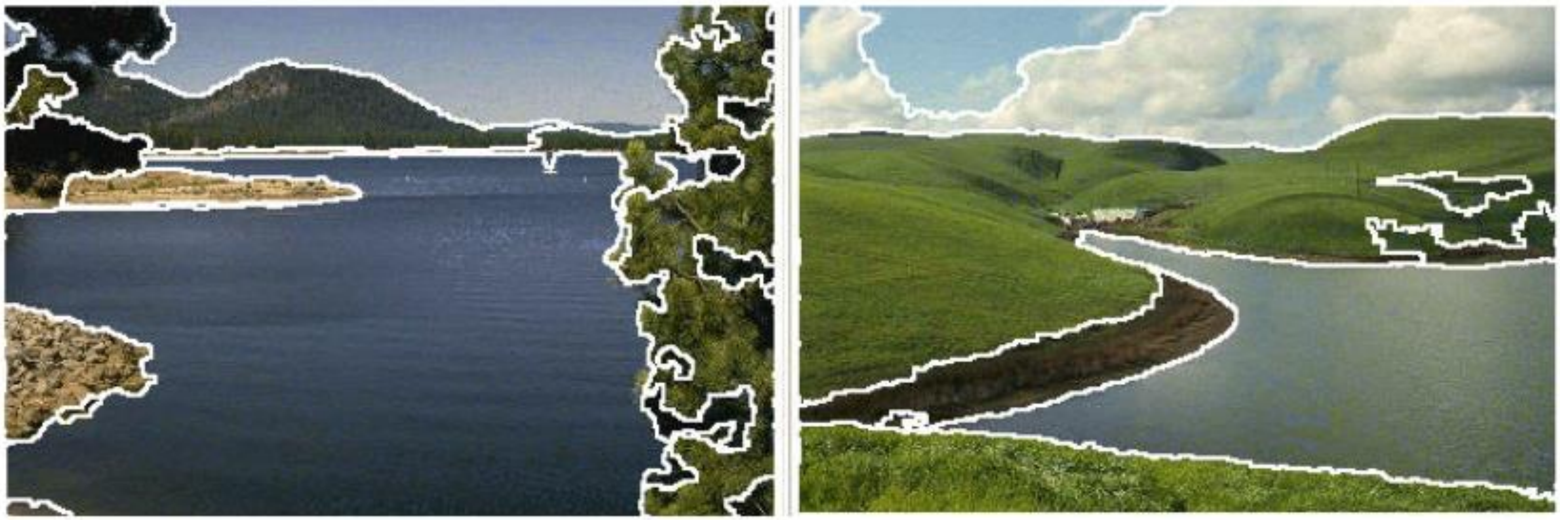
Image Segmentation

Edge segmentation:



Region segmentation:

What is a region?

Definition:

- A group of connected pixels with similar properties.

Image interpretation:

- ´Region` is an important concept in interpreting an image because regions *may* correspond to objects in a scene.
- Consequently, for a correct interpretation of an image, we need to partition an image into regions that correspond to objects or parts of an object.
- Partitioning into regions done often by using gray values of the image pixels.

Region-based segmentation

Definition:

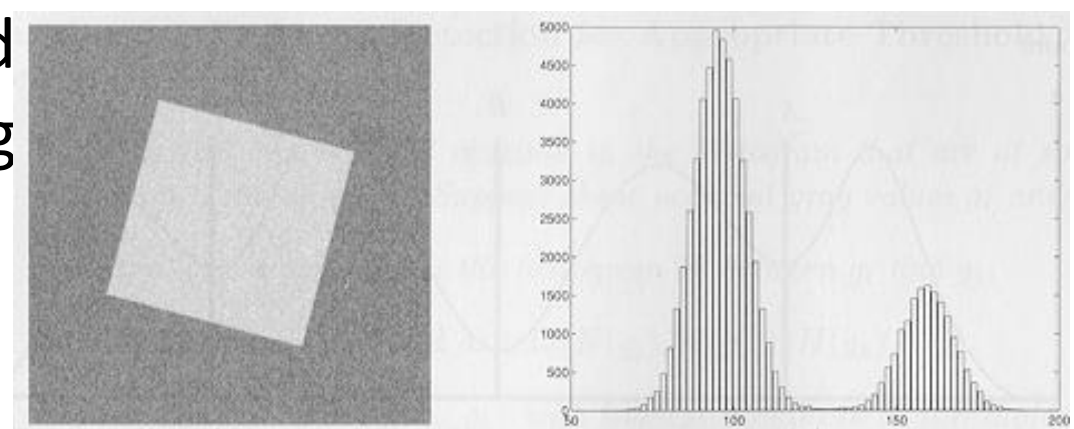
- As any other type of image segmentation, its main goal is to partition an image *I* into regions R_i.
- Unlike edge-based segmentation, which returns boundaries between regions, region-based segmentation is a technique that allows us to determine the regions directly.

Formulation:

- *Completeness*. The segmentation must be complete, i.e., every pixel must be in a region: $\bigcup_{i=1}^{n} R_i = I$
- *Connectedness*. The points of a region must be connected in some sense.
- *Disjointness*. Regions must be disjoint: $R_i \bigcap R_j = \varnothing \quad \forall i = 1,2,\ldots n$
- *Satisfiability*. Pixels of a region must satisfy one common property *P* at least, i.e., any region must satisfy a homogeneity predicate P. $P(R_i) = TRUE, \quad \forall i$
- *Segmentability.* Different regions satisfy different properties, i.e., any two adjacent regions cannot be merged into a single region. $P\left(R_i \bigcup R_j\right) = FALSE$

Region Growing (cont'd)

- **How do we choose the seed(s) in practice ?**
  - It depends on the nature of the problem.
  - If targets need to be detected using infrared images for example, choose the brightest pixel(s).
  - Without a-priori knowledge, compute the histogram and corresponding



- **Region growing** is a simple region-based image segmentation method. It is also classified as a pixel-based image segmentation method since it involves the selection of initial seed points.

- This approach to segmentation examines neighbouring pixels of initial seed points and determines whether the pixel neighbours should be added to the region. The process is iterated on, in the same manner as general data clustering algorithms.

Basic concept of seed points

<u>Region growing algorithm is described below.</u>

- The first step in region growing is to **select a set of seed points.** Seed point selection is **based on some user criterion.**

- The regions are then grown from these **seed points to adjacent points depending on a region membership criterion**. The criterion could be, for example, **pixel intensity, grayscale texture, or color**.

- Since the regions are grown on the basis of the criterion, the image information itself is important.
  - For example, if the criterion were a pixel intensity threshold value, knowledge of the histogram of the image would be of use.

- There is a very simple example followed below. Here we **use 4 or 8 connected neighbourhood** to grow from the seed points. For our pixels adjacent relationship.

- And the criteria we make here is the same pixel value. That is, we keep examining the adjacent pixels of seed points. If they have the **same intensity value with the seed points, we classify them into the seed points.**

- It is an iterated process until there are no change in two successive iterative stages. Of course, we can make other criteria, but the main goal is to classify the similarity of the image into regions.

Seed-based region growing segmentation: *example*

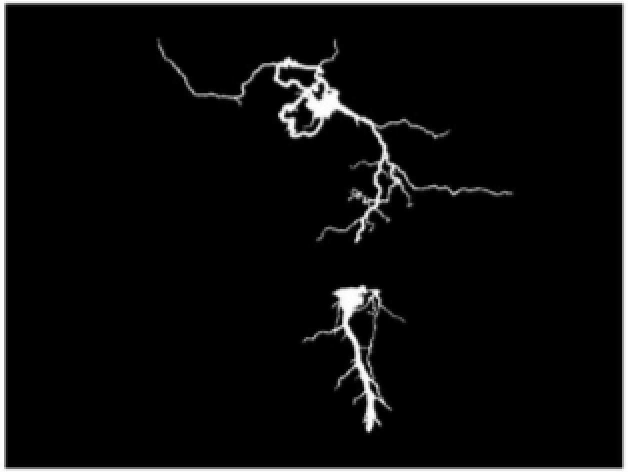
Problem: To isolate the strongest lightning region of the image on the right hand side without splitting it apart.
Solution: To choose the points having the highest gray-scale value which is 255 as the seed points shown in the image immediately below.

original image

threshold = 255 returns multiple seeds

threshold: 225~255

threshold: 190~225



threshold: 155~255

NOTES

Here we show a simple example for region growing.

- Figure 1 is the original image which is a grayscale lightning image. The grayscale value of this image is from 0 to 255. The reason we apply region growing on this image is that we want to mark the strongest lightning part of the image and we also want the result to be connected without being split apart. Therefore, we choose the points having the highest grayscale value which is 255 as the seed points shown in the Figure 2.
- After determining the seed points, we have to determine the range of threshold. Always keep in mind that the objective is to mark the strongest light in the image. The third figure is the region growing result from choosing the threshold between 225 and the value of seed points (which is 255). Hence we only mark out the points whose grayscale values are above 225.
- If we make the range of threshold wider, we will get a result having a bigger area of the lightning region shown as the Figure 4 and the Figure 5.
- We can observe the difference between the last two figures which have different threshold values. Region growing provides the ability for us to separate the part we want connected.
- As we can see in Figure 3 to Figure 5, the segmented results in this example are seed-oriented connected. That means the result grew from the same seed points are the same regions. And the points will not be grown without being connected with the seed points.
- Therefore, there are still lots of points in the original image having the grayscale values above 155 which are not marked in Figure 5.
- This characteristic ensures the reliability of the segmentation and provides the ability to resist noise. For this example, this characteristic prevents us marking out the non-lightning part in the image because the lightning is always connected as one part.

Important issues about region growing

1.The suitable selection of seed points is important.

- The selection of seed points is depending on the users. For example, in a grayscale lightning image, we may want to segment the lightning from the background. Then probably, we can examine the histogram and choose the seed points from the highest range of it.

2.More information of the image is better.

- Obviously, the connectivity or pixel adjacent information is helpful for us to determine the threshold and seed points.

3.The value, “minimum area threshold”.

- No region in region growing method result will be smaller than this threshold in the segmented image.

4.The value, “Similarity threshold value“.

- If the difference of pixel-value or the difference value of average grayscale of a set of pixels less than “Similarity threshold value”, the regions will be considered as a same region.
- The criteria of similarities or so called homogeneity we choose are also important. It usually depends on the original image and the segmentation result we want.

Advantages :

1. Region growing methods can correctly separate the regions that have the same properties we define.

2. Region growing methods can provide the original images which have clear edges with good segmentation results.

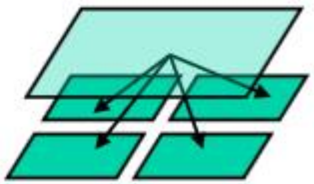
3. The concept is simple. We only need a small number of seed points to represent the property we want, then grow the region.

4. We can determine the seed points and the criteria we want to make.

5. We can choose the multiple criteria at the same time.

6. It performs well with respect to noise.

Region splitting and merging segmentation



Region splitting:

- Unlike region growing which starts from a set of seed points, region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.

Region merging:

- Region merging is the opposite of region splitting, and works as a way of avoiding over-segmentation.
- Start with small regions (e.g. 2x2 or 4x4 regions) and merge the regions that have similar characteristics (such as gray level, variance).

original image *splitting & merging* *thresholding seg.*

Region splitting: *example*

*In this example, the **criterion of homogeneity** is the variance of 1.*

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

original image

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

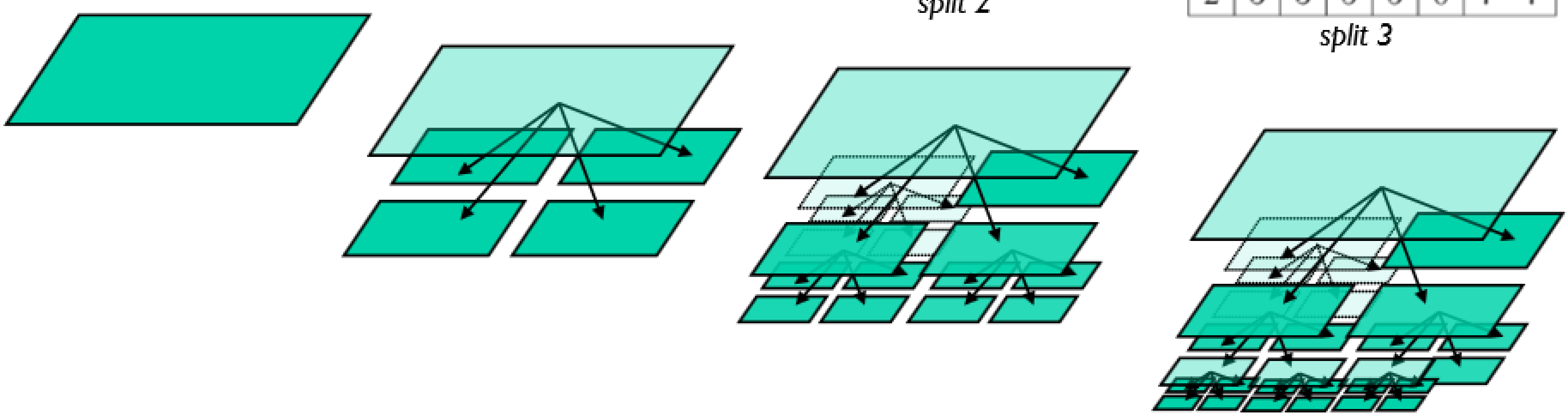
split 1

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 2

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 3



Split-and-merge method and quadtrees

The **split-and-merge** algorithm uses a homogeneity predicate P and has two stages:

- Top-down: Split image into homogeneous quadrant regions
- Bottom-up: Merge similar adjacent regions

Algorithm 2: Split-and-merge algorithm

1. Top-down: Successively subdivide image and regions into smaller quadrant regions until $P(R_i) = TRUE$ for each R_i. Obtain a **quadtree** structure.
2. Bottom-up: At each level, merge any adjacent regions R_i and R_j for which $P(R_i \cup R_j) = TRUE$.
3. Repeat steps 1 and 2 until no further splitting/merging is possible.

- **EXAMPLES**

Graph cuts