Image Processing

Lecture 13: Image Segmentation – II

(Ch10 Image Segmentation)

Zhiguo Zhang

zhiguozhang@hit.edu.cn

Review of Last Lecture

- In the last lecture we learnt:
  - The segmentation problem
  - Edge detection
  - Edge linking and boundary detection

Contents of This Lecture

- In this lecture we will learn:
  - Thresholding
  - Region based segmentation
  - Morphological watersheds approach

Thresholding

- Thresholding is usually the first step in any segmentation approach
- We have talked about simple single value thresholding already
- Here we will look at:
    - Basic Global Thresholding
    - Basic Adaptive Thresholding
    - Minimisation Thresholding

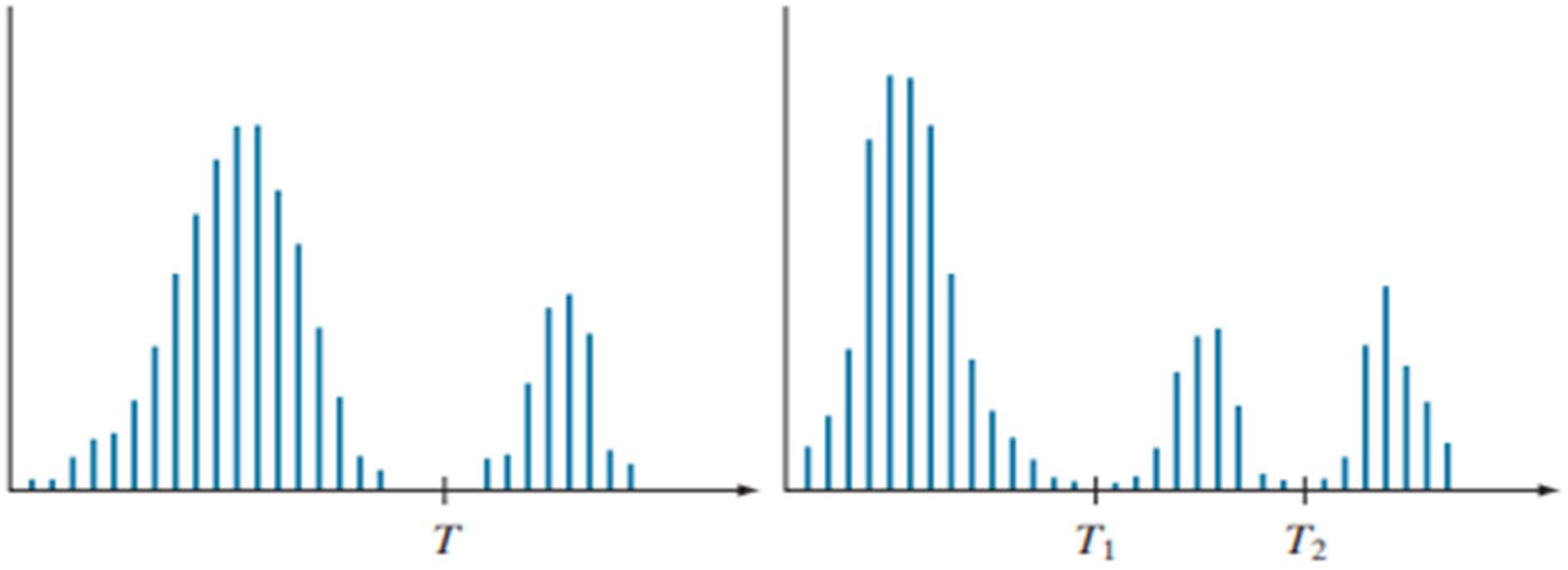
Thresholding

- In this lecture we will discuss light objects in dark background.

- To extract the objects:
  - Select a threshold T that separates the objects from the background.
  - i.e. any (x, y) for which $f(x, y) > T$ is an object point.

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

Thresholding



Background: $f(x,y) \le T$

Object: $f(x,y) > T$

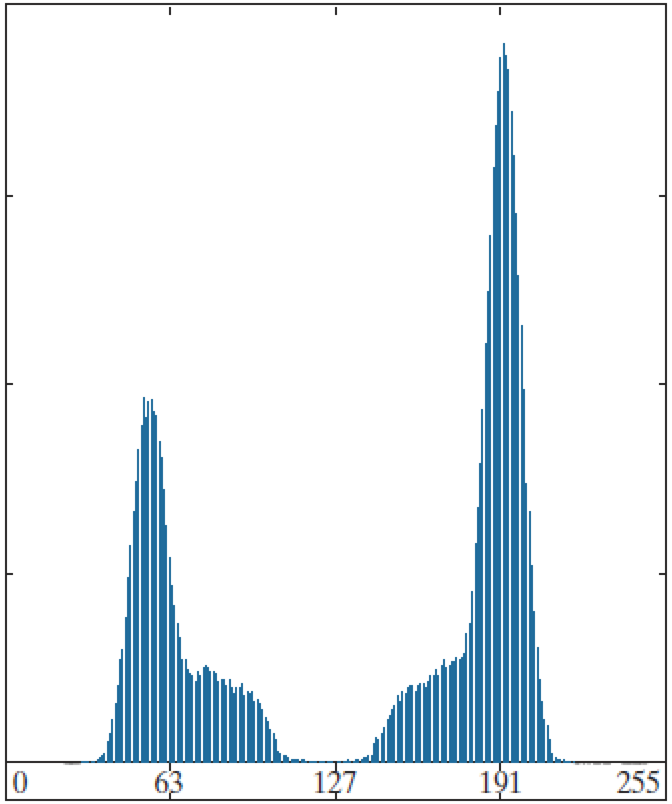
Background: $f(x,y) \le T_1$

Object1: $T_1 < f(x,y) \le T_2$

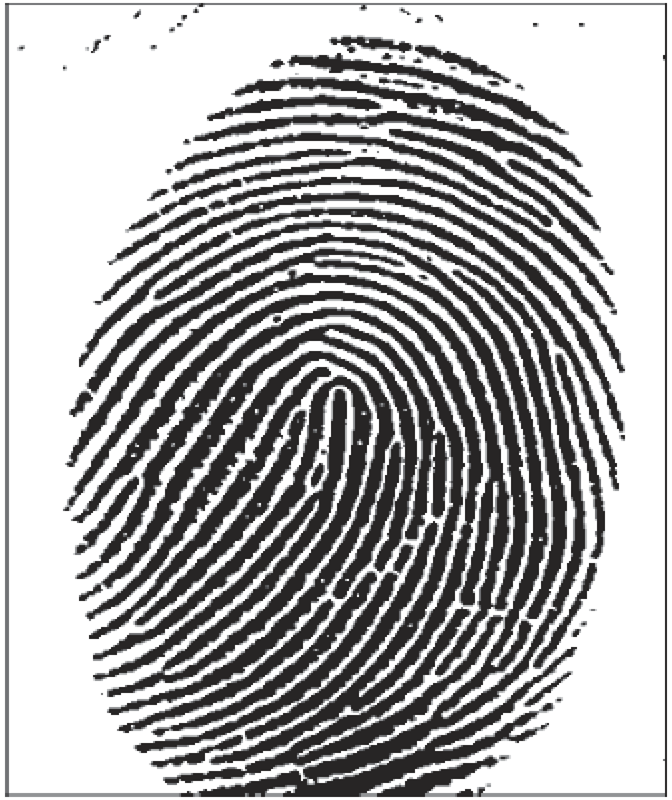
Object2: $f(x,y) > T_2$

Basic Global Thresholding Algorithm

Noisy fingerprint



Histogram



Segmented result using a global threshold.

Thresholding

- A threshold can take a more general form:

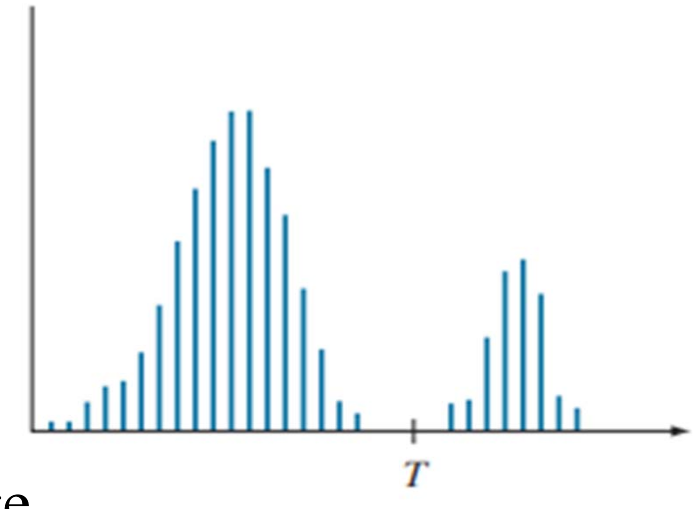
$$T = T[x, y, p(x, y), f(x, y)]$$

where $f(x, y)$ is the gray level of point (x, y), and $p(x, y)$ denotes some local property of this point. For example: the average gray level of a neighborhood centered on (x, y)

- T depends on:
  - only $f(x, y)$: only on gray-level values → **Global threshold**;
  - both $f(x, y)$ and $p(x, y)$ → **Local threshold**;
  - on $f(x, y)$, $p(x, y)$, x and y → **Adaptive threshold**.

Thresholding

- The success of intensity thresholding is related directly to the width and depth of the valley separating the histogram modes.
- In turn, the key factors affecting the properties of the valley(s) are:
    1. The separation between peaks: the further apart the peaks are, the better the chances of separating the modes;
    2. The noise content in the image;
    3. The relative sizes of objects and background;
    4. The uniformity of the illumination source;
    5. The uniformity of the reflectance properties of the image.



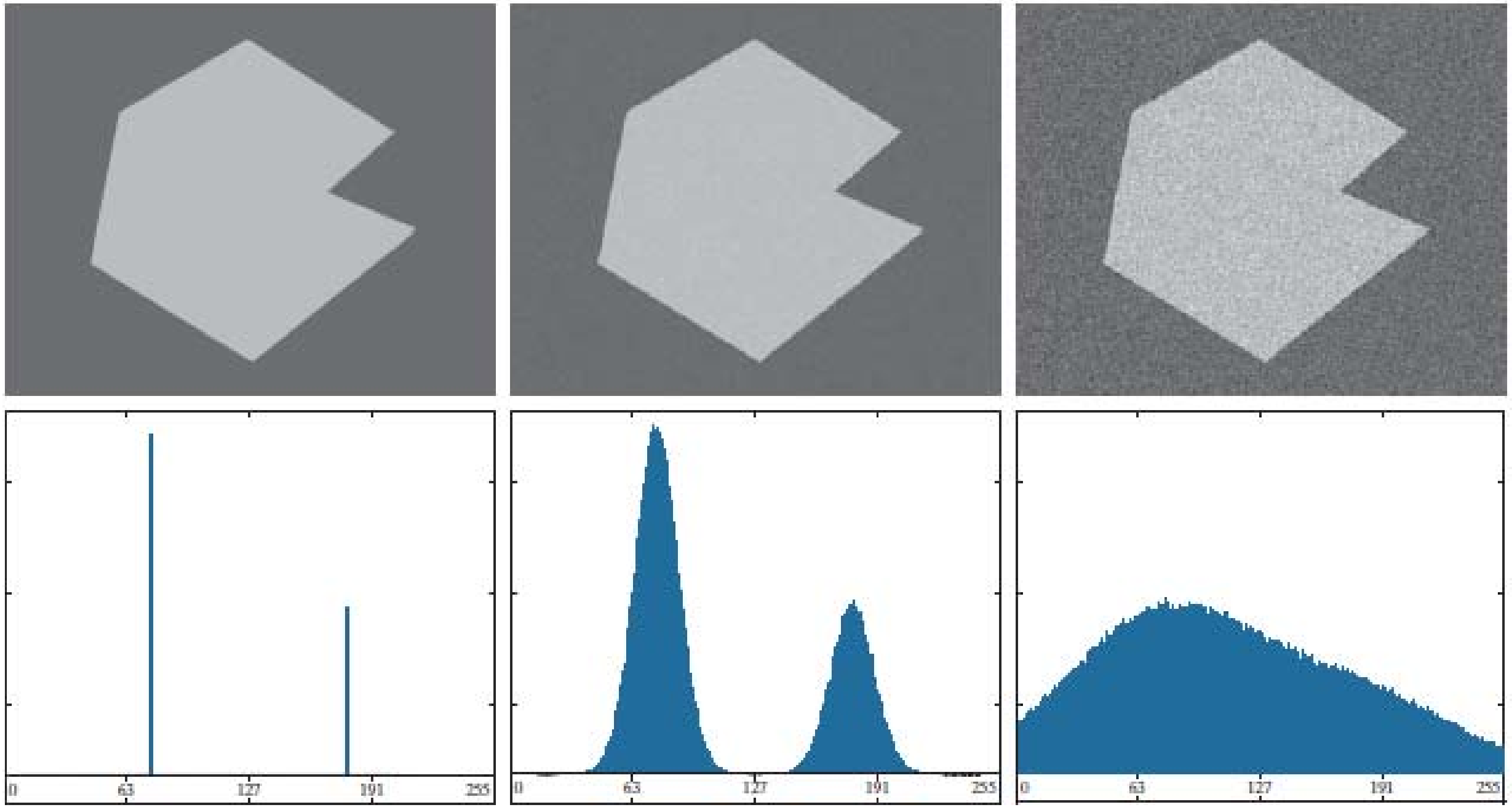
Thresholding

- Influence of noise

(a) Noiseless 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.



Corresponding histograms.

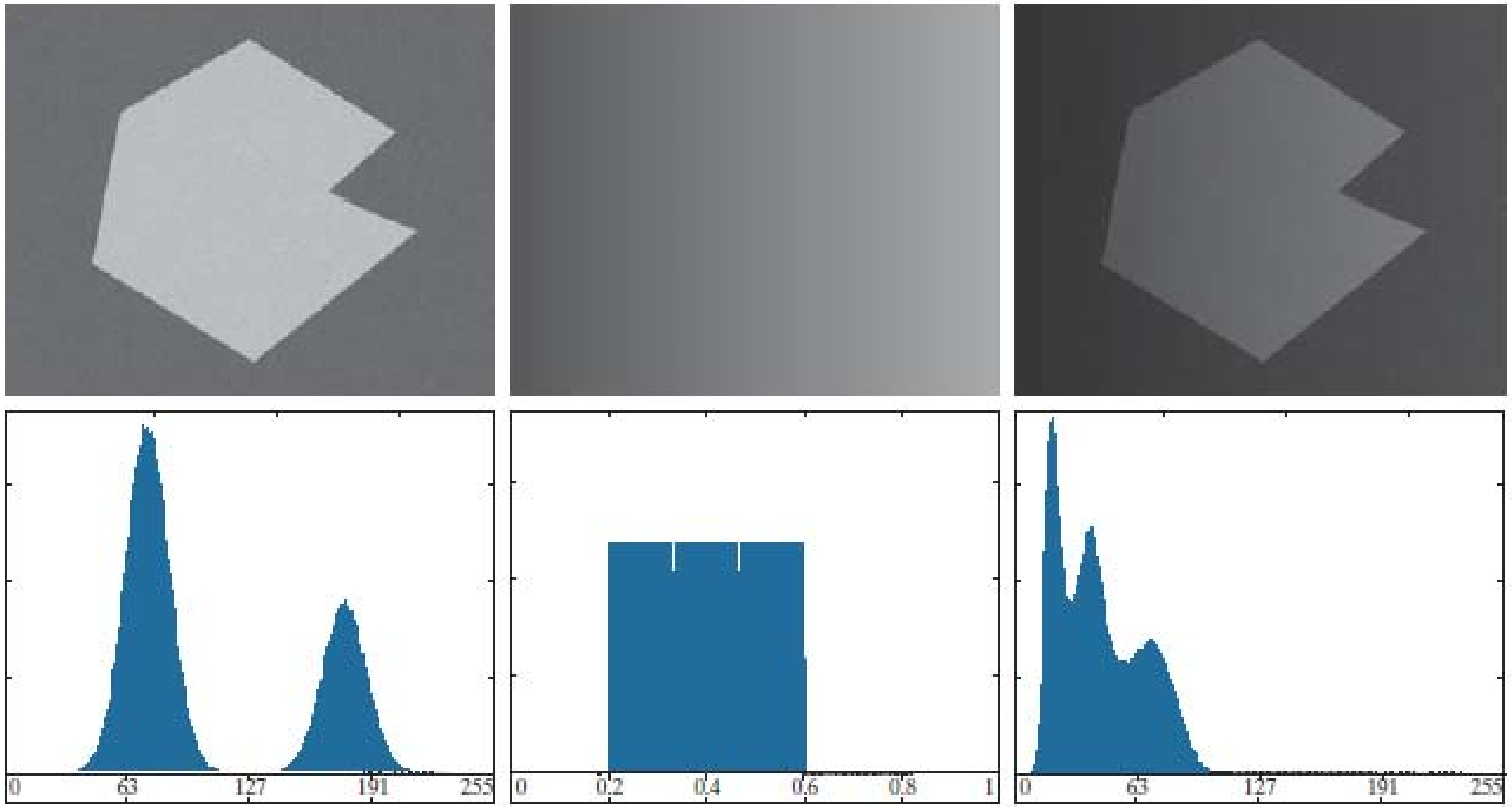
Thresholding

- Influence of non-uniformity

(a) Noisy image.

(b) Intensity ramp in the range [0.2, 0.6].

(c) Product of (a) and (b)



Corresponding histograms.

Basic Global Thresholding

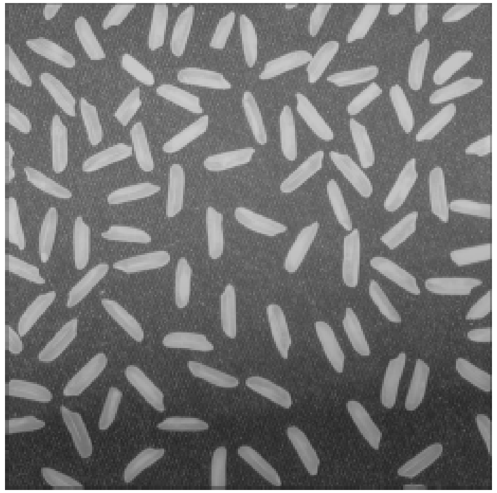
- To partition the image histogram by using a single threshold T.

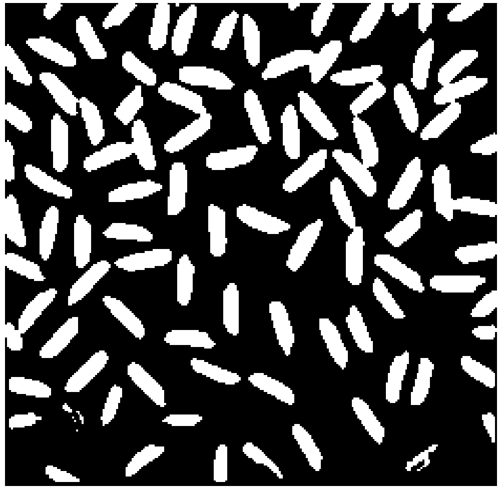
- Then the image is scanned and labels are assigned.

- This technique is successful in highly controlled environments, or depends on how well the histogram can be partitioned.

Basic Global Thresholding

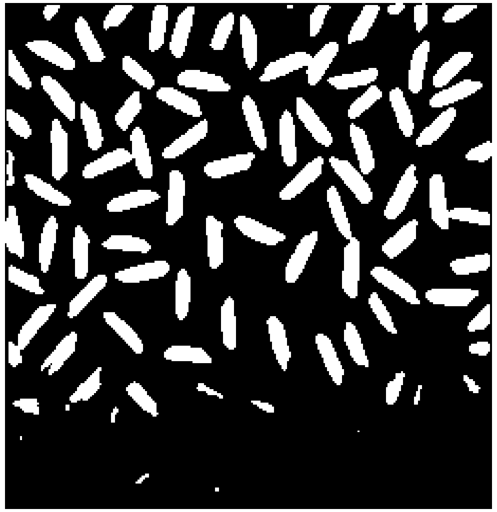
Original Image

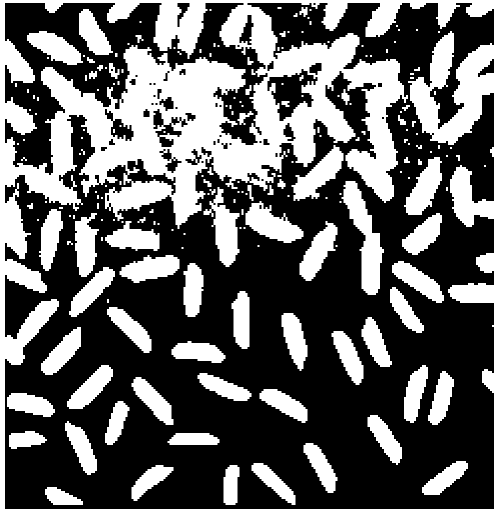




T=125

T=150



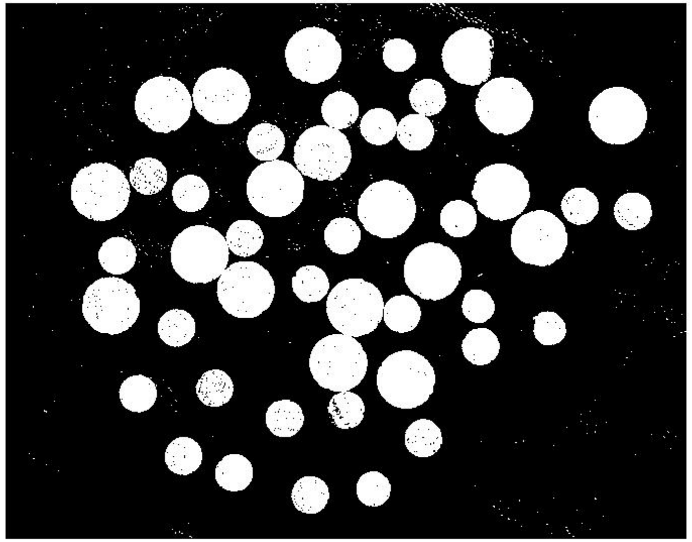


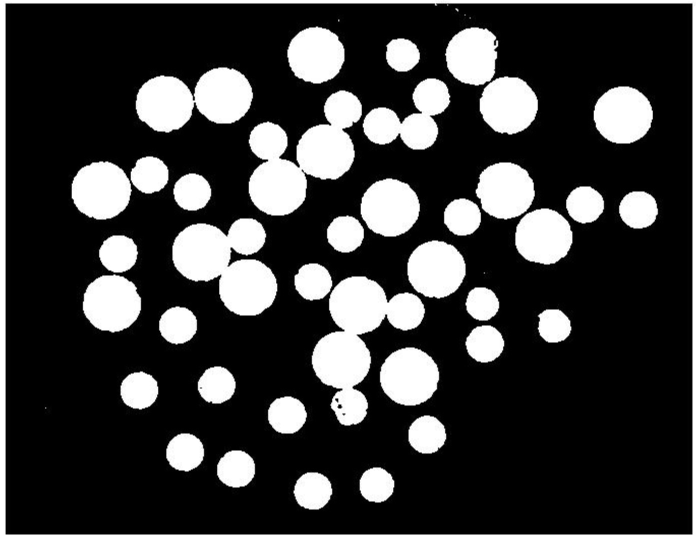
T=110

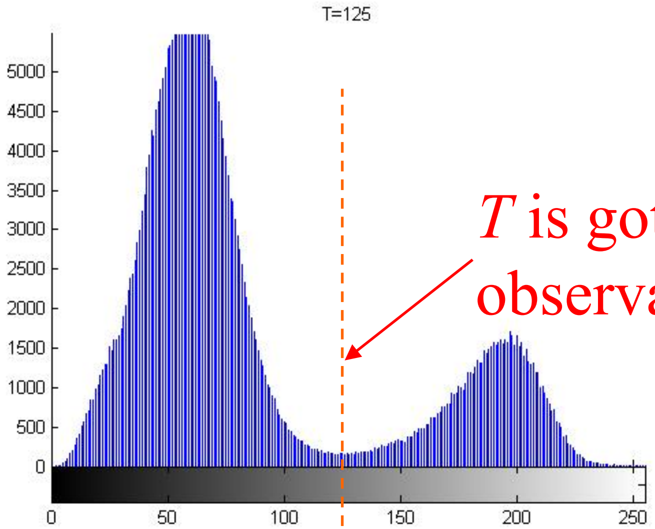
Basic Global Thresholding

?

?







T is gotten by observation

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.
  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_0.

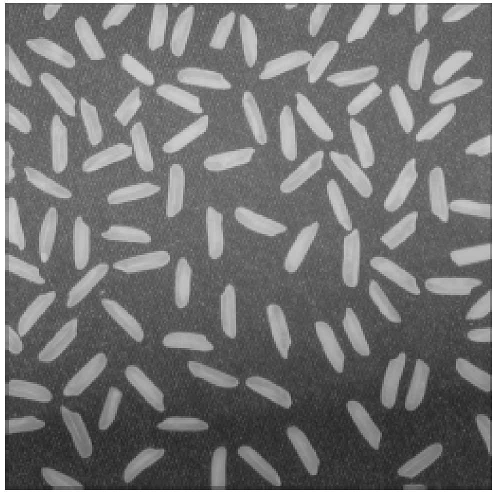
Basic Global Thresholding Algorithm

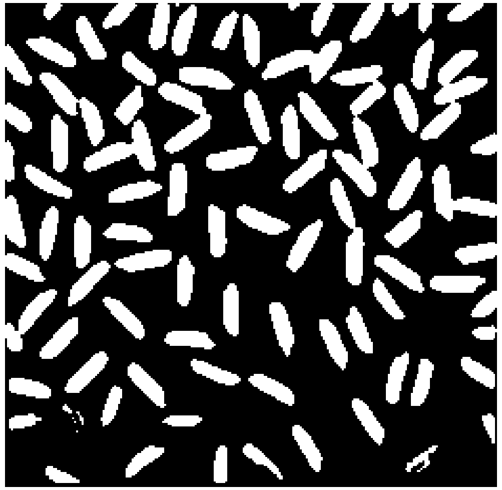
Remarks:

- When the background and object occupy comparable areas in the image, a good initial value for T is the average gray level of the image.
- When objects are smaller than the background (or vice versa), then one group of pixels will dominate the histogram and the average gray level is not as good an initial choice.
- A more appropriate initial value for T in cases is the middle value.

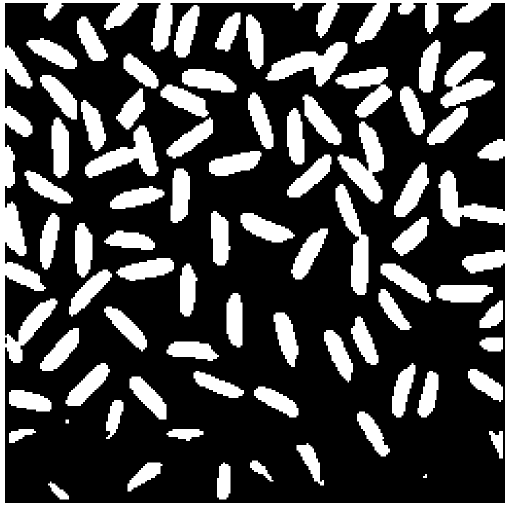
Basic Global Thresholding Algorithm

Original Image



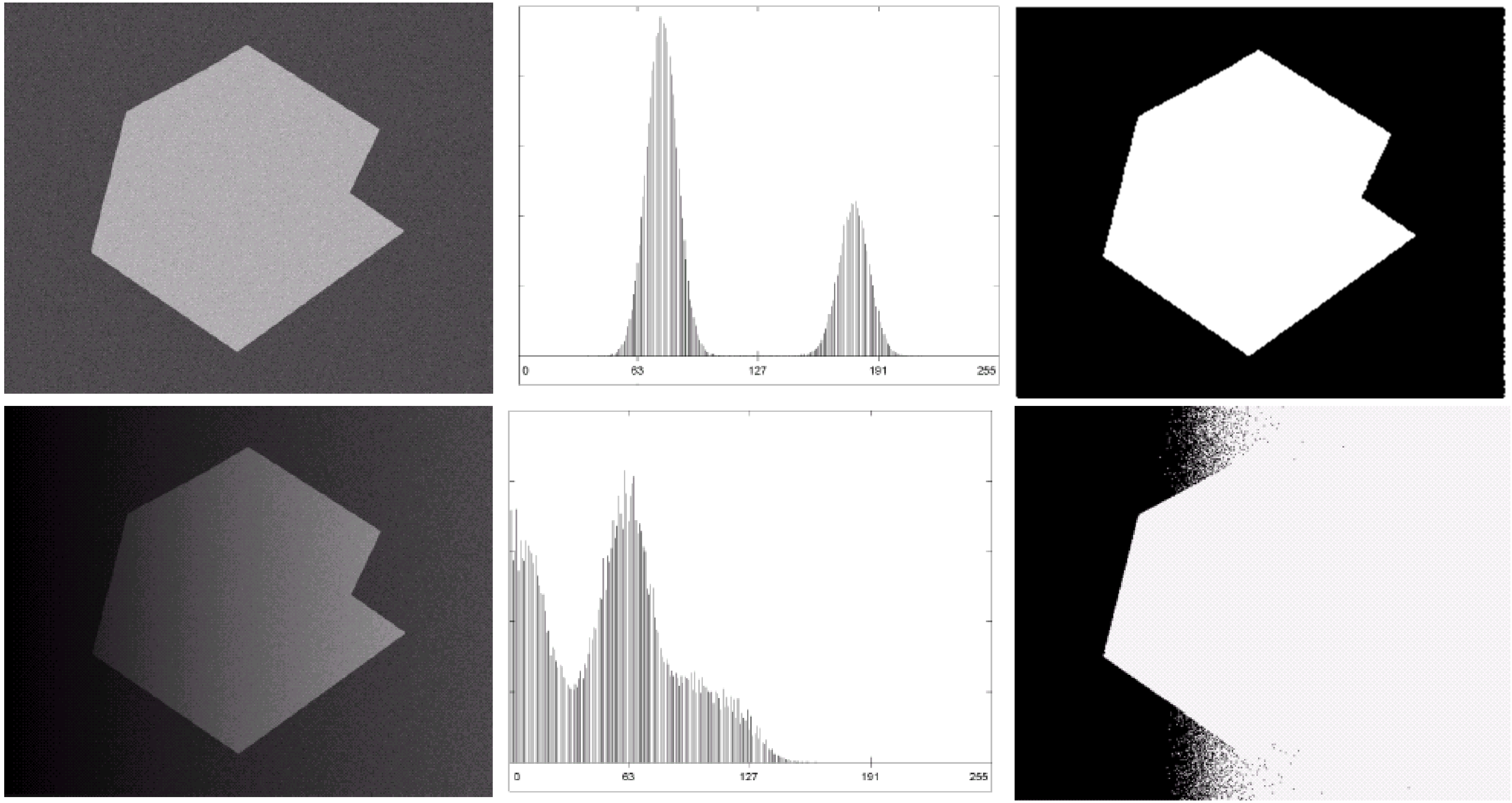


T=125, by visual inspection



T=130.96, by basic global thresholding

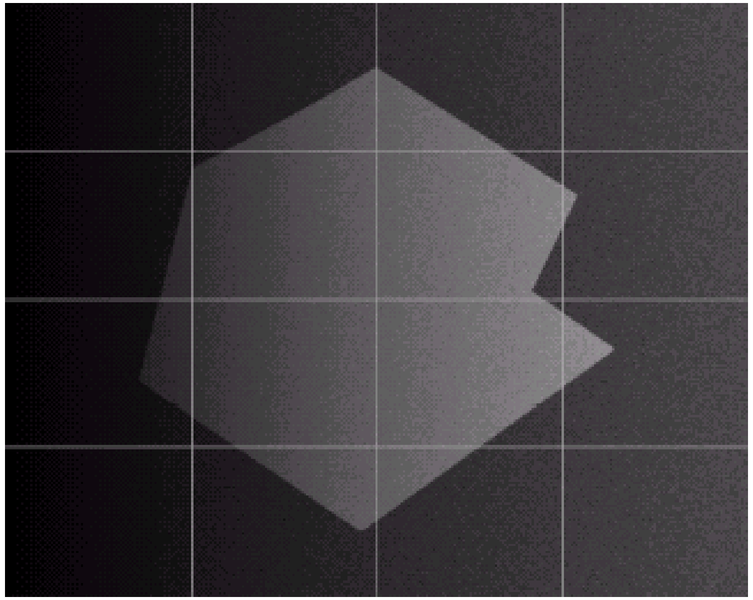
Single Value Thresholding and Illumination

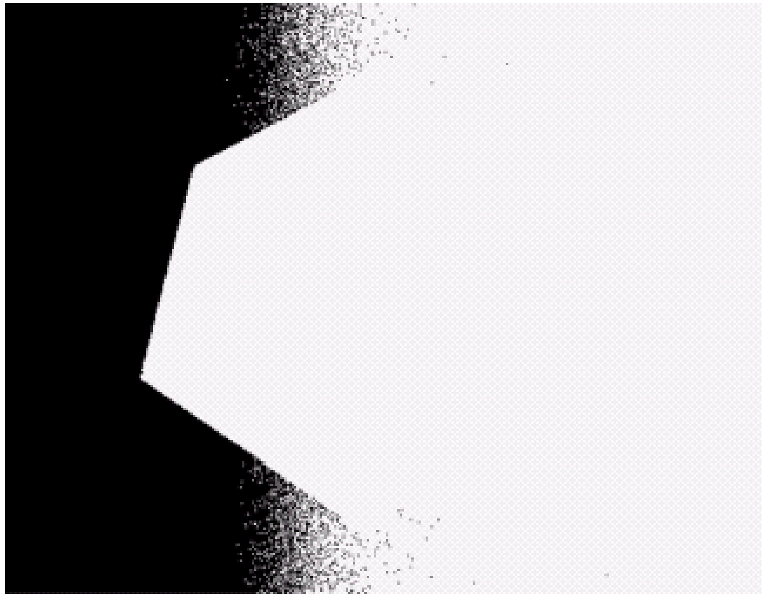


Uneven illumination can upset a single valued thresholding scheme.

Basic Adaptive Thresholding

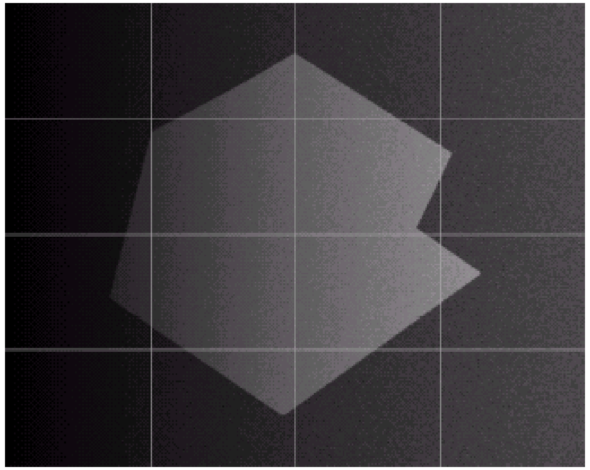
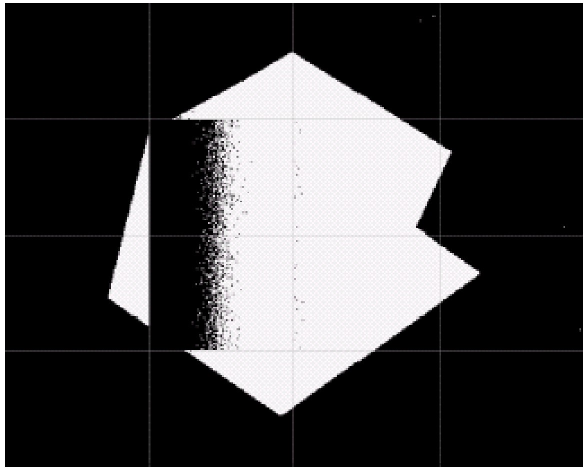
- An approach to handling situations in which single value thresholding will not work is to divide an image into sub-images and threshold them individually.
- Since the threshold for each pixel depends on its location within an image, this technique is said to be **adaptive**.





Basic Adaptive Thresholding Example

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

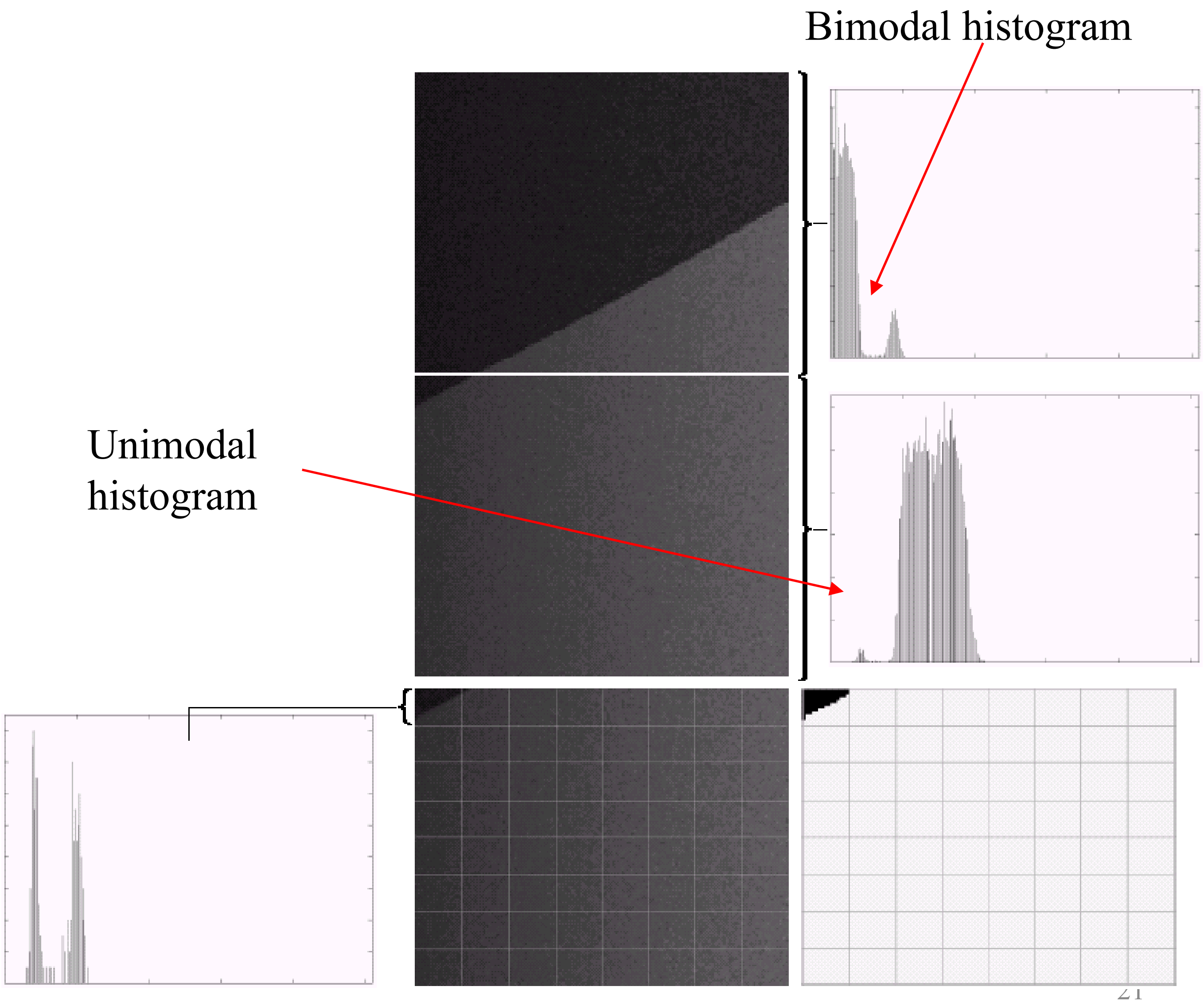



- The results are mixed.
- But, we can further divide the troublesome sub-images.

> ➢ All the subimages that did not contain a boundary had variances less than 75.
>
> ➢ All subimages containing boundaries had variances greater than 100.
>
> ➢ For each subimage, use basic global thresholding to segment it. Original *T* is the middle gray level.

Basic Adaptive Thresholding Example (cont…)

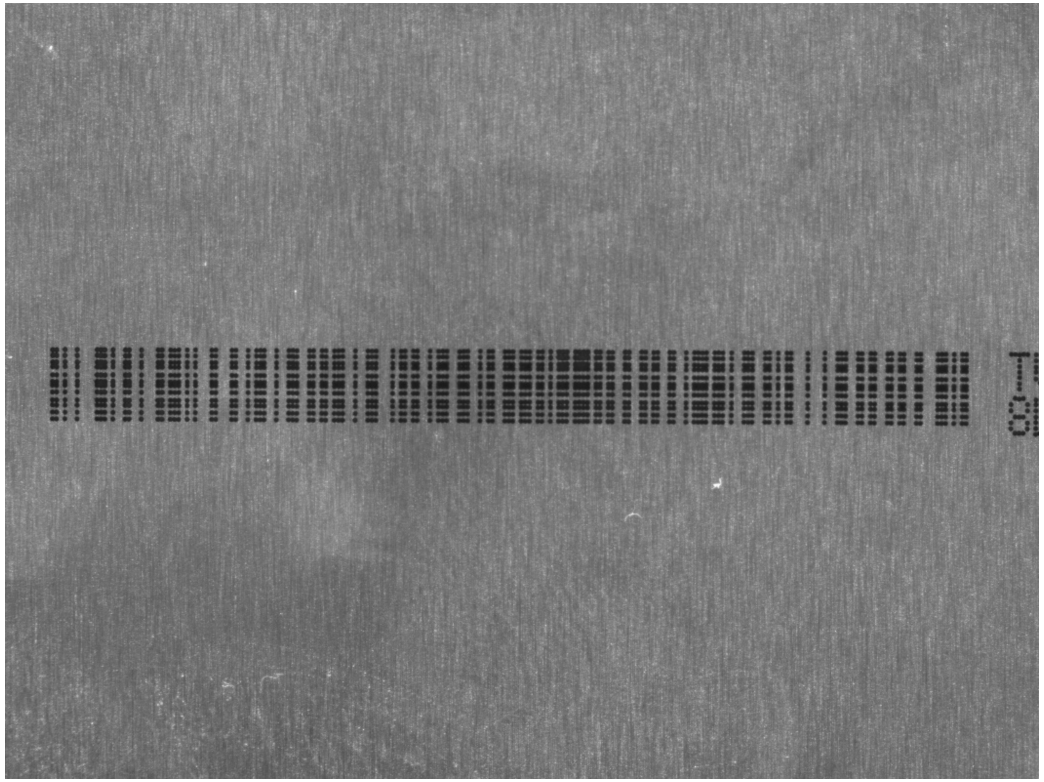
- The left graph shows the troublesome parts of the previous problem, in which the troublesome sub-image is further divided.
- After this sub-division, successful thresholding can be achieved.

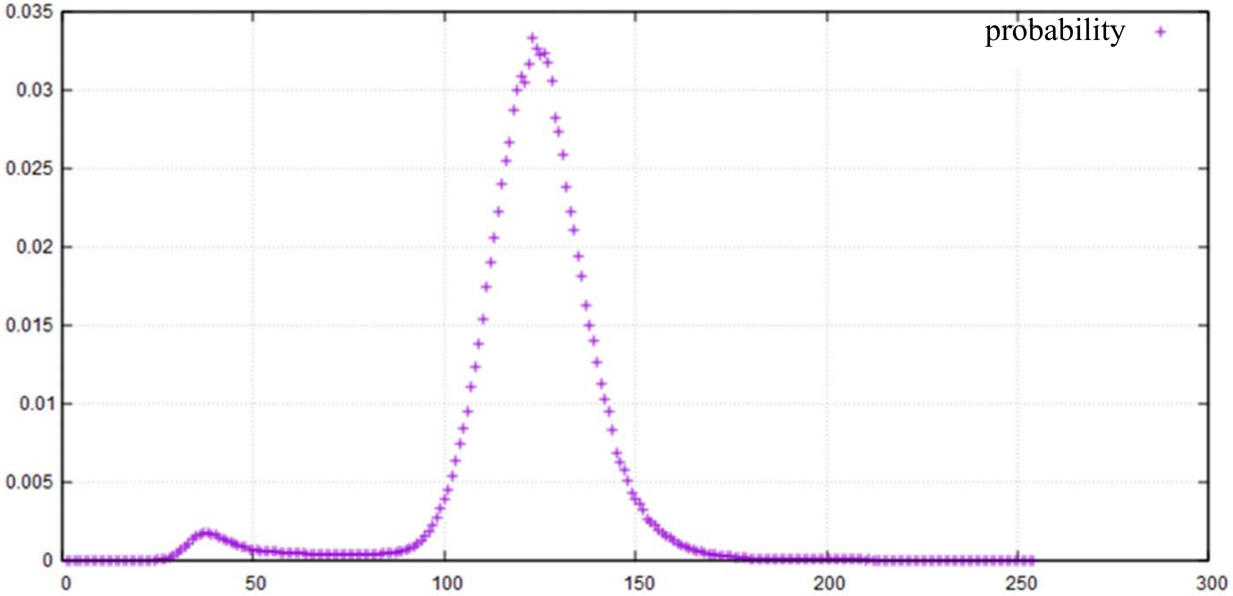


Otsu's Method

- Optimization criterion of Otsu's method: maximize the inter-class variance.
  - Take an optimal threshold to divide the original image into foreground and background.
  - The greater the inter-class (between-class) variance of the two parts, the greater the difference between the two parts, then the image can be effectively segmented.
  - So the key to this algorithm is to find the optimal threshold.
  - Otsu's Method can segment the image in which the histogram peaks and valleys are not obvious.

Otsu’s Method





Otsu's Method

- Suppose the mean value of an image is M.
- A given threshold T can segment the image into two groups: object and background:
  - The probability of group object is $p_o(T)$, and the mean is M_O;
  - The probability of group background is $p_b(T)$, and the mean is M_B.
- The inter-class variance:

$$ICV = p_o(T) \cdot (M_O - M)^2 + p_b(T) \cdot (M_B - M)^2$$

- Object: to find a threshold that makes the maximum ICV.

Otsu’s Method

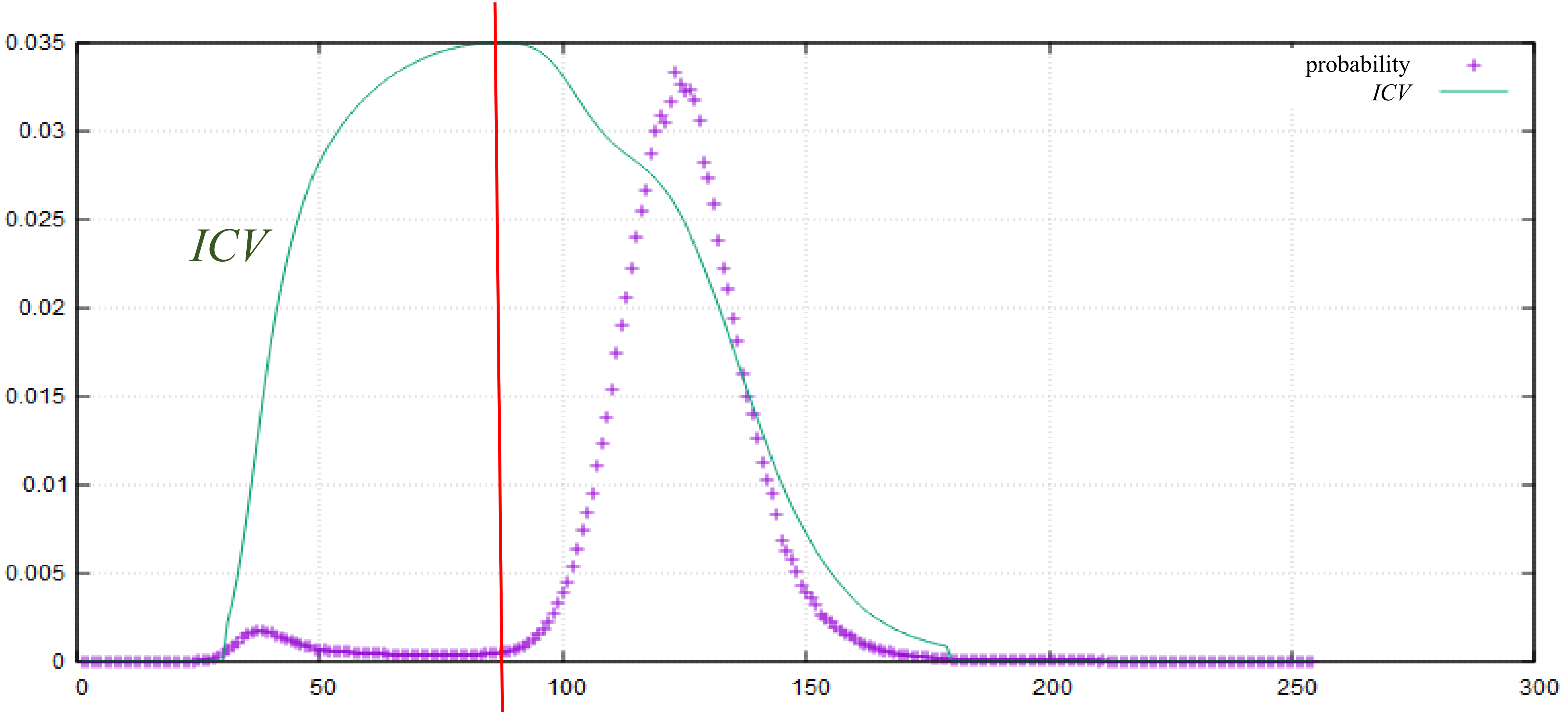
- Otsu algorithm uses the idea of *clustering*:
  - Divide the gray level of the image into 2 parts by a gray level such that the difference in gray value between two parts is maximum and the difference in gray level within each part is minimum;
  - The variance is calculated to find a suitable gray level to divide.

Otsu’s Method

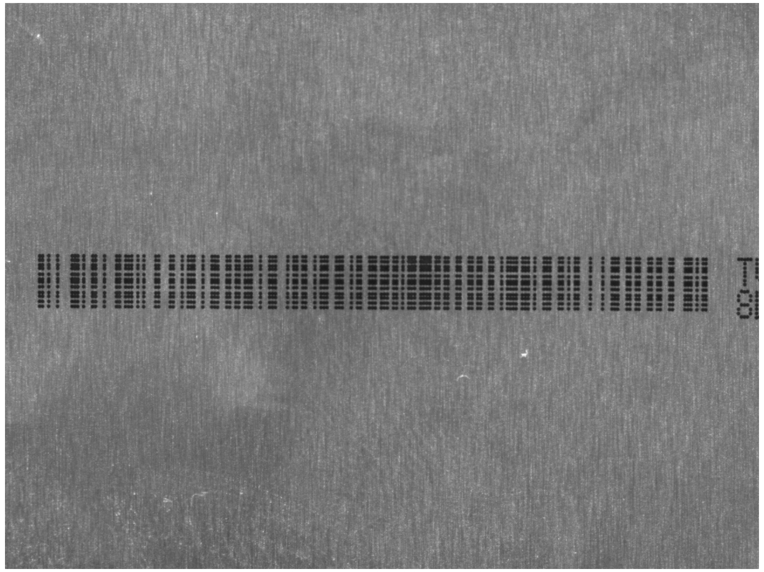
- Since variance is a measure of the uniformity of the gray scale distribution, the larger the inter-class variance between the background and the foreground, the larger the difference between the two parts that make up the image.
- When part of the foreground is wrongly divided into the background, or part of the background is wrongly divided into the foreground, the difference (inter-class variance) between the two parts becomes smaller.
- Therefore, the segmentation that maximizes the interclass variance implies that the probability of misclassification is minimized.

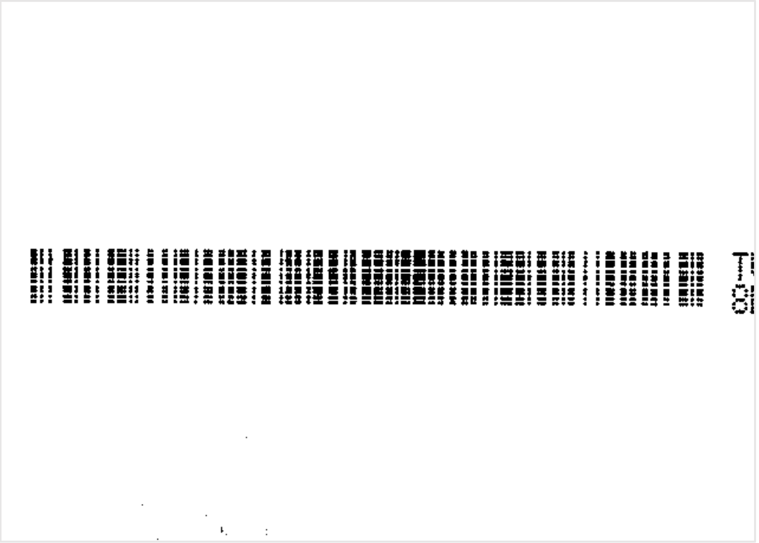
Otsu’s Method

- Try all T to find the maximum ICV:



Otsu's Method





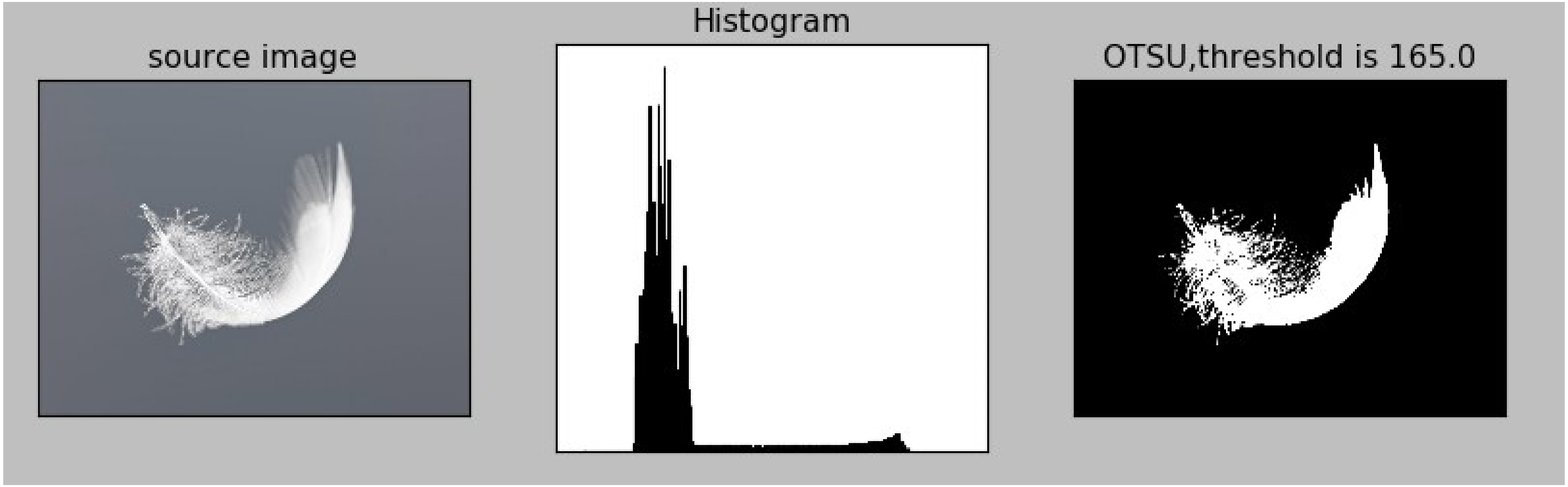
$$ICV = p_o(T)\cdot(M_O - M)^2 + p_b(T)\cdot(M_B - M)^2$$

- Sometimes, when the sizes of object and background are largely different, the method doesn't work.
- We can consider to modify the criterion:

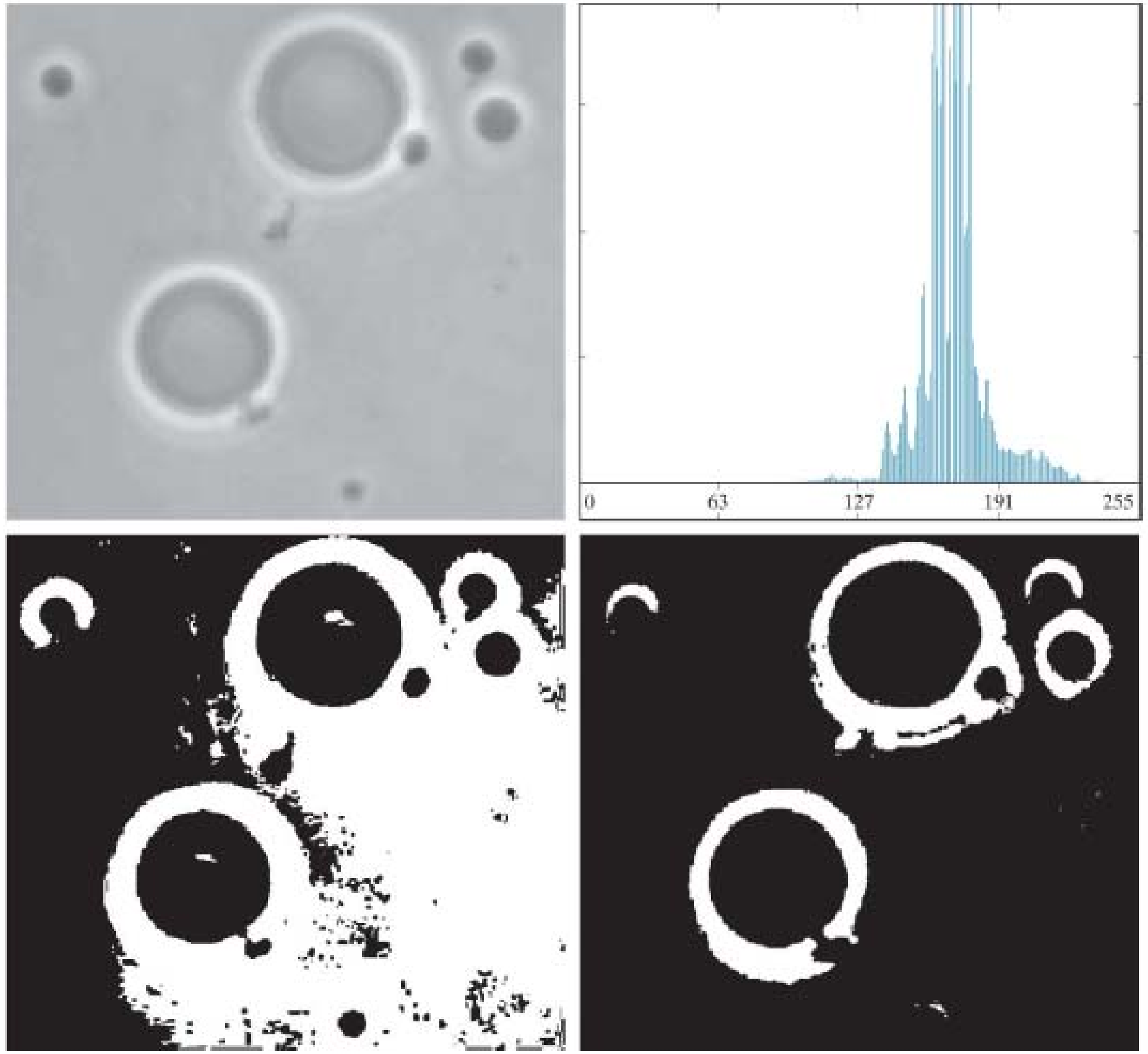
$$ICV = p_o(T)^a \cdot(M_O - M)^2 + p_b(T)^a \cdot(M_B - M)^2$$

Otsu’s Method

- For images with only object-background classes, Otsu’s method yields statistically optimal segmentation.



Otsu’s Method



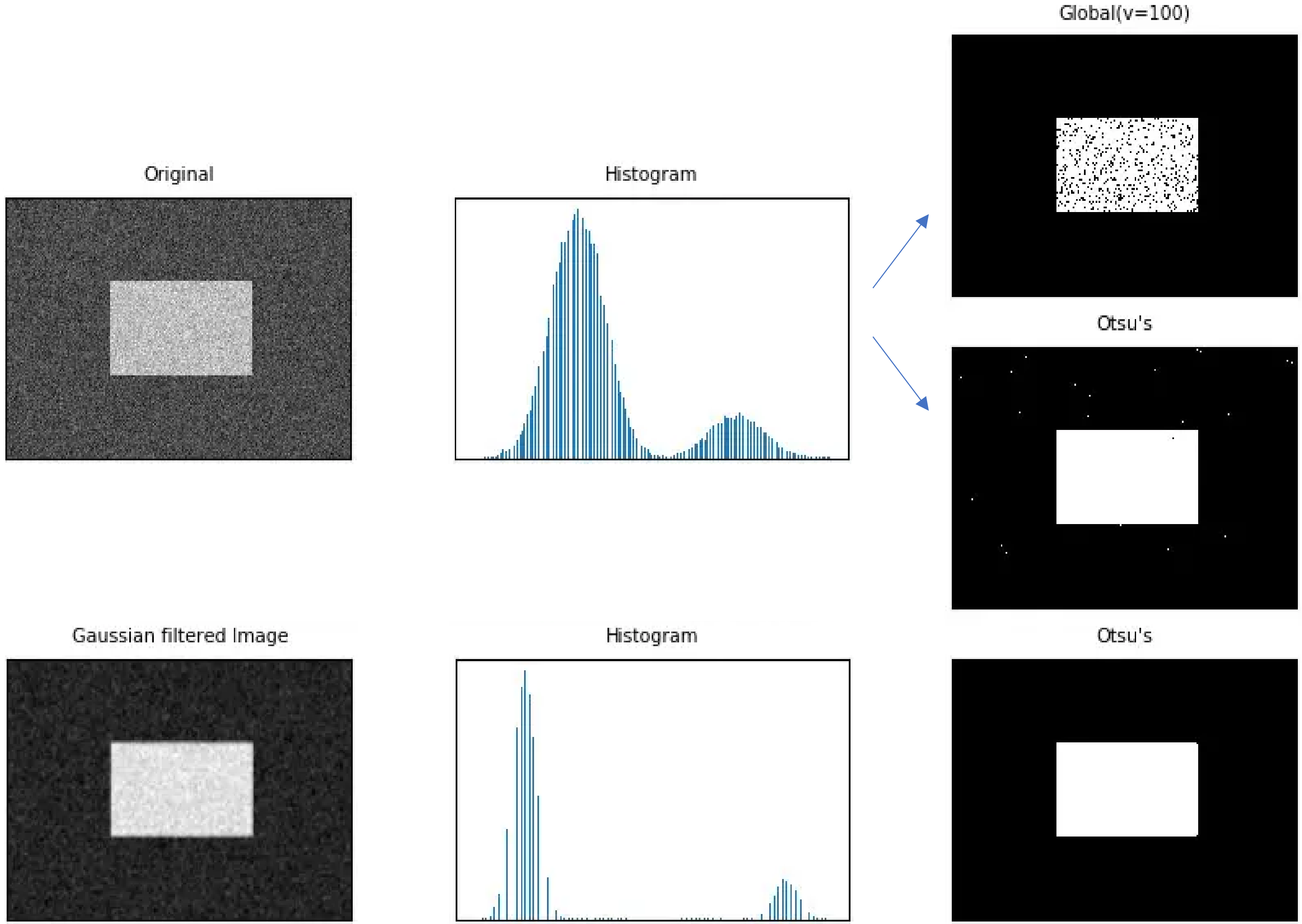
(a) Original image.

(b) Histogram (high peaks were clipped to highlight details in the lower values).

(c) Segmentation result using the basic global thresholding algorithm.

(d) Result using Otsu’s method.

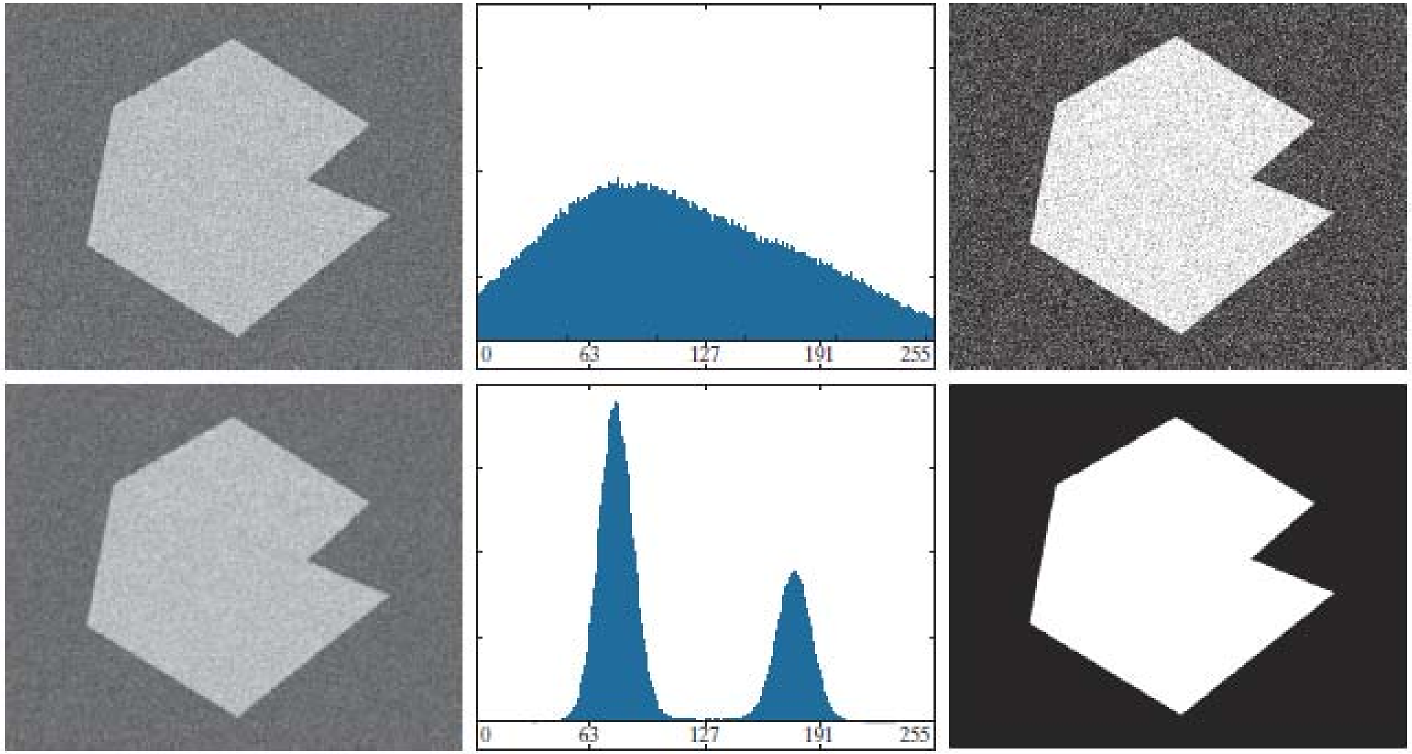
Otsu’s Method



Otsu’s Method

- We can use image smoothing to improve global thresholding.

(a) Noisy image. (b) The histogram of (a) (c) Result obtained using Otsu’s method



(d) Noisy image smoothed using an averaging kernel. (e) The histogram of (d) (f) Result obtained using Otsu’s method

Otsu’s Method

- **Problem:** When there are multiple objects in an image, direct use of the Otsu’s method will fail. Multiple thresholds are often required for correct segmentation of images with multiple objects.
- **Solution:** First use Otsu’s method to classify the image into two categories: background and foreground, and then determine whether the background or foreground contains multiple sub-targets respectively (which can be determined by the variance of the subregions).
- **Problem:** It is more time-consuming!

Region-based Segmentation

- **Points, lines, and edges:** based on discontinuities in gray levels.

- **Thresholding:** based on the distribution of pixel properties, such as gray level values and colors.

- **Region growing:** based on ***similarity***, finding the regions directly.

Region-based Segmentation

- Region-based Segmentation: R is an image, and segmentation is a process that partitions R into n sub-regions R_1, R_2, ... , R_n such that:

(a) $\bigcup_{i=1}^{n} R_i = R$

(b) R_i is a connected region, $i = 1, 2, \ldots, n$

(c) $R_i \cap R_j = \varnothing$ for all i and j, $i \neq j$

(d) $P(R_i) = \text{TRUE}$ for $i = 1, 2, \ldots, n$

(e) $P(R_i \cup R_j) = \text{FALSE}$ for $i \neq j$

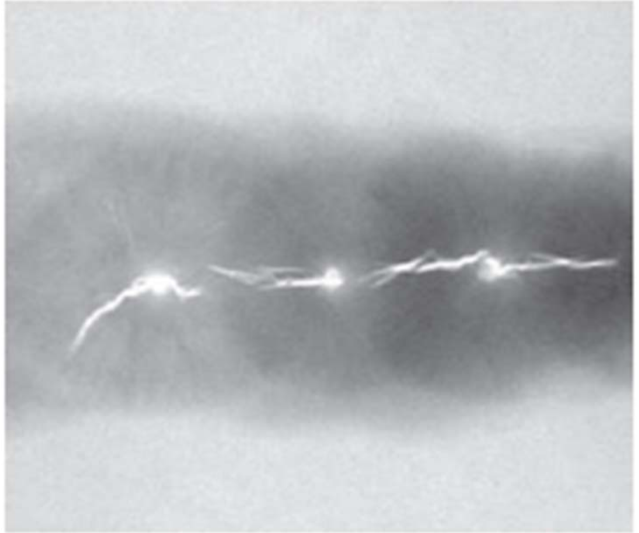
$P(R_i)$ is a logical predicate property defined over the points in set R.

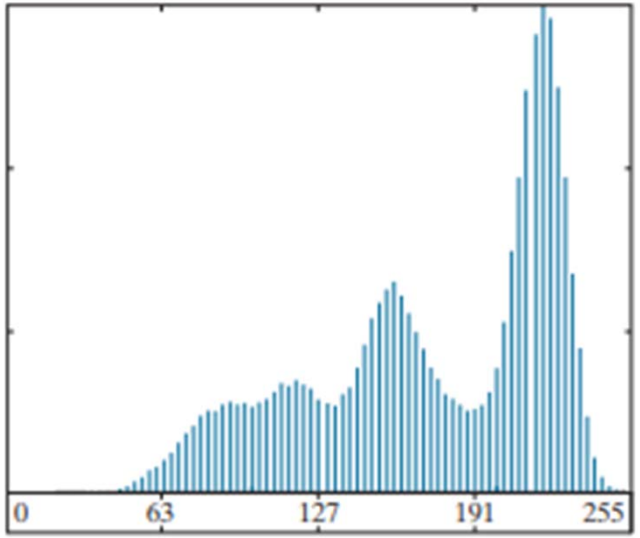
Region Growing

- Region growing is a procedure that groups pixels or sub-regions into larger regions based on predefined criteria.
  - Select a point as a start (seed) point.
  - Grow the point based on a certain criteria.
  - The selection of criteria depends not only on the problem under consideration, but also on the type of image data available. For example:
    - Land-use satellite image: depends heavily on the use of color;
    - Monochrome image: depends on set of descriptors based on gray levels and spatial properties (moments or texture).
  - Connections should be considered.
  - Another problem in region growing is the formulation of a stopping rule. Growing a region should be stop when no more pixels satisfy the criteria in that region.

Region Growing

- Seed points can be determined as those with the maximum gray level.
- Growing criteria:
  - Gray level value difference (with respect to seed points) less than a threshold.
  - Each candidate pixel should be 8-connected to that seed point.

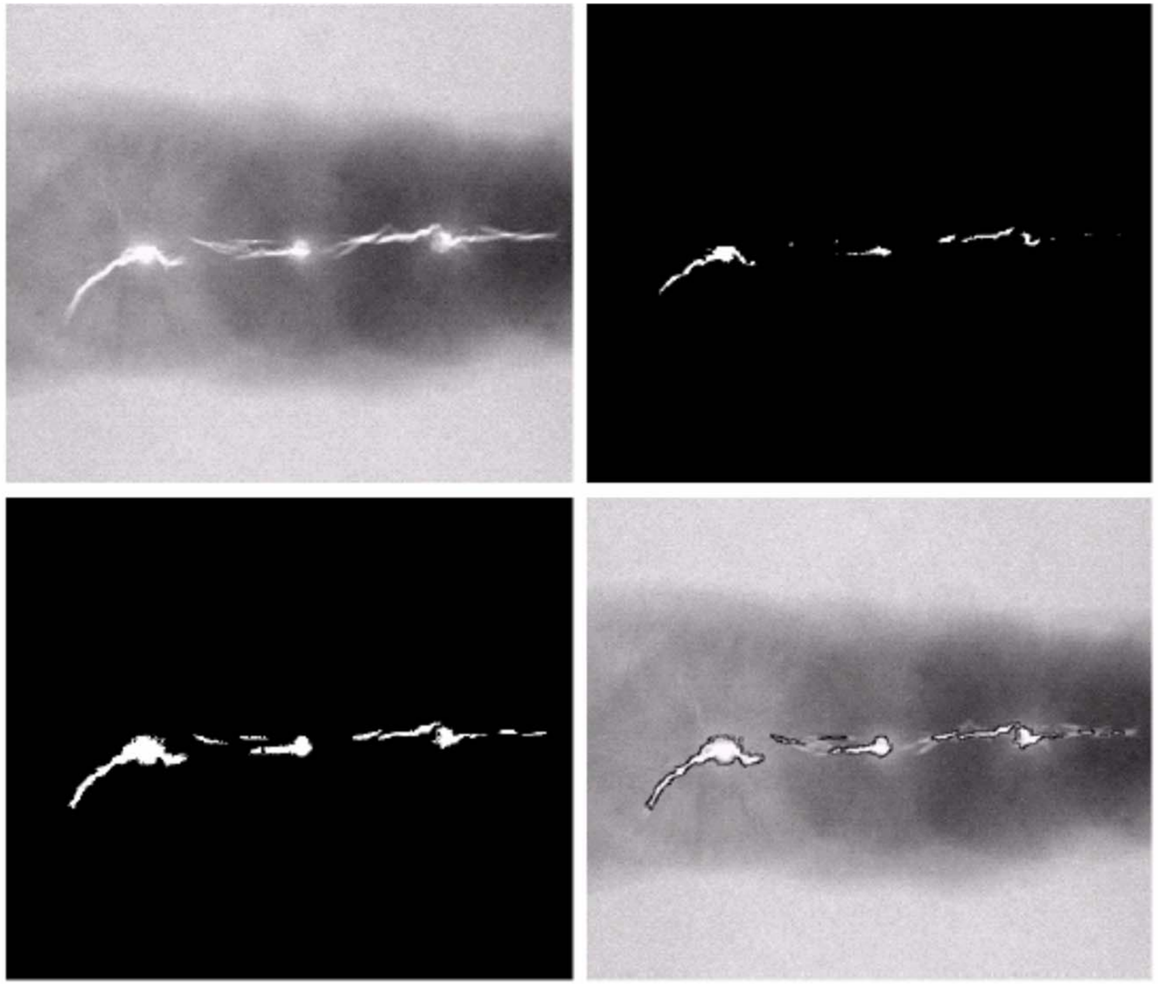




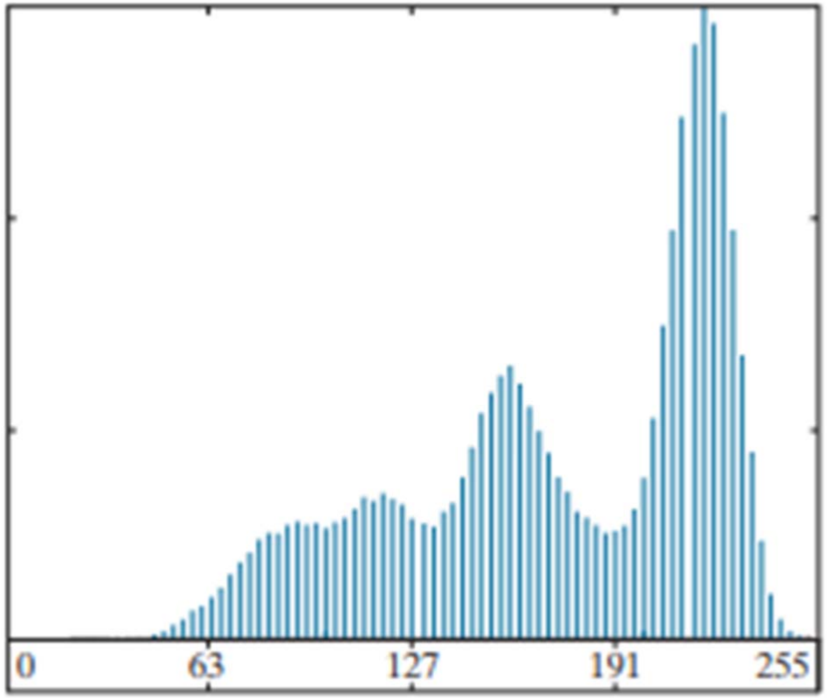
Region Growing Example

- Seed points: the points with 255 gray levels.

- Growing criteria:
  - the absolute gray-level difference between any pixel and the seed has to be less than 65
  - the pixel has to be 8-connected to at least one pixel in that region (if more the regions are merged)



Region Growing



Remark: For a problem having multimodal histograms, it can be best solved using region based approach.

Growth Process:

1. According to the position of the seed point, find the pixel value of the point.
2. With the seed point as the center, loop to find the pixels in its 8-neighbours. If the pixel difference is within a certain threshold interval set, then it satisfies the growth criterion. Press the coordinates of the point into the stack as a point to continue growing later.
3. Remove a pixel point from the stack structure and go back to step 2.
4. Repeat steps 1~3 until the growth ends when there is no eligible point in the image.

Region Growing

Seed point

$$
\begin{bmatrix} 4 & 3 & 7 & 3 & 3 \\ 1 & 7 & (8) & 7 & 5 \\ 0 & 5 & 6 & 1 & 3 \\ 2 & 2 & 6 & 0 & 4 \\ 1 & 2 & 1 & 3 & 1 \end{bmatrix}
\longrightarrow
\begin{bmatrix} 4 & 3 & (7) & 3 & 3 \\ 1 & (7) & (8) & (7) & 5 \\ 0 & 5 & 6 & 1 & 3 \\ 2 & 2 & 6 & 0 & 4 \\ 1 & 2 & 1 & 3 & 1 \end{bmatrix}
$$

Criterion: Gray level value difference <=1

$$
\begin{bmatrix} 4 & 3 & (7) & 3 & 3 \\ 1 & (7) & (8) & (7) & 5 \\ 0 & 5 & (6) & 1 & 3 \\ 2 & 2 & 6 & 0 & 4 \\ 1 & 2 & 1 & 3 & 1 \end{bmatrix}
\longrightarrow
\begin{bmatrix} 4 & 3 & (7) & 3 & 3 \\ 1 & (7) & (8) & (7) & 5 \\ 0 & (5) & (6) & 1 & 3 \\ 2 & 2 & (6) & 0 & 4 \\ 1 & 2 & 1 & 3 & 1 \end{bmatrix}
$$

Region Splitting and Merging

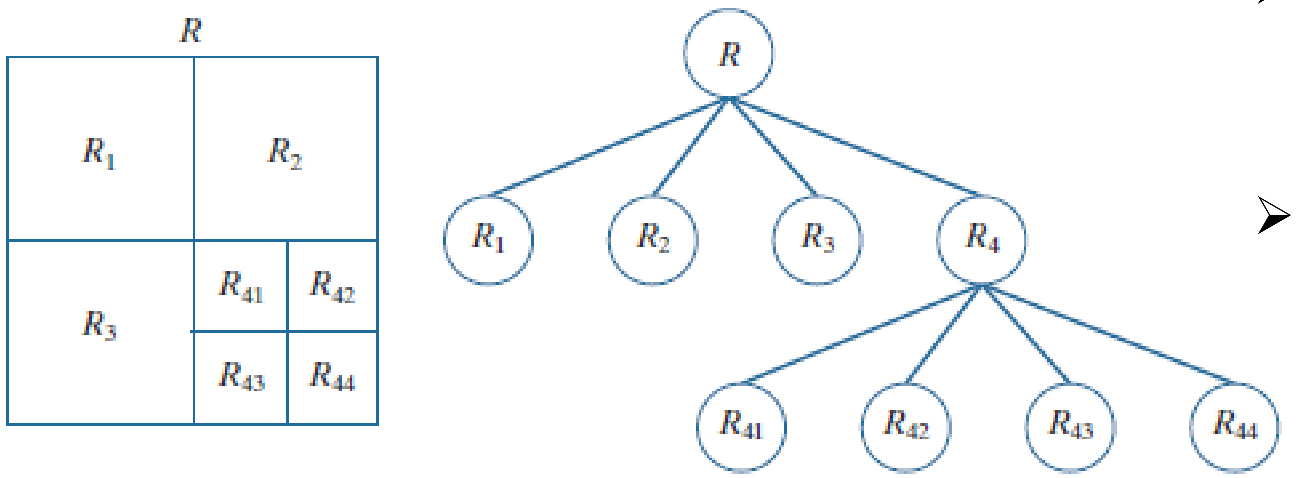
- Subdivide an image initially into a set of arbitrary, disjointed regions, and then merge and/or split the regions in an attempt to satisfy the conditions of region-oriented segmentation.

- Splitting and Merging:
  - Define a criterion for each region to be a valid segment.
  - Split each region which does not satisfy the criteria.
  - Merge two neighboring region based on the criterion.

- Quadtree-based algorithm

Region Splitting and Merging

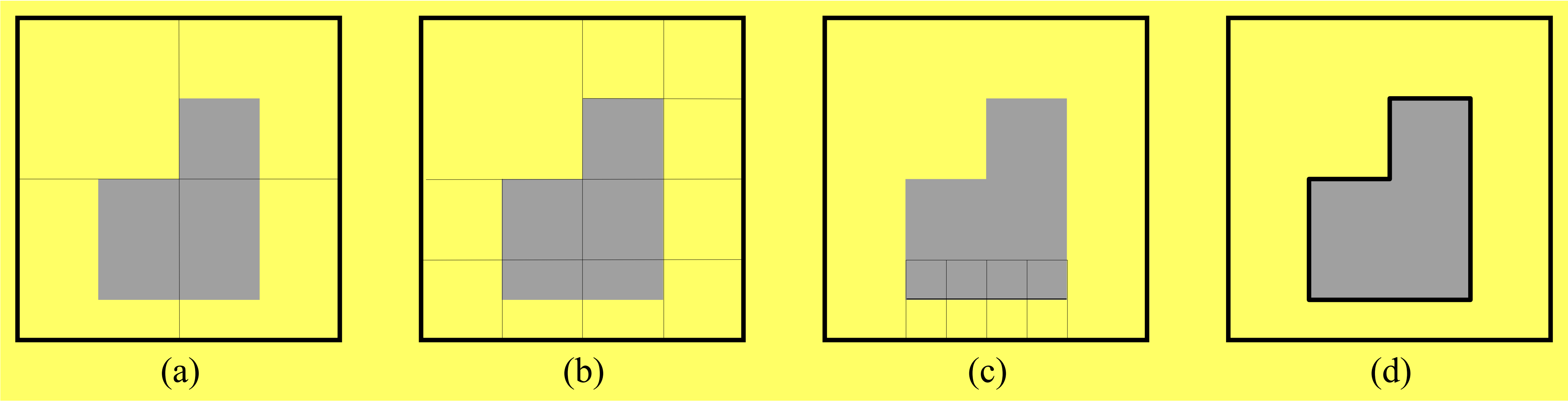
- Procedure:
  - Split into 4 disjointed quadrants, if any region R_i where $P(R_i)$ = FALSE
  - Merge any adjacent regions R_j and R_k for which $P(R_j \cup R_k)$ = TRUE
  - Stop when no further splitting or merging is possible.



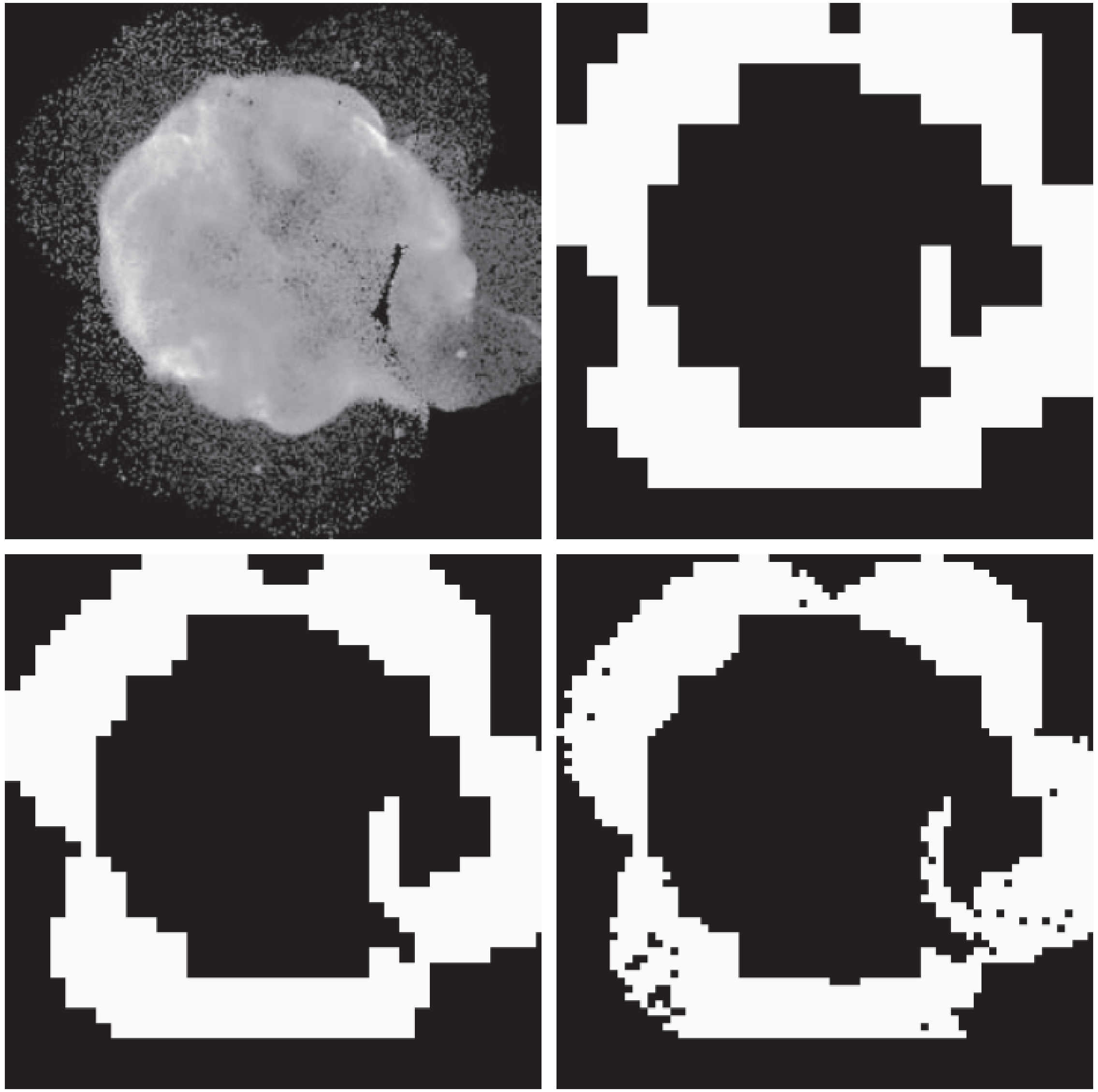
- ➢ If only splitting were used, the final partition likely would contain adjacent regions with identical properties.
- ➢ This drawback may be remedied by allowing merging, as well as splitting.

Region Splitting and Merging

Criteria: If the gray level in the R_i are same, then $P(R_i)$ = TRUE



Region Splitting and Merging



(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope.

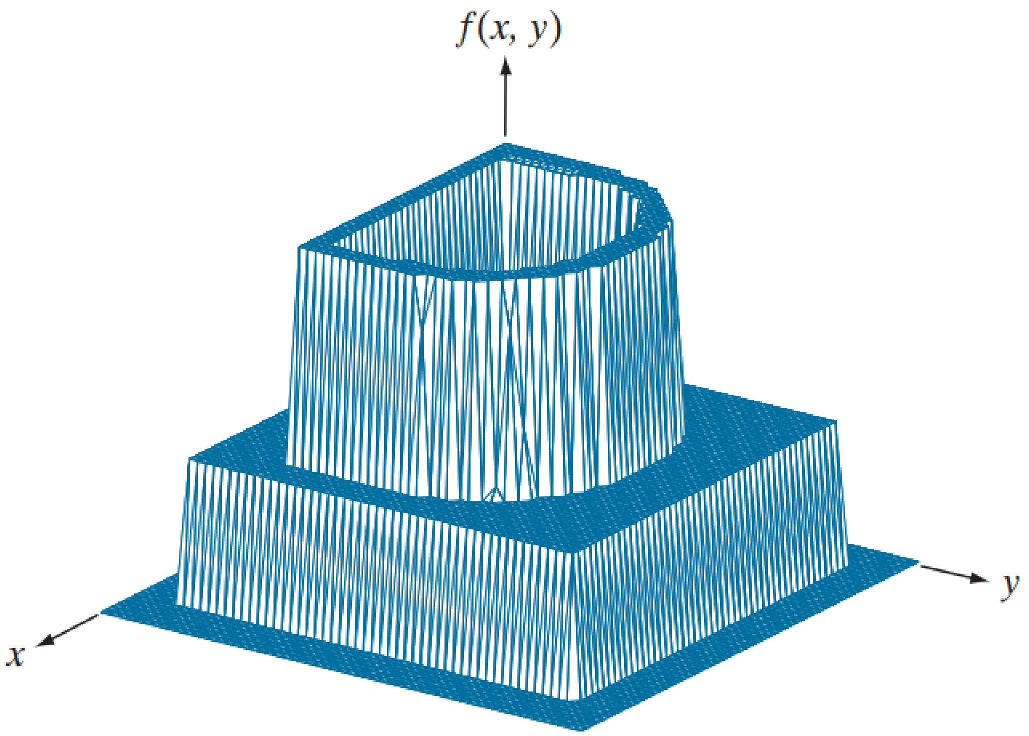
(b) Results of limiting the smallest allowed quadregion to be of sizes of 32 × 32.

(c) Results of limiting the smallest allowed quadregion to be of sizes of 16 × 16.

(d) Results of limiting the smallest allowed quadregion to be of sizes of 8 × 8 pixels.

Morphological Watersheds

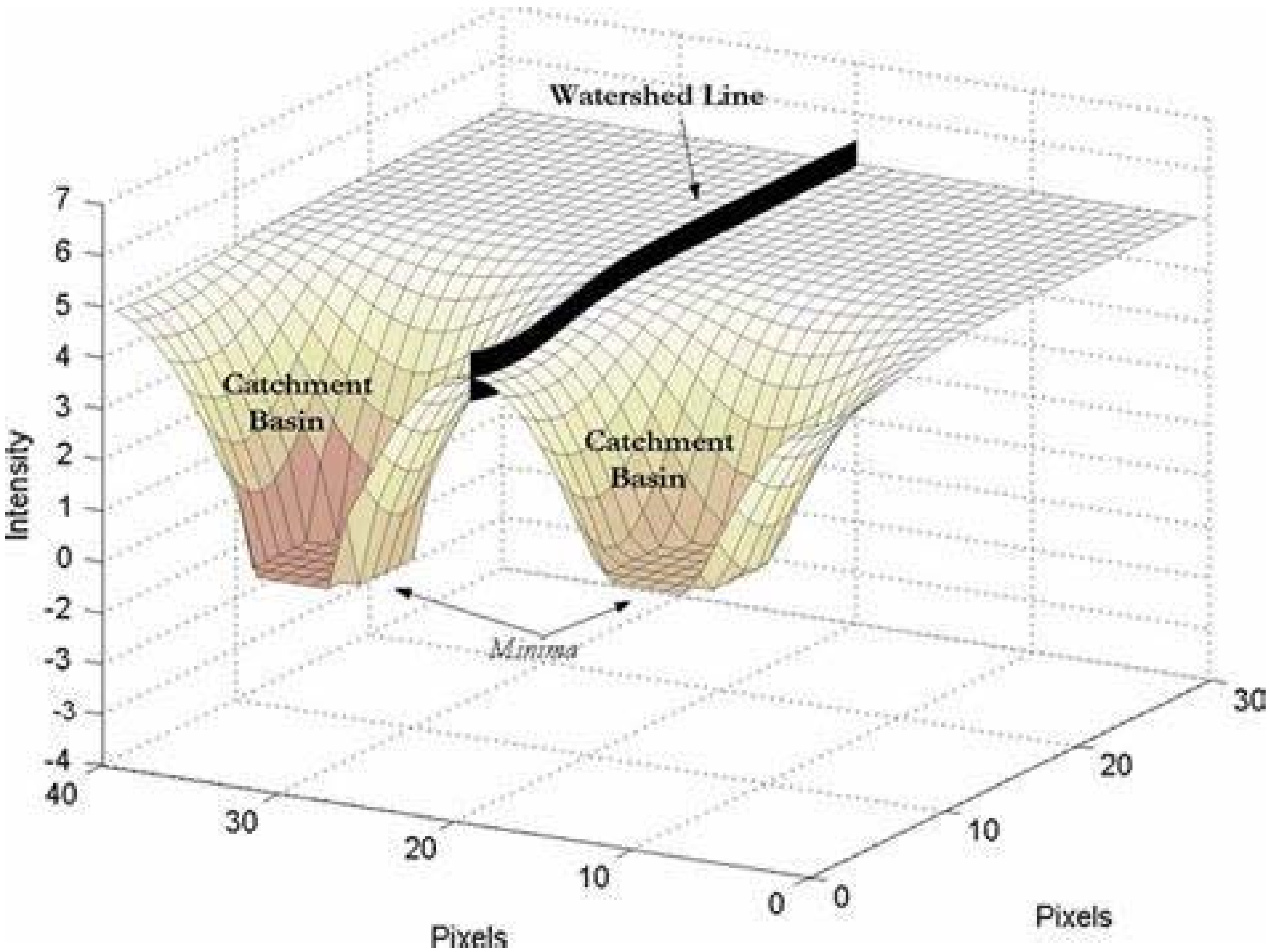
- The concept of a watershed is based on visualizing an image in three dimensions, two spatial coordinates versus intensity, as the figure below.



Morphological Watersheds

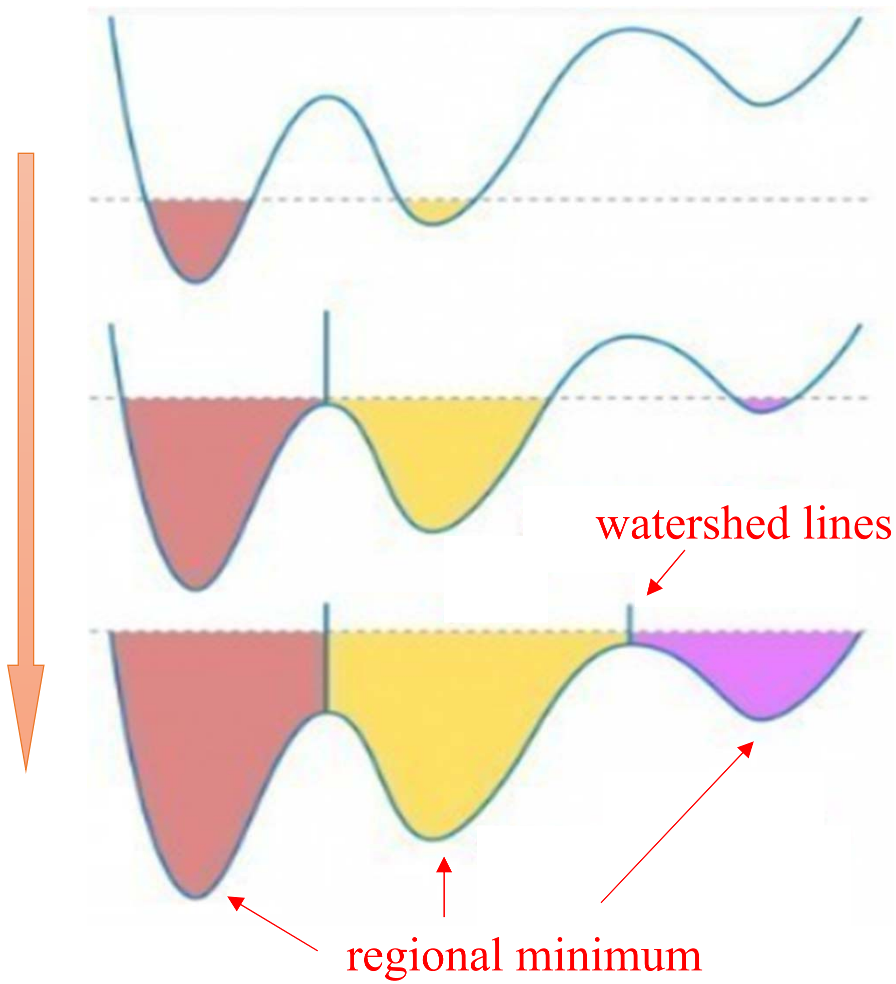
- In such a “topographic” interpretation, we consider three types of points:
  1) points belonging to a regional minimum;
  2) points at which a drop of water, if placed at the location of any of those points, would fall with certainty to a single minimum;
  3) points at which water would be equally likely to fall to more than one such minimum.
- For a particular regional minimum, the set of points satisfying condition (2) is called the *catchment basin* or *watershed* of that minimum. The points satisfying condition (3) form crest lines on the topographic surface, and are referred to as *divide lines* or *watershed lines*.

Morphological Watersheds

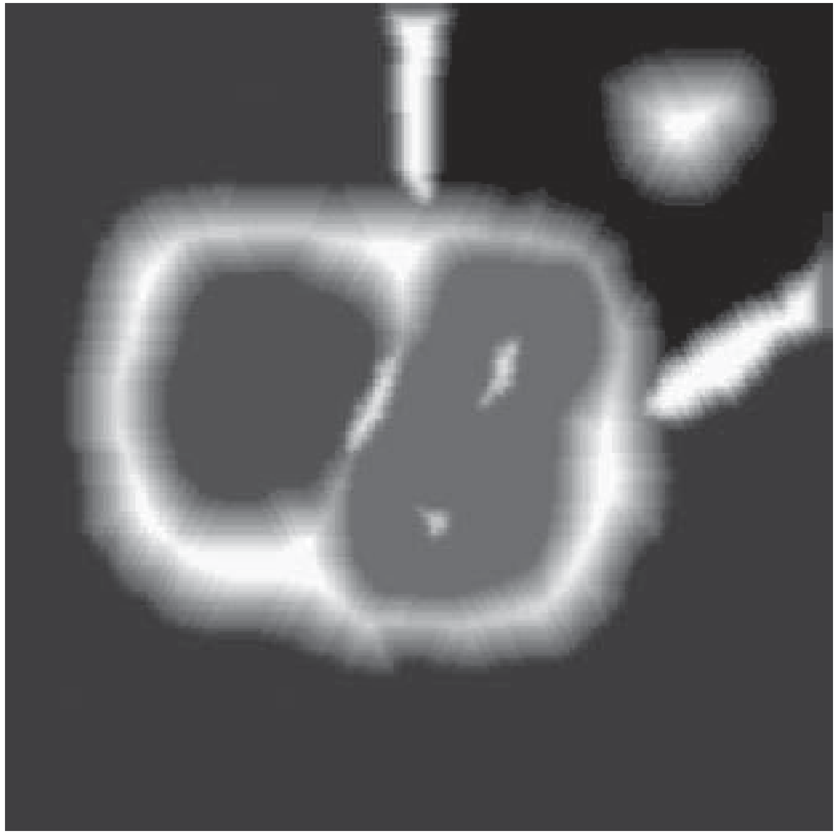


Morphological Watersheds

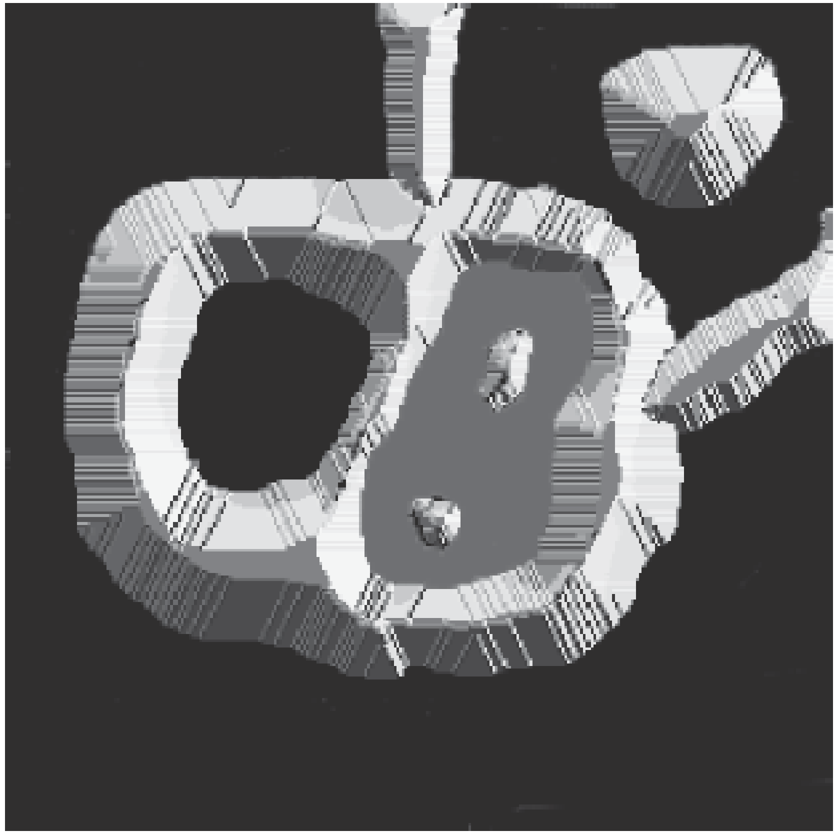
- The principal objective of segmentation algorithms based on these concepts is to find the watershed lines.
  - Suppose that a hole is punched in each regional minimum and that the entire topography is flooded from below by letting water rise through the holes at a uniform rate.
  - Here, water from the lower part of the left basin overflowed into the basin on the right, and a short "dam" (consisting of single pixels) was built to prevent water from merging at that level of flooding.
  - This process is continued until the maximum level of flooding (corresponding to the highest intensity value in the image) is reached.



Morphological Watersheds

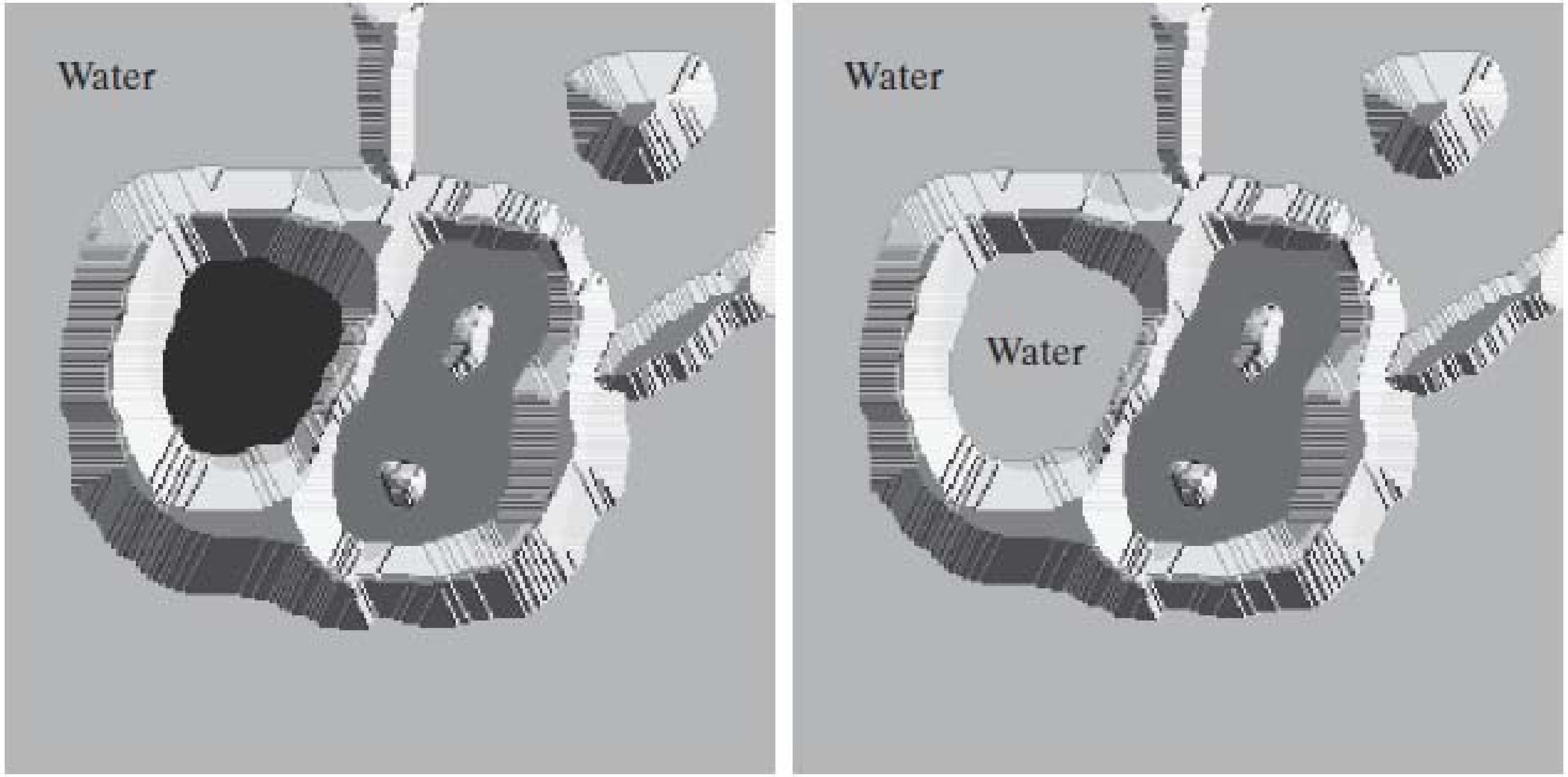


Original image.



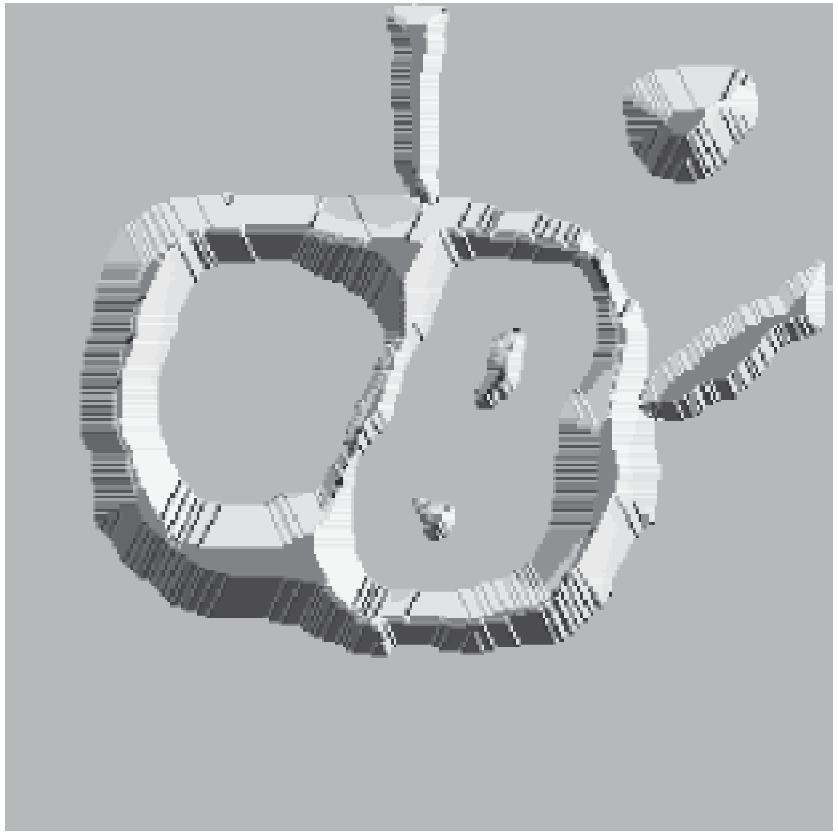
Topographic view. Only the background is *black*. The basin on the left is slightly lighter than black.

Morphological Watersheds

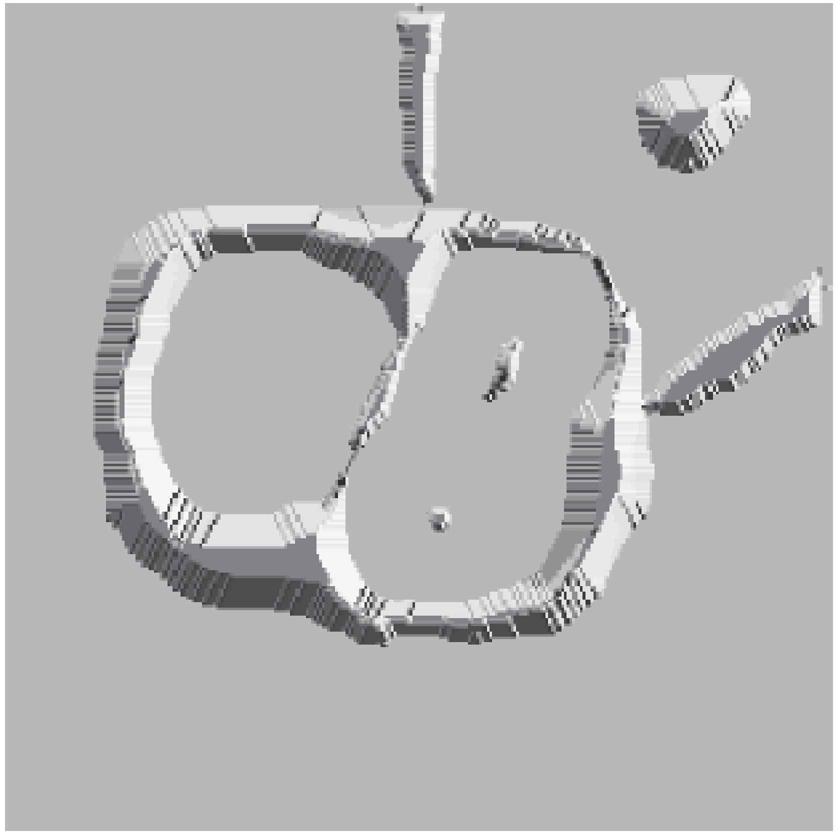


Two stages of flooding. All constant dark values of gray are intensities in the original image. Only constant *light gray* represents “water.”

Morphological Watersheds

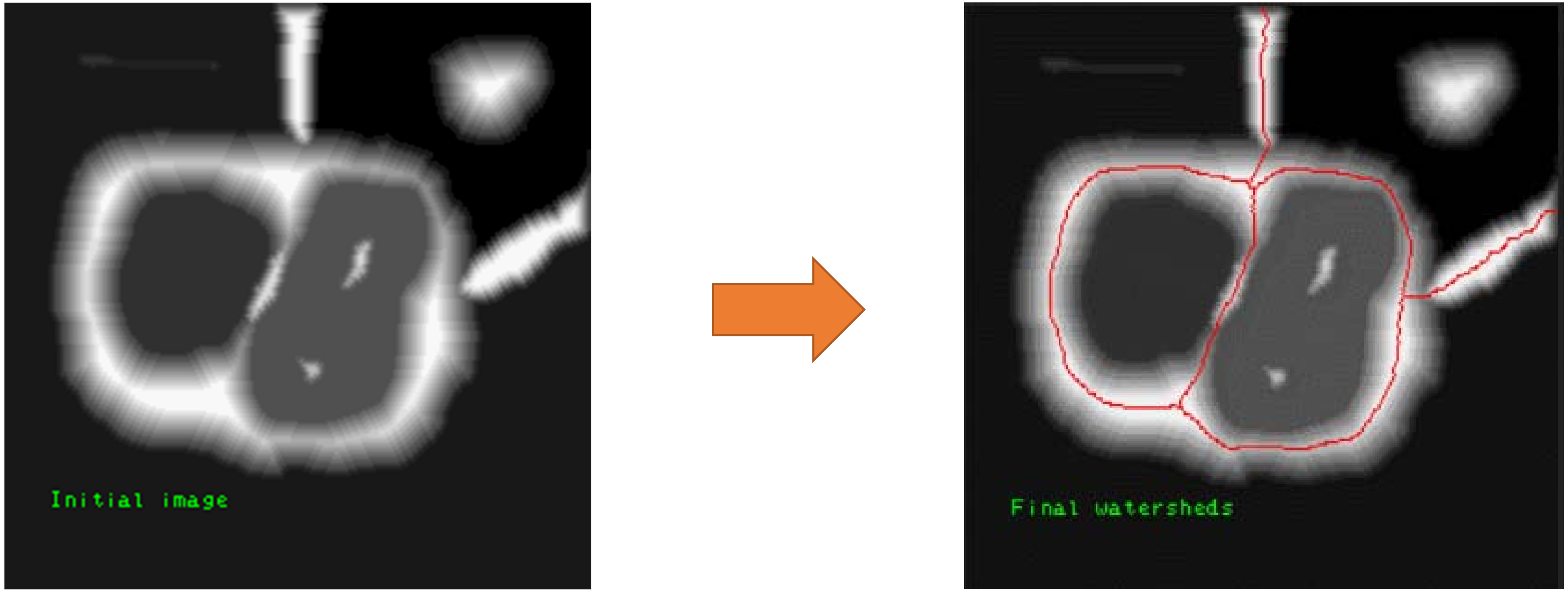


Result of further flooding



Beginning of merging of water from two catchment basins (a short dam was built between them).

Morphological Watersheds



The process of building dams.

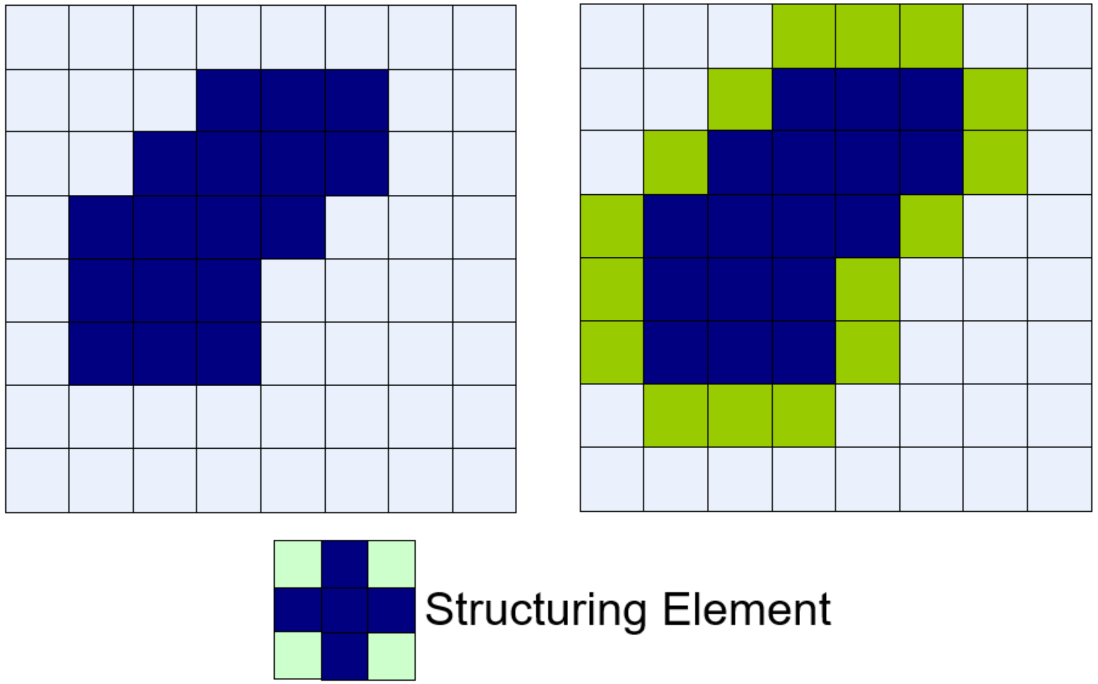
Note the important property that the watershed lines form connected paths, thus giving continuous boundaries between regions.

Morphological Watersheds

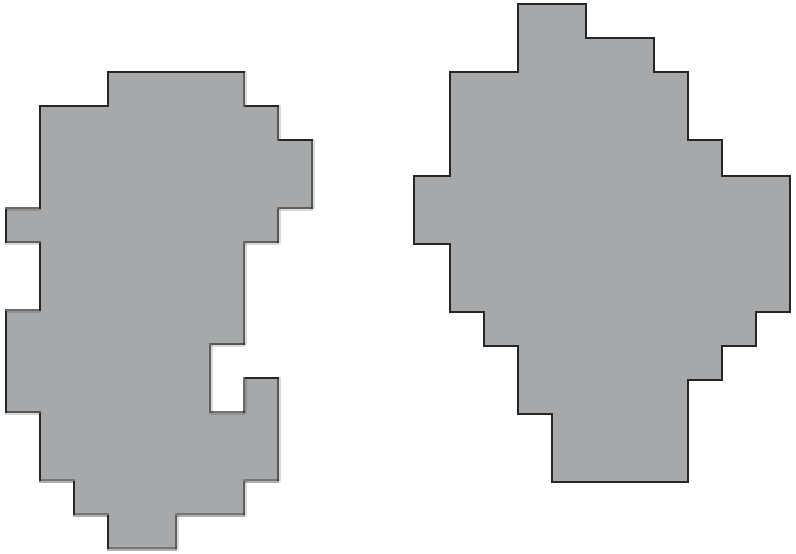
- One of the principal applications of watershed segmentation is in the extraction of nearly uniform (blob-like) objects from the background.
- Regions characterized by small variations in intensity have small gradient values.
- Thus, in practice, we often see watershed segmentation applied to the gradient of an image, rather than to the image itself.
- In this formulation, the regional minima of catchment basins correlate nicely with the small value of the gradient corresponding to the objects of interest.

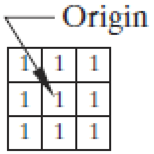
Morphological Watersheds

- Dam construction is based on binary images.
- The simplest way to construct dams separating sets of binary points is to use morphological dilation.

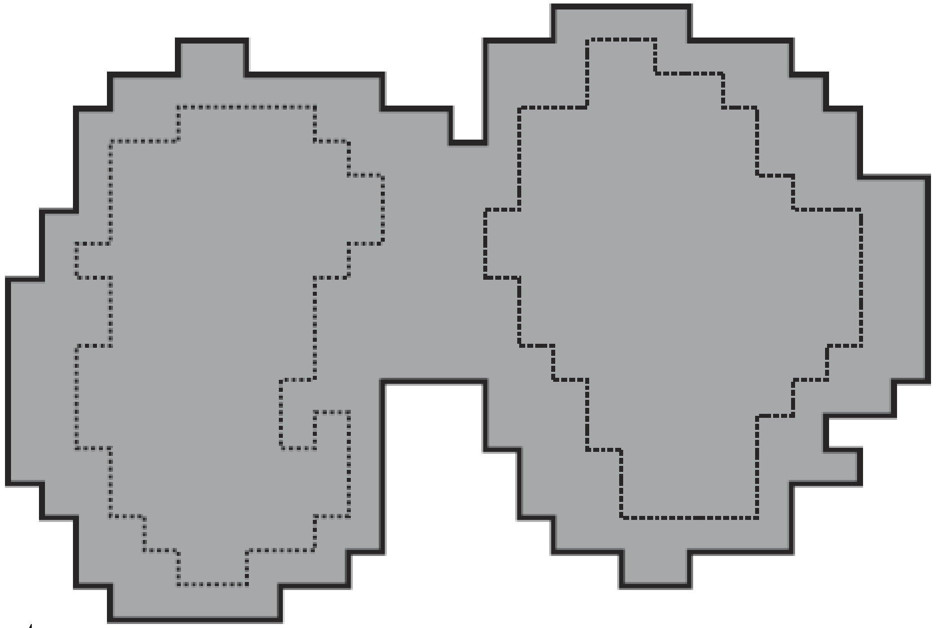


Dam Construction





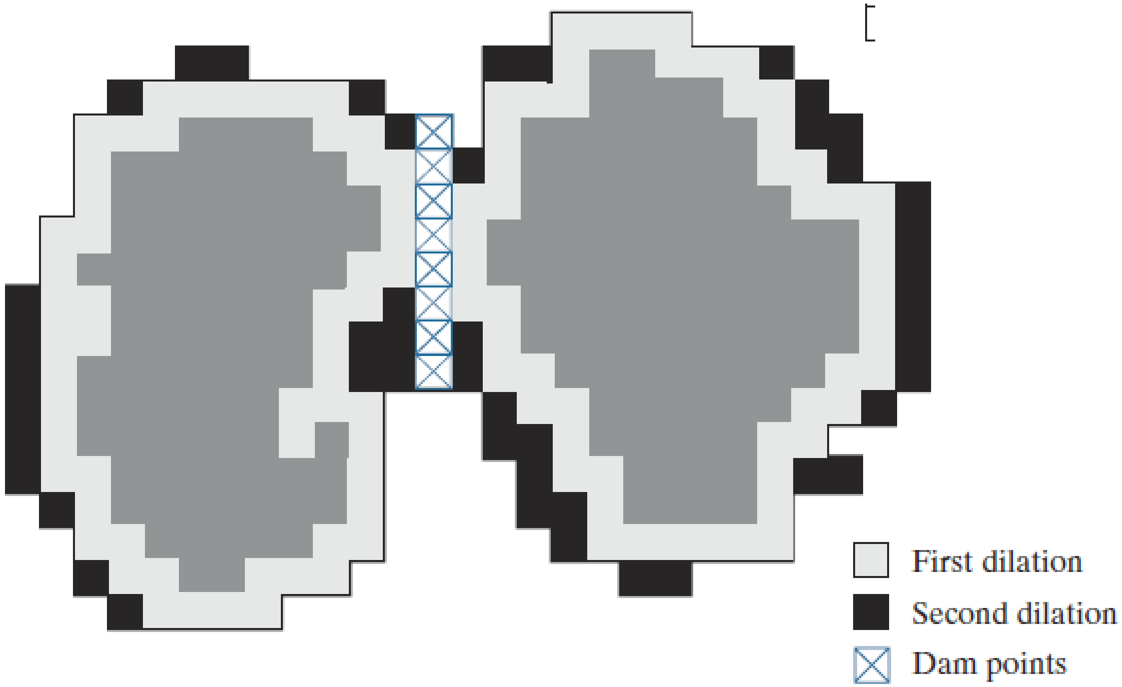
Structuring element used for dilation



Two partially flooded catchment basins at stage $n-1$ of flooding.

Flooding at stage n, showing that water has spilled between basins. Therefore, a dam must be built.

Dam Construction



Result of dilation and dam construction.

- A first dilation pass (in light gray) expanded the boundary of each original connected component.
- In the second dilation, shown in black, the one-pixel-thick connected path shown crossed-hatched is the desired separating dam at stage n of flooding.
- Construction of the dam at this level of flooding is completed by setting all the points in the path just determined to a value greater than the maximum possible intensity value of the image (e.g., greater than 255 for an 8-bit image).

Watershed Segmentation Algorithm

Original Image

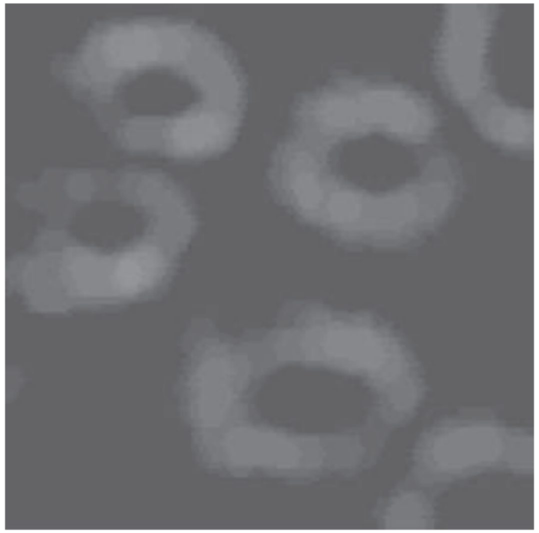
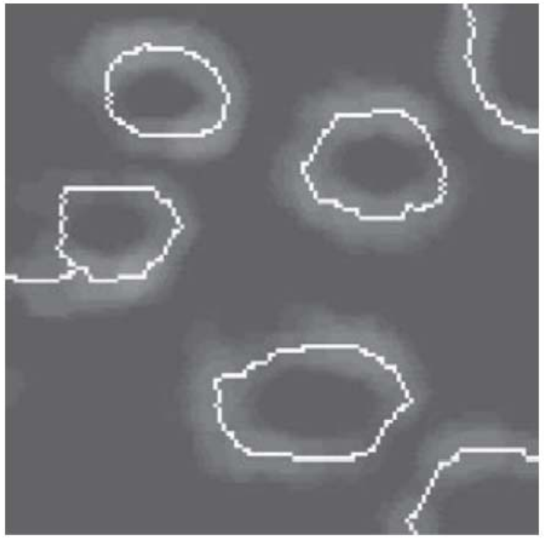
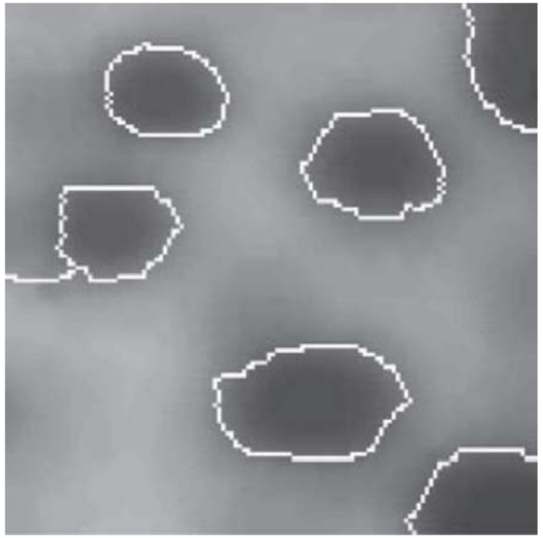


Image gradient

Watershed lines superimposed on the gradient image.

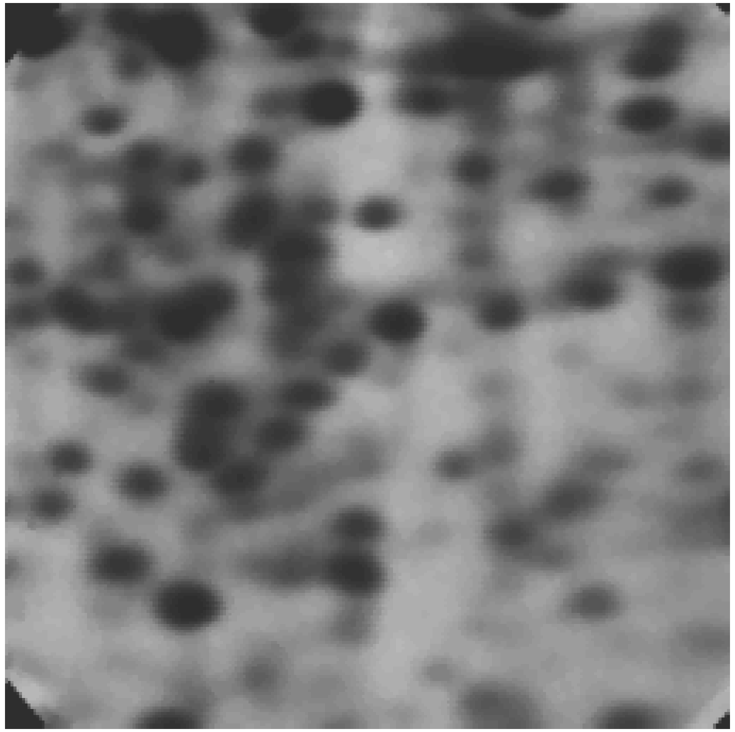


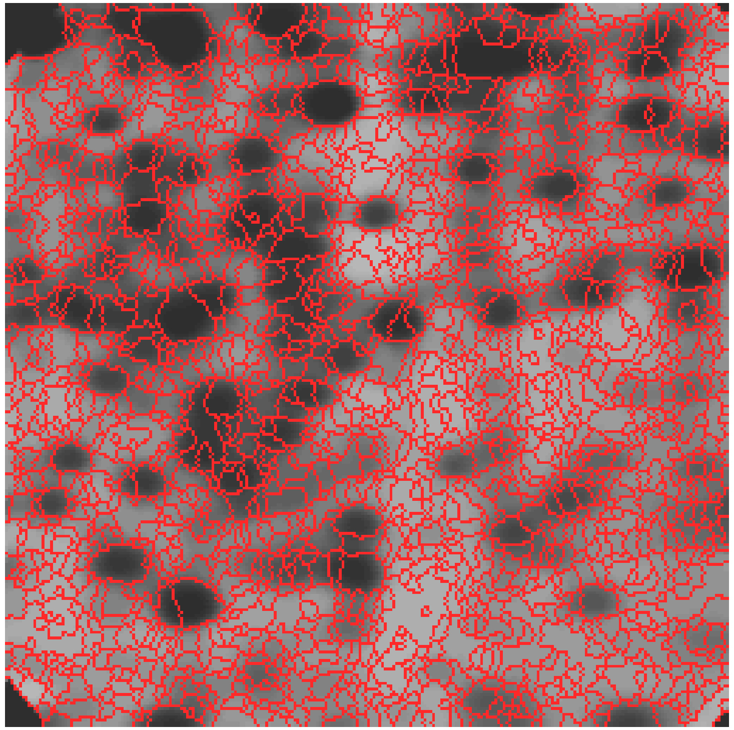


Watershed lines superimposed on the original image

Watershed Segmentation Algorithm

- Direct application of the watershed segmentation algorithm in the form discussed in the previous section generally leads to over-segmentation, caused by noise and other local irregularities of the gradient.





Watershed Segmentation Algorithm

Solutions:

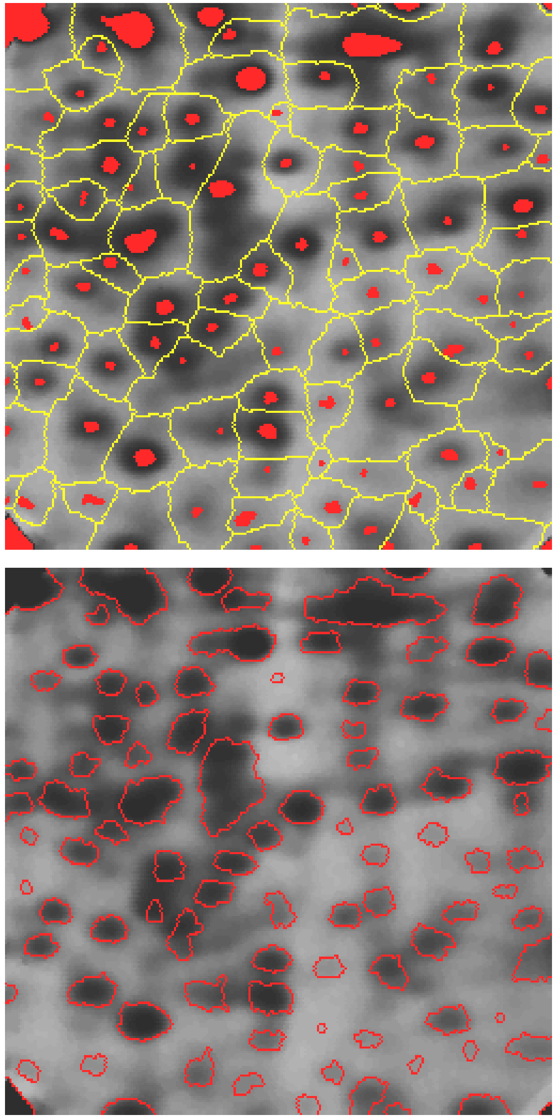
- A practical solution to this problem is to limit the number of allowable regions by incorporating a preprocessing stage designed to bring additional knowledge into the segmentation procedure.
- An effective method for minimizing the effect of small spatial detail is to filter the image with a smoothing filter.

Watershed Segmentation Algorithm

- An approach used to control over-segmentation is based on the concept of markers.
  - A *marker* is a connected component belonging to an image. We have *internal markers*, associated with objects of interest, and *external markers*, associated with the background.

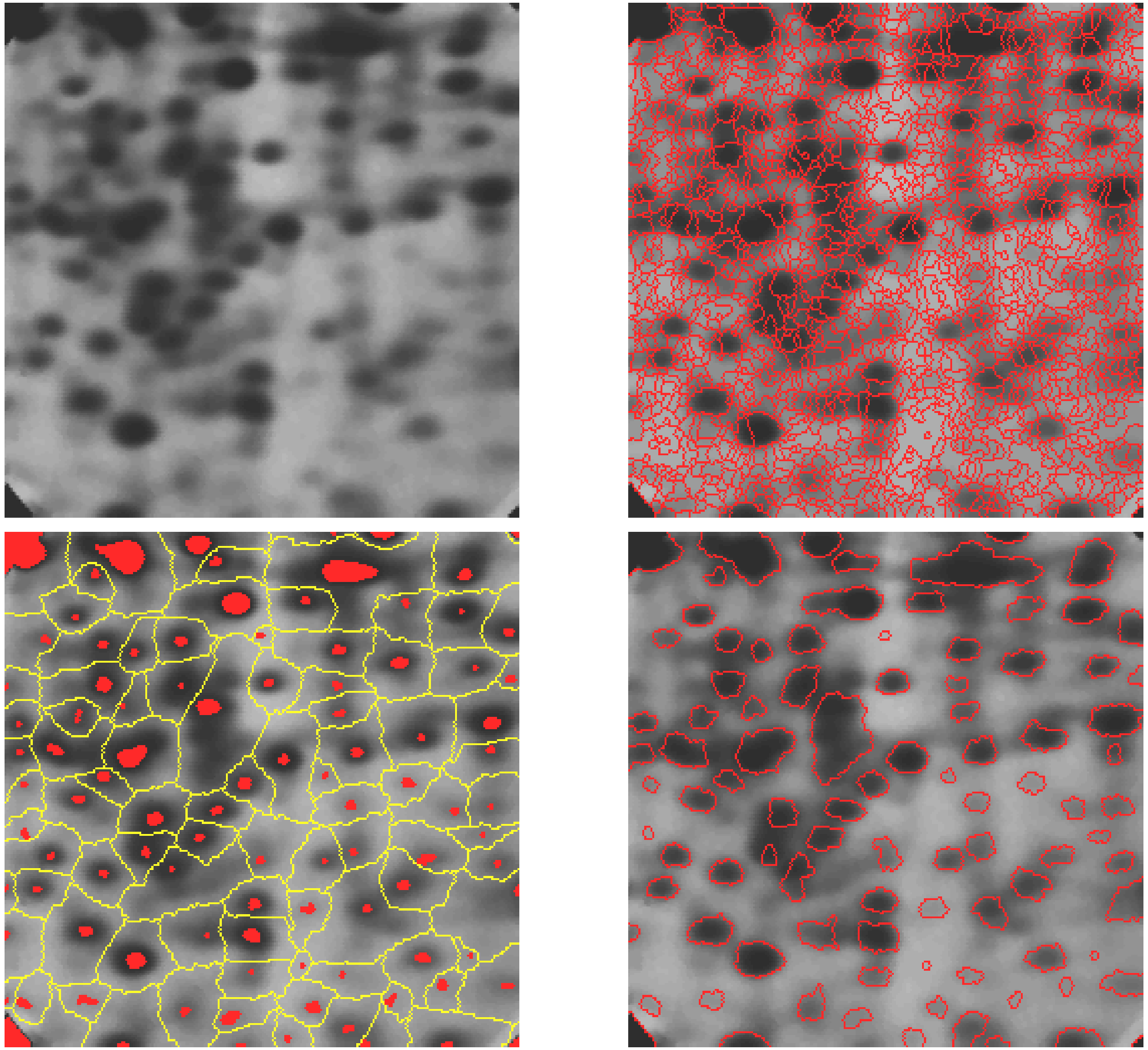
- Suppose that we define an internal marker as
  1) a region that is surrounded by points of higher “altitude”;
  2) such that the points in the region form a connected component;
  3) in which all the points in the connected component have the same intensity value.

Watershed Segmentation Algorithm



- For the previous image, smoothing is performed first to suppress the effect of noise.
- The target of interest is labeled (red region) in the original image using internal labeling, and the upper image result is obtained by using the watershed algorithm on the labeled image.
- Apply the watershed segmentation algorithm again to each region of the upper image segmentation (containing only the object and the background) to get the right segmentation result.

Watershed Segmentation Algorithm



Summary

- In this lecture we have learnt:
  - Thresholding
  - Region based segmentation
  - Morphological watersheds approach

Optional Homework

Check the Textbook!

- **Chapter 10: Problems 10.32, 10.33, 10.41, 10.42, 10.49**

- Homework answers will be provided at the end of each week.