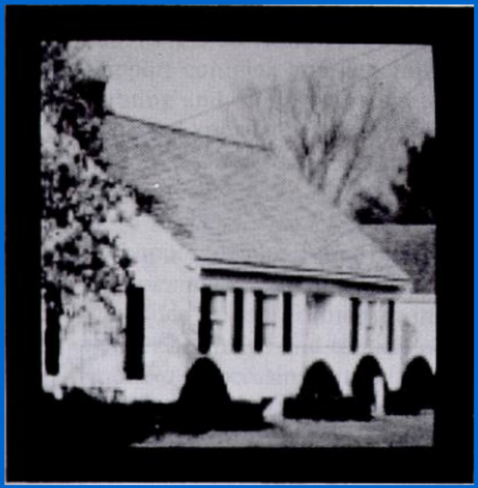
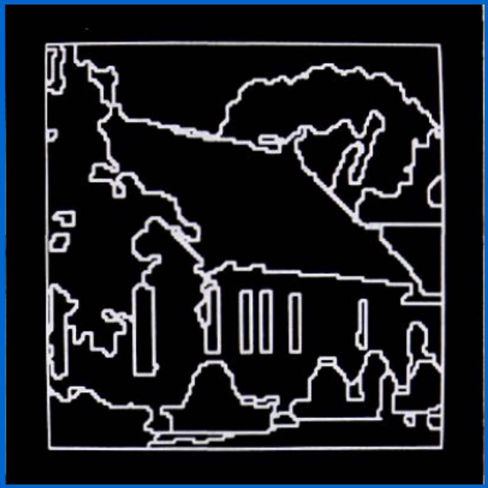
Segmentation (Section 10.3 & 10.4)

Segmentation Approaches

- Edge-based approaches
  - Use the boundaries of regions to segment the image.
  - Detect abrupt changes in intensity (discontinuities).
- Region-based approaches
  - Use similarity among pixels to find different regions.





Main Approaches

- Thresholding (i.e., pixel classification)
- Region growing (i.e., splitting and merging)
- Relaxation

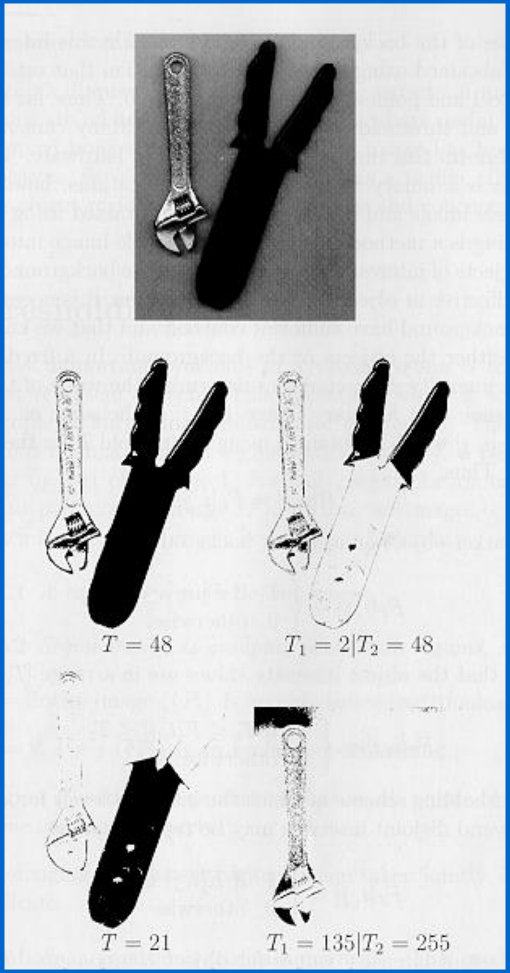
Thresholding

- The simplest approach to segment an image.

If $f(x, y) > T$ then

$f(x, y) = 0$

else $f(x, y) = 255$

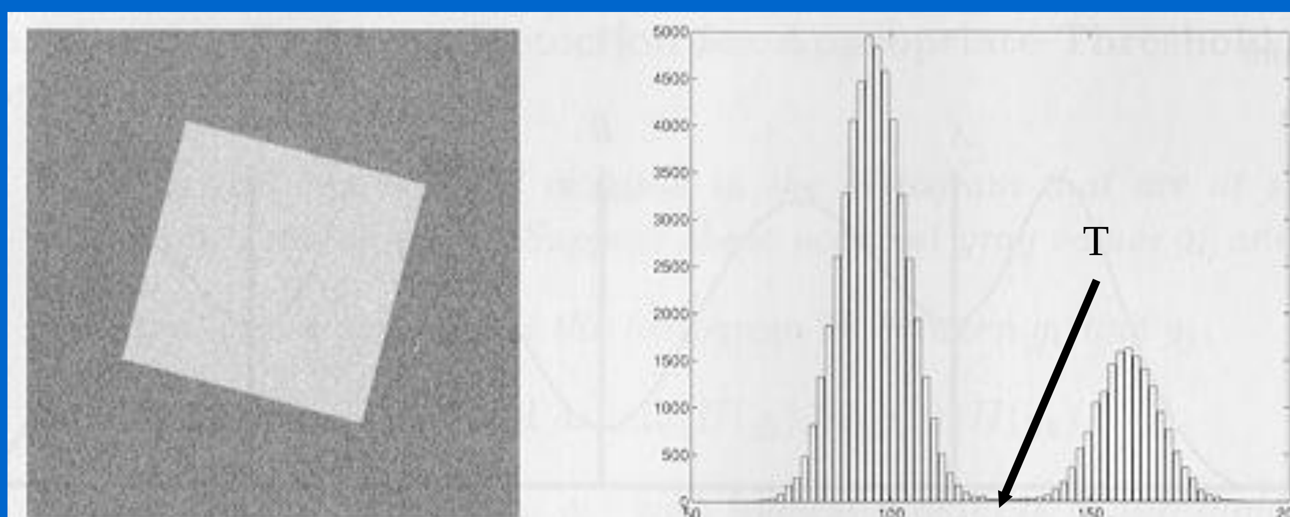


Automatic Thresholding

- To make segmentation more robust, the threshold should be automatically selected by the system.
- Knowledge about the objects, the application, the environment should be used to choose the threshold automatically.
  - Intensity characteristics of the objects
  - Size of the objects.
  - Fractions of an image occupied by the objects
  - Number of different types of objects appearing in an image

Thresholding Using Image Histogram

- Regions with uniform intensity give rise to strong peaks in the histogram.
- In general, a good threshold can be selected if the histogram peaks are tall, narrow, symmetric, and separated by deep valleys.



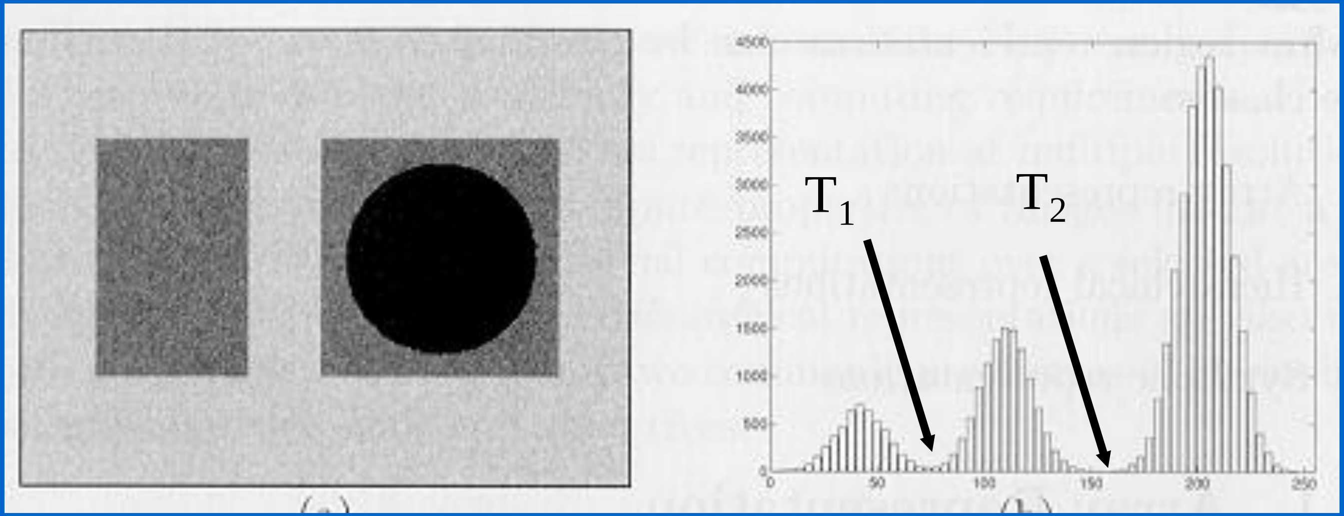
Thresholding Using Image Histogram (cont'd)

- Multiple thresholds are possible

If $f(x, y) < T_1$ then $f(x, y) = 255$

else if $T_1 < f(x, y) < T_2$ then $f(x, y) = 128$

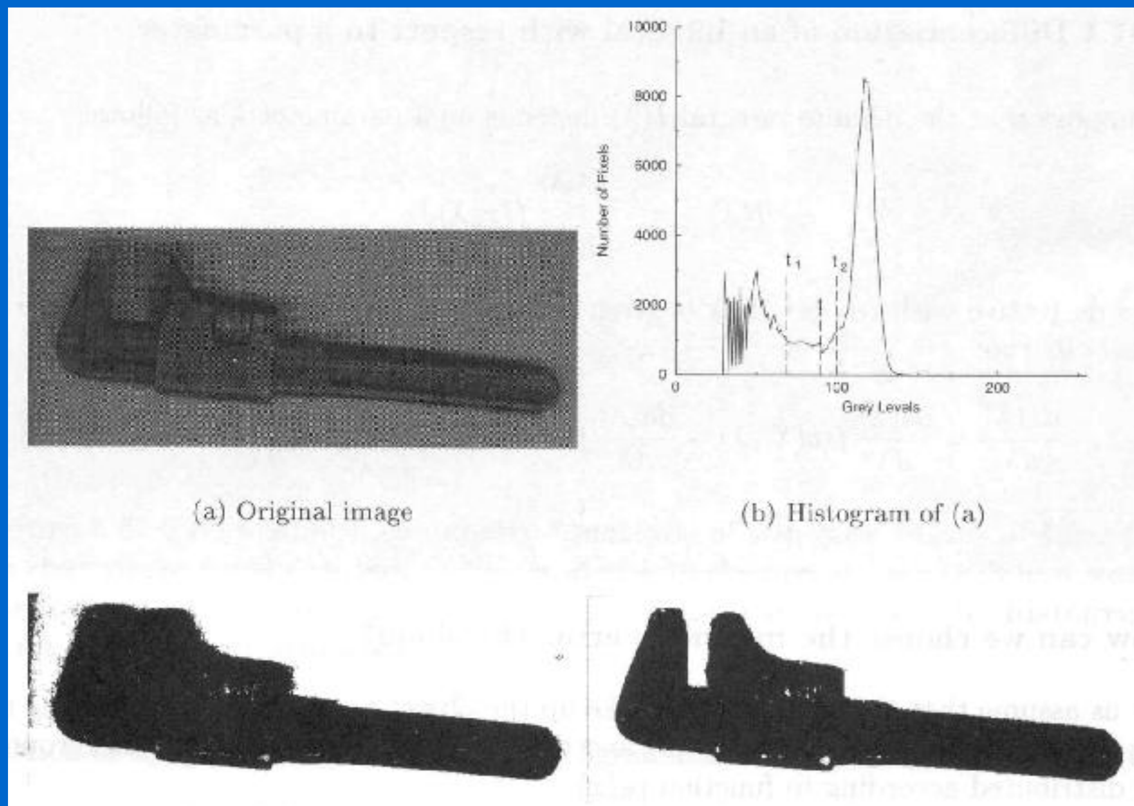
else $f(x, y) = 0$



Hysteresis Thresholding

- If there is no clear valley in the histogram of an image, then there are several background pixels that have similar gray level value with object pixels and vice versa.
- Hystreresis thresholding (i.e., <u>two thresholds</u>, one at each side of the valley) can be used in this case.
  - Pixels above the high threshold are classified as object and below the low threshold as background.
  - Pixels between the low and high thresholds are classified as object only <u>**if**</u> they are adjacent to other object pixels.

Hysteresis Thresholding (cont’d)



(a) Original image

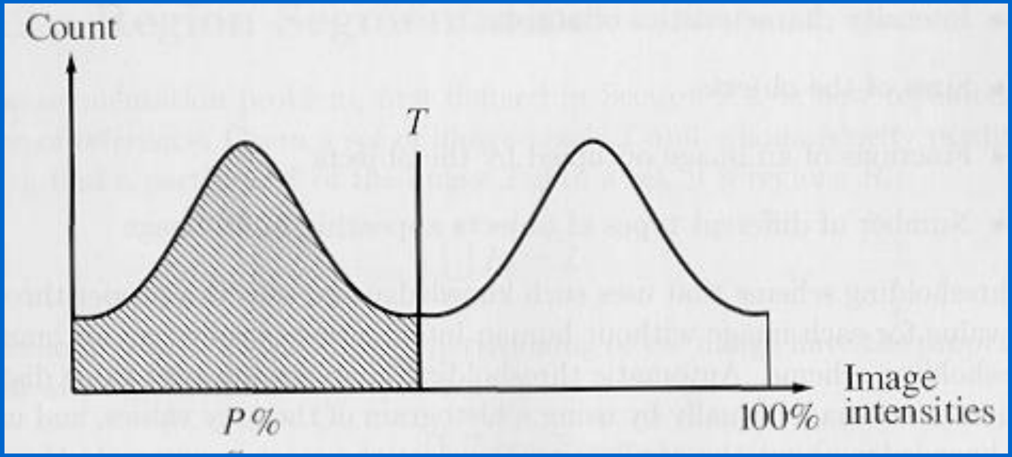
(b) Histogram of (a)

single threshold

hysteresis thresholding

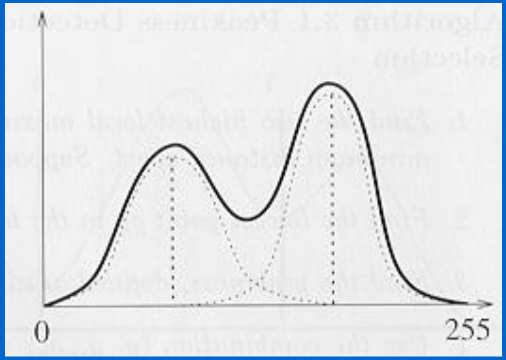
Using prior knowledge for segmentation: **P-Tile method**

- This method requires knowledge about the area or size of the objects present in the image.
  - Assume dark objects against a light background.
  - If, the objects occupy *p*% of the image area, an appropriate threshold can be chosen by partitioning the histogram.



Optimal Thresholding

- Suppose that an image contains only two principal regions (e.g., object and background).
- We can minimize the number of misclassified pixels if we have some prior knowledge about the distributions of the gray level values that make up the object and the background.



e.g., assume that the distribution of gray-level values in each region follows a Gaussian distribution.

Optimal Thresholding (cont'd)

- The probability of a pixel value is then given by the following mixture (i.e., law of "total" probability):

$$P(z) = p(z/background)\ P(background) + p(z/object)\ P(object)$$

$$\text{or } P(z) = P_b p_b(z) + P_o p_o(z)$$

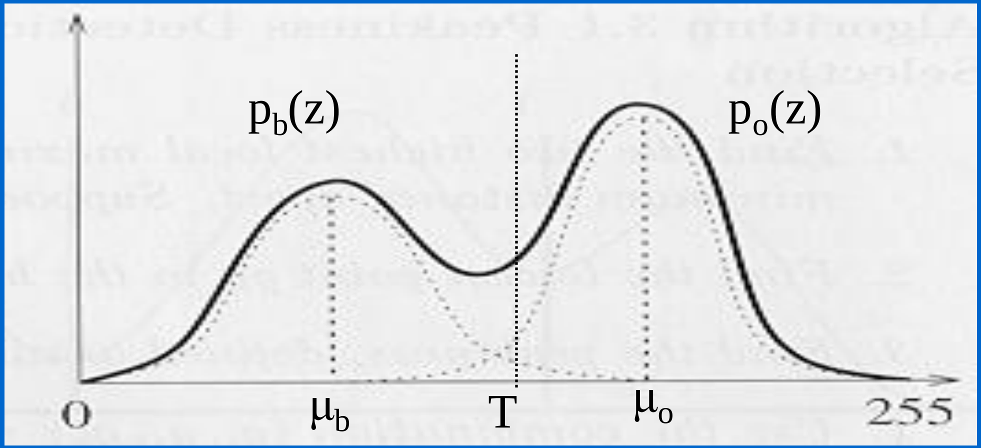
$p_b(z)$, $p_o(z)$, prob. distributions of background, object pixels

P_b, P_o: the a-priori probabilities of background, object pixels

assuming Gaussian distributions:

$$P(z) = P_b \frac{1}{\sqrt{2\pi\sigma_b}} e^{-\frac{(z-\mu_b)^2}{2\sigma_b^2}} + P_o \frac{1}{\sqrt{2\pi\sigma_o}} e^{-\frac{(z-\mu_o)^2}{2\sigma_o^2}}$$

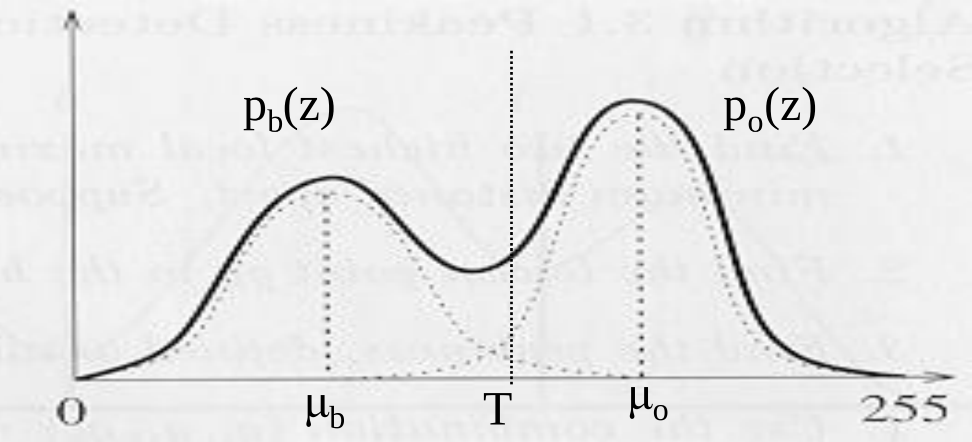
Optimal Thresholding (cont'd)



- Suppose we have chosen a threshold *T*, what is the probability of (erroneously) classifying an object pixel as background ?

$$E_o(T) = \int_{-\infty}^{T} p_o(z)dz$$

Optimal Thresholding (cont'd)



- What is the probability of (erroneously) classifying a background pixel as object ?

$$E_b(T) = \int_T^{\infty} p_b(z)dz$$

Optimal Thresholding (cont'd)

- Overall probability of error: $E(T) = P_b\, E_o(T) + P_o\, E_b(T)$

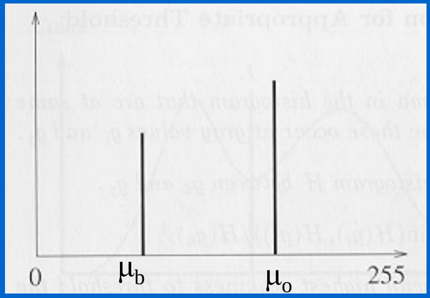
- Minimize $E(T)$ $\quad \dfrac{dE(T)}{dT} = 0$

- The above expression is minimized when

$$T = \frac{\mu_b + \mu_o}{2} + \frac{\sigma^2}{\mu_b - \mu_o}\, ln(P_o/P_b) \quad (\sigma_b = \sigma_o = \sigma)$$

- Special cases when $P_b = P_o$ or $\sigma = 0$:

$$T = \frac{\mu_b + \mu_o}{2}$$



Optimal Thresholding (cont'd)

- Main steps in choosing T

(1) Find the histogram $h(z)$ of the image to be segmented

(2) Choose the parameters (μ_b, μ_o, σ_b, σ_o, P_b, P_o) such that the model $p(z) = P_b p_b(z) + P_o p_o(z)$ fits $h(z)$ satisfactorily

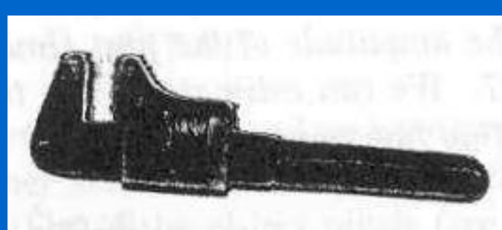
$$\text{e.g., minimize } Error = \frac{1}{N}\sum_{i=1}^{N}(p(z_i) - h(z_i))^2$$

(3) Choose T based on the formula derived above

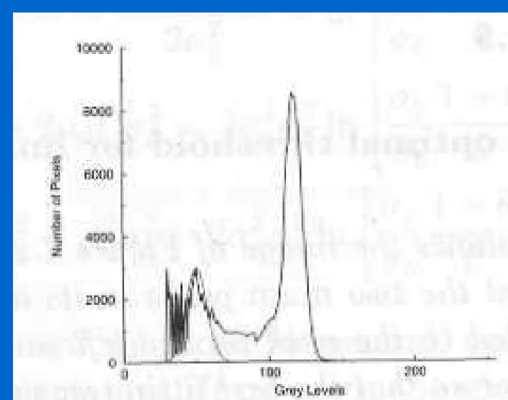
Optimal Thresholding (cont'd)

- Drawbacks of the optimum thresholding method
  - Object/Background distributions might not be known.
  - Prior probabilities might not be known.

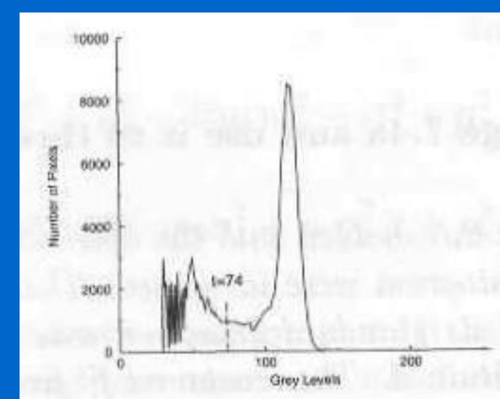
thresholded image



object distribution superimposed on histogram

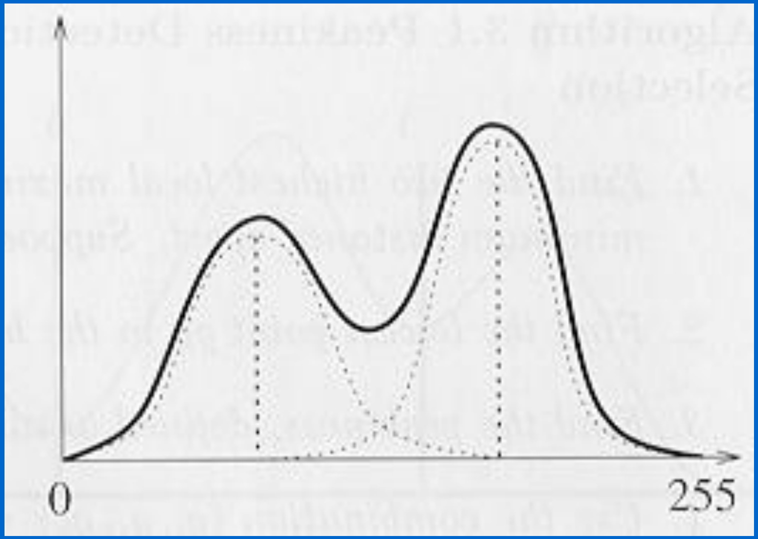


optimal threshold



Otsu's Method

- Assumptions
  - It does not depend on modeling the probability density functions.
  - It **does** assume a bimodal histogram distribution



Otsu's Method

- Segmentation is based on "region homogeneity".
- Region homogeneity can be measured using variance (i.e., regions with high homogeneity will have low variance).

- Otsu's method selects the threshold by minimizing the <u>within-class variance</u>.

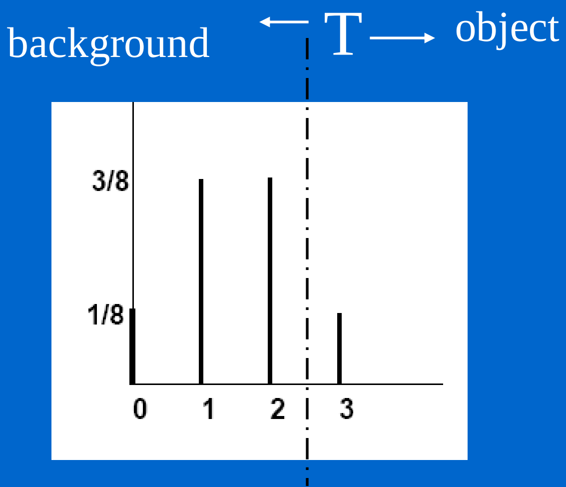
Otsu's Method (cont'd)
Mean and Variance

- Consider an image with L gray levels and its normalized histogram
  - $P(i)$ is the normalized frequency of i.
- Assuming that we have set the threshold at T, the normalized fraction of pixels that will be classified as background and object will be:

$$q_b(T) = \sum_{i=1}^{T} P(i),$$

$$q_o(T) = \sum_{i=T+1}^{L} P(i)$$

$$(q_b(T) + q_o(T) = 1)$$



Otsu's Method (cont'd)
Mean and Variance

- The mean gray-level value of the background and the object pixels will be:

$$\mu_b(T) = \frac{\sum_{i=1}^{T} iP(i)}{\sum_{i=1}^{T} P(i)} = \frac{1}{q_b(T)} \sum_{i=1}^{T} iP(i)$$

$$\mu_o(T) = \frac{\sum_{i=T+1}^{L} iP(i)}{\sum_{i=T+1}^{L} P(i)} = \frac{1}{q_o(T)} \sum_{i=T+1}^{L} iP(i)$$

- The mean gray-level value over the whole image ("grand" mean) is:

$$\mu = \frac{\sum_{i=1}^{L} iP(i)}{\sum_{i=1}^{L} P(i)} = \sum_{i=1}^{L} iP(i)$$

$$E[x] = \sum_{i=1}^{n} x_i P(X = x_i)$$

Otsu's Method (cont'd) Means and Variances

- The variance of the background and the object pixels will be:

$$\sigma_b^2(T) = \frac{\sum_{i=1}^{T}(i-\mu_b)^2 P(i)}{\sum_{i=1}^{T} P(i)} = \frac{1}{q_b(T)}\sum_{i=1}^{T}(i-\mu_b)^2 P(i)$$

$$\sigma_o^2(T) = \frac{\sum_{i=T+1}^{L}(i-\mu_o)^2 P(i)}{\sum_{i=T+1}^{L} P(i)} = \frac{1}{q_o(T)}\sum_{i=T+1}^{L}(i-\mu_o)^2 P(i)$$

- The variance of the whole image is:

$$Var(X) = \sum$$

$$\sigma^2 = \sum_{i=1}^{L}(i-\mu)^2 P(i)$$

Otsu's Method (cont'd)
Within-class and between-class variance

- It can be shown that the variance of the whole image can be written as follows:

$$\sigma^2 = \underbrace{q_b(T)\sigma_b^2(T) + q_o(T)\sigma_o^2(T)}_{\sigma_W^2(T)} \quad + \quad \underbrace{q_b(T)(\mu_b(T)-\mu)^2 + q_o(T)(\mu_o(T)-\mu)^2}_{\sigma_B^2(T)} =$$

$$\sigma_W^2(T) + \sigma_B^2(T)$$

$\sigma_W^2(T)$ within-class variance should be minimized!

$\sigma_B^2(T)$ between-class variance should be maximized!

Otsu's Method (cont'd) Determining the threshold

- Since the total variance does not depend on *T*, the *T* that minimizes σ_W^2 will also maximize σ_B^2
- Let us rewrite σ_B^2 as follows:

$$\sigma_B^2 = \frac{[\mu(T) - \mu q_b(T)]^2}{q_b(T) q_o(T)}$$

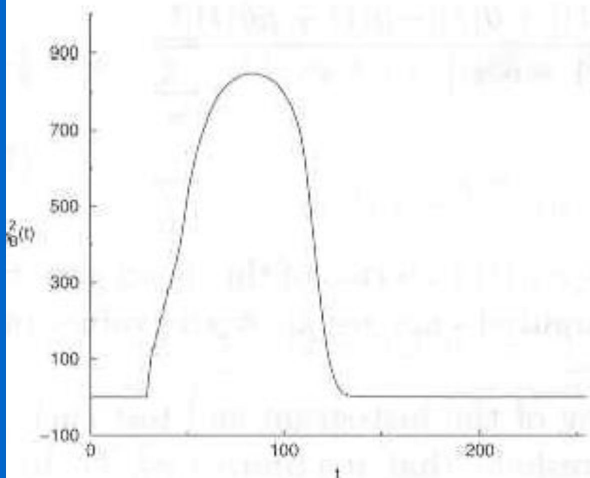
where

$$\mu(T) = \sum_{i=1}^{T} iP(i)$$

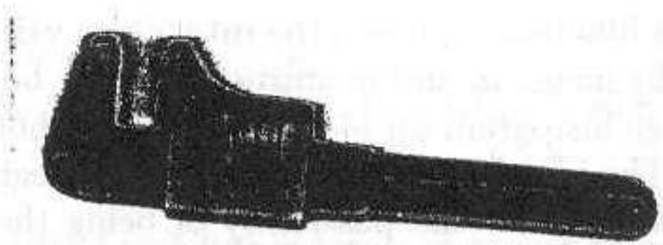
- Find the T value that maximizes σ_B^2

Otsu's Method (cont'd)
Determining the threshold

- Start from the beginning of the histogram and test each gray-level value for the possibility of being the threshold *T* that maximizes σ_B^2



$$\sigma_B^2 = \frac{[\mu(T) - \mu q_b(T)]^2}{q_b(T)q_o(T)}$$



(a) $\sigma_B(t)$ *versus* t.

(b) *Image thresholded with Otsu's threshold.*

Otsu's Method (cont'd)

- **Drawbacks of the Otsu's method**
  - The method assumes that the histogram of the image is bimodal (i.e., two classes).
  - The method breaks down when the two classes are very unequal (i.e., the classes have very different sizes)
    - In this case, σ_B^2 may have two maxima.
    - The correct maximum is not necessary the global one.
  - The method does not work well with variable illumination.

Effect of Illumination on Segmentation

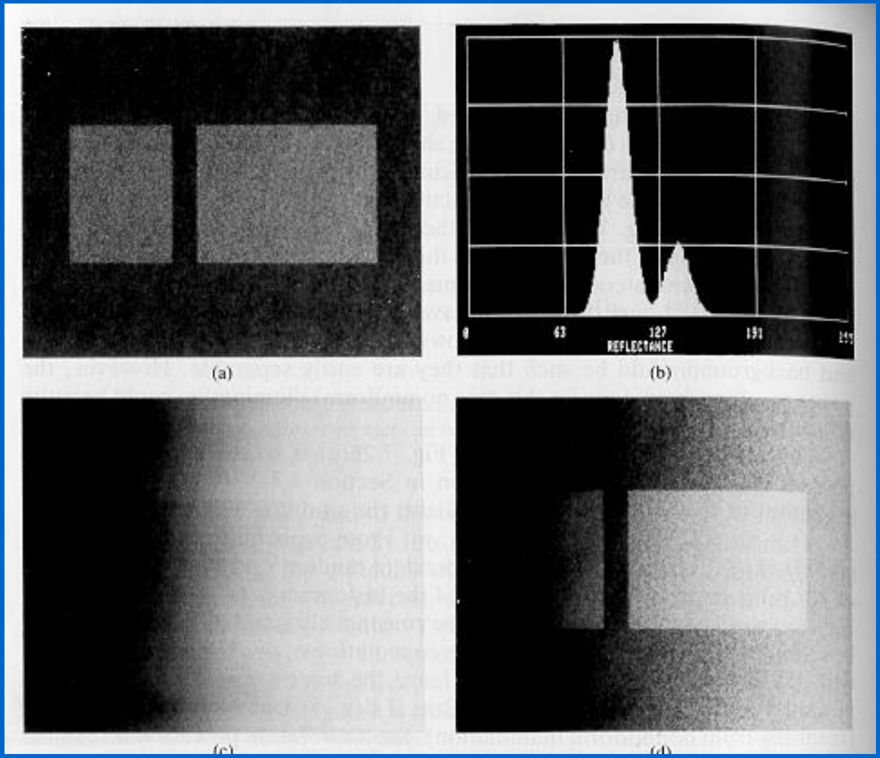
$$f(x, y) = i(x, y)r(x, y)$$

- How does illumination affect the histogram of an image?

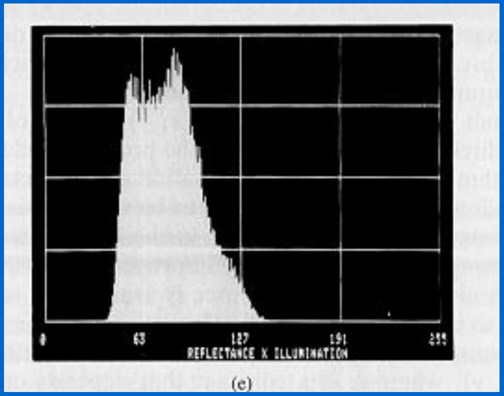
$$ln(f(x, y)) = ln(i(x, y)) + ln(r(x, y))$$

$$hist(ln(f(x, y))) = hist(ln(i(x, y))) + hist(ln(r(x, y)))$$

Effect of Illumination on Segmentation (cont'd)



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



(e)

Handling non-uniform illumination: a laboratory solution

- Suppose that $f(x, y) = i(x, y)r(x, y)$, where $i(x, y)$ is non-uniform
- Obtain an image of the illumination field.
  - e.g., project the illumination pattern on a surface with uniform reflectance (e.g., a white surface)

$$\boxed{g(x, y) = k\, i(x, y)}$$

- Normalize f(x,y)

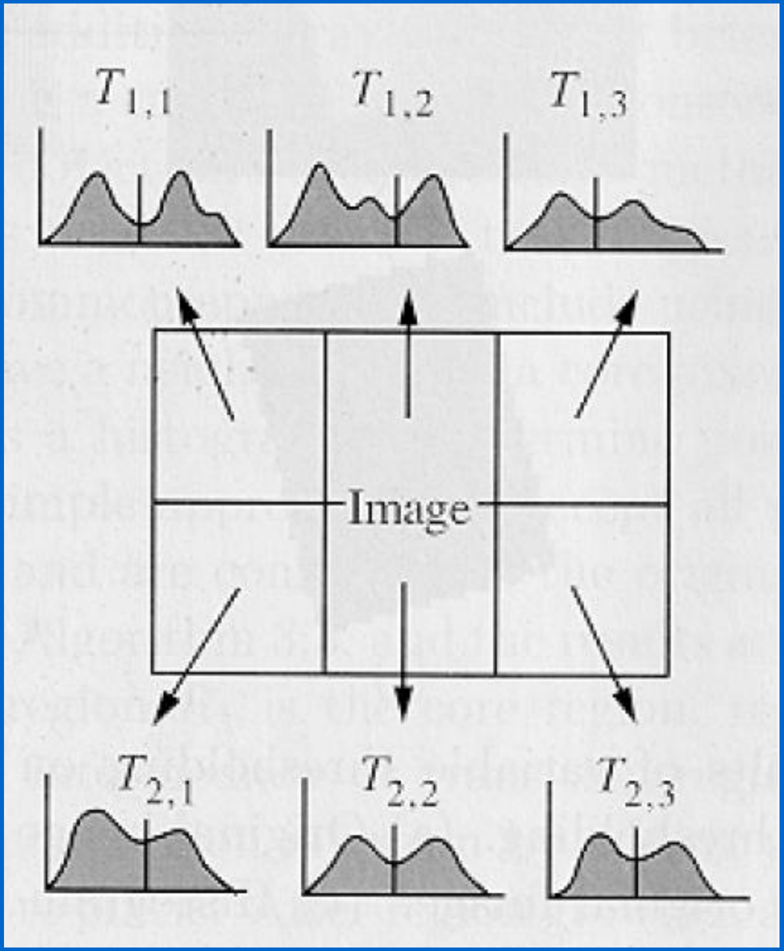
$$\boxed{h(x, y) = f(x, y)/g(x, y) = r(x, y)/k}$$

- If $r(x, y)$ can be segmented using T, then $h(x, y)$ can be segmented using T/k

Handling non-uniform illumination: local thresholding

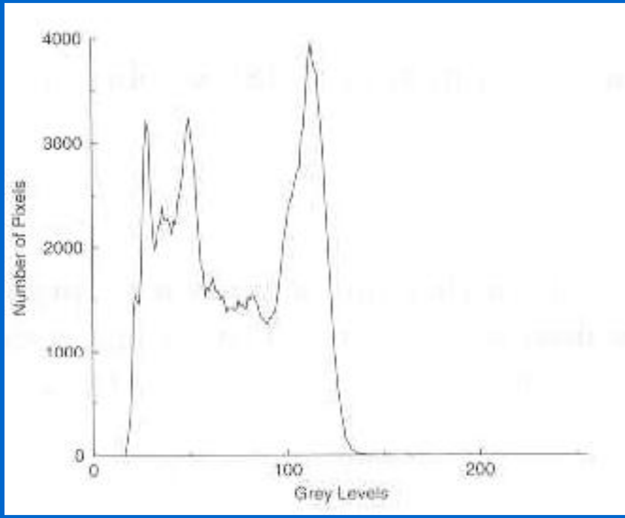
- A single threshold will not work well when we have uneven illumination due to shadows or due to the direction of illumination.
- Idea:
  - Partition the image into *m x m* subimages (i.e., illumination is likely to be uniform in each subimage).
  - Choose a threshold T_{ij} for each subimage.

Handing non-uniform illumination: local thresholding (cont'd)

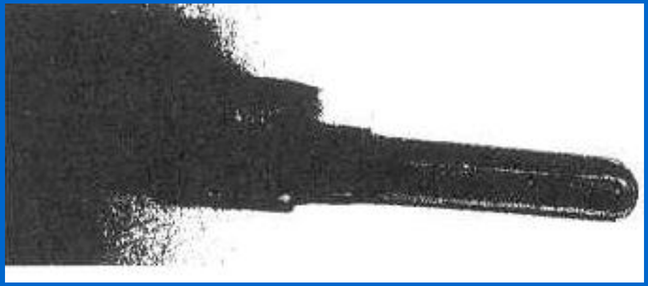


This approach might lead to subimages having simpler histogram (e.g., bimodal)

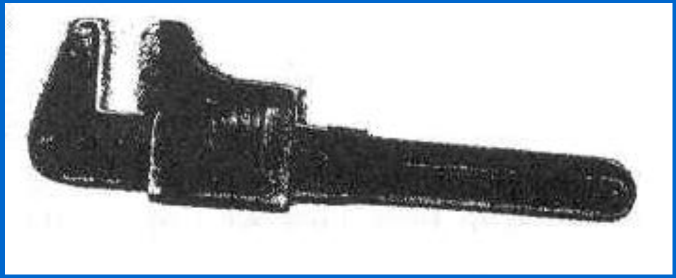
Handling non-uniform illumination: local thresholding (cont'd)



single threshold



local thresholding using Otsu's method



Drawbacks of Thresholding

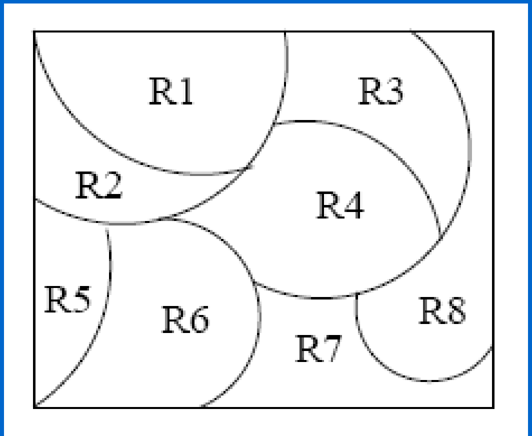
- Threshold selection is not always straightforward.
- Pixels assigned to a single class need not form coherent regions as the spatial locations of pixels are completely ignored.
  - Only hysteresis thresholding considers some form of spatial proximity.

Other Methods

- Region Growing
- Region Merging
- Region Splitting
- Region Splitting and Merging

Properties of region-based segmentation

- Partition an image R into sub-regions $R_1, R_2, \ldots, R_n$



- Suppose $P(R_i)$ is a logical predicate, that is, a property that the pixel values of region R_i satisfy

(e.g., the gray level values are between 100 and 120).

Properties of region-based segmentation (cont'd)

- The following properties must hold true:

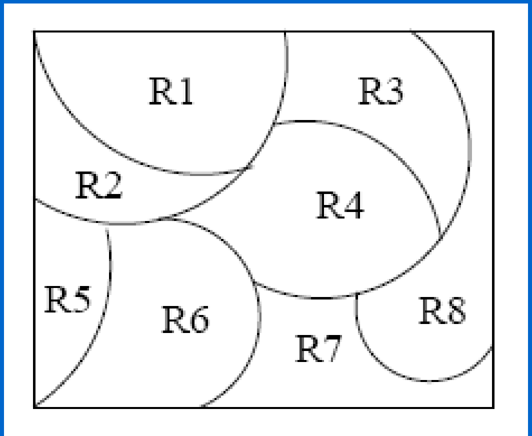
(1) $R_1 \cup R_2 \cup \cdots \cup R_n = R$

(2) R_i is connected

(3) $R_i \cap R_j = empty$

(4) $P(R_i) = \text{True}$

(5) $P(R_i \cup R_j)$=False



Region Growing

- Region-growing approaches exploit the fact that pixels which are close together have similar gray values.
- Start with a single pixel (**seed**) and add new pixels slowly

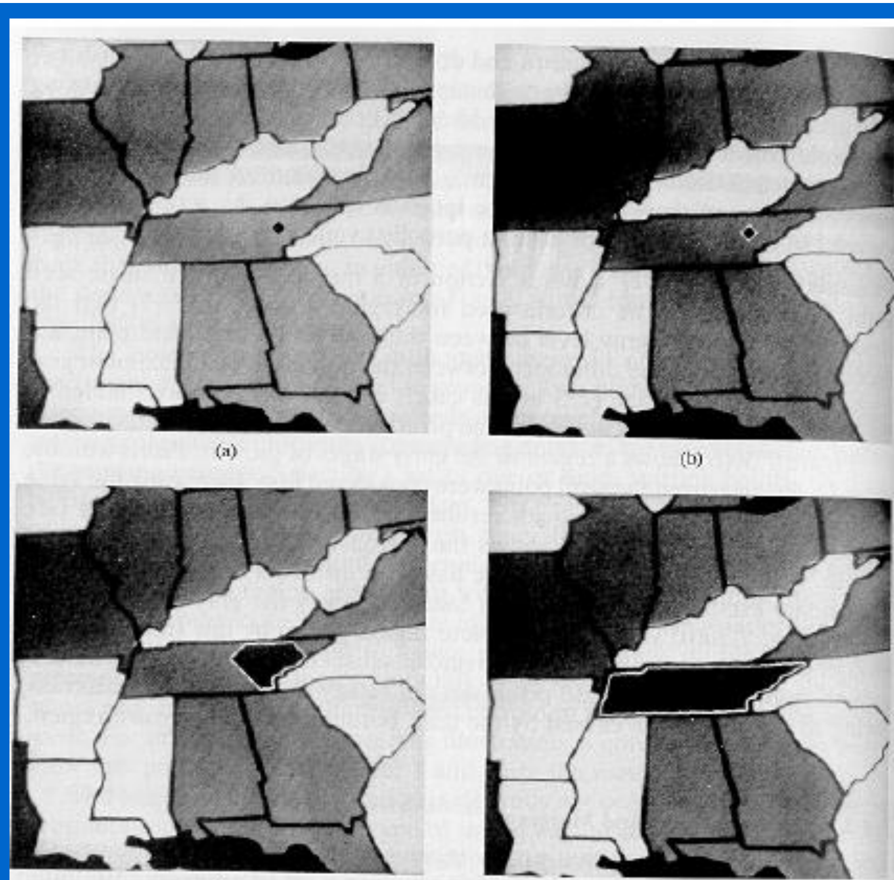
(1) Choose the seed pixel

(2) Check the neighboring pixels and add them to the region if they are similar to the seed

(3) Repeat step 2 for each of the newly added pixels; stop if no more pixels can be added.

Region Growing (cont'd)

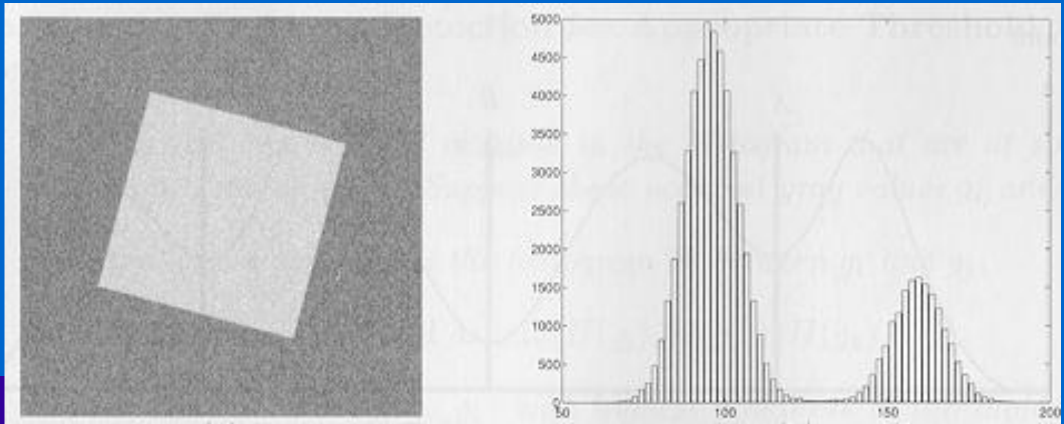
8 neighbors, predicate: $|z - z_{seed}| < 0.1(\max_z - \min_z)$



Multiple regions can be grown in parallel using multiple seeds

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 choose the gray-level values corresponding to the strongest peaks



Region Growing (cont'd)

- **How do we choose the similarity criteria (predicate)?**
  - The homogeneity predicate can be based on any characteristic of the regions in the image such as:
    - average intensity
    - variance
    - color
    - texture

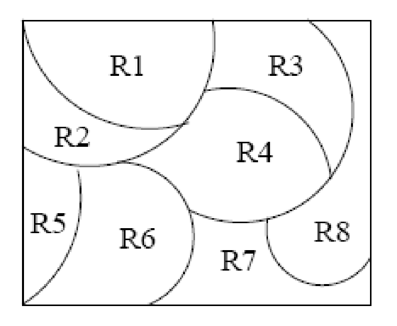
Region Merging

- Region merging operations eliminate false boundaries and spurious regions by merging adjacent regions that belong to the same object.
- Merging schemes begin with a partition satisfying condition (4) (e.g., regions produced using thresholding).

$$(4)\ P(R_i) = \text{True}$$

- Then, they proceed to fulfill condition (5) by gradually merging adjacent image regions.

$$(5)\ P(R_i \cup R_j) = \text{False}$$



Region Merging (cont'd)

(1) Form initial regions in the image.

(2) Build a regions adjacency graph (RAG).

(3) For each region do:

(3.1) Consider its adjacent region and test to see if they are similar.

(3.2) For regions that are similar (i.e., $P(R_i \cup R_j)$=True), merge them and modify the RAG.

(4) Repeat step 3 until no regions are merged.

How to determine region similarity?

(1) Based on the gray values of the regions – examples:

- Compare their mean intensities.
- Use surface fitting to determine whether the regions may be approximated by one surface.
- Use hypothesis testing to judge the similarity of adjacent region

(2) Based on the weakness of boundaries between the regions.

Region merging using hypothesis testing

- This approach considers whether or not to merge adjacent regions based on the probability that they will have the same statistical distribution of intensity values.

- Assume that the gray-level values in an image region are drawn from a Gaussian distribution
  - Parameters can be estimated using sample mean/variance:

$$p(g_i) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(g_i-\mu)^2}{2\sigma^2}}$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} g_i, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(g_i - \hat{\mu})^2$$

Region merging using hypothesis testing (cont'd)

- Given two regions R_1 and R_2 with m_1 and m_2 pixels respectively, there are two possible hypotheses:

H0: Both regions belong to the same object.

The intensities are all drawn from a single Gaussian distribution $N(\mu_0, \sigma_0)$

H1: The regions belong to different objects.

The intensities of each region are drawn from separate Gaussian distributions $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$

Region merging using hypothesis testing (cont'd)

- The joint probability density under H_0, assuming all pixels are independently drawn, is given by:

$$p(g_1, g_2, \ldots, g_{m_1+m_2}|H_0) = \prod_{i=1}^{m_1+m_2} p(g_i|H_0) = \frac{1}{(\sqrt{2\pi\sigma_0})^{m_1+m_2}} e^{-\frac{(m_1+m_2)}{2}}$$

- The joint probability density under H_1 is given by

$$p(g_1, g_2, \ldots, g_{m_1+m_2}|H_1) = \frac{1}{(\sqrt{2\pi\sigma_1})^{m_1}} e^{-\frac{m_1}{2}} \frac{1}{(\sqrt{2\pi\sigma_2})^{m_2}} e^{-\frac{m_2}{2}}$$

Region merging using hypothesis testing (cont'd)

- The likelihood ratio is defined as the ratio of the probability densities under the two hypotheses:

$$L = \frac{p(g_1, g_2, \ldots, g_{m_1+m_2}|H_1)}{p(g_1, g_2, \ldots, g_{m_1+m_2}|H_0)} = \frac{\sigma_0^{m_1+m_2}}{\sigma_1^{m_1}\ \sigma_2^{m_2}}$$

- If the likelihood ratio is below a threshold value, there is strong evidence that there is only one region and the two regions may be merged.

Region merging by removing weak edges

- The idea is to combine two regions if the boundary between them is weak.
- A weak boundary is one for which the intensities on either side differ by less than some threshold.
- The relative lengths between the weak boundary and the region boundaries must be also considered.

Region merging by removing weak edges (cont'd)

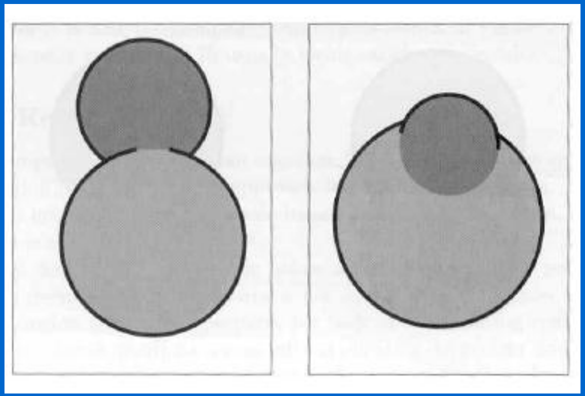
- **Approach 1:** merge adjacent regions R_1 and R_2 if

$$\frac{W}{S} > T_2$$

where:

W is the length of the weak part of the boundary

S = $min(S_1, S_2)$ is the minimum of the perimeter of the two regions.



Region merging by removing weak edges (cont'd)

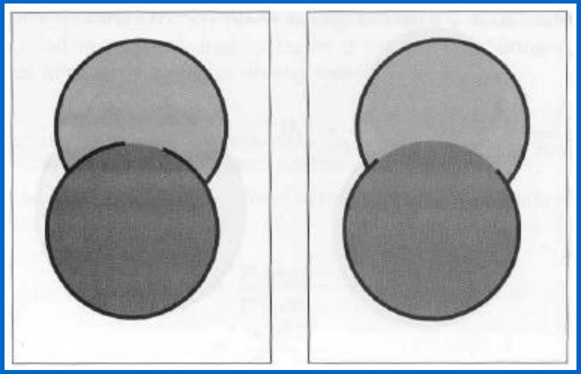
- **Approach 2:** Merge adjacent regions R_1 and R_2 if

$$\frac{W}{S} > T_3$$

where:

W is the length of the weak part of the boundary

S is the common boundary between R_1 and R_2.



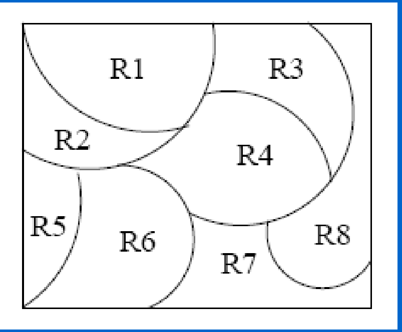
Region Splitting

- Region splitting operations add missing boundaries by splitting regions that contain parts of different objects.
- Splitting schemes begin with a partition satisfying condition (5), for example, the whole image.

$$(5)\ P(R_i \cup R_j) = \text{False}$$

- Then, they proceed to satisfy condition (4) by gradually splitting image regions.

$$(4)\ P(R_i) = \text{True}$$



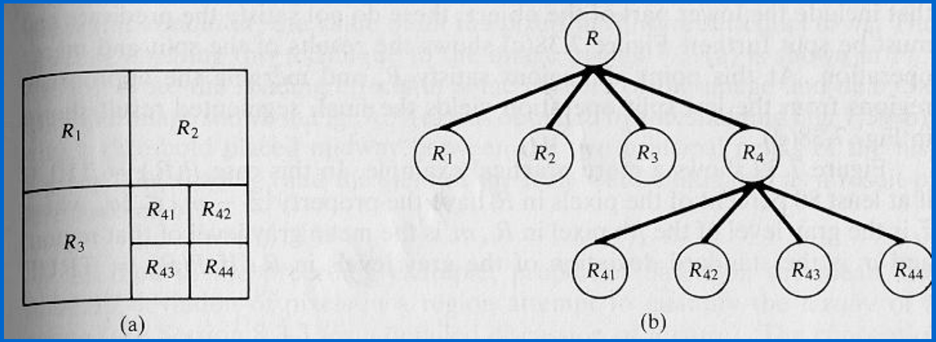
Region Splitting (cont'd)

- Two main difficulties in implementing this approach:
  - Deciding when to split a region (e.g., use variance, surface fitting).
  - Deciding how to split a region.

Regular Decomposition

(1) If $P(R)$=False, split R into four quadrants

(2) If P is false on any quadrant, subsplit



Region Splitting and Merging

- Splitting or merging might not produce good results when applied separately.
- Better results can be obtained by interleaving merge and split operations.
- This strategy takes a partition that possibly satisfies neither condition (4) or (5) with the goal of producing a segmentation that satisfies both conditions.

(1) $R_1 \cup R_2 \cup \cdots \cup R_n = R$

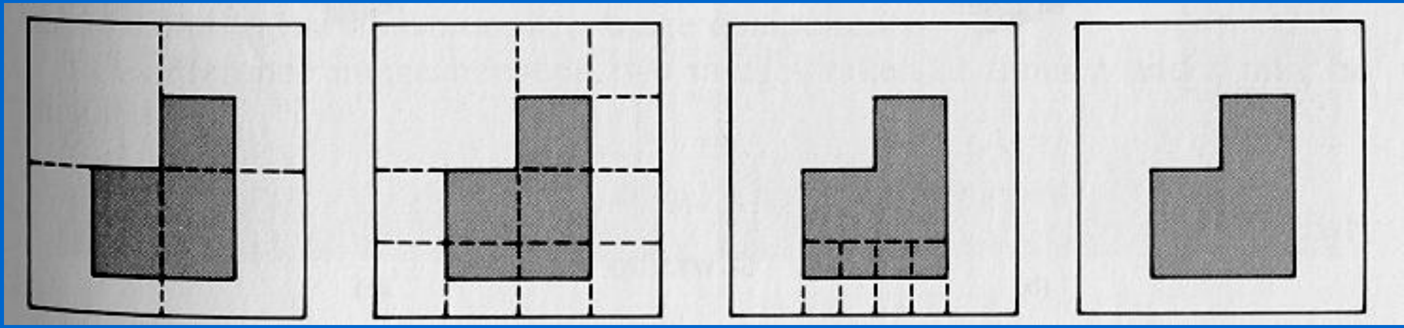
(2) R_i is connected

(3) $R_i \cap R_j = empty$

(4) $P(R_i) = \text{True}$

(5) $P(R_i \cup R_j)$=False

Region Splitting and Merging (cont'd)

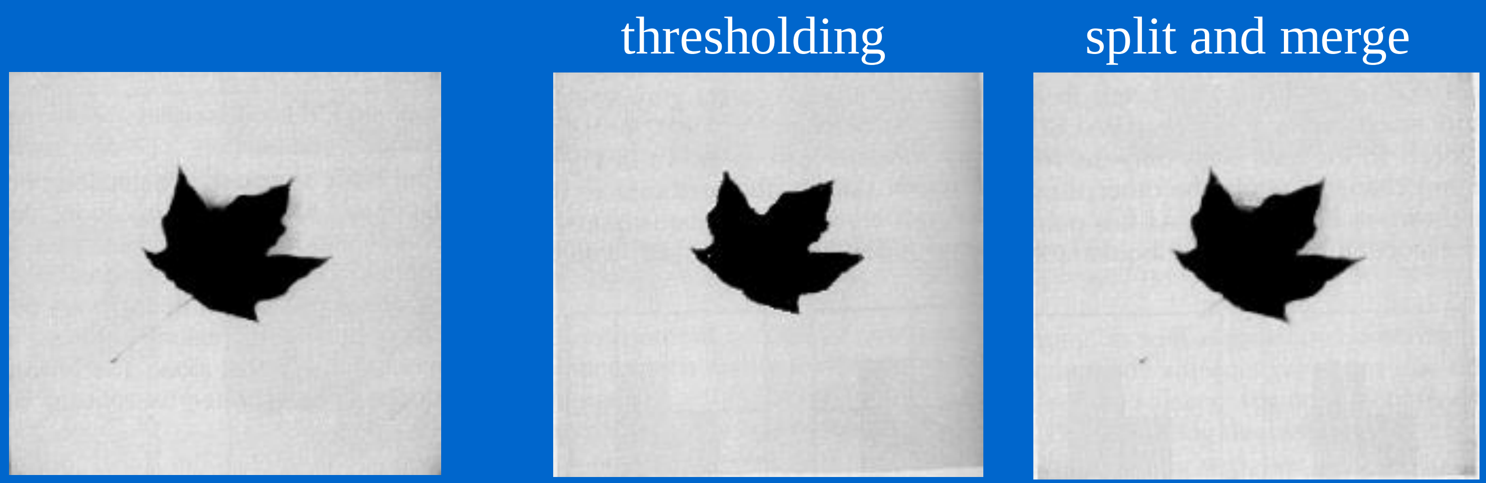


(1) Split into four disjointed quadrants any region R_i where $P(R_i)$=False

(2) Merge any adjacent regions R_j and R_k for which $P(R_j \cup R_k)$=True;

(3) Stop when no further merging or splitting is possible

Region Splitting and Merging (cont'd)



$$P(R_i) = True \text{ if}$$

$$|z_i - m_i| \leq 2\sigma_i \text{ for 80\% of the pixels in } R_i$$

(m_i, σ_i are the mean and standard deviation of pixels in R_i)