Lecture 6
Fourth stage

Medical Imaging Processing II

Image Segmentation- II Region growing

By

Asst. Prof. Dr. Mehdi Ebady Manaa
Asst. Lect. Lubna Ali Jalil

1. SEGMENATION USING REGION GROWING

Region growing is known as a simple and fast method to segment an image. Many papers on medical image segmentation have reported the use of this method in a variety of applications, for example, to detect cardiac disease and breast cancer and to delineate tumor volumes. One approach compares *the initial seed pixels with the unassigned pixels*. Another approach compares *the outermost pixels with their unassigned neighbor pixels at each iteration*. The first leads to consistent segmented areas but is very *sensitive to noise*. The second may result in inaccurate segmentations especially in cases where the pixel attributes change gradually, *but it is robust to noise*.

Region growing refers to the procedure that groups pixels or subregions into larger regions. Starting with a set of seed points, the regions are grown from these points by including to each seed point those neighboring pixels that have similar attributes like intensity, gray level texture, color, etc. It is an iterative process where each seed pixel grows iteratively until every pixel is processed and thereby forms different regions whose boundaries are defined by closed polygons. The important issues in the region growing are:

- *Selection of initial seeds that represent regions and the selection of suitable properties for including the points in various regions during the growing process.*
- Growing the pixels based on certain properties of the image may not ascertain good segmentation. Connectivity or adjacency information should also be used in the region-growing process.
- Similarity: The similarity denotes the minimum difference in the gray level observed between two spatially adjacent pixels or average gray level of a set of pixels, which will yield different

regions. If this difference is less than the similarity threshold value, the pixels belong to the same region.

- Area of region: The minimum area threshold is associated with the smallest region size in pixels. In the segmented image, no region will be smaller than this threshold, defined by the user.

Region growing post-processing: Region growing often results in undergrowing or overgrowing as a result of nonoptimal parameter setting. A variety of postprocessors has been developed. which combine segmentation information obtained from region growing and edge-based segmentation. Simpler postprocessors are based on general heuristics and decrease the number of small regions in the segmented image that cannot be merged with any adjacent region according to the originally applied homogeneity criteria.

A simple way to measure the similarity is a comparison with the original pixel seed intensity. The user can select a single arbitrary pixel and all the neighboring pixels are compared to this. The similarity can be defined as the difference of the intensity. This is a straight forward way to determine a region and leads to consistent segmented regions. However, this method is very sensitive to noise and results in an unsatisfactory segmentation when the original seed is a noise pixel.

An alternative way to remove this effect is to compare all assigned pixel to the neighboring pixel that is already in the region of interest. Another approach is to initialize the region with not only a single pixel but with a small set of pixels to better describe the region statistics.

Region Growing An image can be divided simply into two regions, namely background and foreground, using a region growing algorithm. A simple way to do so is to pick a seed and then expand it.

expand it. Let $x \in I$ be a randomly selected pixel in a 2D image II. Let R_f is a region of interest of the image I, namely foreground, and δ_1 is tolerance of the difference in intensity. A simple way to expand R_f is to compare all neighborhood pixels with the initial seed and assign each pixel to R_f if it satisfies

$$|x - N(p)| \leq \delta_1 \quad (1)$$

where $N(p) \in I$ is a neighborhood pixel of R_F and $|.|$ is the absolute of the difference of intensity. Once a pixel is assigned to R_f , the new neighborhood pixel can be evaluated in the same way. Unfortunately, when a noise pixel is chosen accidently (see Fig. 1(a) and (c)), the result will lead to an undesirable result.

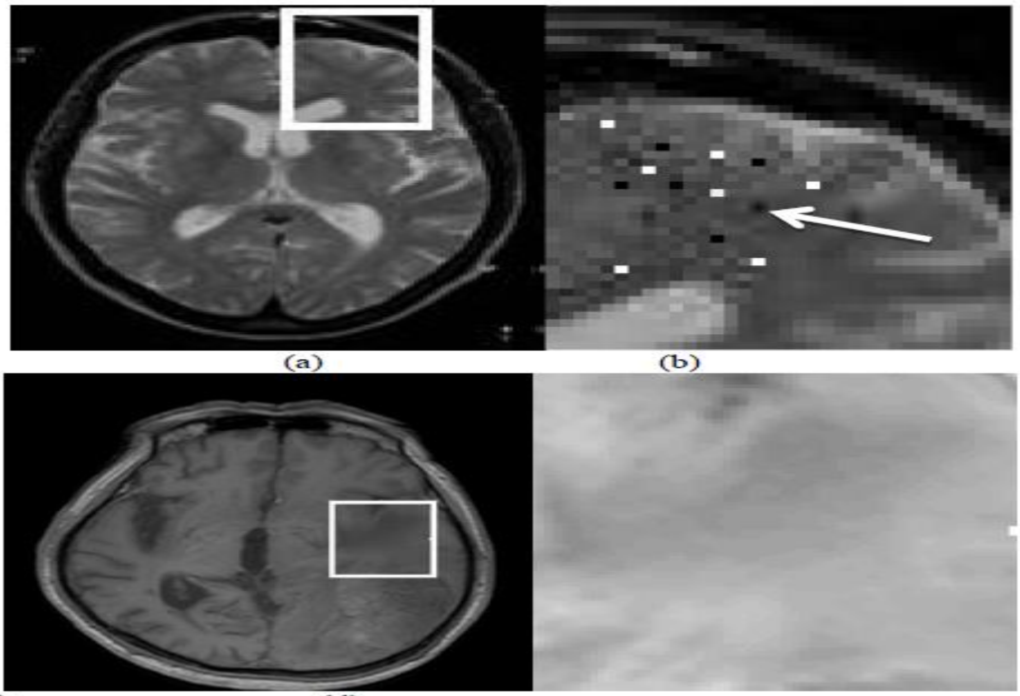

(a) (b)
(c) (d)

Fig 1. Zoom view: (a) and (c) are original images, (b) is a noise pixel chosen in (a), (d) is a gradual area in (c)

Another approach is to compare the pixel at the outer-most region with the closest neighborhood pixel (see Fig. 2). This is similar to eq. (1) but the comparison is between $N(p)$ and outermost pixel of R_F, namely $S(x)$:

$$|S(x) - N(p)| \leq \delta_2 \quad (2)$$

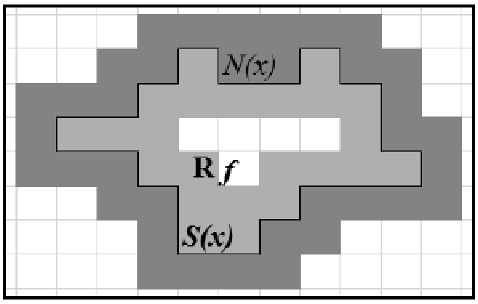


Fig. 2. Outermost pixels R_f in (brighter gray) and neighborhood pixels (darker gray)

The first approach leads to consistent segmented areas but is *very sensitive to noise* while the second approach may result in inaccurate segmentations especially in cases where the pixel attributes change gradually (see Figs 1(c) and (d)), *but is robust to noise.*

Multiple-seeds can be an alternative to solve the problem above. The initialization is not a single pixel but a small set of pixels to better describe the region using statistical tools such as mean or variance. Then the expansion also involves a statistic operation. This is an expensive operation because every neighborhood pixel is evaluated not as a single pixel but as an area. We combine the approaches above by modifying the multiple-seeds as a way to obtain the ground truth value of the area of interest (Fig 3). First, we use 5x5 multiple-seeds to get a better initialization.

34	34	34	34	35
33	34	34	34	34
32	33	32	34	34
32	32	34	34	34
33	33	33	34	33

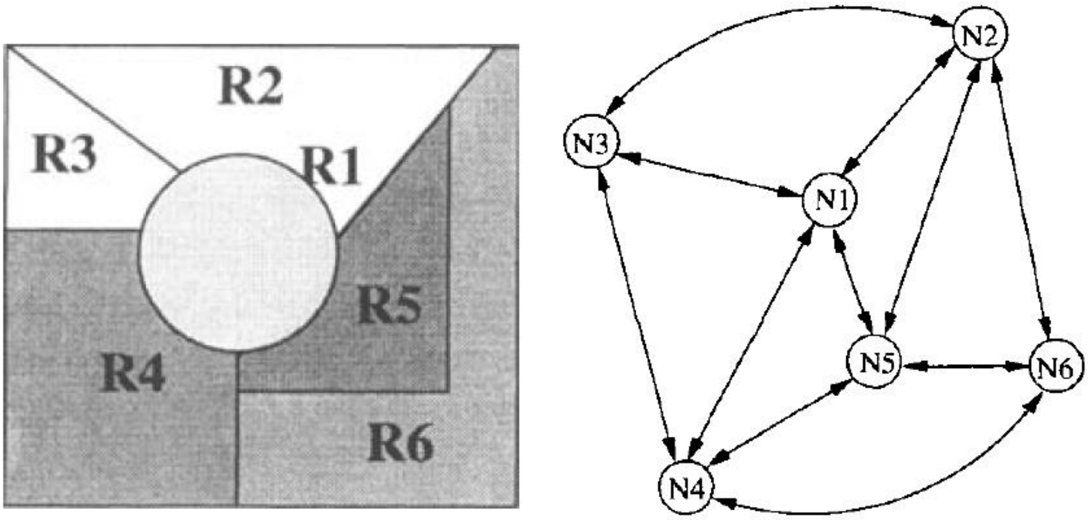
34	34	34	34	0
33	34	34	34	34
32	33	0	34	34
32	32	34	34	34
33	33	33	0	33

Fig 3.Left: 32 is chosen. Right: 0, a noise pixel, is chosen

To handle the case when the chosen pixel is not representative of the area, we compute the median of the initial multiple-seeds. For example, both initializations in Fig. 3 result in 34. In our experiment, applying blurring image using Gaussian results in a better segmentation. Image using Gaussian results in a better segmentation.

Region Adjacency Graph

The adjacency relation among the regions in a scene can be represented by a region adjacency graph (RAG). The regions in the scene are represented by a set of nodes N = { N_1, N_2,. . . , Nm,} in the RAG, where node N_i, represent the region R, in the scene and properties of the region R, is stored in the node N_i,. The edge $e_{i,j,}$ between N_i, and N_j represent the adjacency between the regions R_i, and R_j. Two regions R_i, and R_j are adjacent if there exist a pixel in region R_i, and a pixel in region R_j which are adjacent to each other. The adjacency can be either 4-connected or 8connected. The adjacency relation is reflexes and symmetric, but not necessarily transitive. In Figure 4, we show the adjacency graph of a scene with five distinct regions.



(a) (b)

Fig 5: (a) A scene with 6 distinct regions, (b) the adjacency graph of the scene

A binary matrix A is called an adjacency matrix when it represents a region adjacency graph (RAG). When the nodes N_i and N_j in RAG are adjacent, ai,j in A is 1. Since adjacency relation is reflexive, the diagonal elements of the matrix are all 1. The adjacency matrix (relation) of a multiregion scene in Figure 5(a) is shown below.

$$A = \begin{bmatrix} & N1 & N2 & N3 & N4 & N5 & N6 \\ N1 & 1 & 1 & 1 & 1 & 1 & 0 \\ N2 & 1 & 1 & 1 & 0 & 1 & 1 \\ N3 & 1 & 1 & 1 & 1 & 0 & 0 \\ N4 & 1 & 0 & 1 & 1 & 1 & 1 \\ N5 & 1 & 1 & 0 & 1 & 1 & 1 \\ N6 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Region Merging and Splitting

A segmentation algorithm can produce too many small regions because of fragmentation of a single large region in the scene. In such a situation, the smaller regions need to be merged based on similarity and compactness of the smaller regions. A simple region merging algorithm is presented below.

Step 1: Segment the image into R1, Rz,. . . , R, using a set of thresholds.

Step 2: Create a region adjacency graph (RAG) from the segmented description of the image.

Step 3: For every Ri, i = 1 , 2 , . . . , m, identify all Rj, j # i from the RAG such that Ri is adjacent to Rj.

Step 4: Compute an appropriate similarity measure Si,j between R, and R,, for all i and j .

Step 5: If S_{ij} > T,then merge R_i, and R_j,.

Step 6: Repeat steps 3 to 5 until there is no region to be merged according to the similarity criteria.

There are situations, when too little regions are generated because of inaccurate preliminary segmentation. This is due to wrong merger of different regions into a single one. In such situation, the variances of the gray values in a segmented region may be above a threshold (T) and hence the region needs to be split in smaller regions such that each of the smaller regions has uniform small variances.

Splitting and merging can be combined together for segmenting complex scenes where a rule-based may guide the applications of split and merge operations.