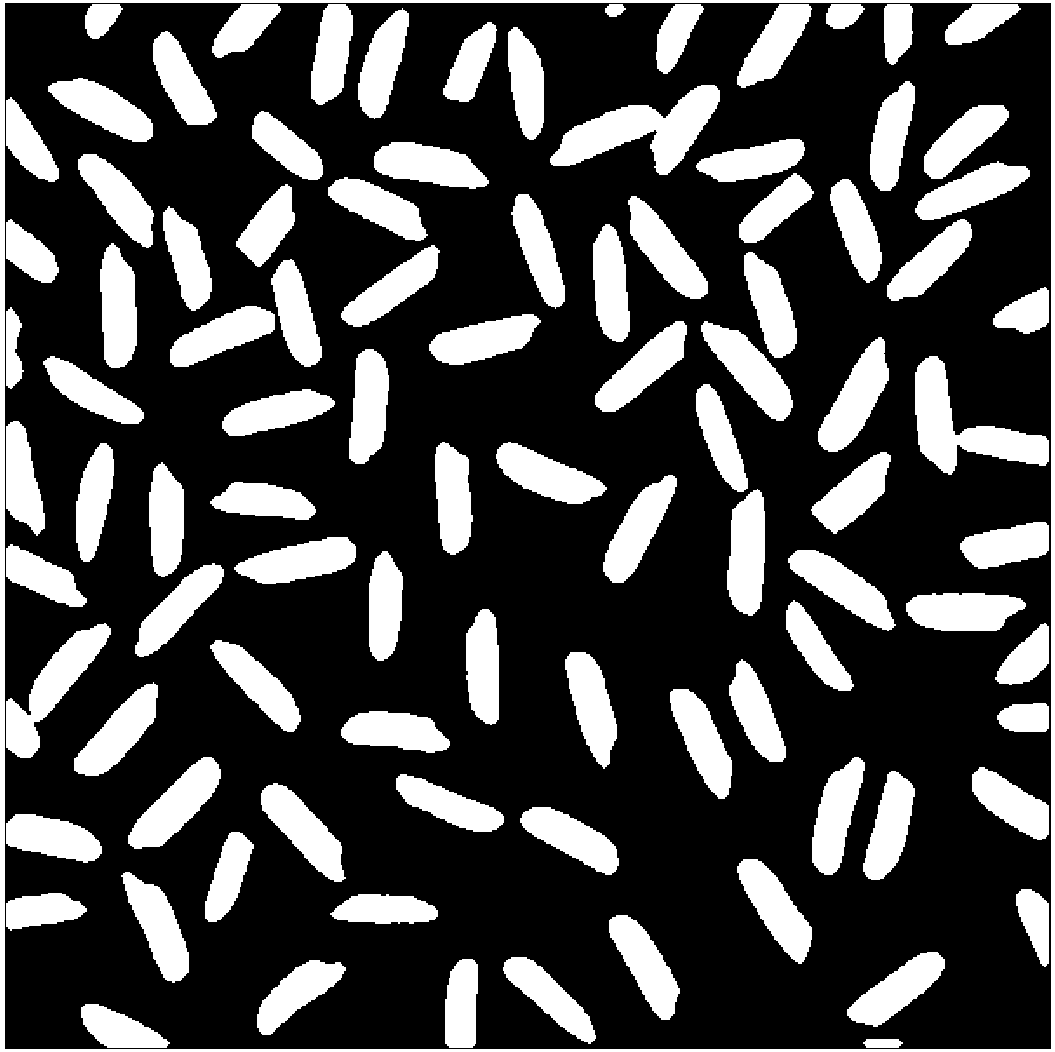
2. Smoothing with 3x3 Gaussian

Segmentation example



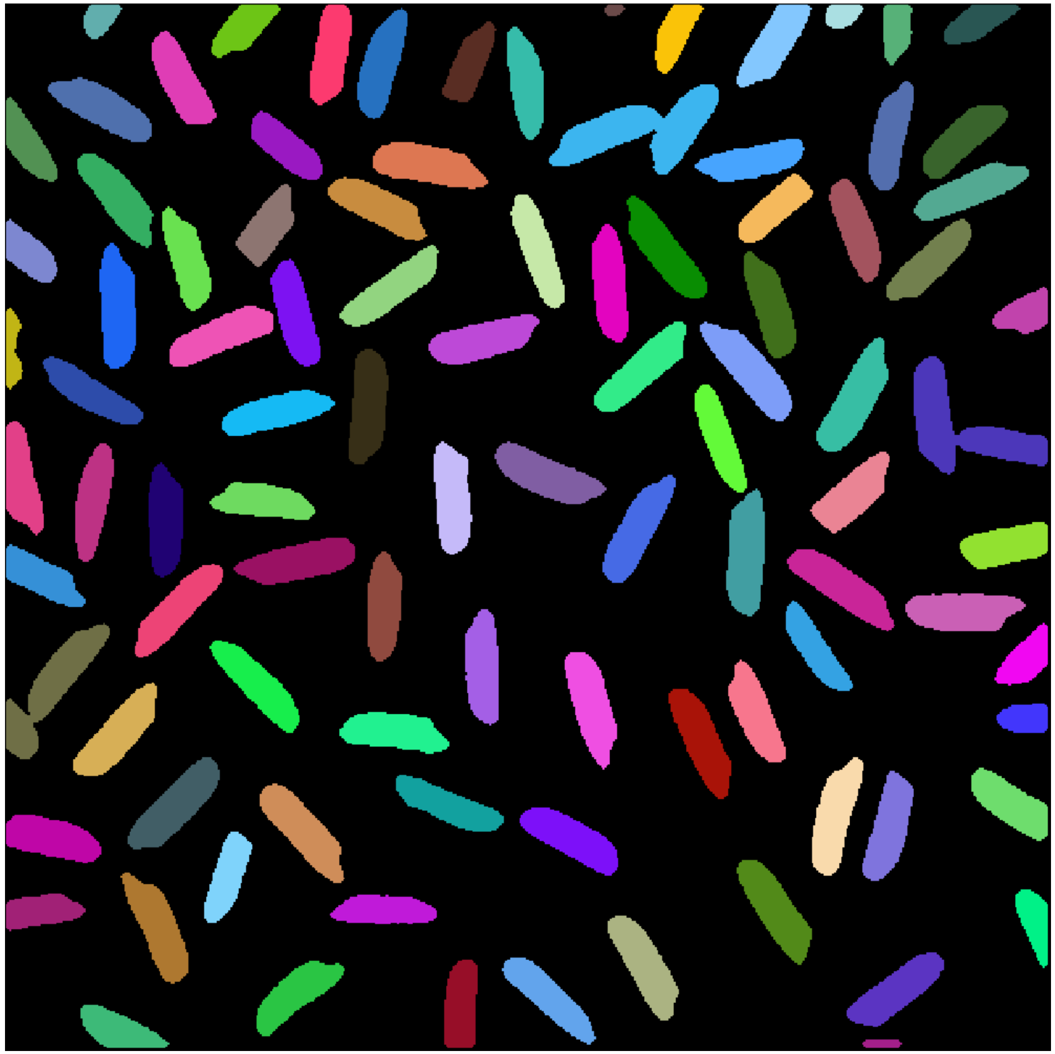
4. Automatic Adaptive Thresholding

Segmentation example



5. Morphological Postprocessing

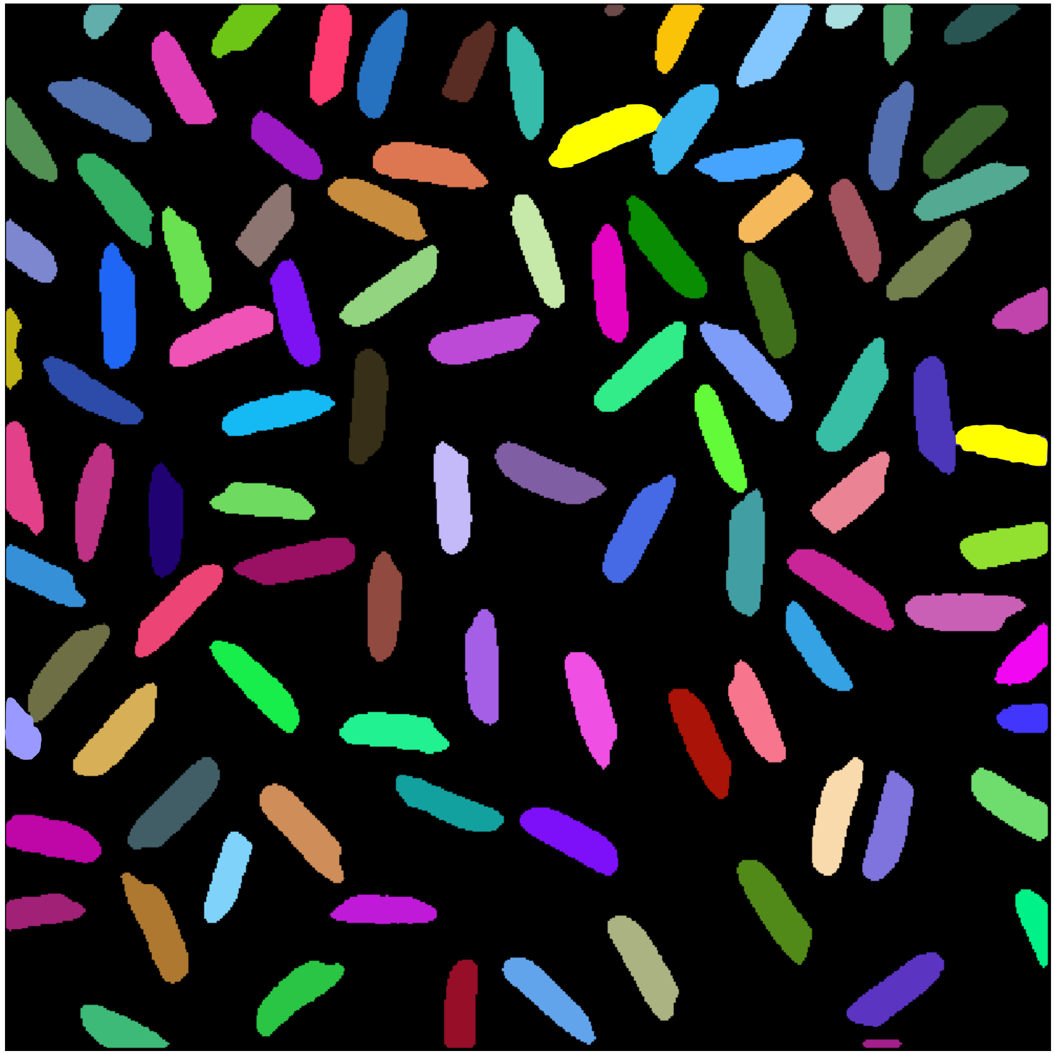
Segmentation example



97 objects
3 wrongly
merged

6. Connected Component Labelling

Outlook Watershed Segmentation

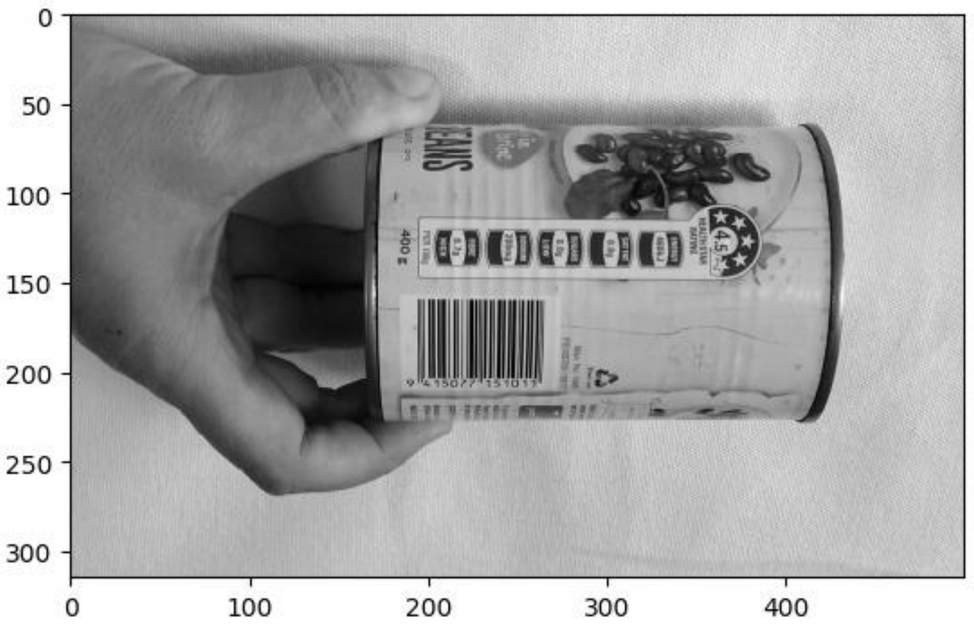


Correct
100 objects

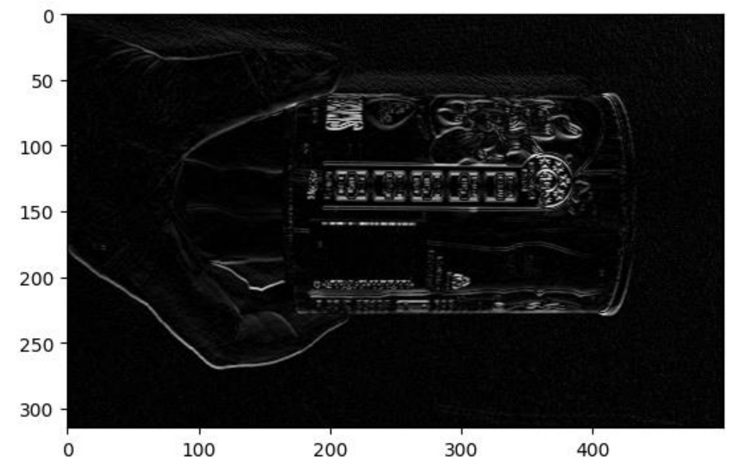
See e.g. Gonzalez & Woods, Digital Image Processing for Watershed segmentation algorithm

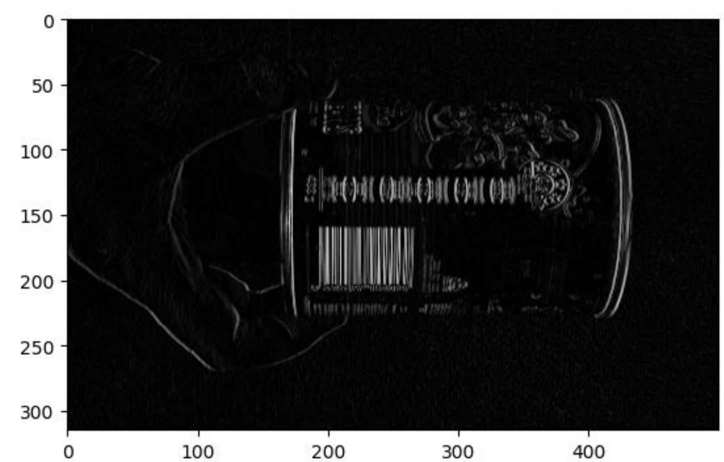
Barcode detection assignment

- Workflow (see Python code skeleton)



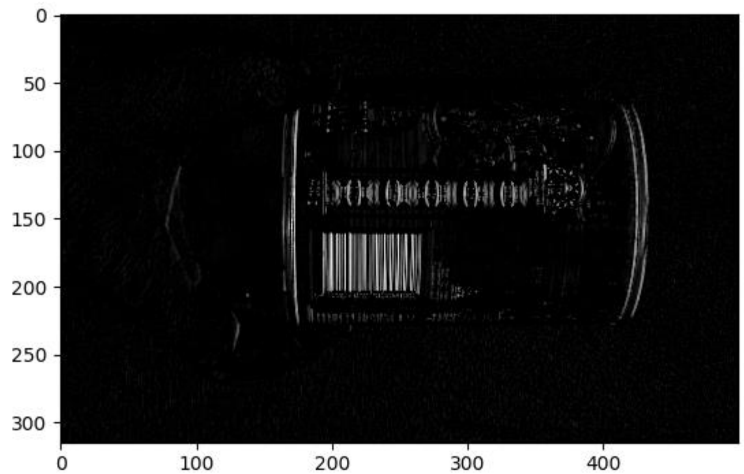
RGB color input, convert to greyscale,
stretch contrast



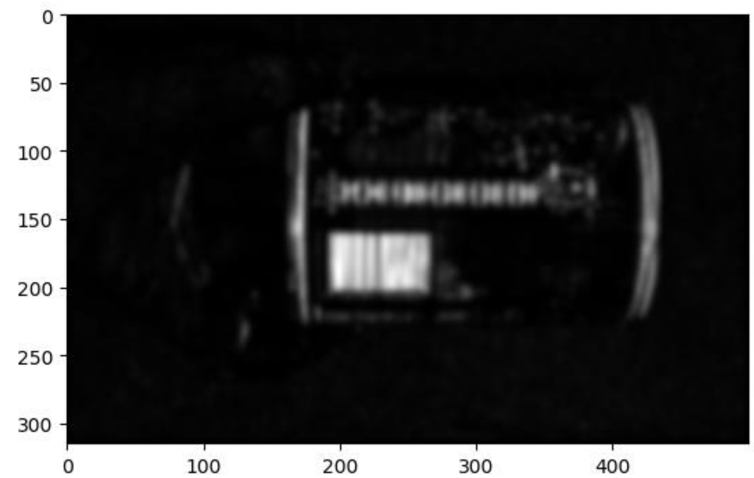


Horizontal & vertical edges (absolute, stretched)

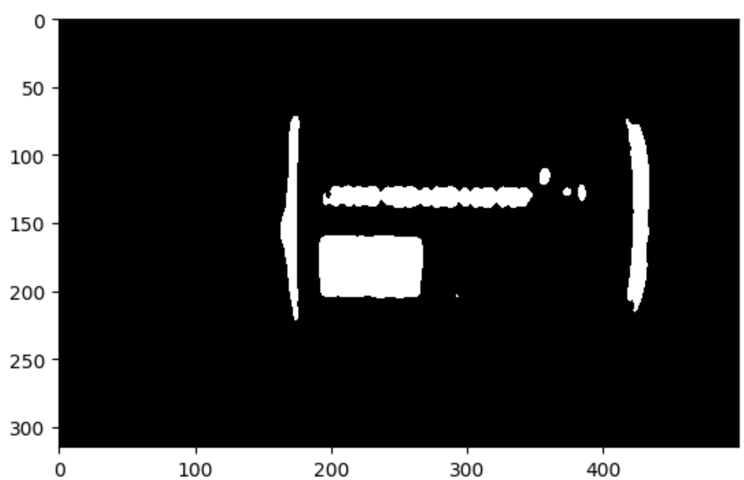
Barcode detection assignment



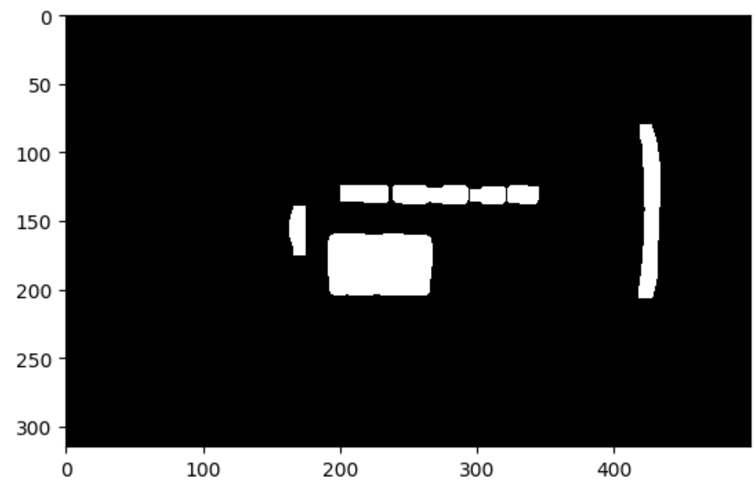
Strong vertical edges



Mean averaged edges (10 times 3x3, stretched)

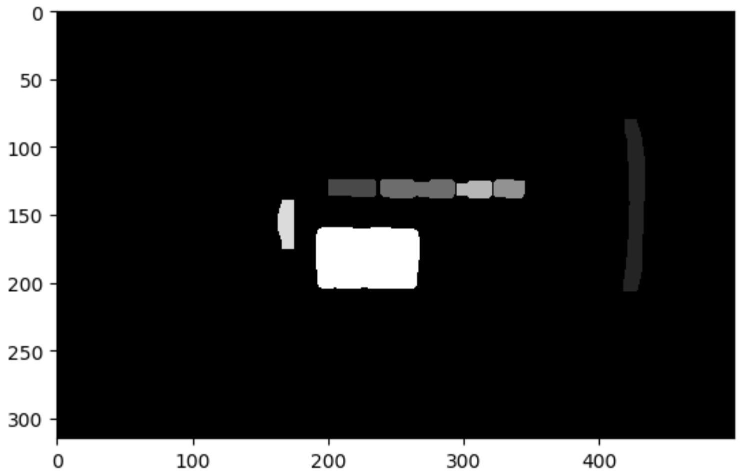


Simple threshold (intensity 70)

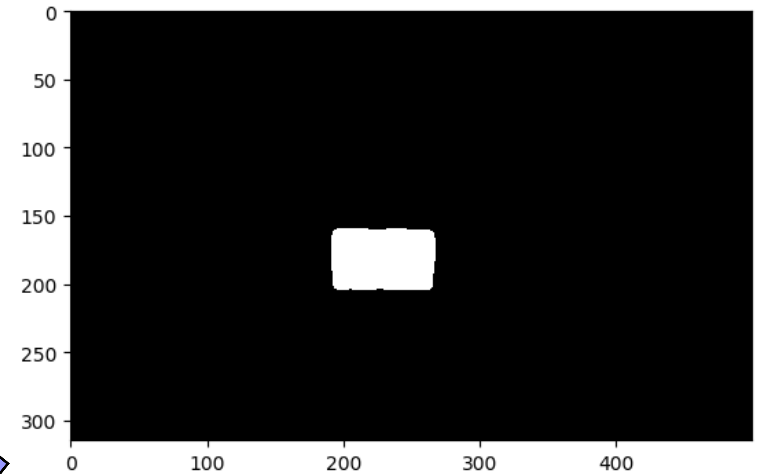


Morphological processing (4 erosions, 4 dilations)

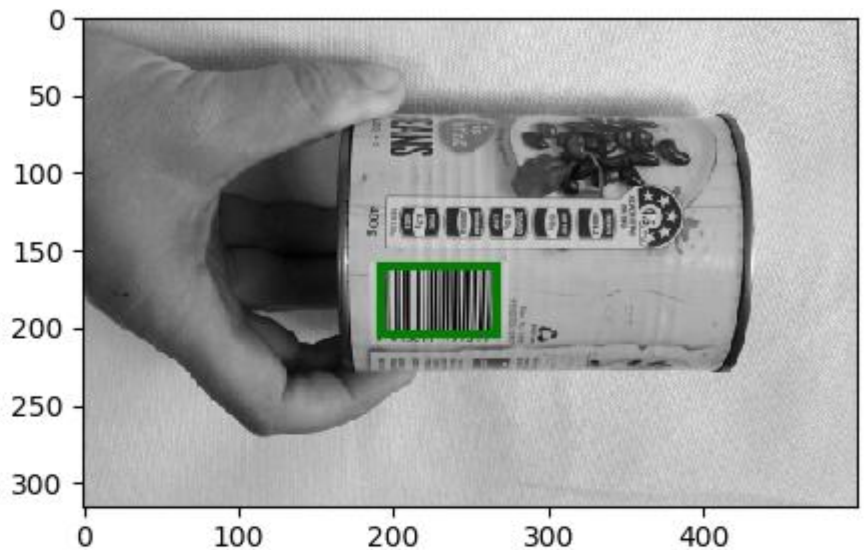
Barcode detection assignment



Connected component labelling (7 components)



Select largest connected component



Final result as rectangle around largest connected component!