



# Segmentation (3): region-based

---

Reading: section 6.3 of chapter 6.

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/region.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/region.pdf)



# Outline

---

- Introduction of the concept
  - Region growing by pixel aggregation
  - Region merging
  - Region splitting
  - Split and merge
- 
- We will study the watershed transform in the chapter about mathematical morphology



# Region-based segmentation

---

- Goal: find coherent (homogeneous) regions in the image
  - Coherent regions contain pixels which share some similar property
- Advantages
  - Region-based techniques are generally better in noisy images (where borders are difficult to detect)
- Drawbacks:
- The output of region-growing techniques is either **oversegmented** (too many regions) or **undersegmented** (too few regions)
- Can't find objects that span multiple disconnected regions



# Basic formulation of the problem

---

- A complete segmentation of an image  $R$  is a finite set of regions  $R_1, R_2, \dots, R_S$  such that

$$R = \bigcup_{i=1}^S R_i \text{ and } R_i \cap R_j = \Phi, i \neq j$$

- Region-based segmentation verifies a homogeneity criterion

$$H(R_i) = \text{true} \quad i = 1, 2, \dots, S$$

$$H(R_i \cup R_j) = \text{false} \quad \text{if } i \neq j, R_i \text{ adjacent to } R_j$$



# Region homogeneity

---

- Examples of homogeneity criteria for region  $R$ 
  - diff. between max and min grey-values in  $R$  is small
  - diff. between any pixel and mean grey-value in  $R$  is small
  - variance of grey-values in  $R$  is small
- Segmentation depends on
  - properties used
  - measure of similarity between properties
  - similarity variation tolerance (threshold)

# Connectivity in digital images

- Depends on how many neighbours are considered connected to pixel
  - 8-connectivity: all 8 neighbours
  - 4-connectivity: only 4 neighbours (vertical, horizontal)
- Pixels  $p$  in  $S$  and  $q$  in  $S$  are connected in set  $S$  if
  - there is a sequence of pixels  $p_0 = p; p_1; \dots; p_n = q$ ,  $p_i$  in  $S$ , that connects  $p$  to  $q$  so that

$p_i$  is neighbour of  $p_{i-1}$ ,  $i = 1..n$

- Set  $S$  is a **connected region** if all its pixels are connected



Connected pixels and sets.  
Set 2 is only connected for 8-connectivity.



## Region growing by pixel aggregation

---

- Start from one seed pixel  $p$  located inside region  $R$ .
- Define a similarity measure  $S(i; j)$  for all pixels  $i$  and  $j$  in the image.
- Add adjacent pixel  $q$  to pixel  $p$ 's region iff  $S(p; q) > T$  for some threshold  $T$ .
- Evaluate the other neighbors of  $p$  as above.
- We can now consider  $q$  as a new seed
- Continue until all pixels in the currently investigated neighbourhood do not satisfy the inclusion criteria.



# Example of pixel aggregation

---

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

image, 2 seeds

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

result for  $T = 4$

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

result for  $T = 8$

Homogeneity criterion: maximum allowed absolute difference  $T$  within region



## Design of a region growing algorithm

---

- How do we choose the seed pixel?
- How do we define the similarity measure  $S$ ?
- Choice of the threshold  $T$  (variable or fixed?)
- When we evaluate  $q$ 's neighbor  $r$ , should we use  $S(p; r)$  or  $S(q; r)$ ?

Adapted from

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/region.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/region.pdf)



## Comparing to original seed pixel

---

- $S(p,r)$
- Advantage: uses a single basis of comparison across all pixels in the region
- Drawback: the region produced is very sensitive to the choice of the seed pixel

Adapted from

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/region.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/region.pdf)



## Compare to neighbour in region

---

- $S(q,r)$
- Advantage: it respects the transitivity of the similarity relationship
- Disadvantage: drift as one grows farther away from the seed pixel

Adapted from

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/region.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/region.pdf)



## Compare to region statistics

---

- Compare candidate  $r$  to the entire region already collected
- Example: compare  $r$  to the average property (i.e. intensity) of all pixels in the region.
- Mean is updated after each aggregation

Adapted from

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/region.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/region.pdf)



## Other techniques

---

- Multiple seeds
- Cumulative differences
- Counterexamples

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MORSE/region.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/region.pdf)

# Region growing example

---



Gray scale lightning image: segment the lightning

# Region growing example

---



Seed points:  $I=255$



T: 125-255



T: 190-255



T: 155-255



# Region merging algorithms

---

- START with an oversegmented image
- Define a criterion for merging two adjacent regions
- Merge all adjacent regions satisfying the merging criterion
- STOP when no two regions can be merged



**Algorithm 6.18: Region merging via boundary melting**

1. Define a starting image segmentation into regions of constant gray-level. Construct a supergrid edge data structure in which to store the crack edge information.
2. Remove all weak crack edges from the edge data structure (using equation (6.32) and threshold  $T_1$ ).
3. Recursively remove common boundaries of adjacent regions  $R_i, R_j$ , if

$$\frac{W}{\min(l_i, l_j)} \geq T_2 ,$$

where  $W$  is the number of weak edges on the common boundary,  $l_i, l_j$  are the perimeter lengths of regions  $R_i, R_j$ , and  $T_2$  is another preset threshold.

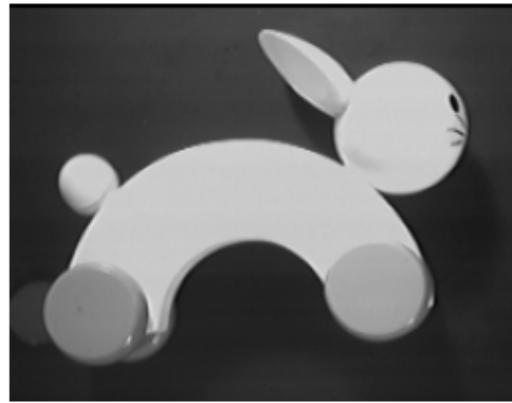
4. Recursively remove common boundaries of adjacent regions  $R_i, R_j$  if

$$\frac{W}{l} \geq T_3 \tag{6.33}$$

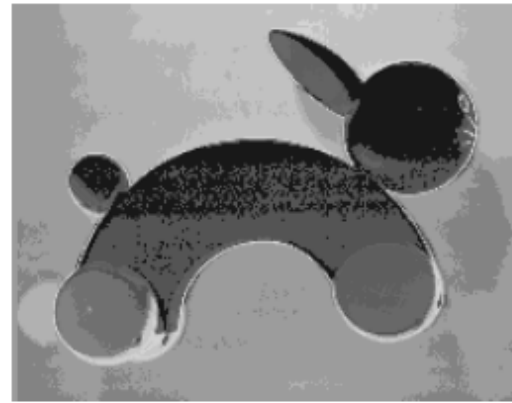
or, using a weaker criterion [Ballard and Brown, 1982]

$$W \geq T_3 , \tag{6.34}$$

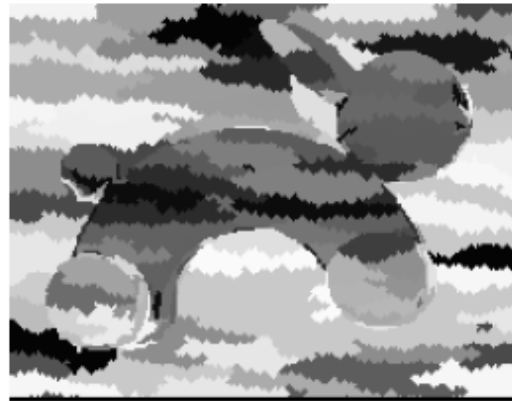
where  $l$  is the length of the common boundary and  $T_3$  is a third threshold.



(a)



(b)



(c)



(d)

**Figure 6.43:** Region merging segmentation. (a) Original image. (b) Pseudo-color representation of the original image (in grayscale). (c) Recursive region merging. (d) Region merging via boundary melting. *Courtesy of R. Marik, Czech Technical University.*



## Region splitting

---

- Opposite of region merging
- Begins with the whole image represented as a single region which does not verify the homogeneity condition



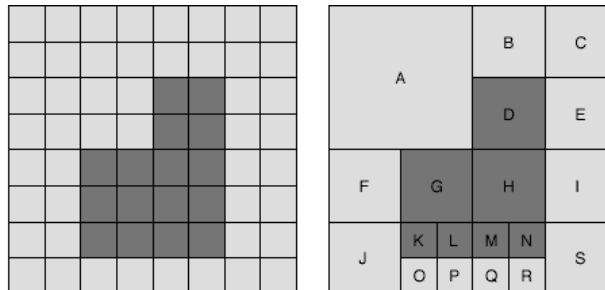
# Split/Merge

---

- If a region  $R$  is inhomogeneous ( $P(R) = \text{False}$ ) then is split into four sub regions
- If two adjacent regions  $R_i, R_j$  are homogeneous ( $P(R_i \cup R_j) = \text{TRUE}$ ), they are merged
- The algorithm stops when no further splitting or merging is possible

# Split and merge algorithm

- START: consider entire image as one region
  1. If region satisfies homogeneity criteria, leave it unmodified
  2. If not, split it into four quadrants and recursively apply 1 and 2 to each newly generated region  
STOP when all regions in the quadtree satisfy the homogeneity criterion
  3. If any two adjacent regions  $R_i, R_j$  can be merged into a homogeneous region, merge them.  
STOP when no merging is possible any more.



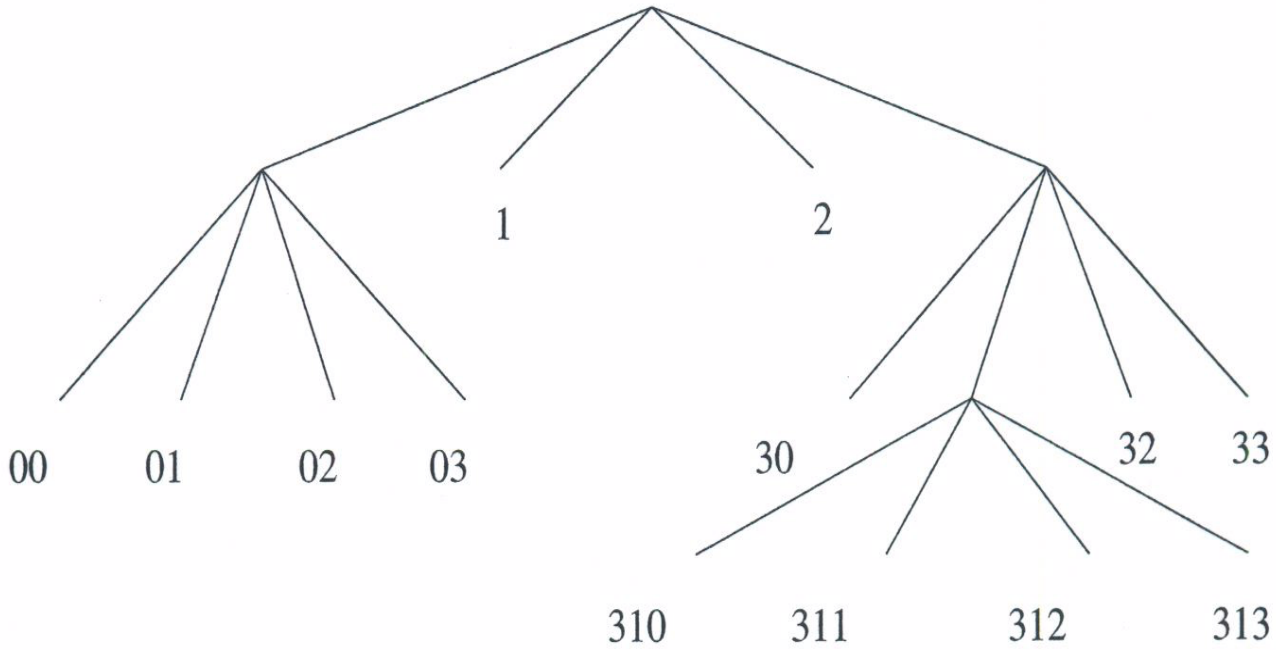
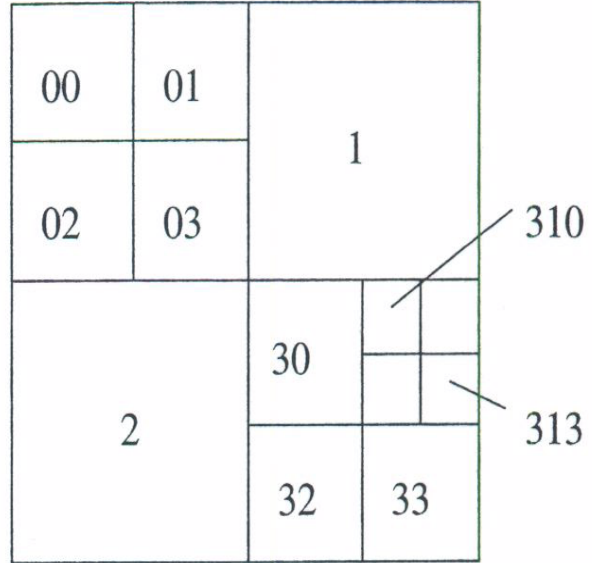
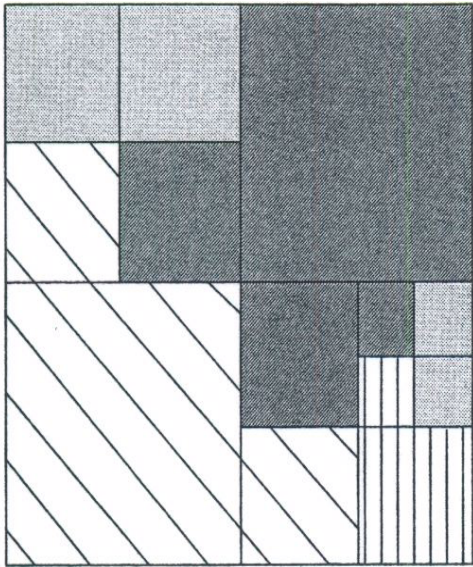


Figure 6.46: Segmentation quadtree.

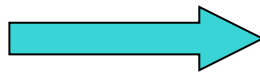
# Results – Region grow

---



# Results – Region Split

---



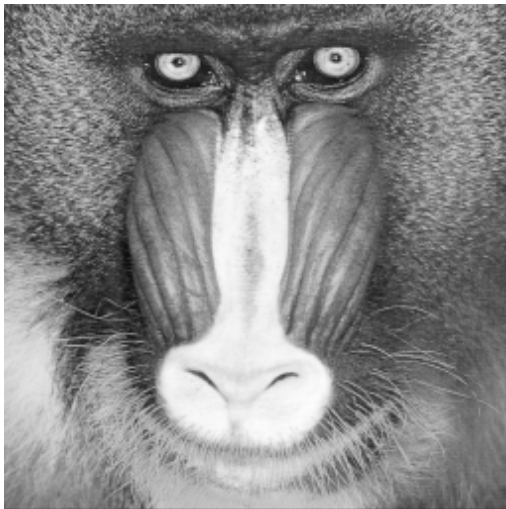
# Results – Region Split and Merge

---



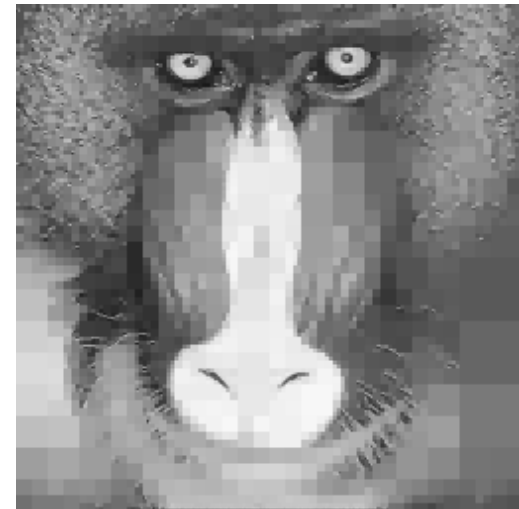
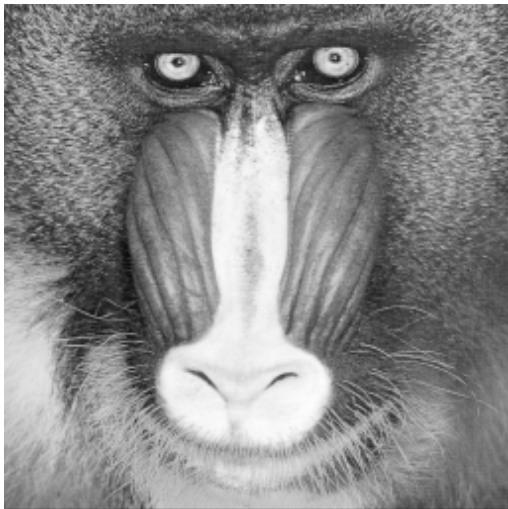
# Results – Region growing

---



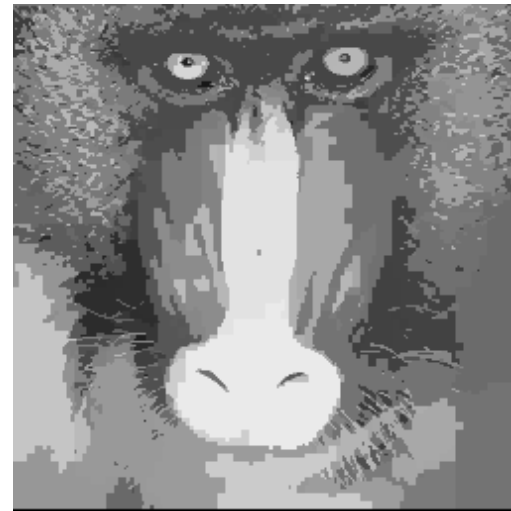
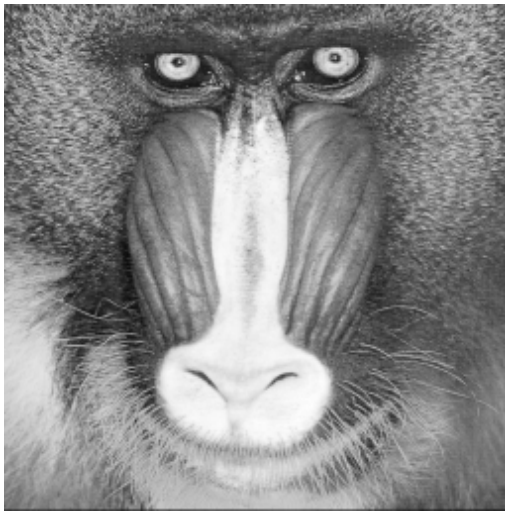
# Results – Region Split

---



# Results – Region Split and Merge

---



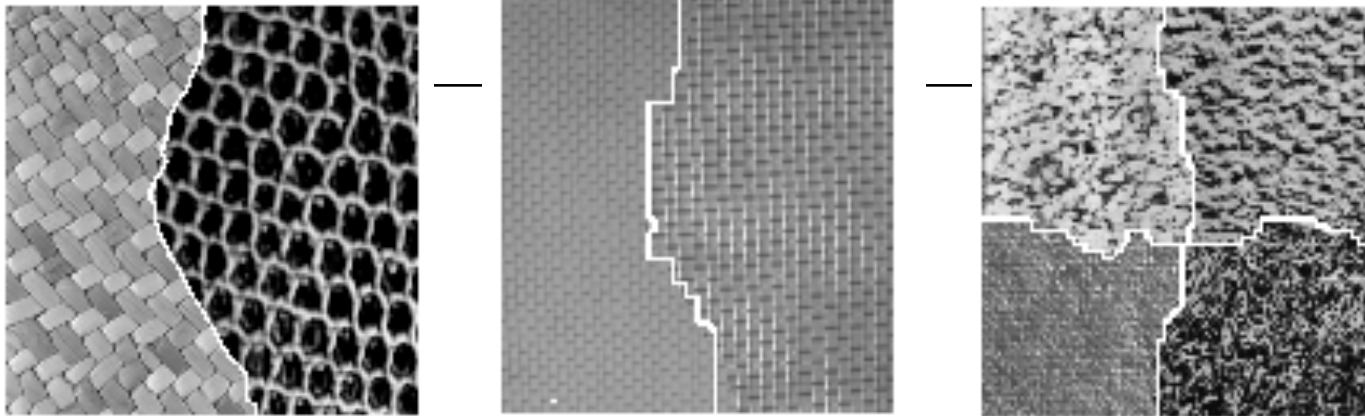


# Summary

---

- Region growing should satisfy the conditions of complete segmentation and the maximum region homogeneity conditions
- Three basic approaches:
  - Pixel aggregation : good for background subtraction
  - Region merging
  - Region splitting
  - Split and merge
  - Postprocessing is often necessary to remove small regions in the segmented image

# Hard Problem: Textures



Similarity measure makes the difference