Simple Global Thresholding

Thresholding:

Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation.

Global Thresholding:

The simplest of all thresholding techniques is to partition the image histogram by using a single global threshold, T. Segmentation is then accomplished by scanning the image pixel by pixel and labeling each pixel as object or background, depending on whether the gray level of that pixel is greater or less than the value of T. As indicated earlier, the success of this method depends entirely on how well the histogram can be partitioned.

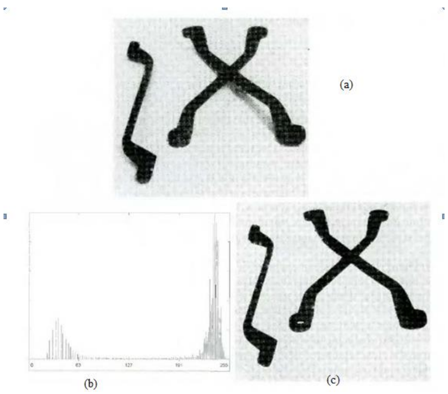


FIGURE (a) Original image, (b) Image histogram, (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

Figure (a) shows a simple image, and Fig. 4.1(b) shows its histogram. Figure 4.1(c) shows the result of segmenting Fig. 4.1(a) by using a threshold T midway between the maximum and minimum gray levels. This threshold achieved a "clean" segmentation by eliminating the shadows and leaving only the objects themselves. The objects of interest in this case are darker than the background, so any pixel with a gray level $\leq T$ was labeled black (0), and any pixel with a gray level $\geq T$ was labeled white (255).The key objective is merely to generate a binary image, so the black-white relationship could be reversed. The type of global thresholding just described can be expected to be successful in highly controlled environments. One of the areas in which this often is possible is in industrial inspection applications, where control of the illumination usually is feasible.

The threshold in the preceding example was specified by using a heuristic approach, based on visual inspection of the histogram. The following algorithm can be used to obtain T automatically:

1. Select an initial estimate for
2. Segment the image using T. This will produce two groups of pixels: G_1 consisting of all pixels with gray level values $>T$ and G_2 consisting of pixels with values $<$
3. Compute the average gray level values μ_1 and μ_2 for the pixels in regions $G_1$and G
4. Compute a new threshold value:

$$T = \frac{1}{2}(\mu_1 + \mu_2).$$

The threshold in the preceding example was specified by using a heuristic approach, based on visual inspection of the histogram. The following algorithm can be used to obtain T automatically:

1. Select an initial estimate for
2. Segment the image using T. This will produce two groups of pixels: G_1 consisting of all pixels with gray level values >T and G_2 consisting of pixels with values <
3. Compute the average gray level values μ_1 and μ_2 for the pixels in regions $G_1$and G
4. Compute a new threshold value:

Region Oriented Segmentation

The goal of the region oriented segmentation finds coherent (homogeneous) regions in the image. The Coherent regions contain pixels which share some similar property. The advantages of Region-based techniques are generally better in noisy images (where borders are difficult to detect). The drawbacks may be the output of region-growing techniques is either over segmented (too many regions) or under segmented (too few regions). Thus it can't find objects that span multiple disconnected regions.

Basic Formulation

A complete segmentation of an image R is a finite set of regions R1, R2,…, RS such that

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

The Region-based segmentation verifies a homogeneity criterion

$H(R_i) = true \qquad i = 1, 2,..., S$

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

Region growing by pixel aggregation

It starts from one seed pixel p located inside region R. Next step is to define a similarity measure S(i; j) for all pixels i and j in the image. Now 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 and continue until all pixels in the currently investigated neighbourhood do not satisfy the inclusion criteria.

<u>Example of pixel aggregation</u>

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

Design of a region growing algorithm must have some specific questions as follows:

- 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) ?

Comparing to original seed pixel

Comparing to original seed pixel with S(p,r). The advantage finds uses a single basis of comparison across all pixels in the region. The drawback is the region produced is very sensitive to the choice of the seed pixel.

Compare to neihbour in region

Compare to neihbour in region with S(q,r. The advantage may be termed as it respects the transitivity of the similarity relationship. The disadvantage is drift as one grows farther away from the seed pixel.

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. The Mean is updated after each aggregation.

Region growing example

Fig. Gray scale lightning image: segment the lightning.

Fig. Seed points: I=255

Fig. T: 125-255

Fig. T: 155-255

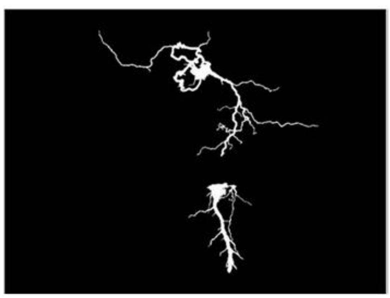


Fig. T: 190-255

Region merging algorithms

The Region merging algorithm STARTs with an oversegmented image. Next part is to 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.

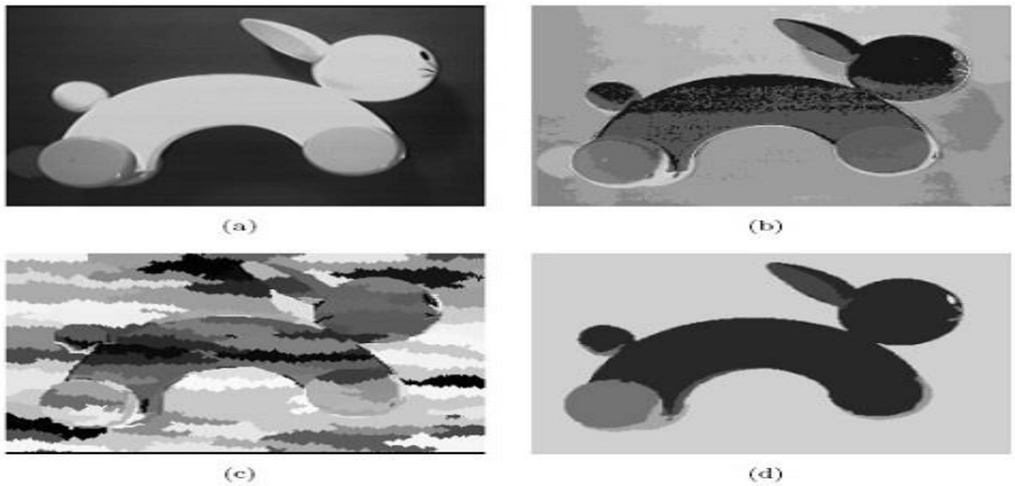


Figure 6.43: Region merging segmentation. (a) Original image. (b) Pseudo-color representation of the original image (in grayscale). (c) Recursive region merging. (d) Region merging via boundary melting. *Courtesy of R. Marik, Czech Technical University.*

Region splitting

The opposite of region merging is Region splitting. It begins with the whole image represented as a single region which does not verify the homogeneity condition.

Split/Merge

If a region R is in homogeneous (P(R)= False) then is split into four sub regions. Now If two adjacent regions Ri,Rj are homogeneous (P(Ri U Rj) = TRUE), they are merged. The algorithm stops when no further splitting or merging is possible.

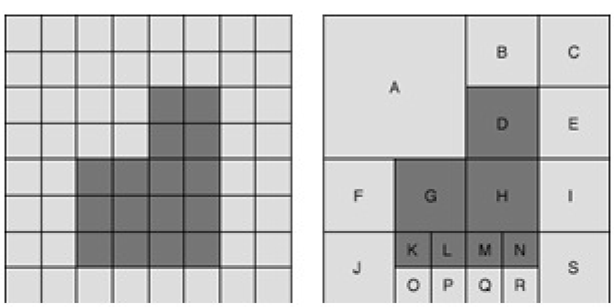
Split and merge algorithm

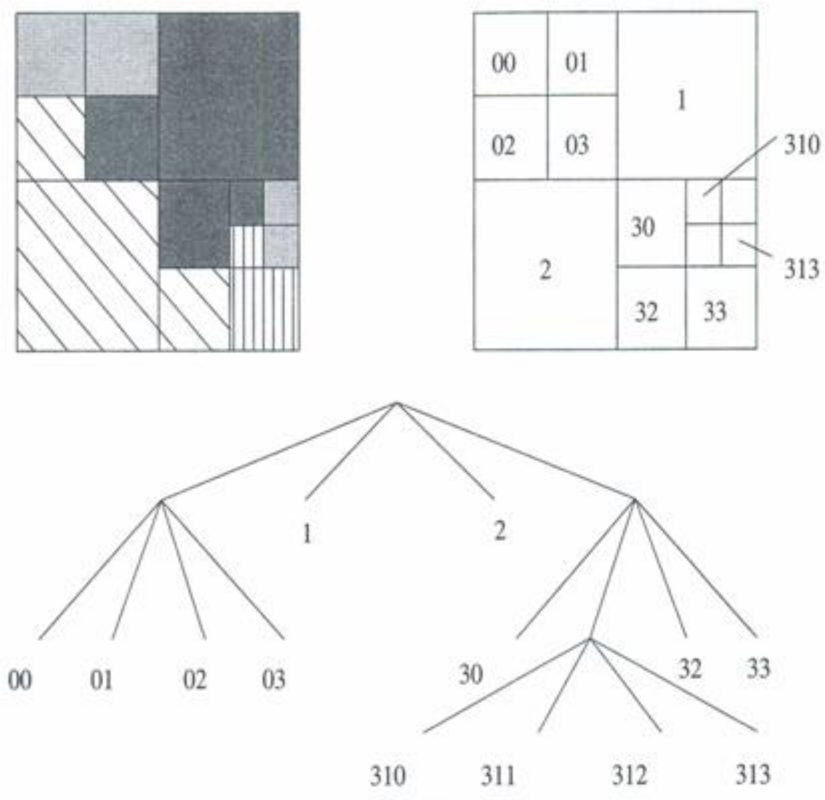
Step 1: consider entire image as one region

Step 2: If region satisfies homogeneity criteria, leave it unmodified

Step 3: If not, split it into four quadrants and recursively apply 2 and 3 to each newly generated region STOP when all regions in the quadtree satisfy the homogeneity criterion

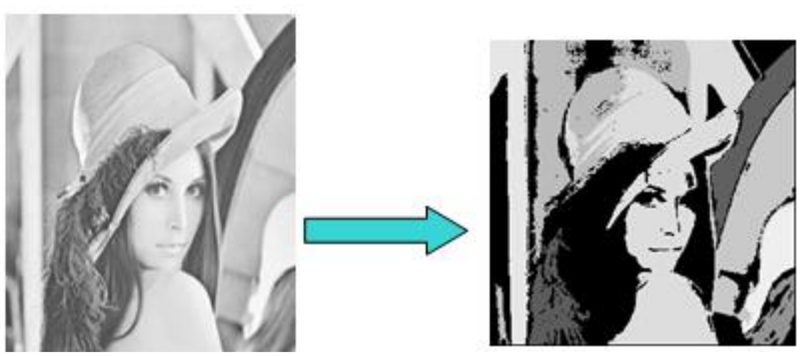
Step 4: If any two adjacent regions Ri, Rj can be merged into a homogeneous region, merge them. STOP when no merging is possible any more.



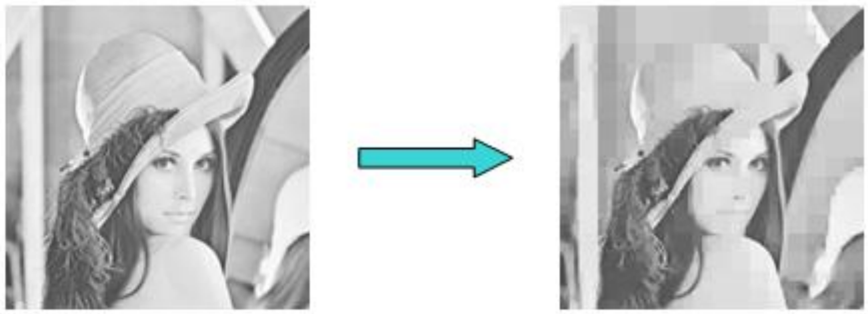


Segmentation quadtree.

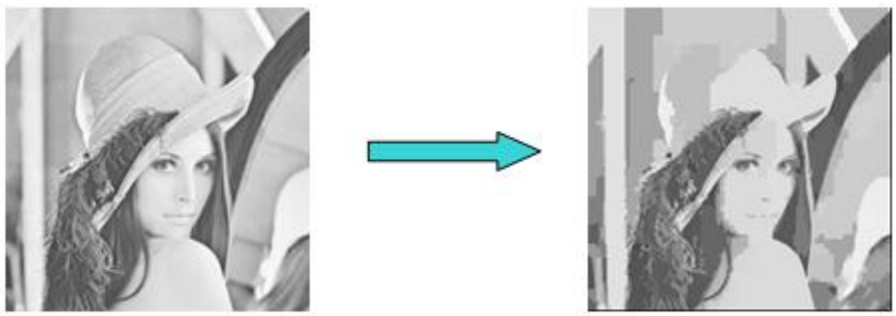
Results – Region grow



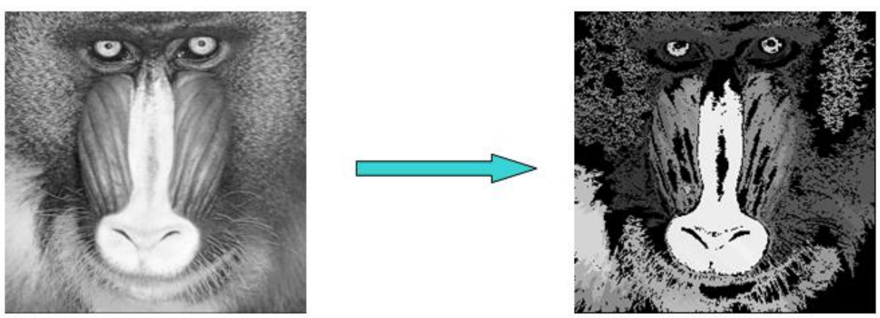
Results – Region Split



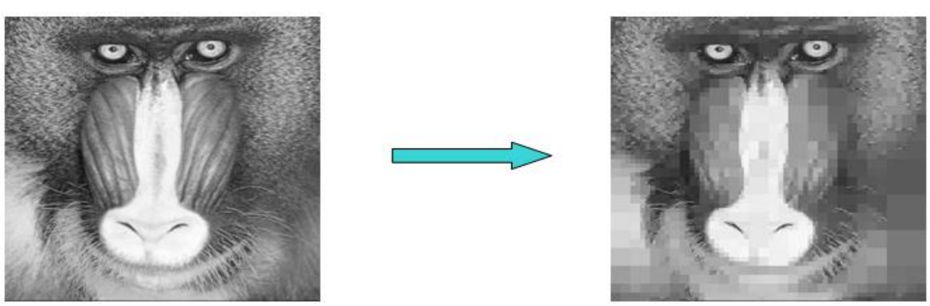
Results – Region Split and Merge



Results – Region growing



Results – Region Split



Results – Region Split and Merge