Thresholding based segmentation

- Now we will continue to look at the problem of segmentation, this time though in terms of thresholding
- In particular we will look at:
  - What is thresholding?
  - Simple Global thresholding
  - Optimal Global thresholding: Otsu's Algorithm
  - Adaptive/dynamic/local thresholding

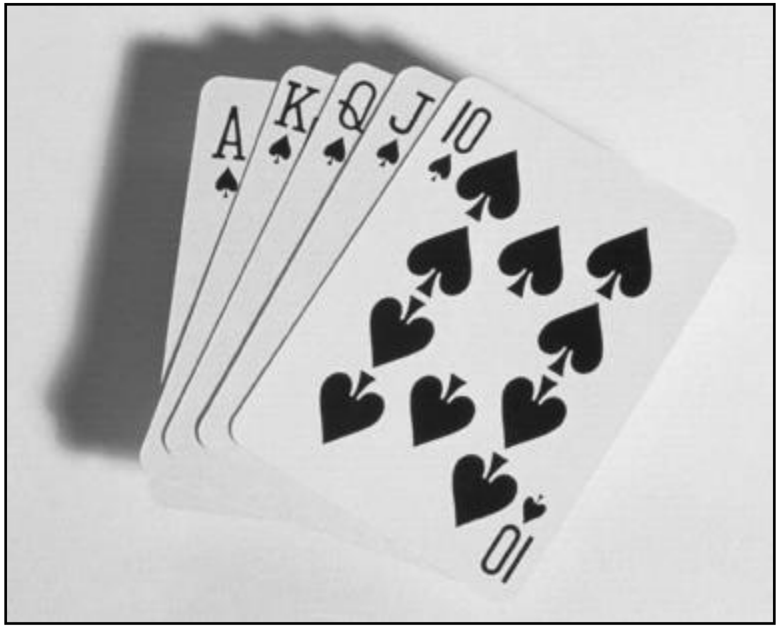
Thresholding

- Thresholding is popular approach because of its intuitive property, simplicity of implementation and computation speed
- We have talked about simple single value thresholding already
- Single value thresholding can be given mathematically as follows:

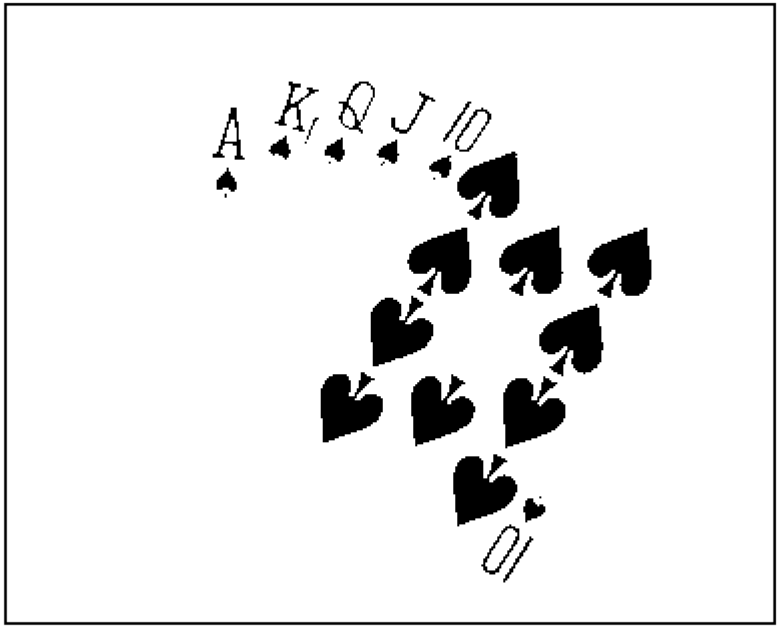
$$g(x,y)=\begin{cases}1\,if\;\;f(x,y)>T\\0\,if\;\;f(x,y)\le T\end{cases}$$

Thresholding Example

- Imagine a poker playing robot that needs to visually interpret the cards in its hand



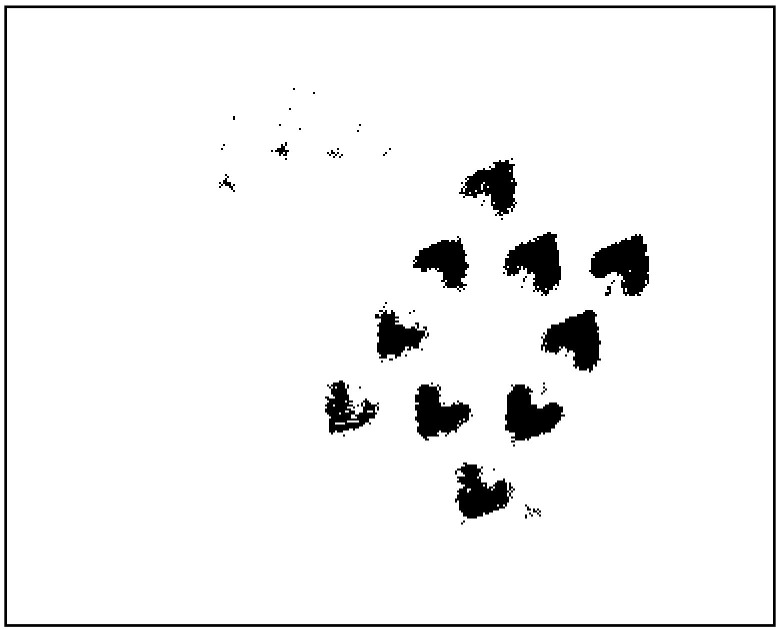
Original Image



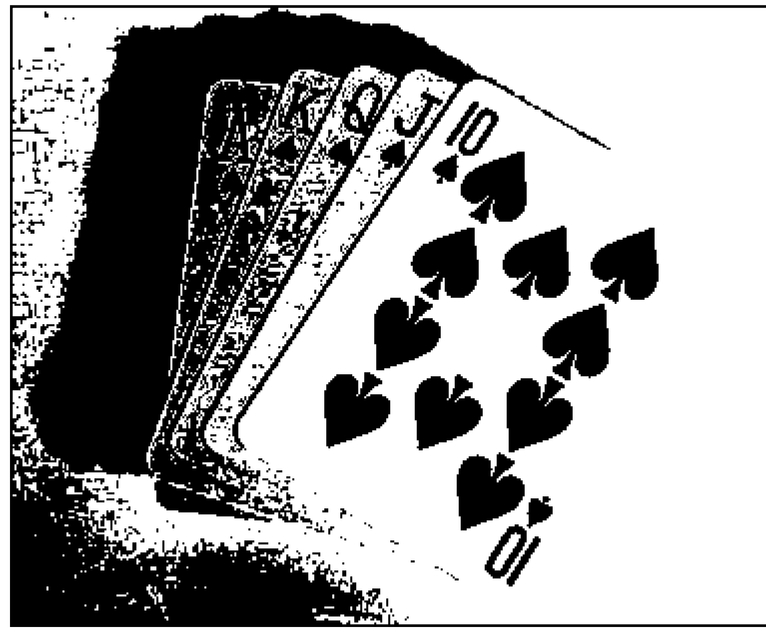
Thresholded Image

But Be Careful

- If you get the threshold wrong the results can be disastrous



Threshold Too Low



Threshold Too High

Noise in Thresholding

- Difficulty in determining the threshold due to noise

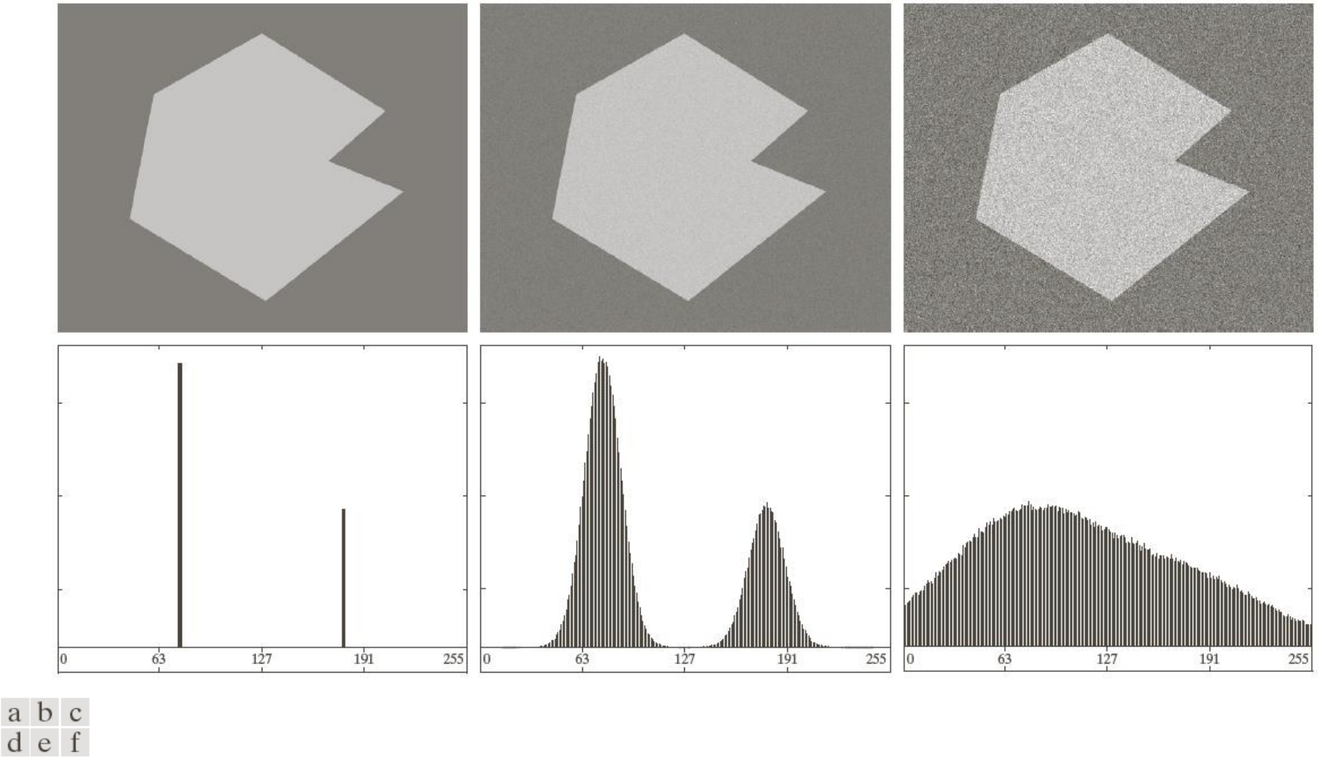
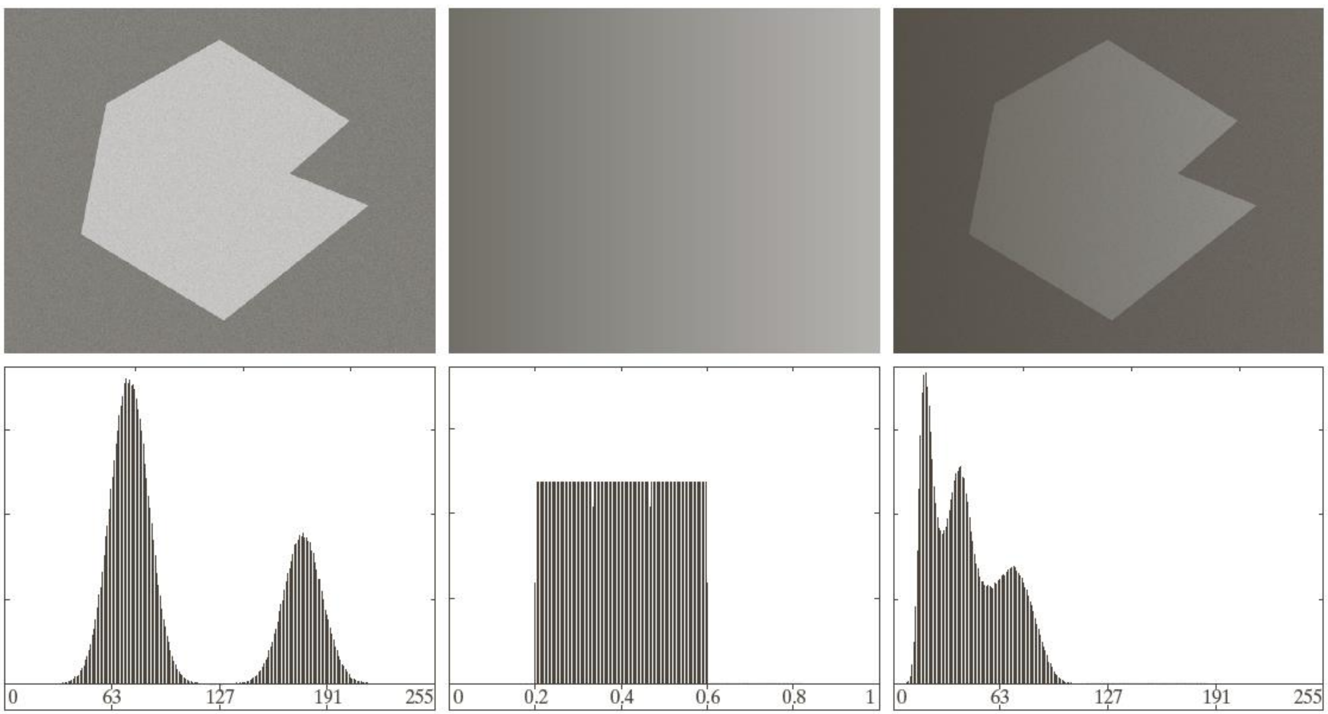


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Illumination in Thresholding

- Difficulty in determining the threshold due to non uniform illumination or reflectance



a b c
d e f

FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Basic Global Thresholding

- Based on the histogram of an image
- Partition the image histogram using a single global threshold
- The success of this technique very strongly depends on how well the histogram can be partitioned

Basic Global Thresholding Algorithm

- The basic global threshold, T, is calculated as follows:
  1. Select an initial estimate for T (typically the average grey level in the image)
  2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels $\leq$ T
  3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

Basic Global Thresholding Algorithm

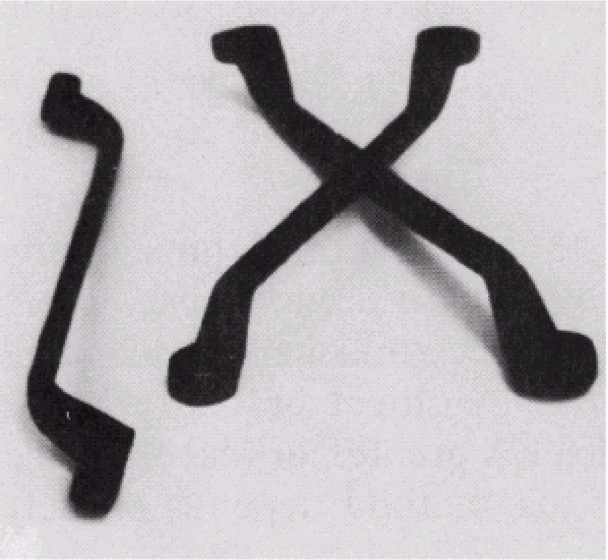
4. Compute a new threshold value:

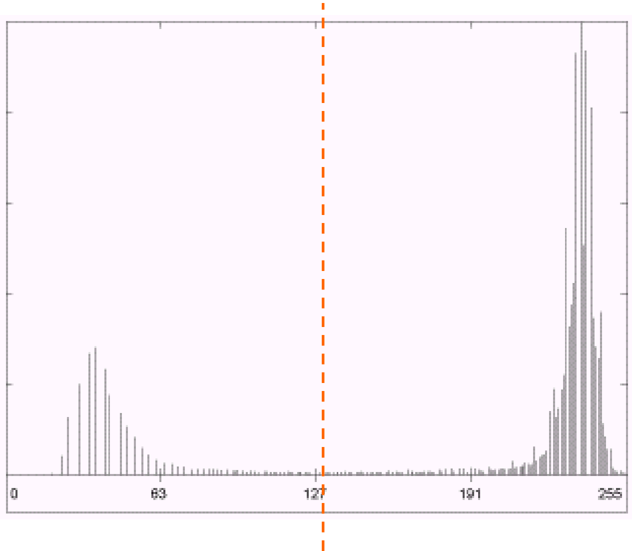
$$T = \frac{\mu_1 + \mu_2}{2}$$

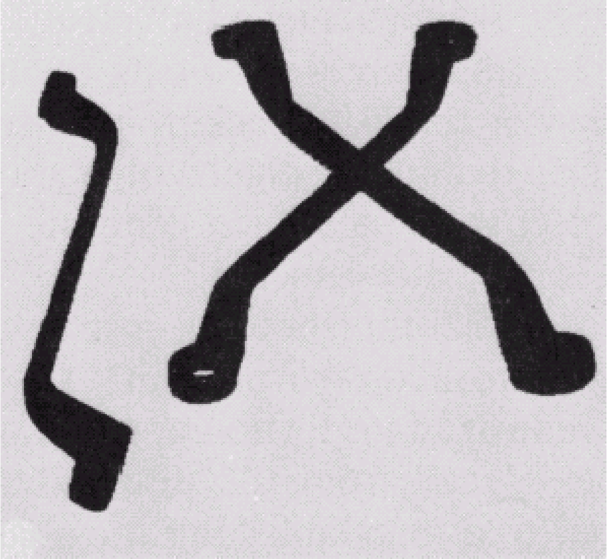
5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞

- This algorithm works very well for finding thresholds when the histogram is suitable

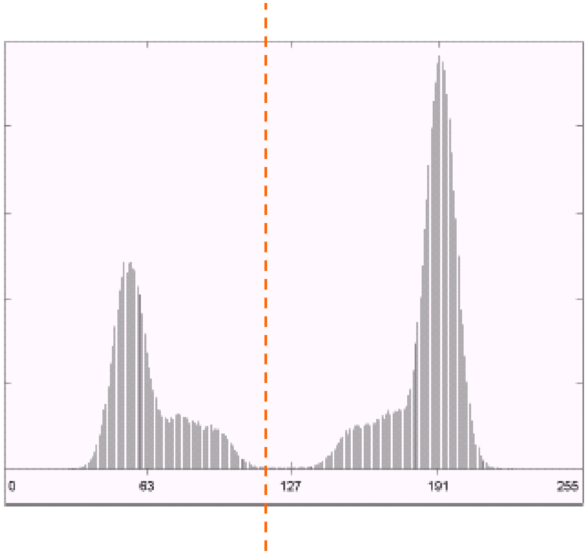
Thresholding Example 1







Thresholding Example 2



Optimum Global Thresholding (Otsu's Method)

- Otsu's method selects the threshold by minimizing the within-class variance
  - Try to make each cluster as tight as possible thus (hopefully!) minimizing their overlap.
- Assumptions: Histogram (and the image) are *bimodal and* uniform illumination

Otsu's Method (contd..)

The *weighted within-class variance* is:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Where the class probabilities are estimated as:

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

And the class means are given by:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

Otsu's Method (contd..)

Finally, the individual class variances are:

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \qquad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

- Run through the full range of *t* values [1,256] and pick the value that minimizes $\sigma_w^2(t)$.
- But the relationship between the within-class and between-class variances can be exploited to generate a recursion relation that permits a much faster calculation.
- *Between-class variance* is the sum of weighted squared distances between the class means and the grand mean

Otsu's Method (contd..)

The total variance is the sum of the within-class variances (weighted) and the between class variance. After some algebra, we can express the total variance as...

$$\sigma^2 = \underbrace{\sigma_w^2(t)}_{\text{Within-class, from before}} + \underbrace{q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2}_{\text{Between-class, } \sigma_B^2(t)}$$

Since the total is constant and independent of *t*, the effect of changing the threshold is merely to move the contributions of the two terms back and forth.

So, *minimizing the within-class variance is the same as maximizing the between-class variance*.

The nice thing about this is that we can compute the quantities in $\sigma_B^2(t)$ *recursively* as we run through the range of *t* values.

Otsu's Method (Finally...)

Initialization... $q_1(1) = P(1)$; $\mu_1(0) = 0$

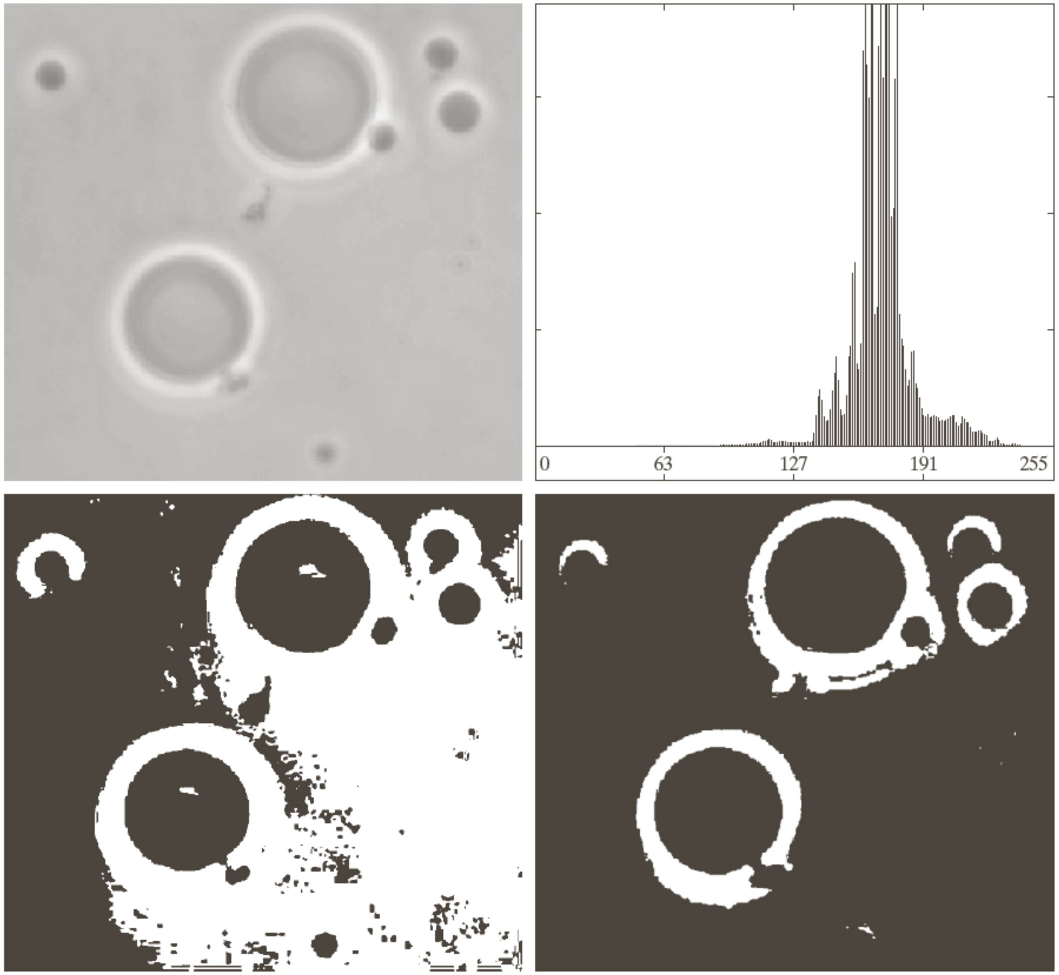
Recursion...

$$q_1(t+1) = q_1(t) + P(t+1)$$

$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)}$$

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$

Otsu's example

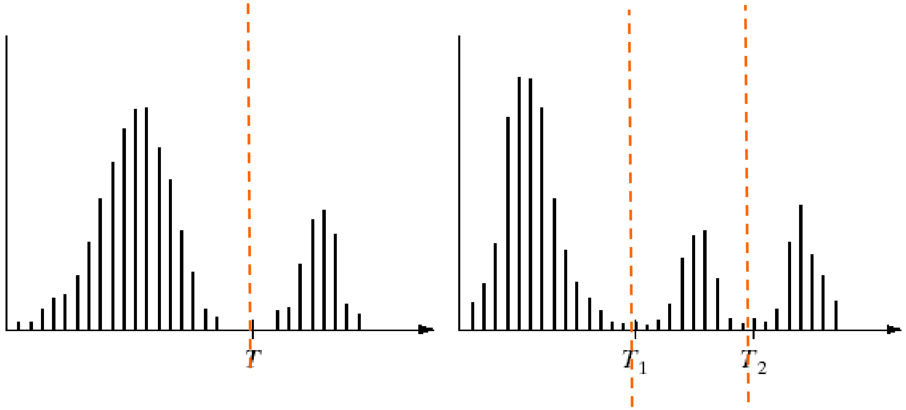


a b
c d

FIGURE 10.39
(a) Original image.
(b) Histogram (high peaks were clipped to highlight details in the lower values).
(c) Segmentation result using the basic global algorithm from Section 10.3.2.
(d) Result obtained using Otsu's method.
(Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Problems With Single Value Thresholding

- Single value thresholding only works for bimodal histograms
- Images with other kinds of histograms need more than a single threshold



Otsu's method extended

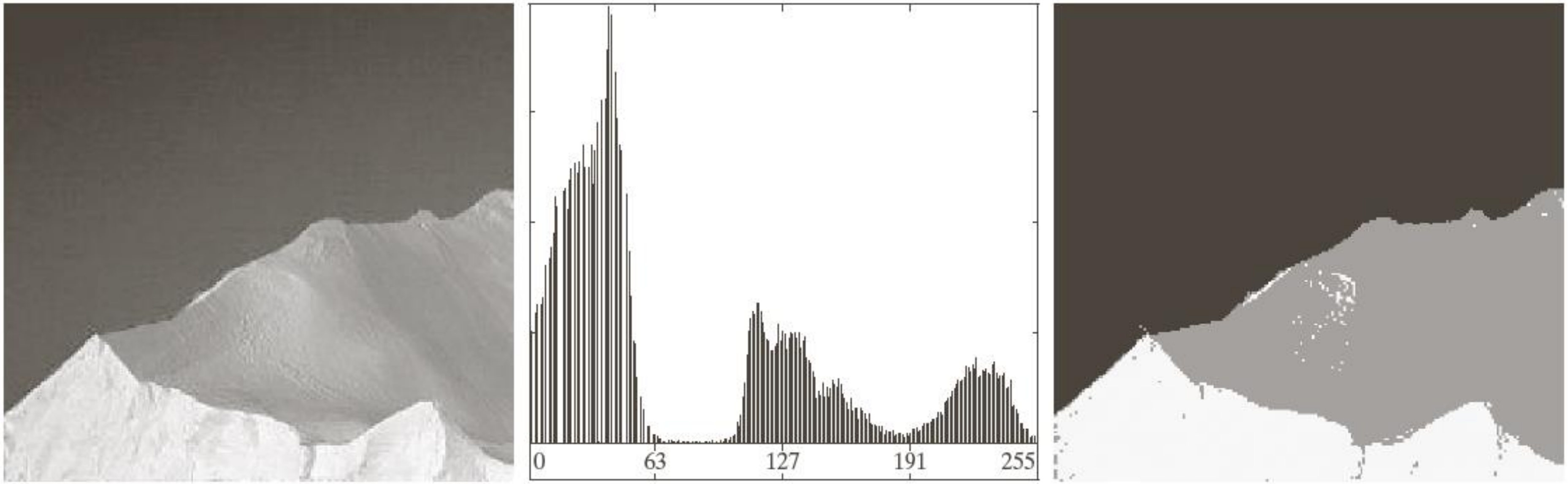
- The method may be extended to multiple thresholds
- In practice for more than 2 thresholds (3 segments) more advanced methods are employed.
- For three classes, the between-class variance is:

$$\sigma_B^2(k_1,k_2) = P_1(k_1)[m_1(k_1) - m_G] + P_2(k_1,k_2)[m_2(k_1,k_2) - m_G] + P_3(k_2)[m_3(k_2) - m_G]$$

- The thresholds are computed by searching all pairs for values:

$$(k_1^*, k_2^*) = \max_{0 \le k_1 < k_2 \le L-1} \{\sigma_B^2(k_1,k_2)\}$$

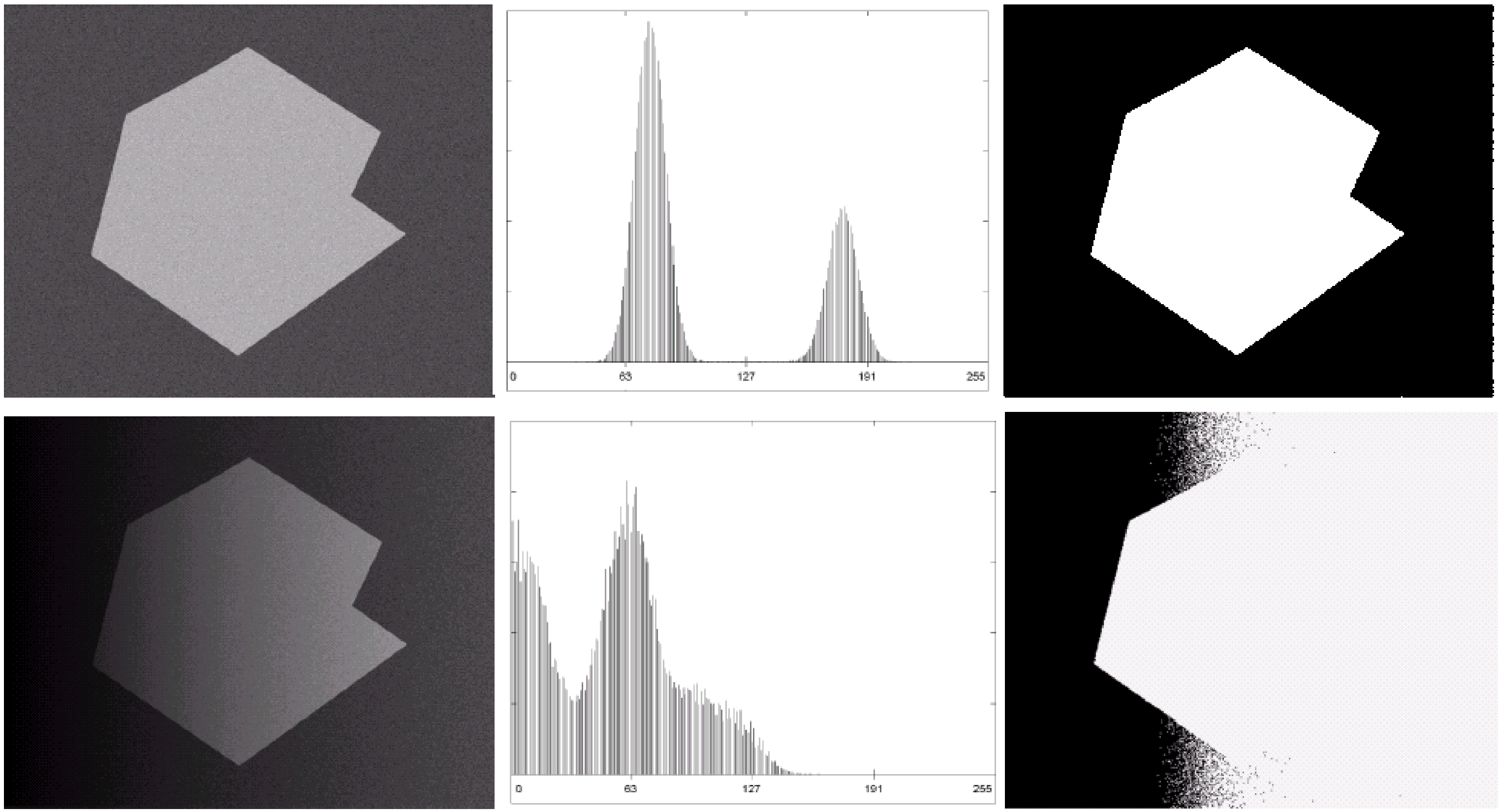
Otsu's dual threshold example



a b c

FIGURE 10.45 (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

Single Value Thresholding and Illumination



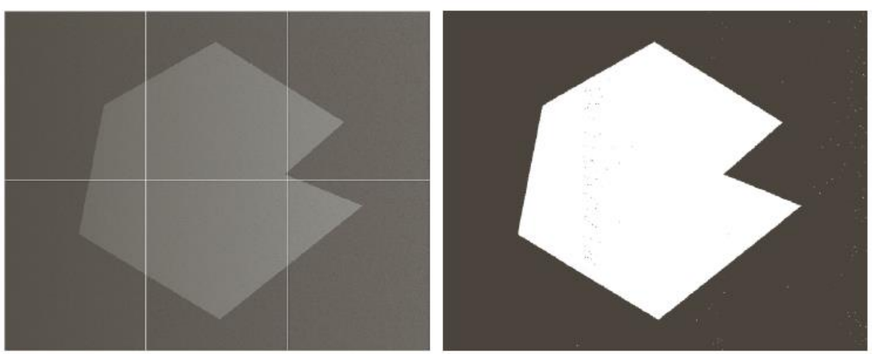
- Uneven illumination can really upset a single valued thresholding scheme

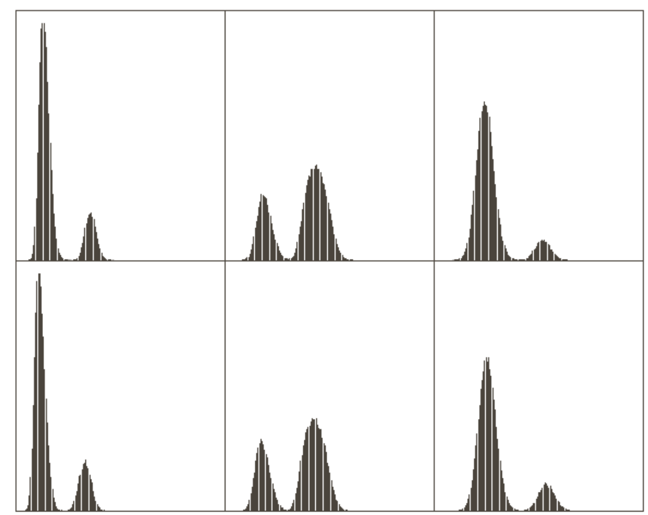
Basic Adaptive Thresholding

- An approach to handling situations in which single value thresholding will not work is to divide an image into sub images and threshold these individually
- Since the threshold for each pixel depends on its location within an image this technique is said to *adaptive*

Basic Adaptive Thresholding Example

- The image below shows an example of using adaptive thresholding with the image shown previously





Variable thresholding based on local processing

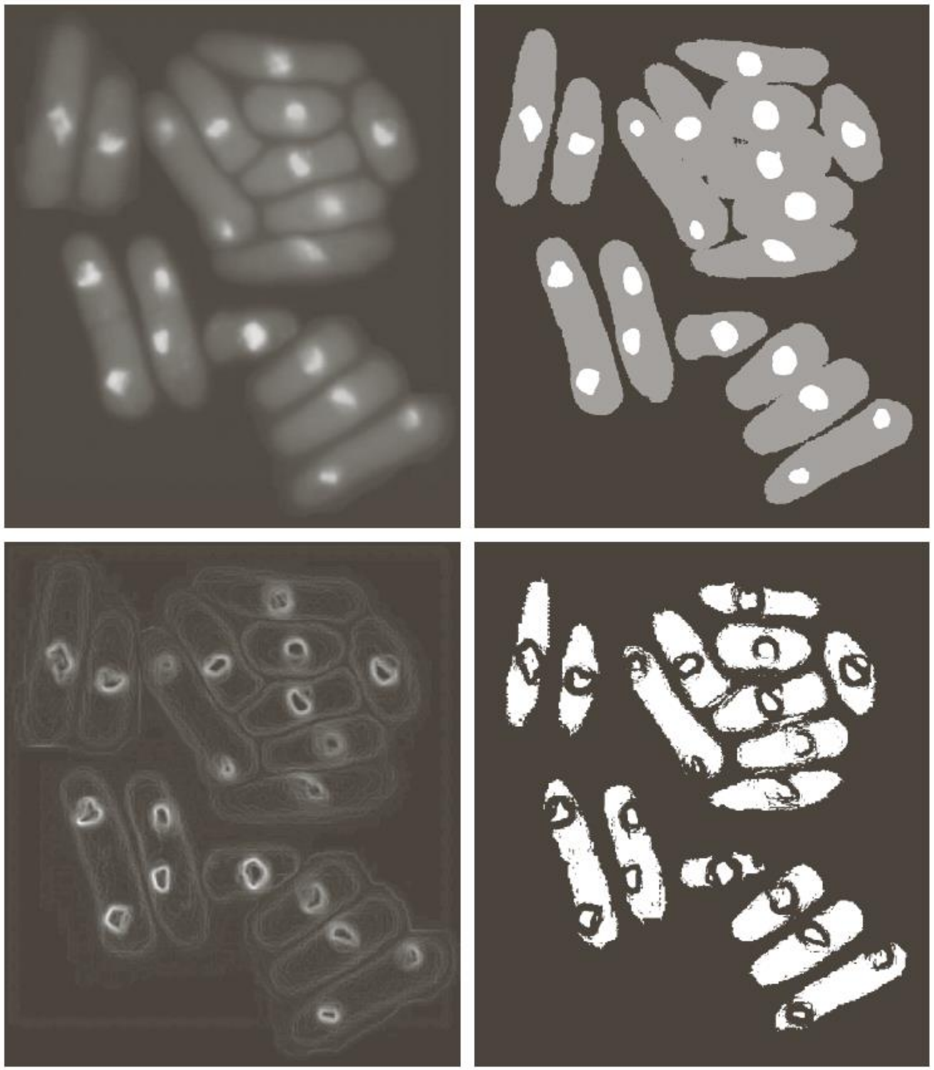
Use of local image properties.

- Compute a threshold for every single pixel in the image based on its neighborhood (m_{xy}, $\sigma_{xy}, \ldots$).

$$g(x,y)=\begin{cases}1 & Q(\text{local properties}) \text{ is true}\\ 0 & Q(\text{local properties}) \text{ is false}\end{cases}$$

$$Q(\sigma_{xy},m_{xy})=\begin{cases}\text{true} & f(x,y)>a\sigma_{xy}\ \ AND\ \ f(x,y)>bm_{xy}\\ \text{false} & \text{otherwise}\end{cases}$$

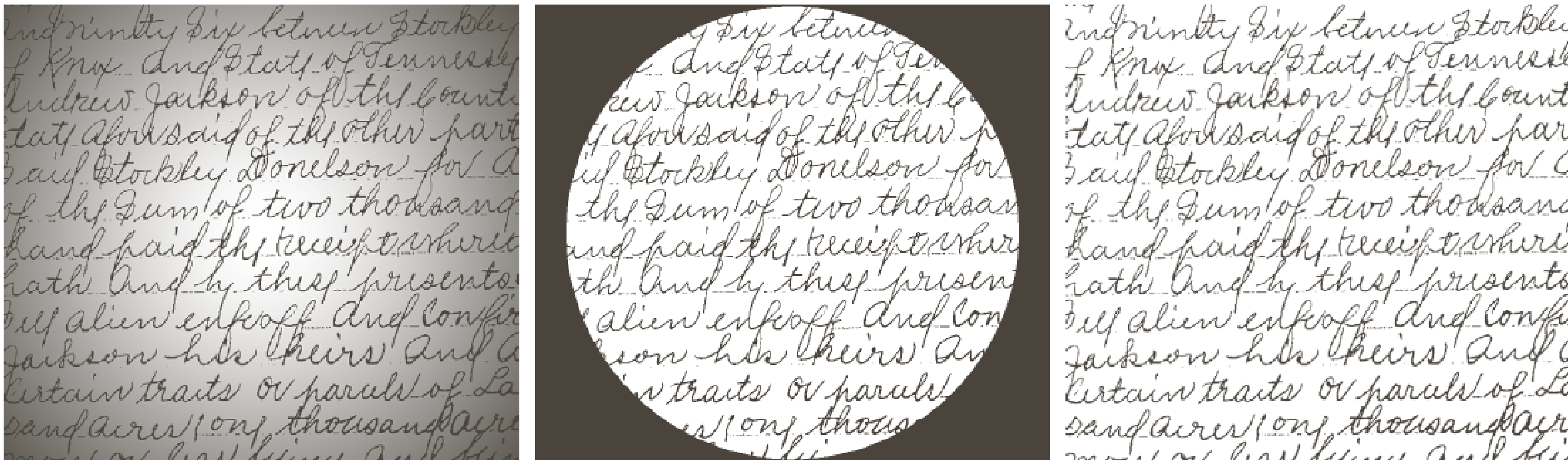
Example of local thresholding



a b
c d

FIGURE 10.48 (a) Image from Fig. 10.43. (b) Image segmented using the dual thresholding approach discussed in Section 10.3.6. (c) Image of local standard deviations. (d) Result obtained using local thresholding.

local thresholding using moving average



a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

Summary

- In this lecture we looked at thresholding based segmentation techniques
- We saw the basic global and ostu's global thresholding algorithm
- We also saw adaptive thresholding methods

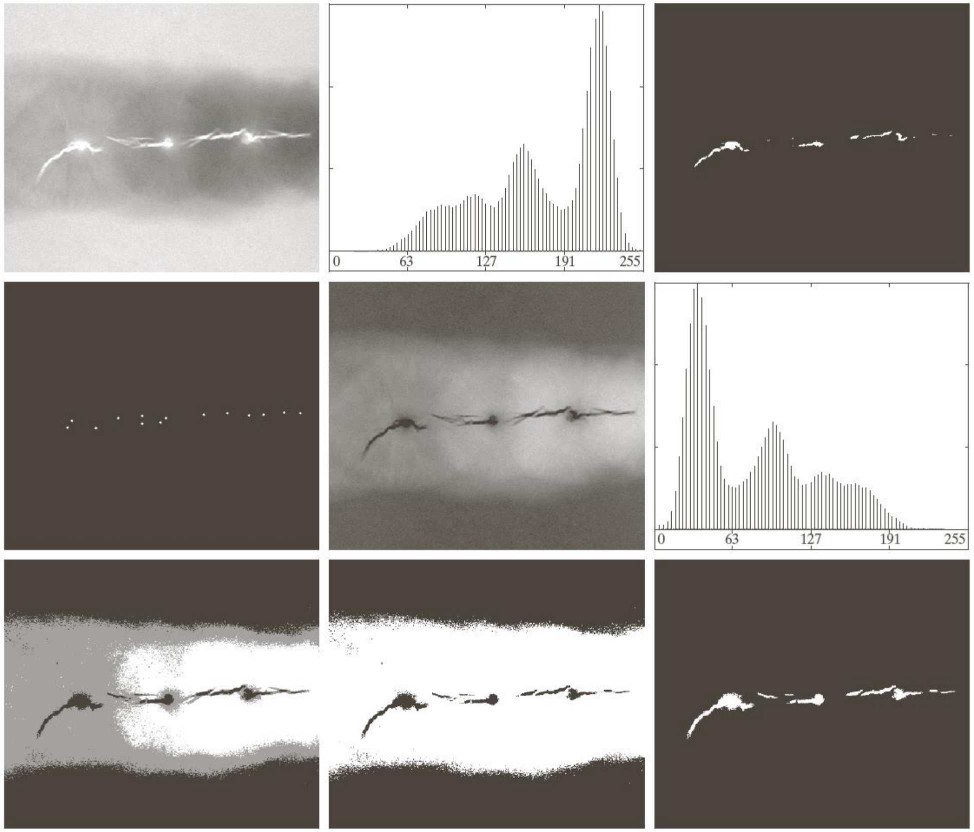
Region based segmentation

- **Thresholding** used global criterion to decide whether a pixel belong to a region
- **Region growing:** Conversely, local methods will accumulate pixels into a region: starting from a single pixel we would inspect the neighbours, adding them to the region if they are similar.
- Issues to be addressed: 1) What region properties are to be used 2) How do we decide that a pixel should be part of the region.

Basic region growing algorithm

- Let f(x,y) be input image, *S(x,y) denote seed array, and Q be the predicate*
- **Algorithm**
  - Find all connected components in *S(x,y)and erode them to 1 pixel.*
  - Form image $f_q(x,y)=1$ *if f(x,y)satisfies the predicate Q.*
  - Form image *g(x,y)=1 for all pixels in* $f_q(x,y)$ *that are 8-connected to any seed point in S(x,y).*
  - Label each connected component in *g(x,y)with a different label.*

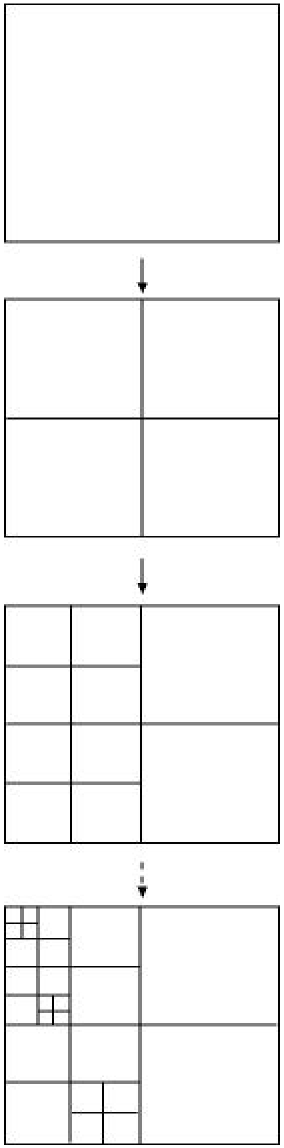
Region Growing



The weld is very bright. The predicate used for region growing is to compare the absolute difference between a seed point and a pixel to a threshold. If the difference is below it we accept the pixel as crack.

Region Splitting and Merging

- Previous method is accurate to pixel level but slow.
- Alternate is to split the image to sub-images that do not satisfy a predicate
- If only splitting was used, the final partition would contain adjacent regions with identical properties.
- A merging step follows that merges adjacent regions satisfying the predicate.
- Execution is more rapid but the outlines it generates are less accurate.



Region Splitting and Merging

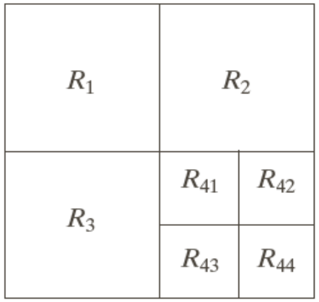
- **Algorithm (**Based on quadtrees/*quadimages.* The root of the tree corresponds to the image)

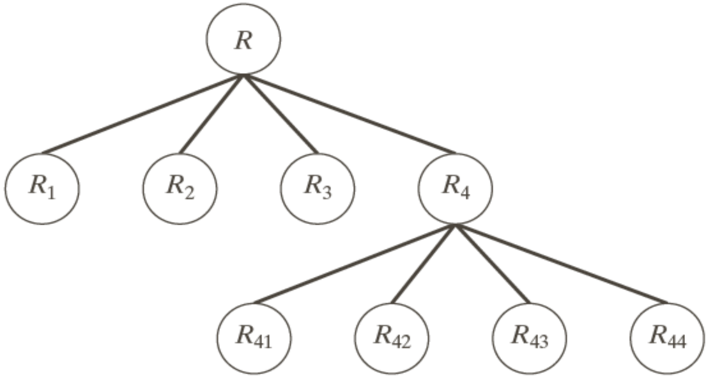
– Split into four disjoint quadrants any region R_i *for which* $Q(R_i)=FALSE$.

– When no further splitting is possible, merge any adjacent regions R_i *and* R_k *for which* $Q(R_{i\ union}\ R_k)=TRUE$.

– Stop when no further merging is possible.

- A maximum quadregion size is specified beyond which no further splitting is carried out.





Region Splitting and Merging

Characteristics of the region of interest:

- Standard deviation grater than the background (which is near zero) and the central region (which is smoother).
- Mean value greater than the mean of background and less than the mean of the central region.
- Predicate:

$$Q = \begin{cases} \text{true} & \sigma > \alpha \ AND \ 0 < m < b \\ \text{false} & \text{otherwise} \end{cases}$$

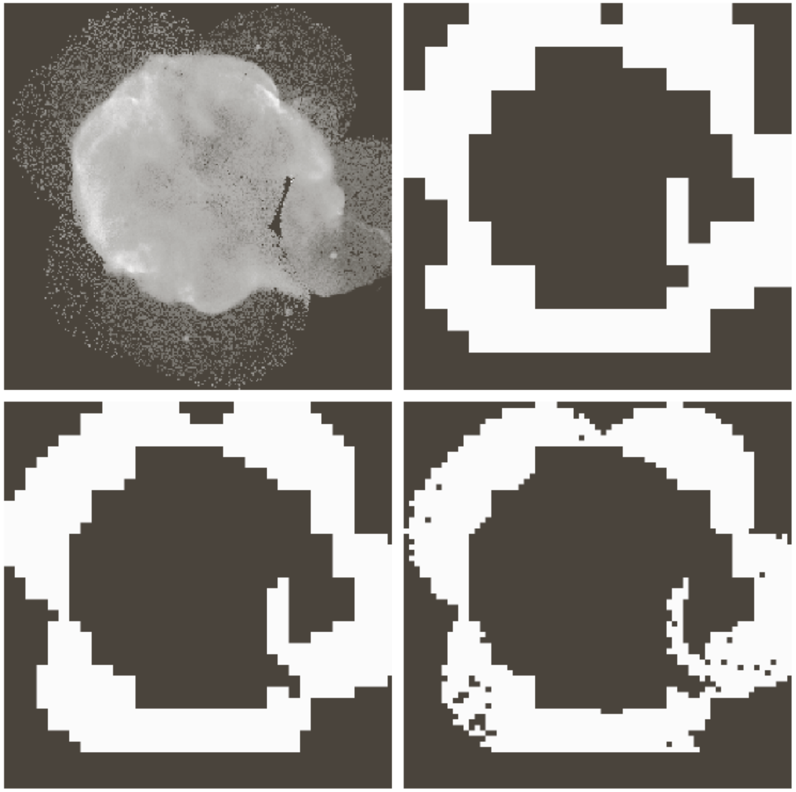
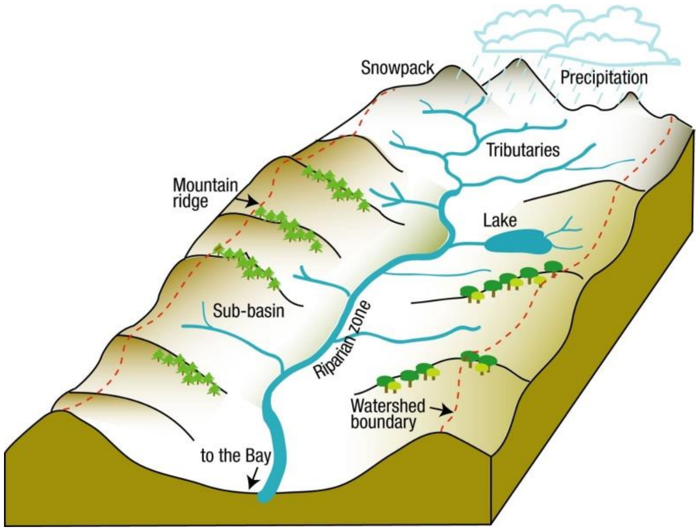


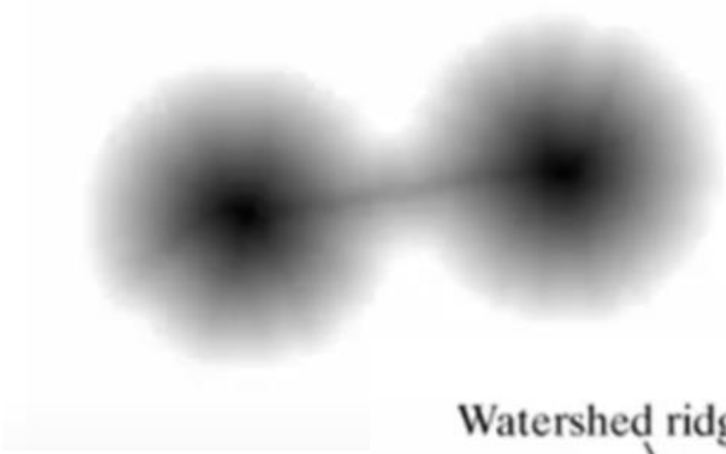
Image of the Cygnus Loop. We want to segment the outer ring of less dense matter

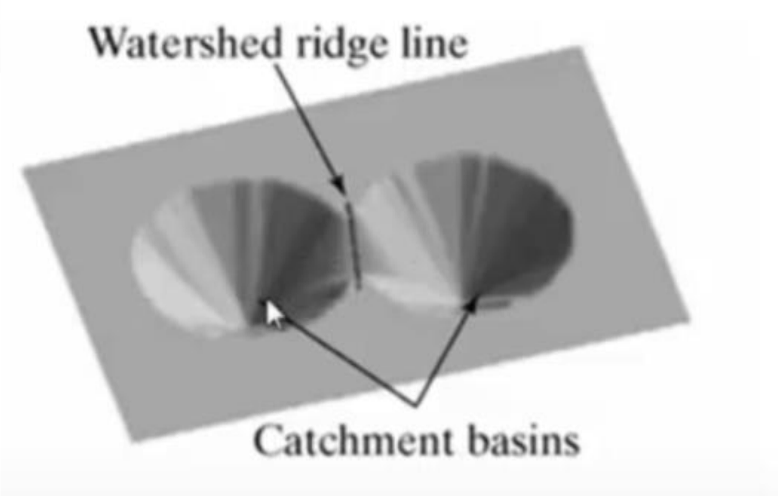
Watershed Transformation

Watershed is a transformation on grayscale images. The aim of this technique is to segment the image, typically when two regions-of-interest are close to each other—i.e, their edges touch.

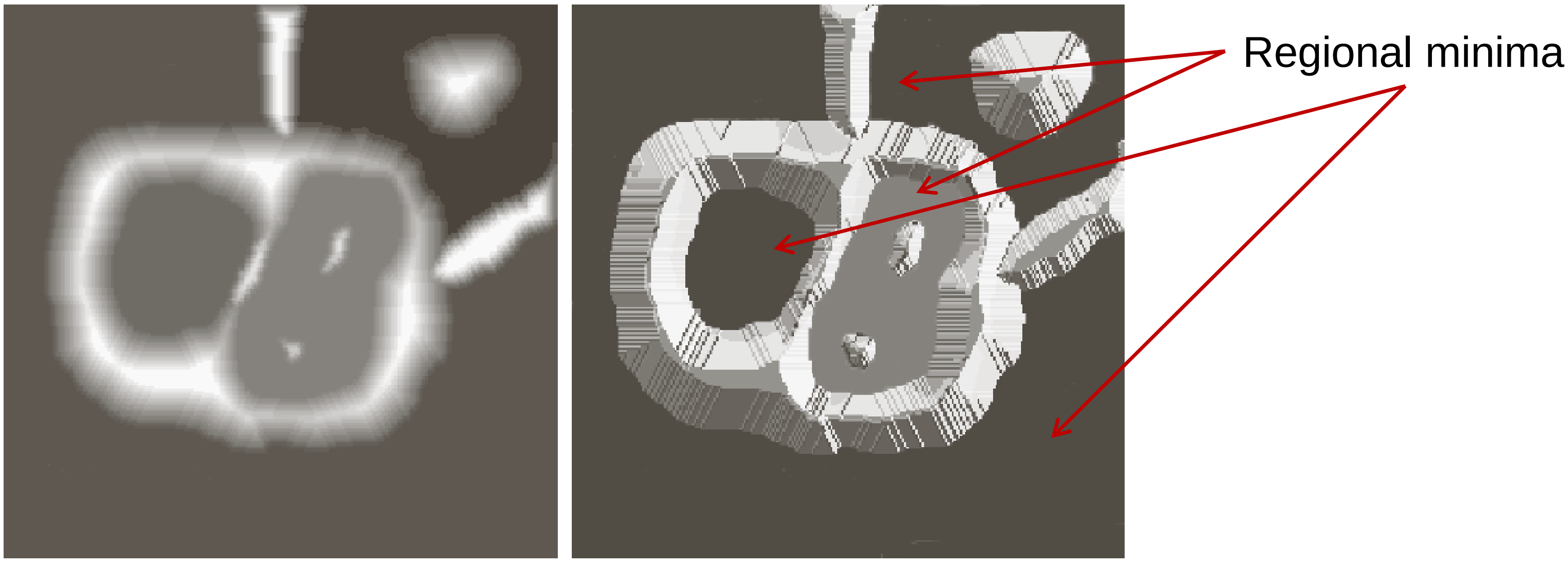


A geographical watershed





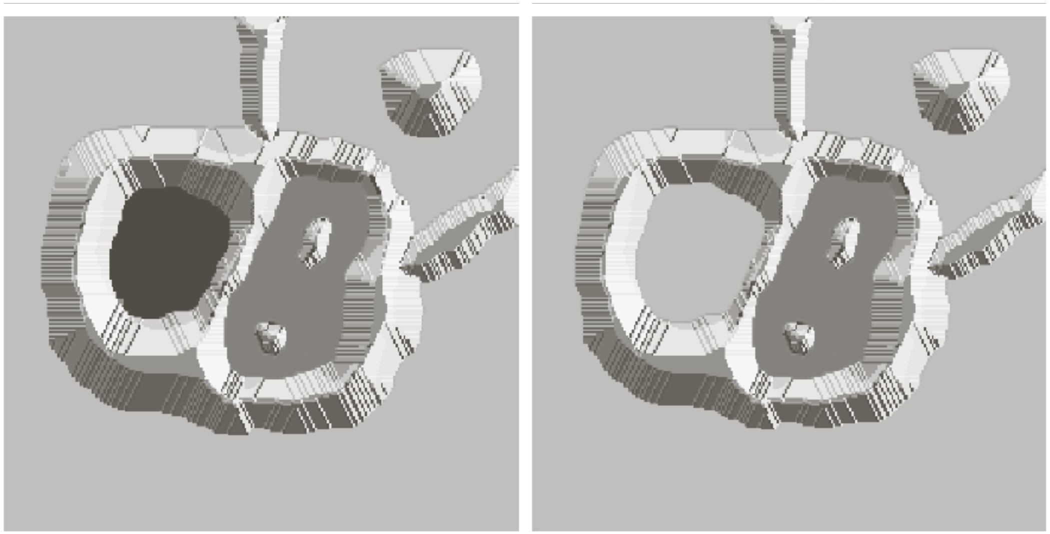
Visualising Watershed



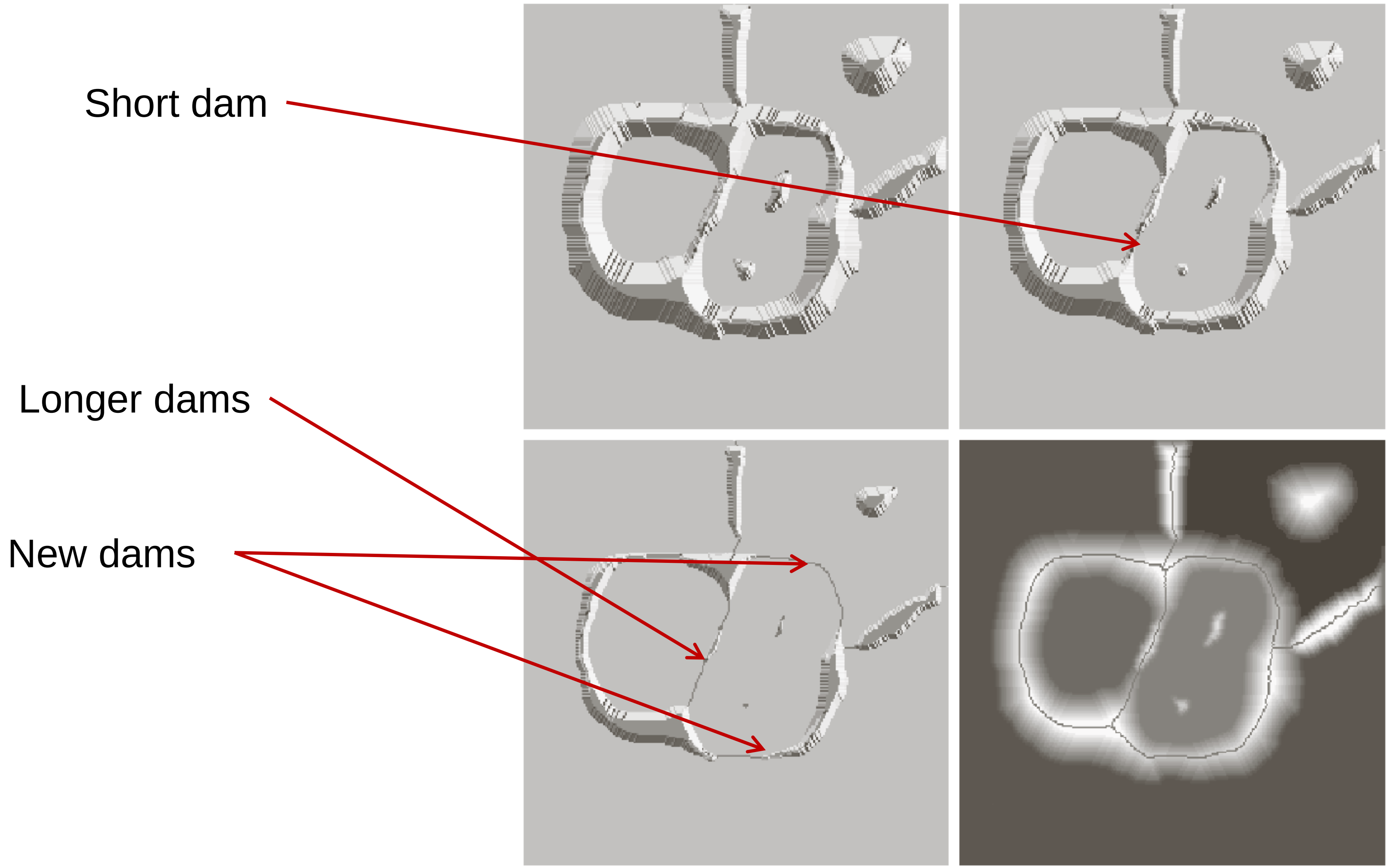
- Image data may be interpreted as a topographic surface.
- The height is proportional to the image intensity.
- Backsides of structures are shaded for better visualization.

Watershed by flooding

- A hole is punched in each regional minimum and the topography is flooded by water from below through the holes.
- When the rising water is about to merge in catchment basins, a dam (barriers in the form of pixels) is built to prevent merging.
- At the max level of flooding only the tops of the dams will be visible. These continuous and connected boundaries act as partition and image is said to be segmentation.

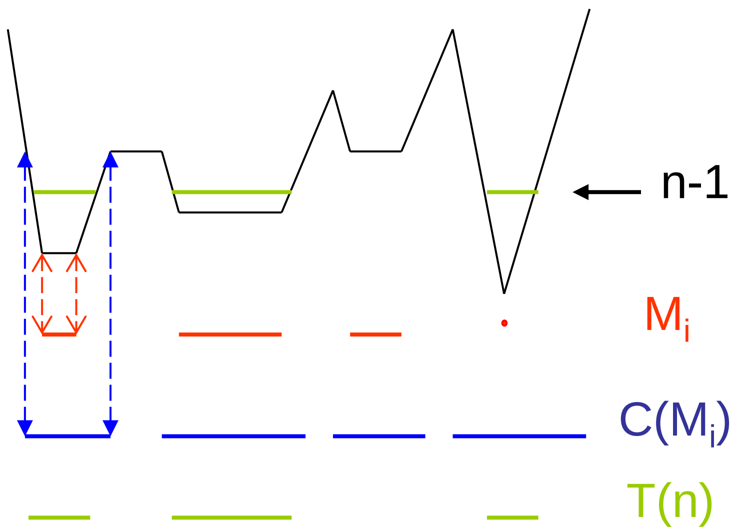


Morphological Watershed



Watershed Transform

- Denote $M_1, M_2, \ldots, M_R$ as the sets of the coordinates of the points in the regional minima of an image g(x,y)
- Denote $C(M_i)$ as the coordinates of the points in the catchment basin associated with regional minimum M_i.
- Denote T[n] as the set of coordinates (s,t) for which g(s,t) < n
- Flood the topography in integer flood increments from n=*min*+1 to n=*max*+1
- At each flooding, the topography is viewed as a binary image

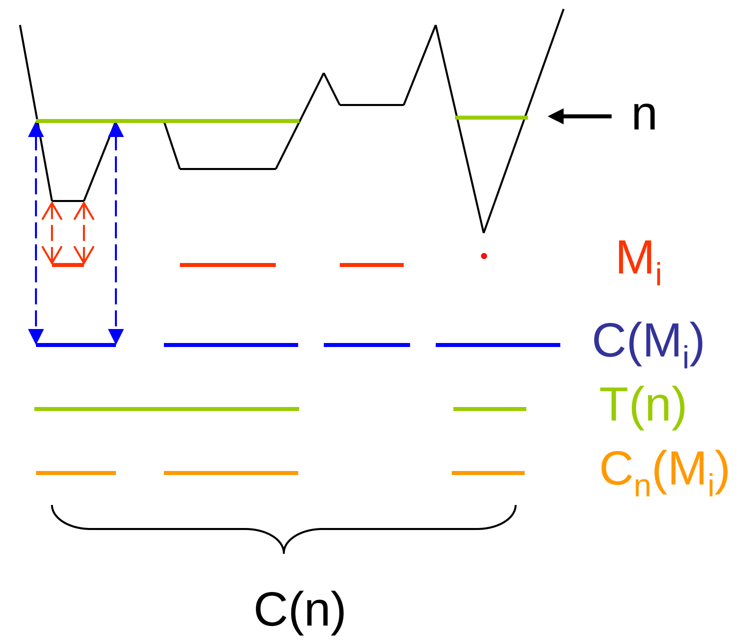


Watershed Transform

- Denote $C_n(M_i)$ as the set of coordinates of points in the catchment basin associated with minimum M_i at flooding stage n.
  - $C_n(M_i) = C(M_i) \cap T[n]$
  - $C_n(M_i) \subseteq T[n]$
- Denote C[n] as the union of the flooded catchment basin portions at stage n:

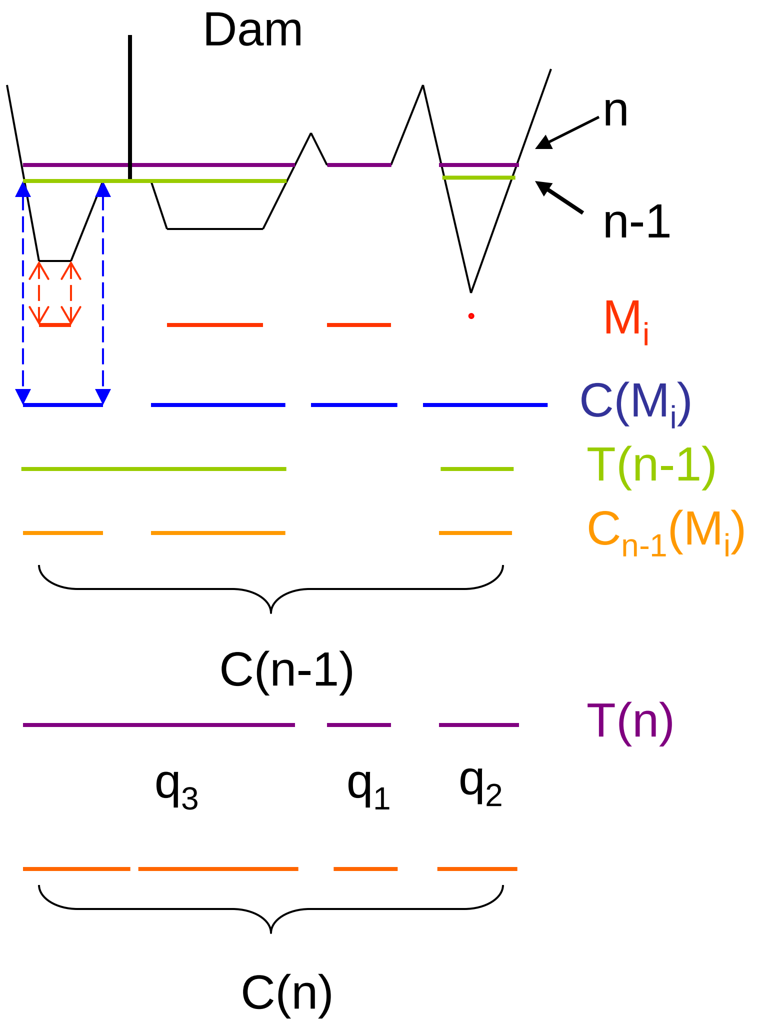
$$C[n] = \bigcup_{i=1}^{R} C_n(M_i) \text{ and } C[\max+1] = \bigcup_{i=1}^{R} C(M_i)$$

- Initialization
  - Let $C[min+1] = T[min+1]$
- At each step n, assume C[n-1] has been constructed. The goal is to obtain C[n] from C[n-1]

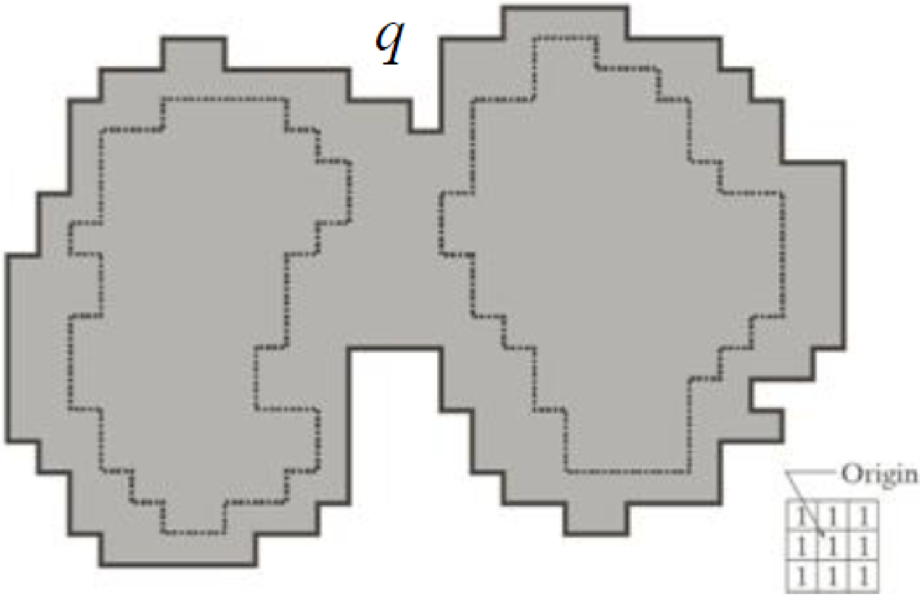


Watershed Transform

- Denote Q[n] as the set of connected components in T[n].
- For each $q \in Q[n]$, there are three possibilities
  1. $q \cap C[n-1]$ is empty (q_1)
     - A new minimum is encountered
     - q is incorporated into C[n-1] to form C[n]
  2. $q \cap C[n-1]$ contains one connected component of C[n-1] (q_2)
     - q is incorporated into C[n-1] to form C[n]
  3. $q \cap C[n-1]$ contains more than one connected components of C[n-1] (q_3)
     - A ridge separating two or more catchment basins has been encountered
     - A dam has to be built within q to prevent overflow between the catchment basins
  4. Repeat the procedure until n=max+1

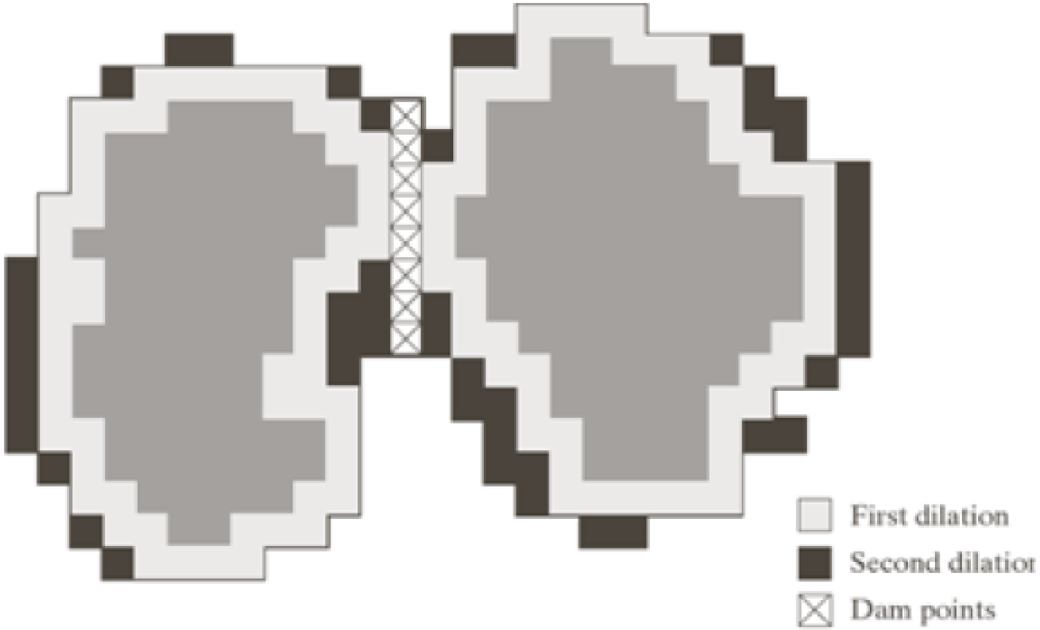


Dam Construction



- Each of the connected components is dilated by the SE shown, subject to:
  1. The center of the SE has to be contained in q.
  2. The dilation cannot be performed on points that would cause the sets being dilated to merge.

Dam Construction



Conditions

1. Center of SE in q.
2. No dilation if merging.

- In the first dilation, condition 1 was satisfied by every point and condition 2 did not apply to any point.
- In the second dilation, several points failed condition 1 while meeting condition 2 (the points in the perimeter which is broken).

Application

- A common application is the extraction of nearly uniform, blob-like objects from their background.
- For this reason it is generally applied to the gradient of the image and the catchment basins correspond to the blob like objects.

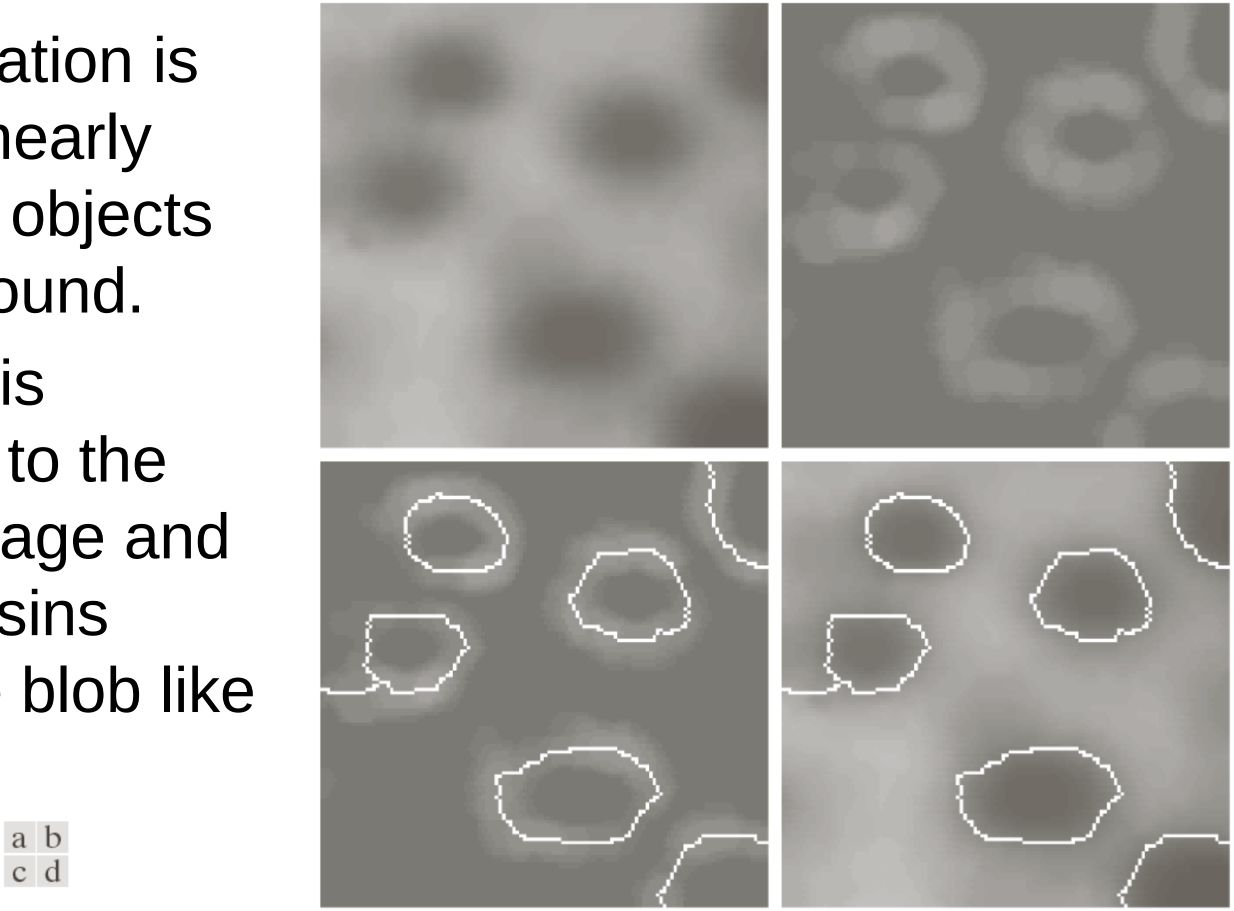
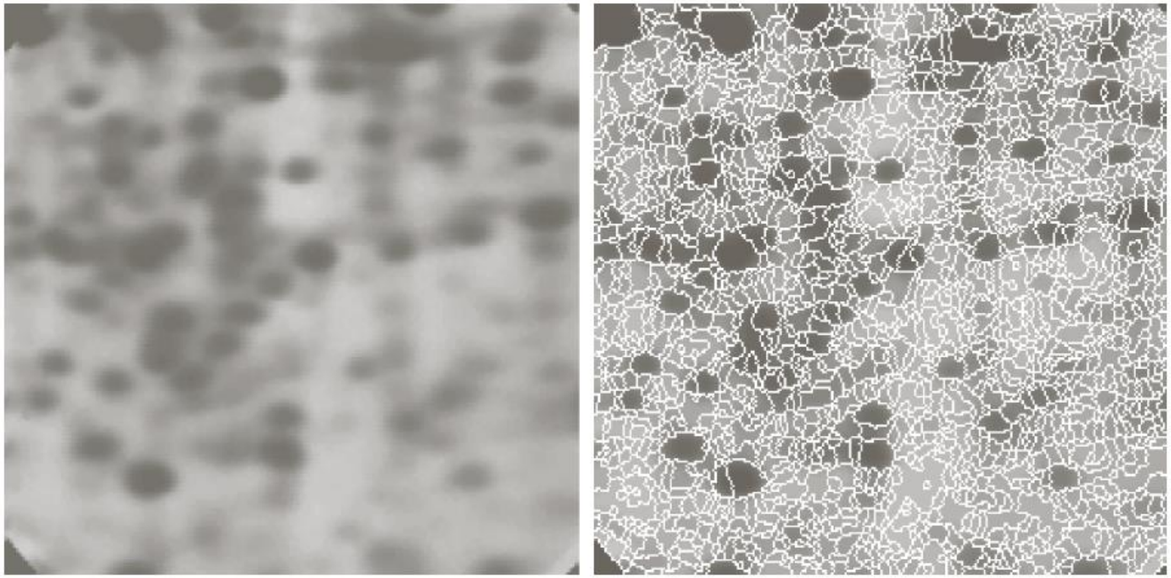


FIGURE 10.56 (a) Image of blobs. (b) Image gradient. (c) Watershed lines. (d) Watershed lines superimposed on original image.

Problems



- Noise and local minima lead generally to *oversegmentation.*
- The result is not useful.
- Solution: limit the number of allowable regions by additional knowledge.

Use of Markers

Markers (connected components):

– *internal, associated with the objects*

– *external, associated with the background.*

- Here the problem is the large number of local minima.
- Smoothing may eliminate them.
- Define an *internal marker (after smoothing):*
  – Region surrounded by points of higher altitude.
  – They form connected components.
  – All points in the connected component have the same intensity.