# Segmentation

Raúl Alonso Calvo

# Purpose

- Segmentation partitions an image into distinct regions containing each pixels with similar properties
- Regions should strongly relate to depicted objects or features of interest
  - For interpreting correctly an image, a meaningful segmentation transforming image into terms of features, objects, and scenes

The success of image analysis depends on reliability of segmentation, but an accurate partitioning of an image is generally a very challenging problem

# Image segmentation

- A complete segmentation must satisfy:
  1. Completeness:
     - All pixels have to be assigned to regions
  2. Disjointness:
     - Each pixel has to belong to a single region only
  3. Connectedness:
     - Each region is a connected set of pixels
  4. Satisfiability:
     - Each region has to be uniform with respect to a given predicate
  5. Segmentability:
     - Any merged pair of adjacent regions has to be non-uniform

# Previous lessons

- Basic concepts from Image analysis and Mathematical Morphology
- Erosion, Dilation, Opening, Closing, …
- How can we obtain the contours of shapes using this basic operators?
- But this contours are not really image segmentation

# boofCV functions

- Erosion - generally decreases the sizes of objects and removes small anomalies by subtracting objects with a radius smaller than the structuring element.

erode4

```
public static GrayU8 erode4(GrayU8 input,
                            int numTimes,
                            GrayU8 output)
```

Parameters:

input - Input image. Not modified.

numTimes - How many times the operation will be applied to the image.

output - If not null, the output image. If null a new image is declared and returned. Modified.

# boofCV functions

- Dilation - generally increases the sizes of objects, filling in holes and broken areas, and connecting areas that are separated by spaces smaller than the size of the structuring element.

```
public static GrayU8 dilate4(GrayU8 input,
                             int numTimes,
                             GrayU8 output)
```

Parameters:
input - Input image. Not modified.
numTimes - How many times the operation will be applied to the image.
output - If not null, the output image. If null a new image is declared and returned. Modified.

# Boundary Extraction

- $I' = I - \varepsilon_B(I)$

# Practice 1

- Boundary extraction
  - $I' = I - \varepsilon_B ( I )$

Auxiliary functions:

PixelMath:

subtract(GrayU8 imgA, GrayU8 imgB, GrayI16 output)

Performs pixel-wise subtraction.
output(x,y) = imgA(x,y) - imgB(x,y)

# Region growing

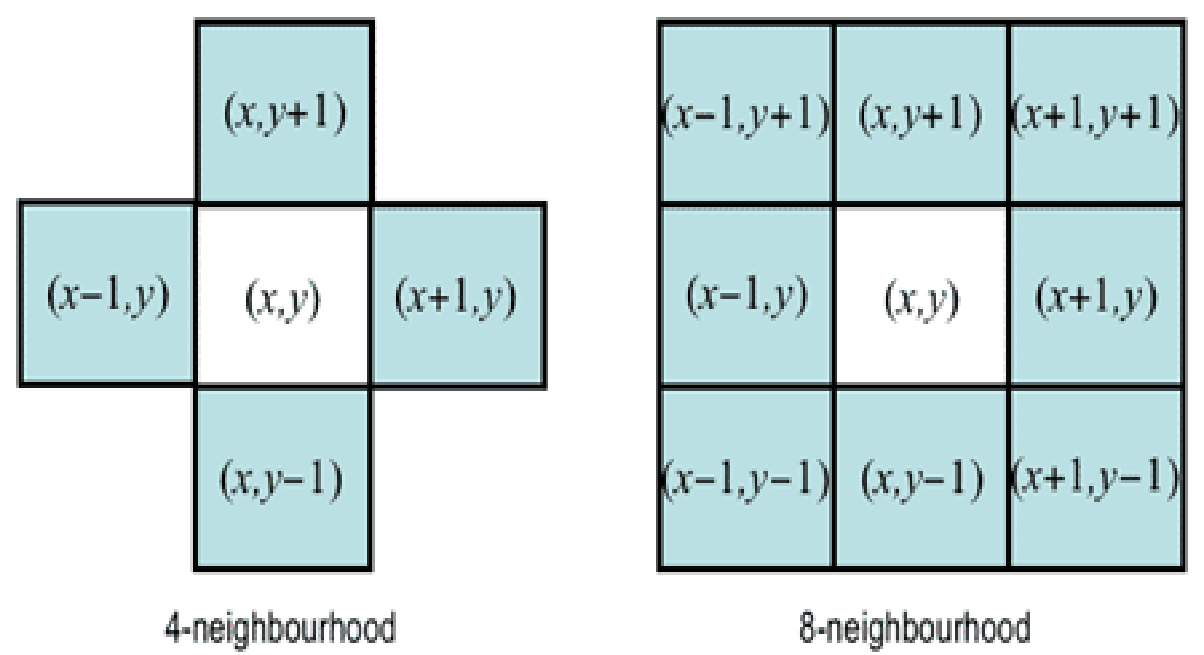
- Contextual segmentation

- Discontinuity-based techniques attempt to find complete boundaries enclosing relatively uniform regions assuming abrupt signal changes across each boundary

- Similarity-based techniques attempt to directly create these uniform regions by grouping together connected pixels that satisfy certain similarity criteria

# Pixel connectivity

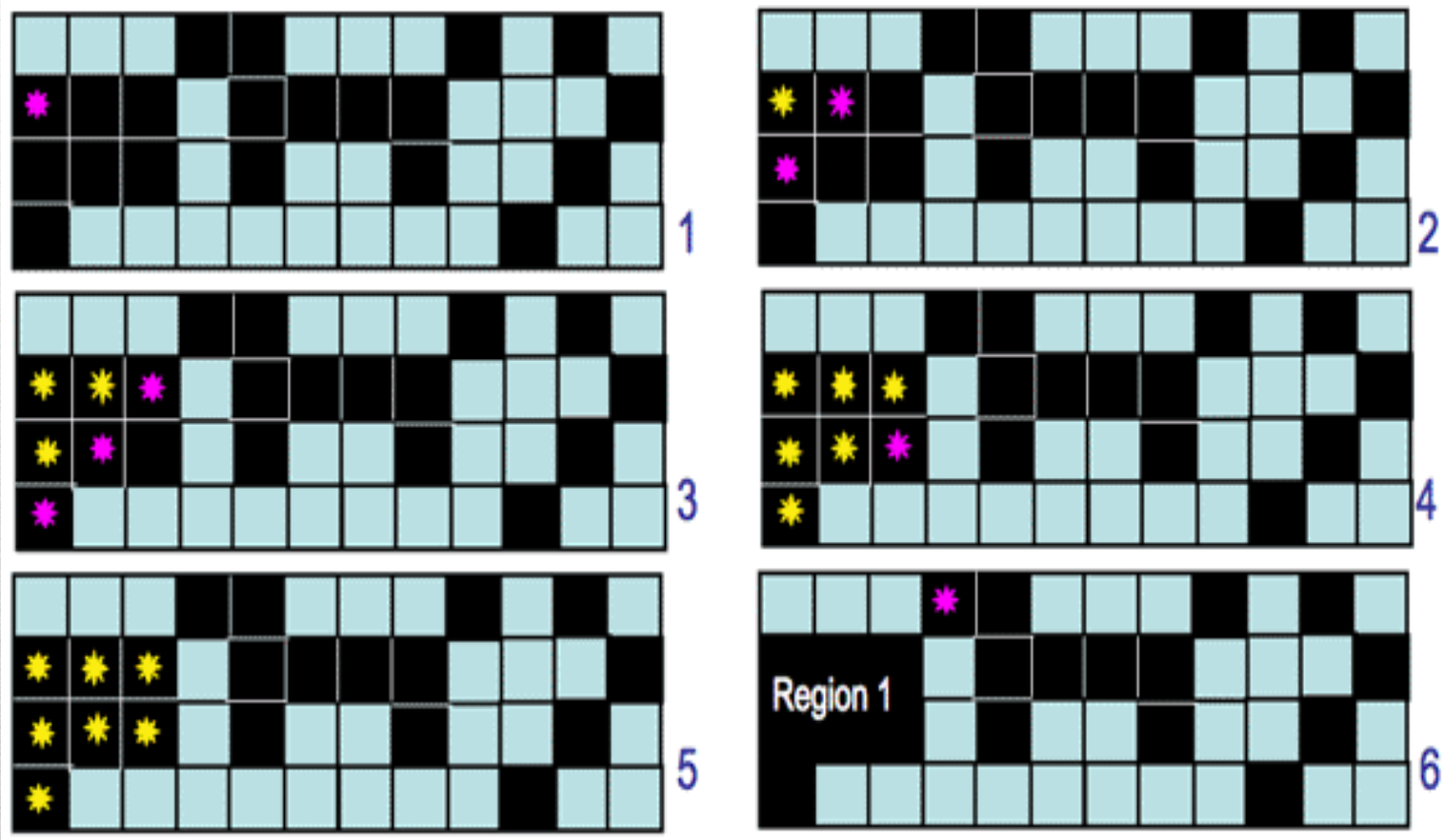
- Context - is defined in terms of pixel neighborhoods



# Region growing

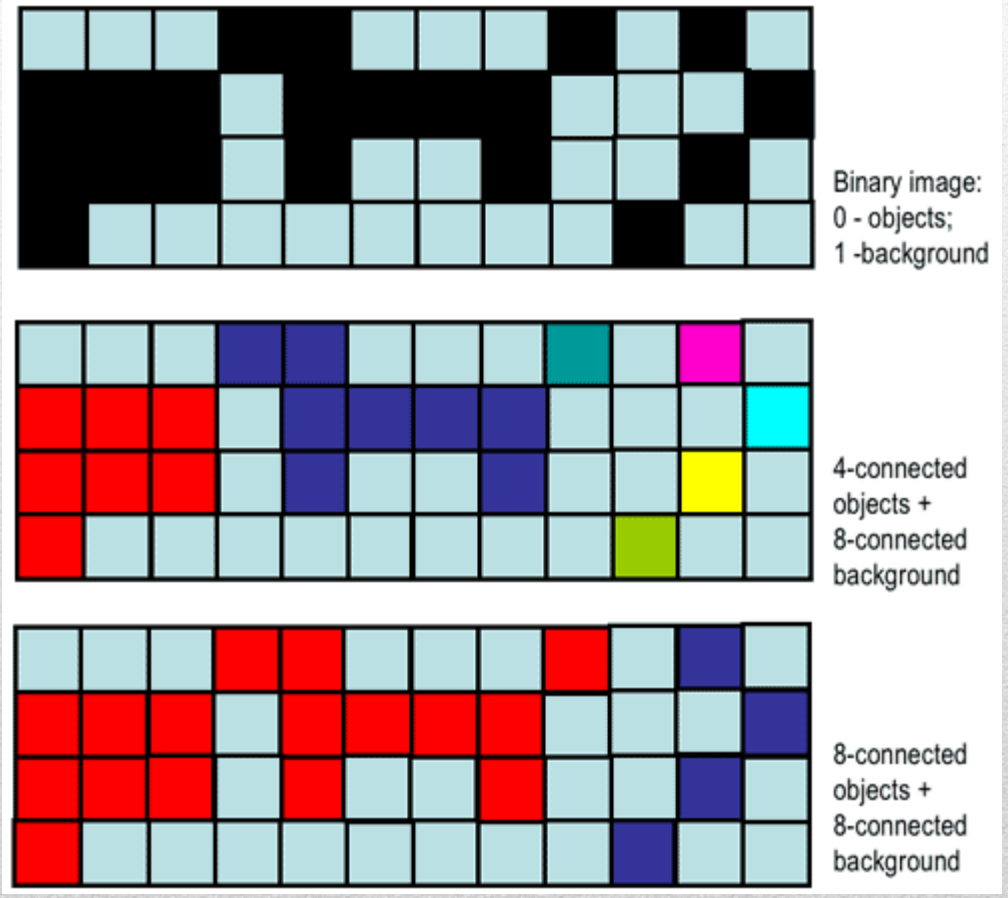
- Simplest algorithms for labeling connected regions: the "grassfire" or "wave propagation"
- "fire" or "wave" starts at one pixel, it propagates to any of the pixel's 4- or 8-neighbours detected by thresholding
- Each already visited (i.e. "burnt away" or "wet") pixel cannot be visited again
- After the entire connected region is labeled, its pixels are assigned a region number

# Region growing



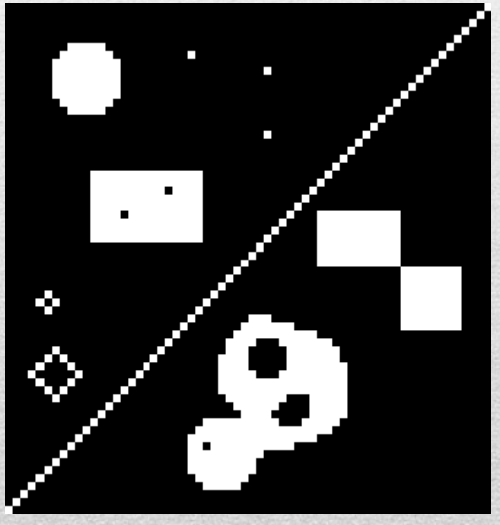
# Region growing

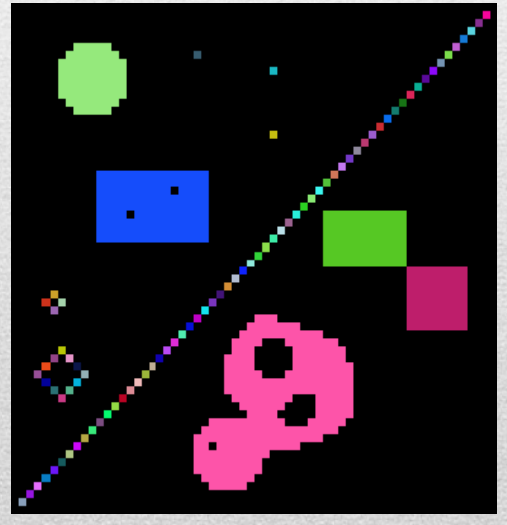
- Different structuring element produces different results

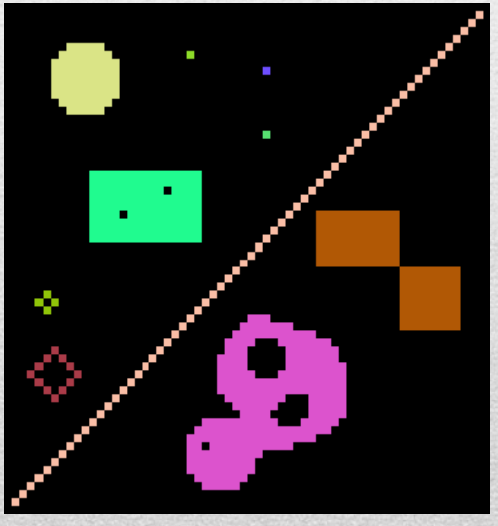


# Region growing

- In many practical cases the connectivity is defined variously for objects (foreground pixels) and background
  - Avoid line splitting
  - Avoid joining close regions







# Practice 2

- Implement grassfire algorith using boofCV
- Allowing 4 / 8 connectivity with a given size

Size n implies a square 2*n+1

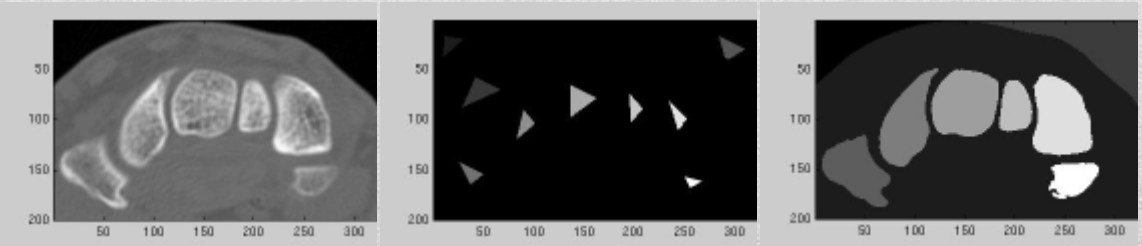
- Encapsulate propagation condition in a method

# Region similarity

- The uniformity or non-uniformity of pixels is represented by a **uniformity predicate**
- Usually a logical statement with respect to some property: color, grey level, edge strength, etc
- Common predicate restricts signal variations over a neighborhood

# Region growing

- The bottom-up **region growing** algorithm starts from a set of seed pixels defined by the user
- Sequentially more pixels are added to a region provided that the pixel has not been assigned to any other region
- Very sensitive to a chosen uniformity predicate



Greyscale image Seed regions Region growing results

(http://www.lems.brown.edu/~msj/cs292/assign5/segment.html)

# Image growing segmentation

- A complete segmentation must satisfy:

1. All pixels have to be assigned to regions
2. Each pixel has to belong to a single region only
3. **Each region is a connected set of pixels**
4. **Each region has to be uniform with respect to a given predicate**
5. Any merged pair of adjacent regions has to be non-uniform