SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

COURSE NAME : 19EC513 – IMAGE PROCESSING AND COMPUTER VISION

III YEAR / V SEMESTER

Unit III- **IMAGE COMPRESSION AND IMAGE SEGMENTATION**

Topic : Region based segmentation , Region growing, splitting and merging

Region Based Segmentation

we approached this problem by attempting to find boundaries between regions based on discontinuities in intensity levels, segmentation was accomplished via thresholds based on the distribution of pixel properties, such as intensity values or color. In this section, we discuss segmentation techniques that are based on finding the regions directly

Region Growing

As its name implies,region growing is a procedure that groups pixels or subregions into larger regions based on predefined criteria for growth.The basic approach is to start with a set of “seed” points and from these grow regions by appending to each seed those neighboring pixels that have predefined properties similar to the seed (such as specific ranges of intensity or color).

Selecting a set of one or more starting points often can be based on the nature of the problem, as shown later in Example 10.23. When a priori information is not available, the procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations shows clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds.

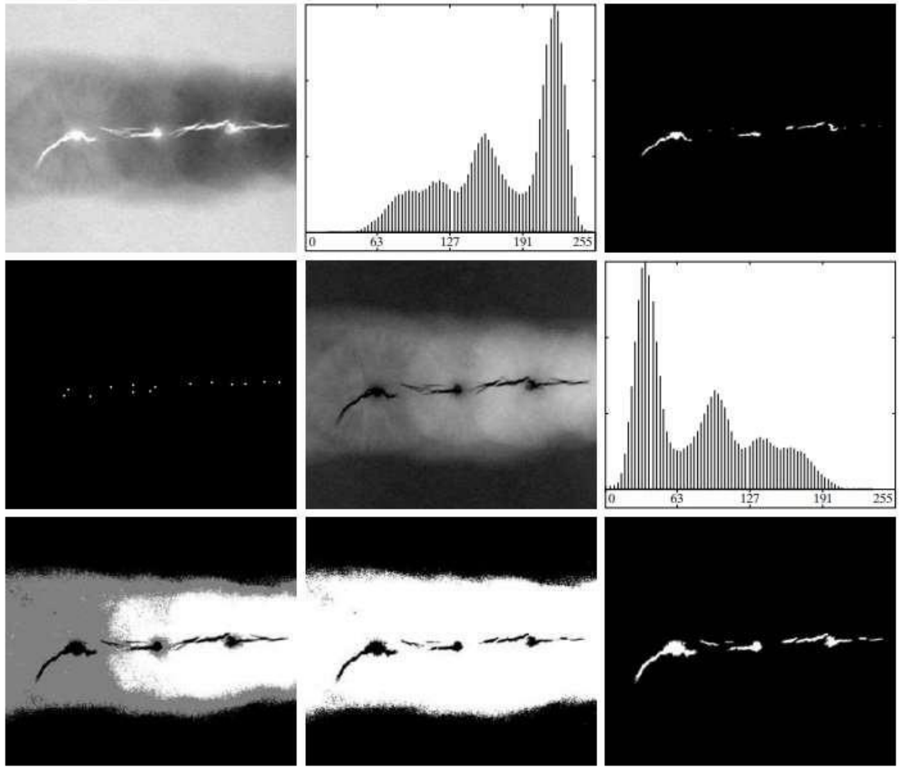
The selection of similarity criteria depends not only on the problem under consideration, but also on the type of image data available. For example, the analysis of land-use satellite imagery depends heavily on the use of color. This problem would be significantly more difficult, or even impossible, to solve without the inherent information available in color images. When the images are monochrome, region analysis must be carried out with a set of descriptors based on intensity levels and spatial properties (such as moments or texture).

Descriptors alone can yield misleading results if connectivity properties are not used in the region-growing process. For example, visualize a random arrangement of pixels with only three distinct intensity values. Grouping pixels with the same intensity level to form a “region” without paying attention to connectivity would yield a segmentation result that is meaningless in the context of this discussion.

Another problem in region growing is the formulation of a stopping rule. Region growth should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as intensity values, texture, and color are local in nature and do not take into account the “history” of region growth. Additional criteria that increase the power of a region-growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far (such as a comparison of the intensity of a candidate and the average intensity of the grown region), and the shape of the region being grown. The use of these types of descriptors is based on the assumption that a model of expected results is at least partially available

Let: denote an input image array; denote a seed array containing 1s at the locations of seed points and 0s elsewhere; and denote a predicate to be applied at each location Arrays and are assumed to be of the same size. A basic region-growing algorithm based on 8-connectivity may be stated as follows.

Find all connected components in and erode each connected component to one pixel; label all such pixels found as 1. All other pixels in are labeled 0. 2. Form an image such that, at a pair of coordinates let if the input image satisfies the given predicate, at those coordinates; otherwise, let 3. Let be an image formed by appending to each seed point in all the 1-valued points in that are 8-connected to that seed point. 4. Label each connected component in with a different region label (e.g.,). This is the segmented image obtained by region growing.



Next, we have to specify a predicate. In this example, we are interested in appending to each seed all the pixels that (a) are 8-connected to that seed and (b) are "similar" to it. Using intensity differences as a measure of similarity, our predicate applied at each location is

$$Q = \begin{cases} \text{TRUE} & \text{if the absolute difference of the intensities} \\ & \text{between the seed and the pixel at } (x, y) \text{ is} \leq T \\ \text{FALSE} & \text{otherwise} \end{cases}$$

where is a specified threshold. Although this predicate is based on intensity differences and uses a single threshold, we could specify more complex schemes in which a different threshold is applied to each pixel, and properties other than differences are used. In this case, the preceding predicate is sufficient to solve the problem, as the rest of this example shows

From the previous paragraph, we know that the smallest seed value is 255 because the image was thresholded with a threshold of 254. Figure 10.51(e) shows the absolute value of the difference between the images in Figs. 10.51(a) and (c). The image in Fig. 10.51(e) contains all the differences needed to compute the predicate at each location Figure 10.51(f) shows the corresponding histogram. We need a threshold to use in the predicate to establish similarity. The histogram has three principal modes, so we can start by applying to the difference image the dual thresholding technique discussed in Section 10.3.6. The resulting two thresholds in this case were and which we see correspond closely to the valleys of the histogram. (As a brief digression, we segmented the image using these two thresholds. The result in Fig. 10.51(g) shows that the problem of segmenting the defects cannot be solved using dual thresholds, even though the thresholds are in the main valleys.)

Figure 10.51(h) shows the result of thresholding the difference image with only The black points are the pixels for which the predicate was TRUE; the others failed the predicate. The important result here is that the points in the good regions of the weld failed the predicate, so they will not be included in the final result.The points in the outer region will be considered by the regiongrowing algorithm as candidates. However, Step 3 will reject the outer points, because they are not 8-connected to the seeds. In fact, as Fig. 10.51(i) shows, this step resulted in the correct segmentation, indicating that the use of connectivity was a fundamental requirement in this case. Finally, note that in Step 4 we used the same value for all the regions found by the algorithm. In this case, it was visually preferable to do so

Region Splitting and Merging

The procedure discussed in the last section grows regions from a set of seed points. An alternative is to subdivide an image initially into a set of arbitrary, disjoint regions and then merge and/or split the regions in an attempt to satisfy the conditions of segmentation stated in Section 10.1. The basics of splitting and merging are discussed next.

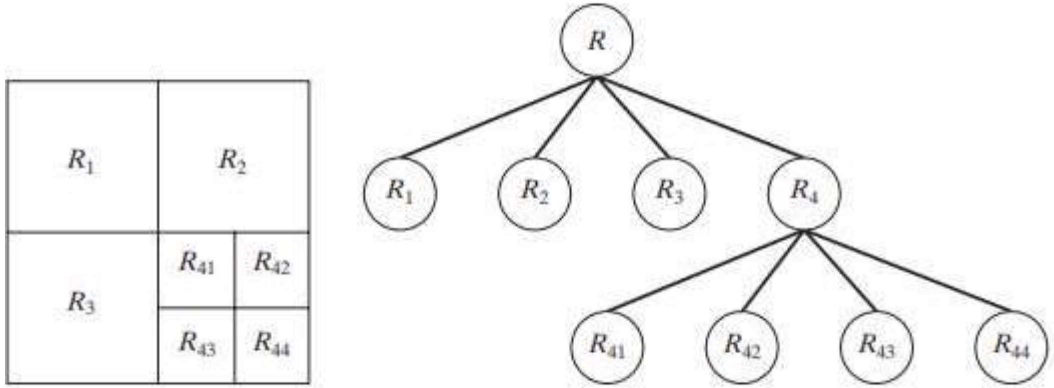
Let represent the entire image region and select a predicate One approach for segmenting is to subdivide it successively into smaller and smaller quadrant regions so that, for any region We start with the entire region. If we divide the image into quadrants. If is FALSE for any quadrant, we subdivide that quadrant into subquadrants, and so on.

This particular splitting technique has a convenient representation in the form of so-called quadtrees, that is, trees in which each node has exactly four descendants, as Fig. 10.52 shows (the images corresponding to the nodes of a quadtree sometimes are called quadregions or quadimages). Note that the root of the tree corresponds to the entire image and that each node corresponds to the subdivision of a node into four descendant nodes. In this case, only was subdivided further. If only splitting is used, the final partition normally contains adjacent regions with identical properties. This drawback can be remedied by allowing merging as well as splitting. Satisfying the constraints of segmentation outlined in Section 10.1 requires merging only adjacent regions whose combined pixels satisfy the predicate That is, two adjacent regions and are merged only if

The preceding discussion can be summarized by the following procedure in which, at any step, we

1. Split into four disjoint quadrants any region R_i for which $Q(R_i)$ = FALSE.
2. When no further splitting is possible, merge any adjacent regions R_j and R_k for which $Q(R_j \cup R_k)$ = TRUE.
3. Stop when no further merging is possible.

It is customary to specify a minimum quadregion size beyond which no further splitting is carried out. Numerous variations of the preceding basic theme are possible. For example, a significant simplification results if in Step 2 we allow merging of any two adjacent regions and if each one satisfies the predicate individually.This results in a much simpler (and faster) algorithm, because testing of the predicate is limited to individual quadregions. As the following example shows, this simplification is still capable of yielding good segmentation re



a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

Figure 10.53(a) shows a X-ray band image of the Cygnus Loop. The objective of this example is to segment out of the image the “ring” of less dense matter surrounding the dense center. The region of interest has some obvious characteristics that should help in its segmentation. First, we note that the data in this region has a random nature, indicating that its standard deviation should be greater than the standard deviation of the background (which is near 0) and of the large central region, which is fairly smooth. Similarly, the mean value (average intensity) of a region containing data from the outer ring should be greater than the mean of the darker background and less than the mean of the large, lighter central region. Thus, we should be able to segment the region of interest using the following predicate

$$Q = \begin{cases} \text{TRUE} & \text{if } \sigma > a \text{ AND } 0 < m < b \\ \text{FALSE} & \text{otherwise} \end{cases}$$

where and are the mean and standard deviation of the pixels in a quadregion, and and are constants. Analysis of several regions in the outer area of interest revealed that the mean intensity of pixels in those regions did not exceed 125 and the standard deviation was always greater than 10. Figures 10.53(b) through (d) show the results obtained using these values for and and varying the minimum size allowed for the quadregions from 32 to 8. The pixels in a quadregion whose

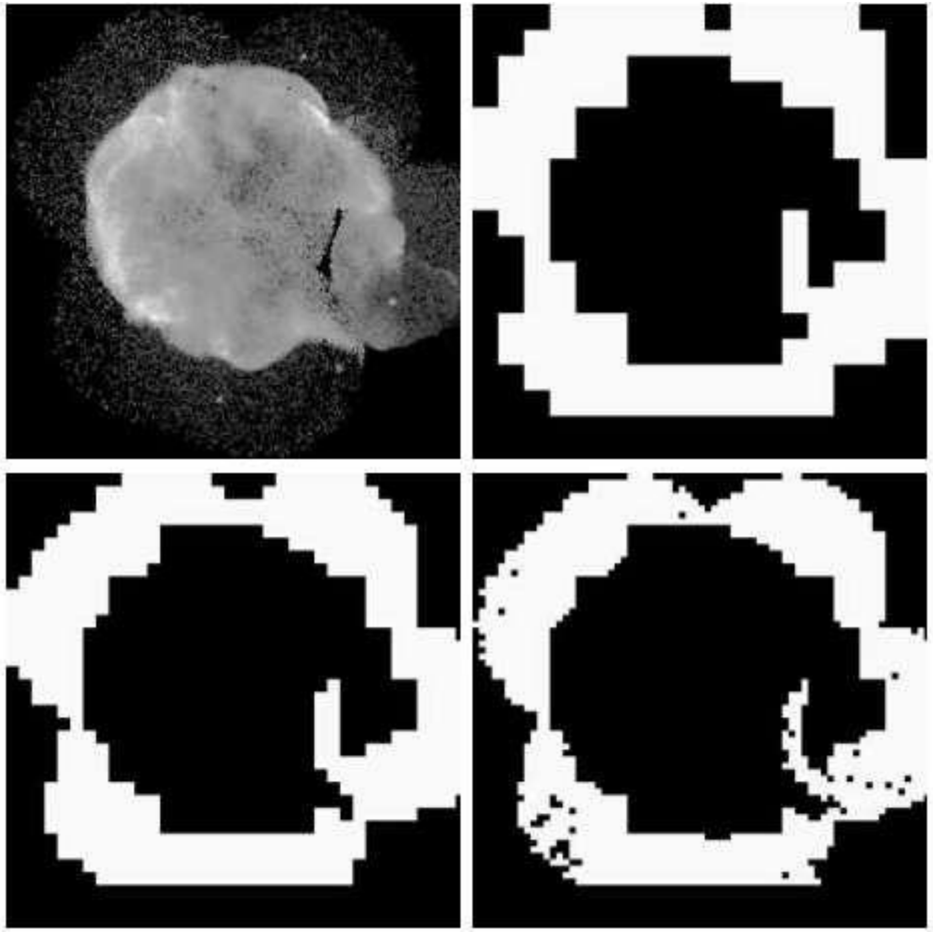


FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

pixels satisfied the predicate were set to white; all others in that region were set to black.The best result in terms of capturing the shape of the outer region was obtained using quadregions of size The black squares in Fig. 10.53(d) are quadregions of size whose pixels did not satisfied the predicate. Using smaller quadregions would result in increasing numbers of such black regions. Using regions larger than the one illustrated here results in a more “blocklike” segmentation. Note that in all cases the segmented regions (white pixels) completely separate the inner, smoother region from the background. Thus, the segmentation effectively partitioned the image into three distinct areas that correspond to the three principal features in the image: background, dense, and sparse regions. Using any of the white regions in Fig. 10.53 as a mask would make it a relatively simple task to extract these regions from the original image (Problem 10.40). As in Example 10.23, these results could not have been obtained using edge- or threshold-based segmentation

As used in the preceding example, properties based on the mean and standard deviation of pixel intensities in a region attempt to quantify the texture of the region (see Section 11.3.3 for a discussion on texture). The concept of texture segmentation is based on using measures of texture in the predicates. In other words, we can perform texture segmentation by any of the methods discussed in this section simply by specifying predicates based on texture content.

THANK YOU !!!