EIE4512 - Digital Image Processing
Image Segmentation

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Zhen Li
lizhen@cuhk.edu.cn
School of Science and Engineering
The Chinese University of Hong Kong, Shen Zhen
April 25, 2019

Image Segmentation

- Obtain a compact representation of the image to be used for further processing.
- Group together similar pixels
- Image intensity is not sufficient to perform *semantic* segmentation
  - Object recognition
    - Decompose objects to simple tokens (line segments, spots, corners)
  - Finding buildings in images
    - Fit polygons and determine surface orientations.
  - Video summarization
    - Shot detection

Image Segmentation (cont.)

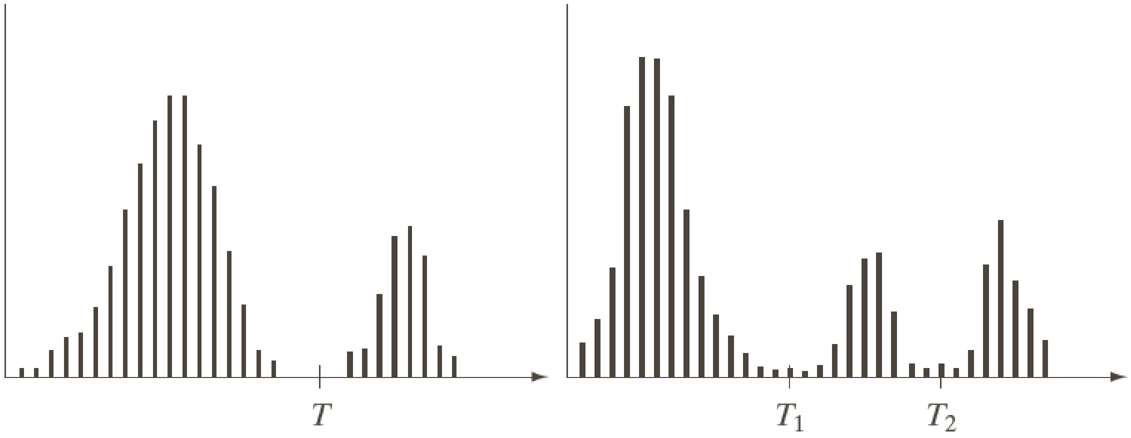
Goal: separate an image into “coherent” regions.

- Basic methods
  - point, line, edge detection
  - thresholding
  - region growing
  - morphological watersheds
- Advanced methods
  - clustering
  - model fitting.
  - probabilistic methods.
  - …

Thresholding

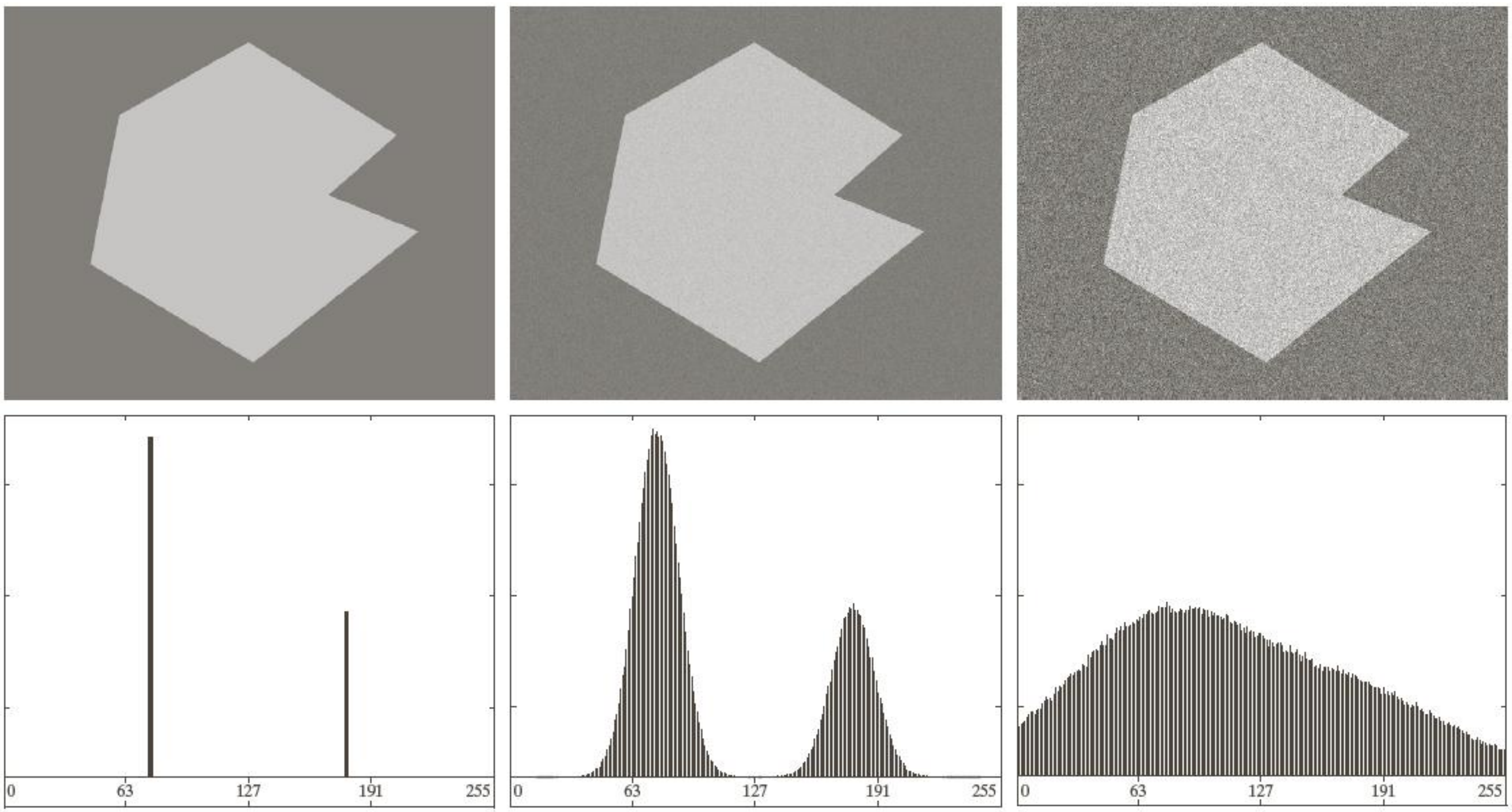
- Image partitioning into regions directly from their intensity values.

$$g(x,y)=\begin{cases}1 & \text{if } f(x,y)>T\\ 0 & \text{if } f(x,y)\le T\end{cases}\qquad g(x,y)=\begin{cases}0 & \text{if } f(x,y)\le T_1\\ 1 & \text{if } T_1< f(x,y)\le T_2\\ 2 & \text{if } f(x,y)>T_2\end{cases}$$



Noise in Thresholding

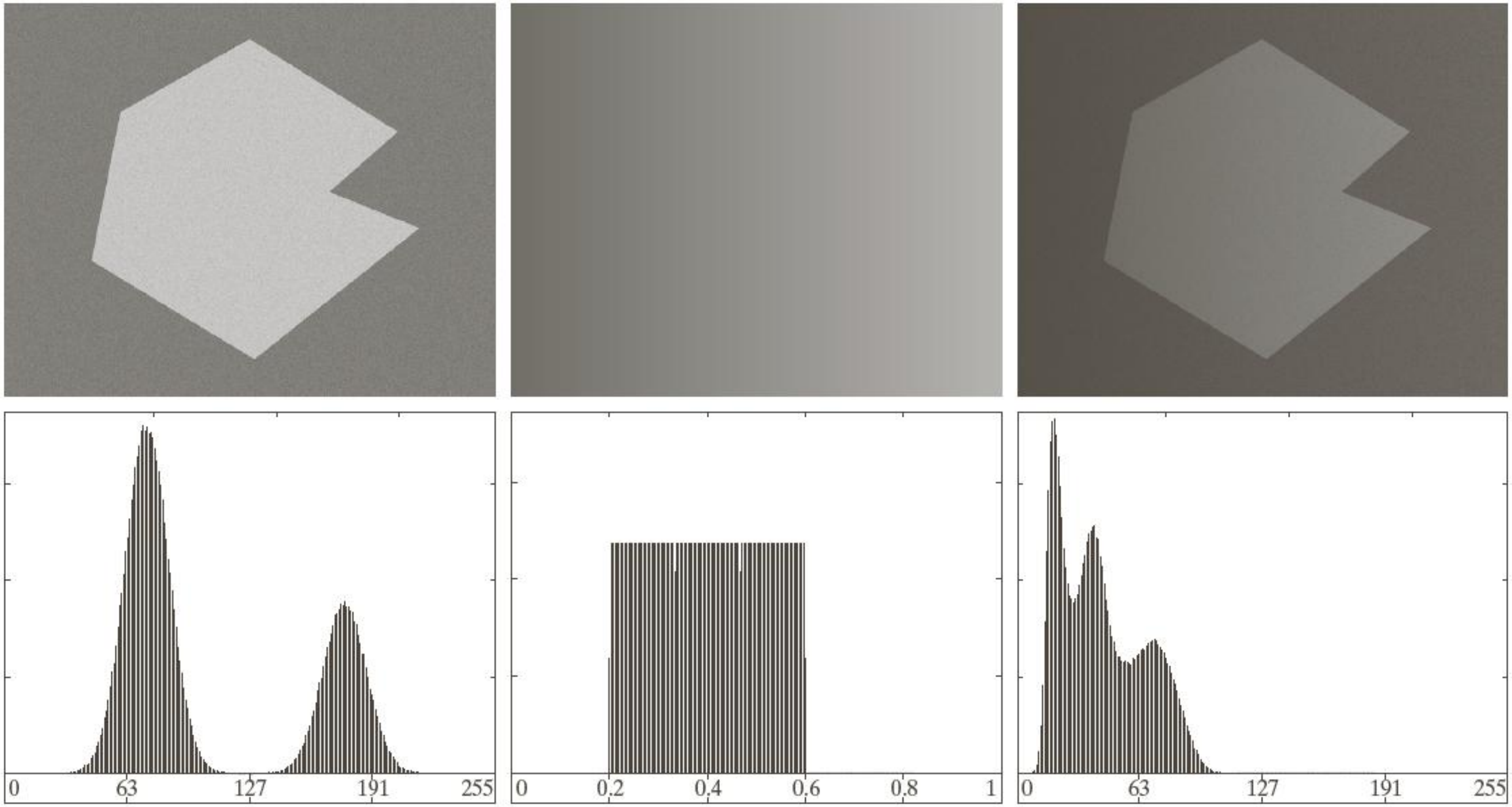
Difficulty in determining the threshold due to noise



Noiseless Gaussian (μ=0, σ=10) Gaussian (μ=0, σ=10)

Illumination in Thresholding

Difficulty in determining the threshold due to non-uniform illumination



(a) Noisy image

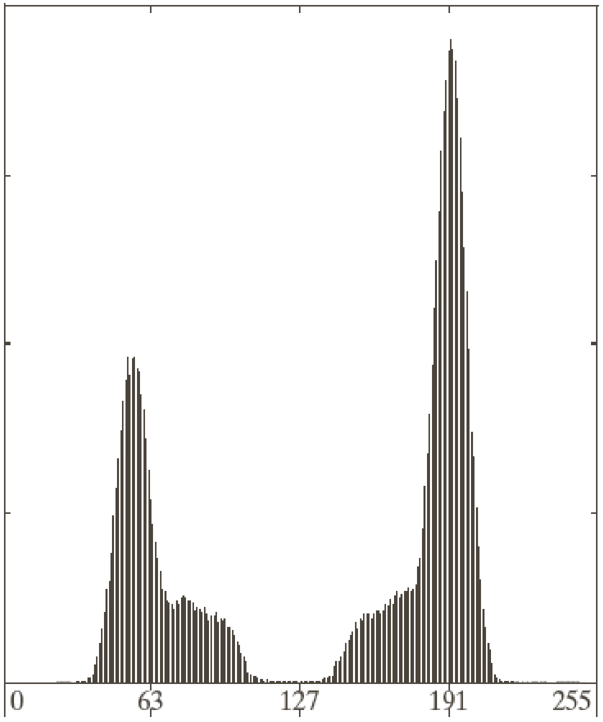
(b) Intensity ramp image

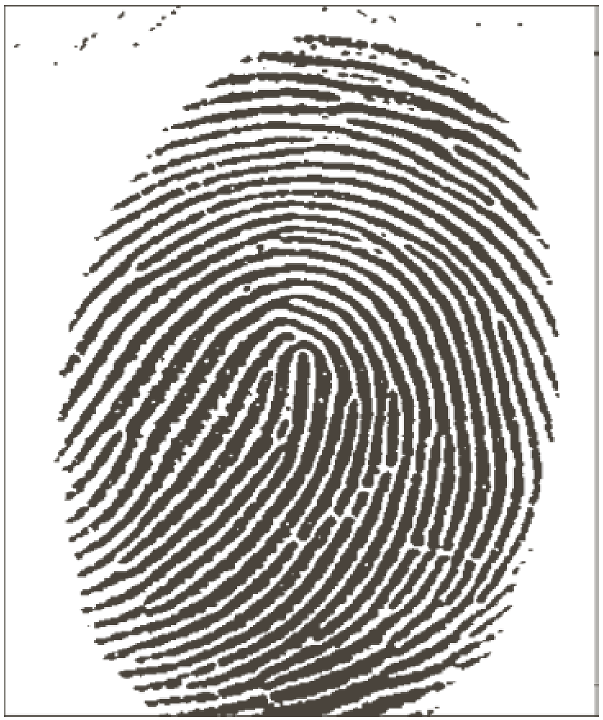
Multiplication

Basic Global Thresholding

- **Algorithm**
  - Select initial threshold estimate T.
  - Segment the image using T
    - Region G_1 (values > T) and region G_2 (values < T).
  - Compute the average intensities m_1 and m_2 of regions G_1 and G_2 respectively.
  - Set $\mathrm{T}=(m_1+m_2)/2$
  - Repeat until the change of T in successive iterations is less than ΔT.

Basic Global Thresholding (cont.)





T=125

Optimum Global Thresholding using Otsu’s Method

- The method is based on statistical decision theory.
- Minimization of the average error of assignment of pixels to two (or more) classes.
- Bayes decision rule may have a nice closed form solution to this problem provided
  - The pdf of each class.
  - The probability of class occurence.
- Pdf estimation is not trivial and assumptions are made (Gaussian pdfs).
- Otsu (1979) proposed an attractive alternative maximizing the *between-class variance*.
  - Only the histogram of the image is used.

Otsu's Method (cont.)

- Let $\{0, 1, 2, \ldots, L\text{-}1\}$ denote the intensities of a $M\mathrm{x}N$ image and n_i the number of pixels with intensity i.
- The normalized histogram has components:

$$p_i = \frac{n_i}{MN}, \quad \sum_{i=0}^{L-1} p_i = 1, \quad p_i \geq 0$$

- Suppose we choose a threshold k to segment the image into two classes:
  - C_1 with intensities in the range $[0, k]$,
  - C_2 with intensities in the range $[k+1, L\text{-}1]$.

Otsu's Method (cont.)

- The probabilities of classes C_1 and C_2:

$$P_1(k) = \sum_{i=0}^{k} p_i, \quad P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

- The mean intensity of class C_1 :

$$m_1(k) = \sum_{i=0}^{k} iP(i \mid C_1) = \sum_{i=0}^{k} i \frac{P(C_1 \mid i)P(i)}{P(C_1)} = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i$$

Intensity i belongs to class C_1 and $P(C_1 \mid i) = 1$

Otsu's Method (cont.)

- Similarly, the mean intensity of class C_2:

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i \mid C_2) = \sum_{i=k+1}^{L-1} i \frac{P(C_2 \mid i)P(i)}{P(C_2)} = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

- Mean image intensity:

$$m_G = \sum_{i=0}^{L-1} ip_i = P_1(k)m_1(k) + P_2(k)m_2(k)$$

- Cumulative mean image intensity (up to k):

$$m(k) = \sum_{i=0}^{k} ip_i$$

Otsu's Method (cont.)

- Between class variance:

$$\sigma_B^2(k) = P_1(k)[m_1(k) - m_G]^2 + P_2(k)[m_2(k) - m_G]^2$$

- With some manipulation we get:

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

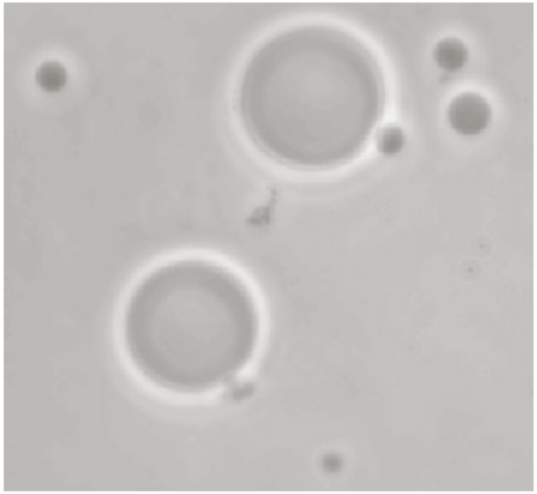
- The value of k is selected by sequential search as the one maximazing:

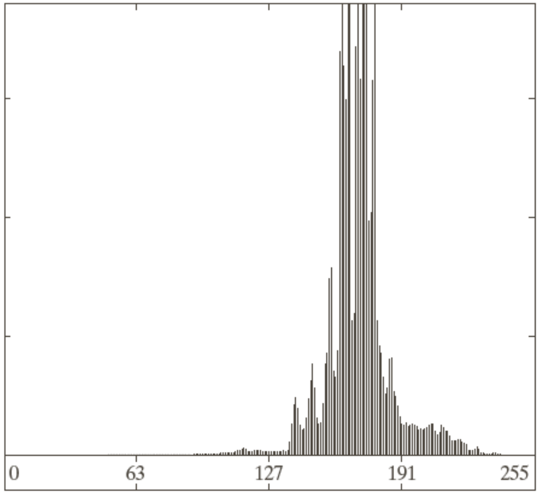
$$k^* = \max_{0 \le k \le L-1} \{\sigma_B^2(k)\}$$

Otsu's Method (cont.)

Example (no obvious valley in the histogram)

Image





Histogram

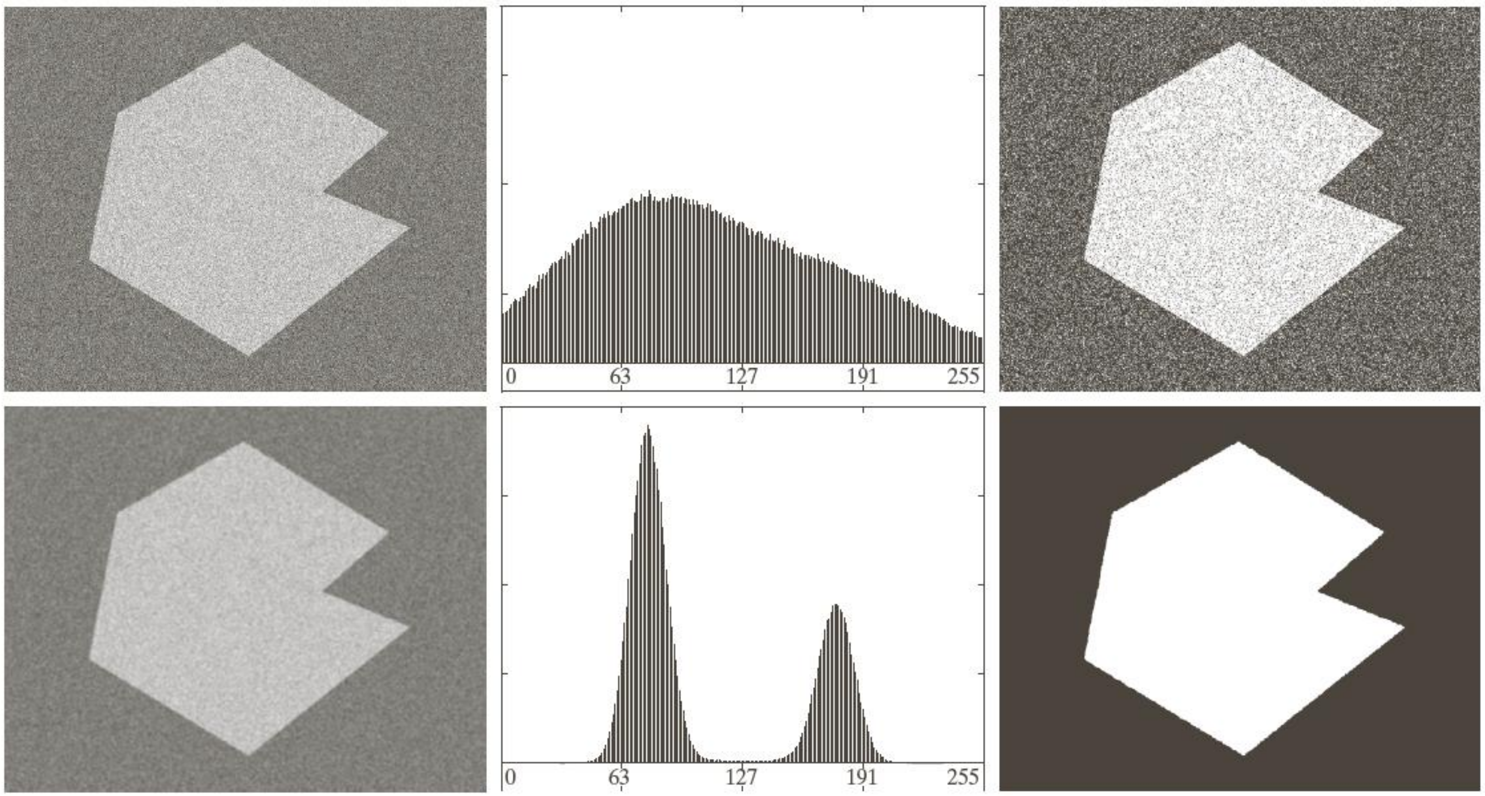
Basic method

Otsu's method

Otsu's Method (cont.)

Smoothing helps thresholding (may create histogram valley)

Noisy image Histogram Otsu no smoothing

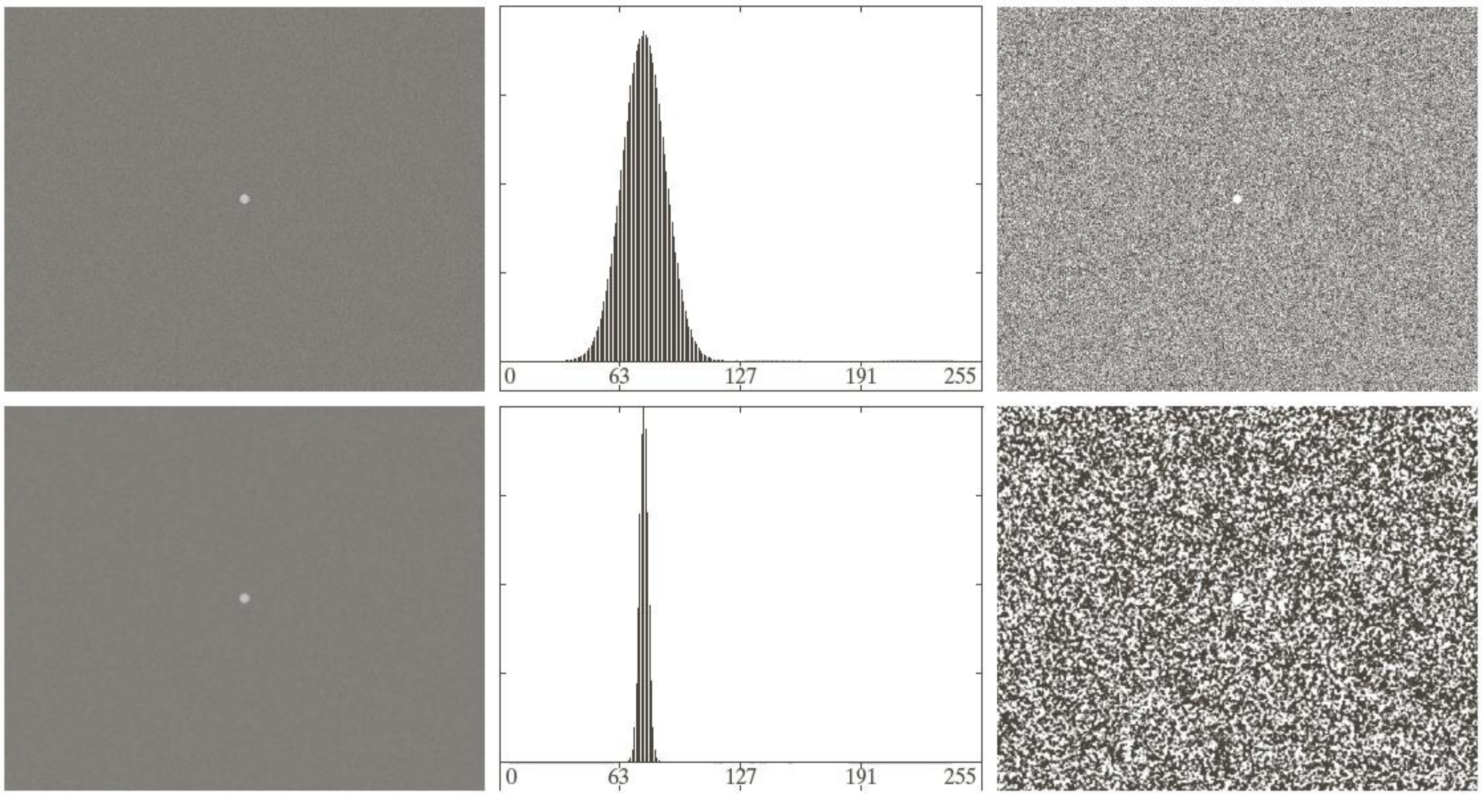


Noisy image smoothed Histogram Otsu with smoothing

Otsu’s Method (cont.)

Otsu’s method, even with smoothing cannot extract small structures

Noisy image, σ=10 | Histogram | Otsu no smoothing



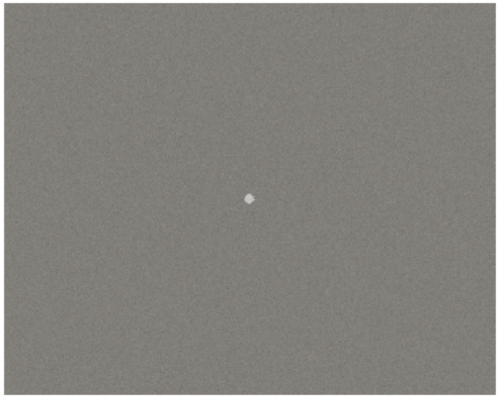
Noisy image smoothed | Histogram | Otsu with smoothing

Otsu's Method (cont.)

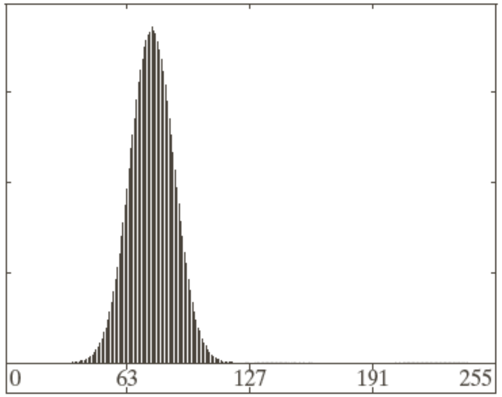
- Dealing with small structures
  - The histogram is unbalanced
    - The background dominates
    - No valleys indicating the small structure.
  - Idea: consider only the pixels around edges
    - Both structure of interest and background are present equally.
    - More balanced histograms
  - Use gradient magnitude or zero crossings of the Laplacian.
    - Threshold it at a percentage of the maximum value.
    - Use it as a mask on the original image.
    - Only pixels around edges will be employed.

Otsu's Method (cont.)

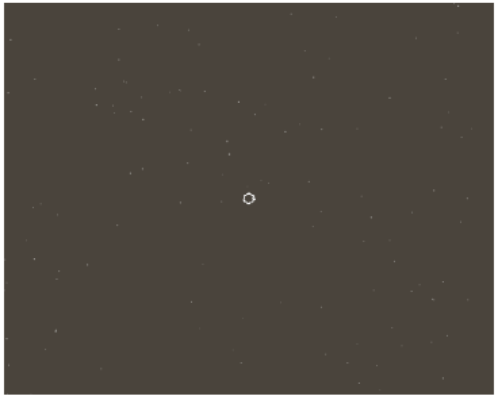
Noisy image, σ=10

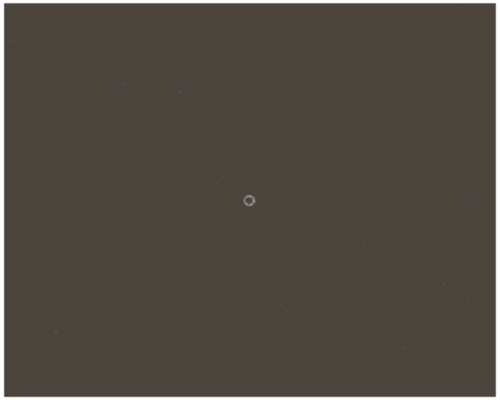


Histogram

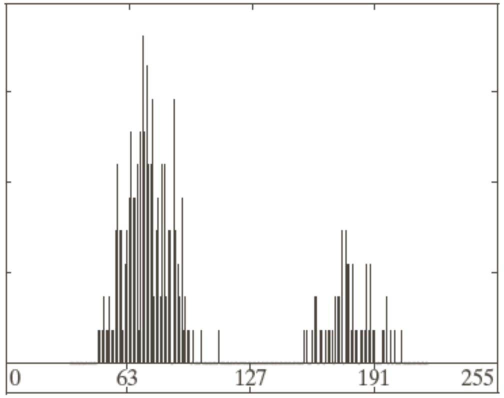


Gradient magnitude mask (at 99.7% of max)

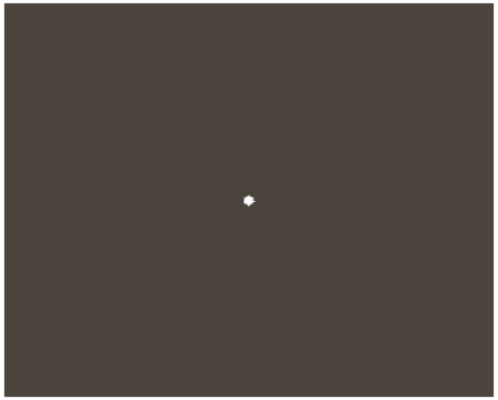




Multiplication (Mask x Image)



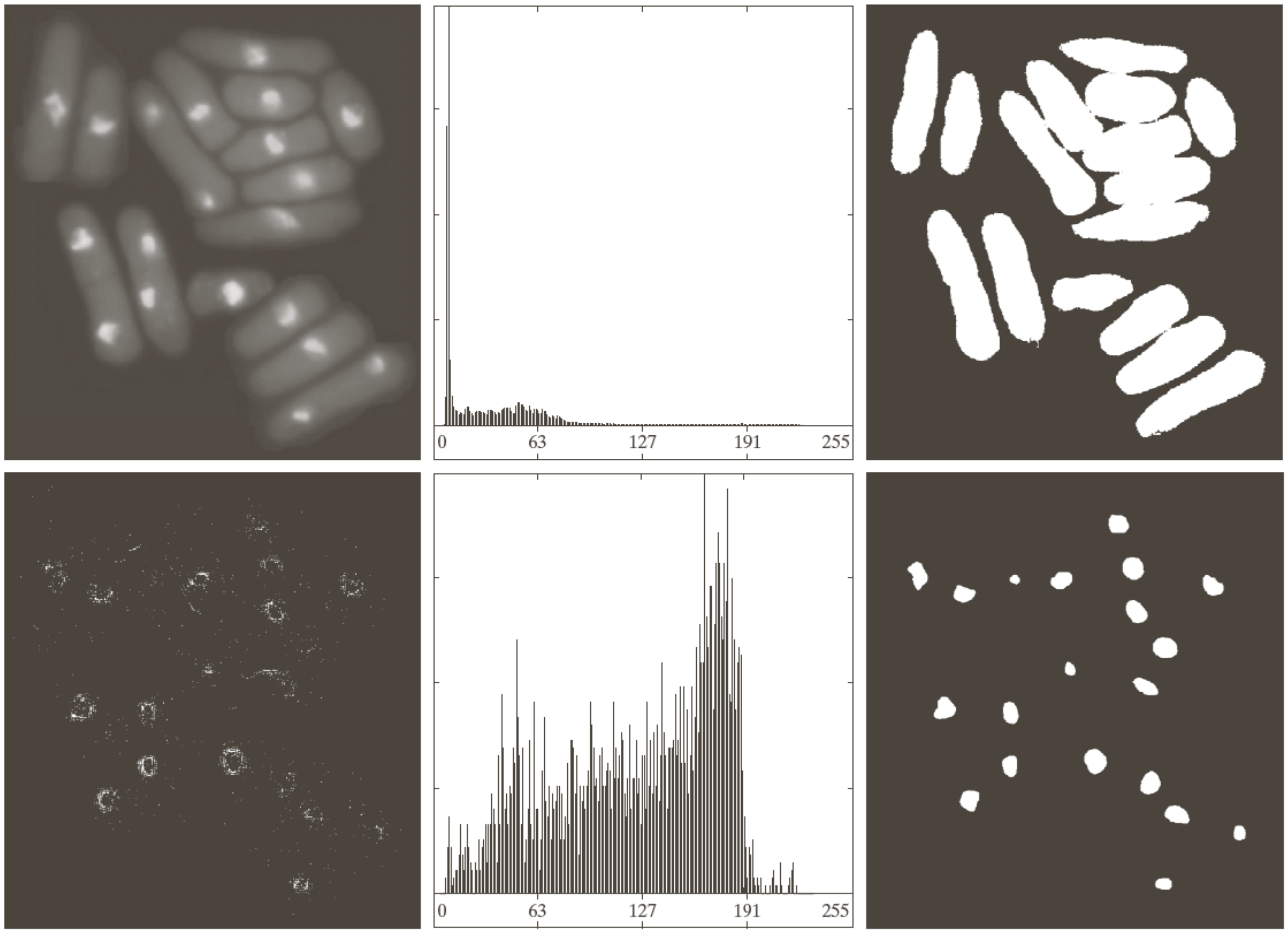
Histogram



Otsu

Otsu's Method (cont.)

Image | Histogram | Otsu on original histogram



Thresholded absolute Laplacian | Histogram | Otsu

We wish to extract the bright spots (nuclei) of the cells

Otsu's Method (cont.)

- 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:

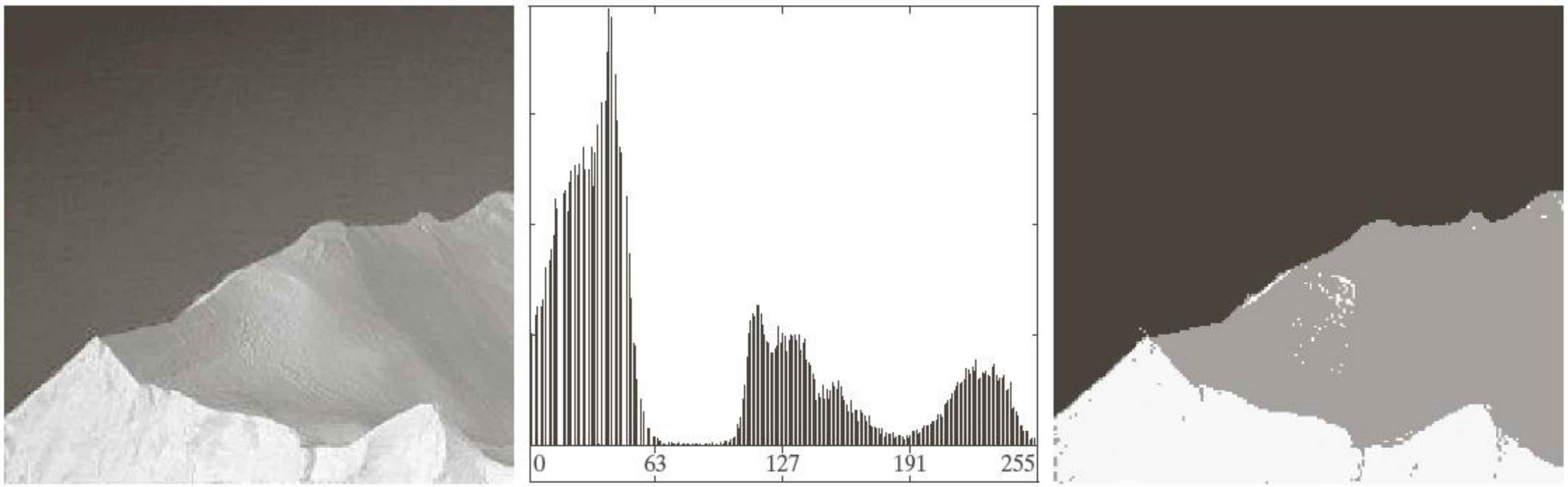
$$\sigma_B^2(k_1,k_2) = P_1(k_1)[m_1(k_1) - m_G] + P_2(k_1,k_2)[m_2(k_1,k_2) - m_G] + P_3(k_2)[m_3(k_2) - m_G]$$

- 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 Method (cont.)

Image of iceberg segmented into three regions.



k_1=80, k_2=177

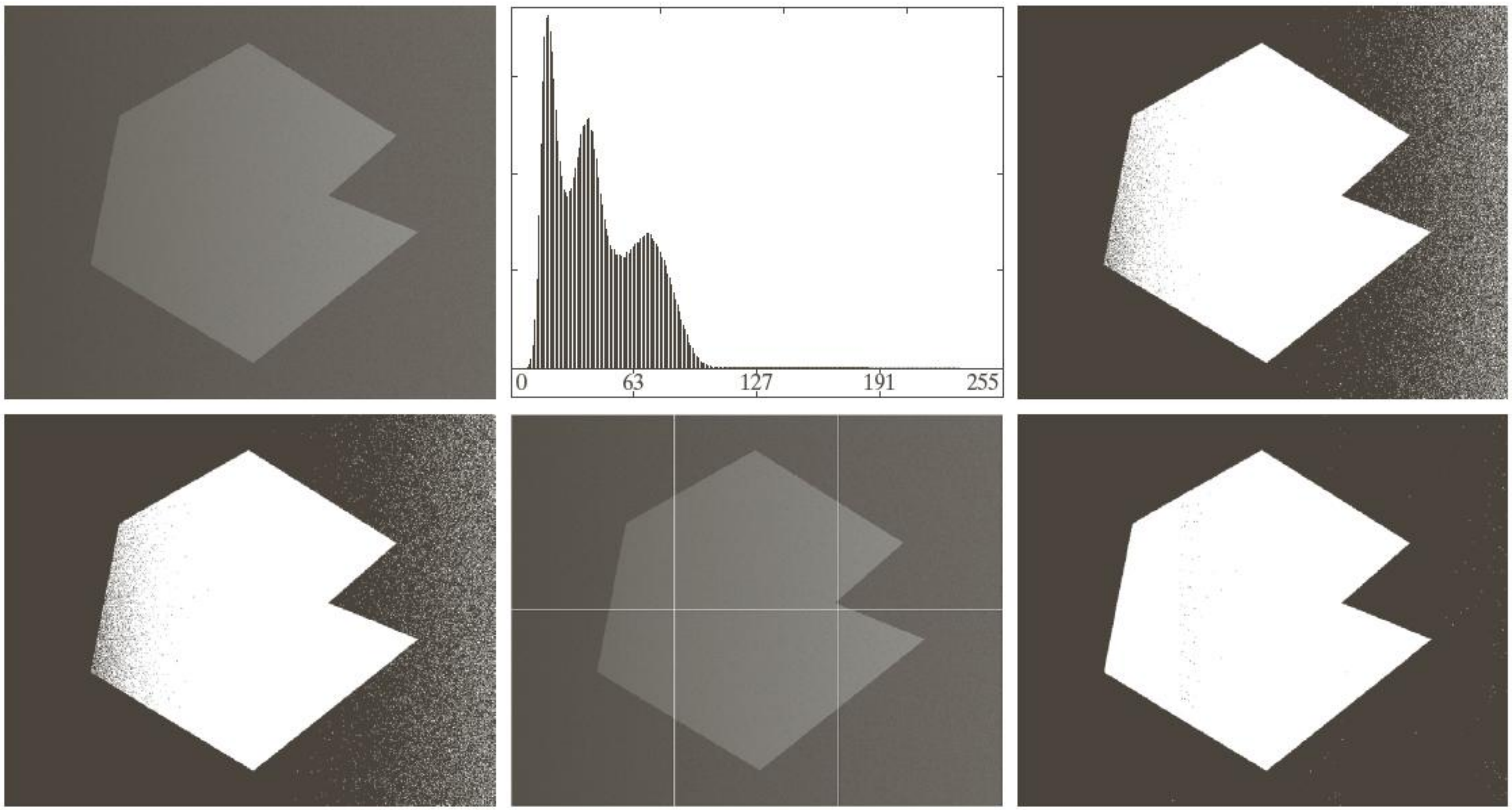
Variable Thresholding

Image partitioning.

The image is sub-divided and the method is applied to every sub-image.

Useful for illumination non-uniformity correction.

Shaded image | Histogram | Simple thresholding



Otsu | Subdivision | Otsu at each sub-image

Variable Thresholding(cont.)

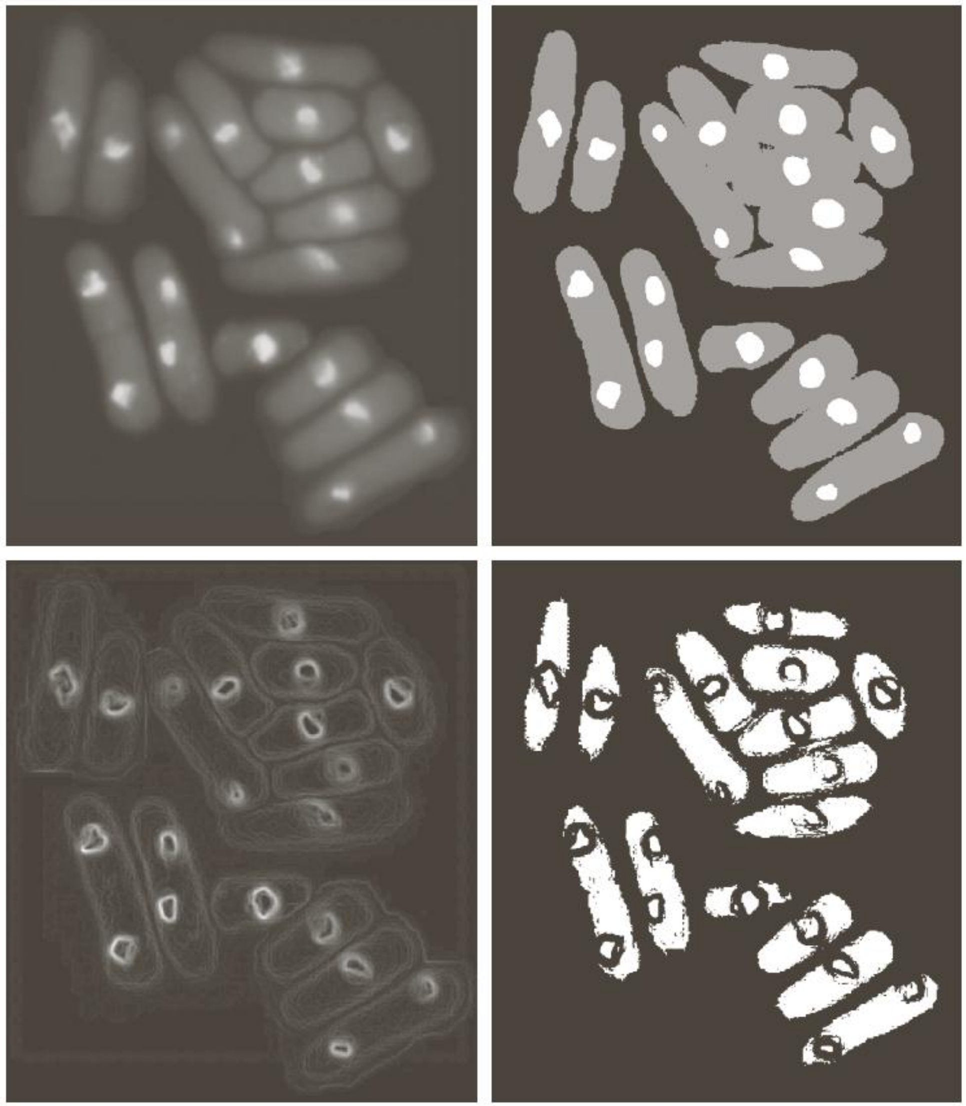
Use of local image properties.

- Compute a threshold for every single pixel in the image based on its neighborhood (m_{xy}, σ_{xy},…).

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

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

Variable Thresholding (cont.)



Image

Otsu with
2 thresholds

Image of local
standard deviations

Local
Thresholding

More accurate
nuclei extraction.

Variable Thresholding (cont.)

Moving averages.

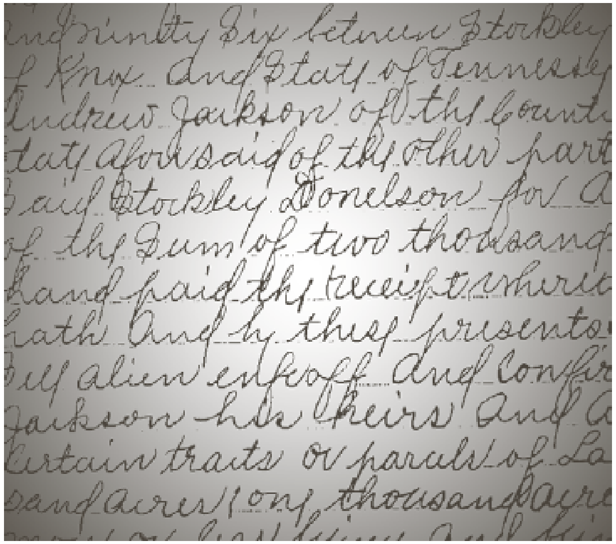
- Generally used along lines, columns or in zigzag .
- Useful in document image processing.
- Let z_{k+1} be the intensity of a pixel in the scanning sequence at step $k+1$. The moving average (mean intensity) at this point is:

$$m(k+1) = \frac{1}{n} \sum_{i=k+2-n}^{k+1} z_i = m(k) + \frac{1}{n}(z_{k+1} - z_{k-n})$$

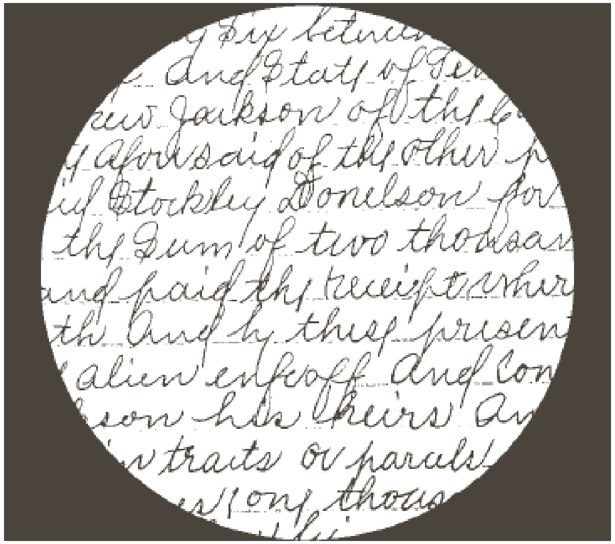
n is the number of points used in the average

- Segmentation is then performed at each point comparing the pixel value to a fraction of the moving average.

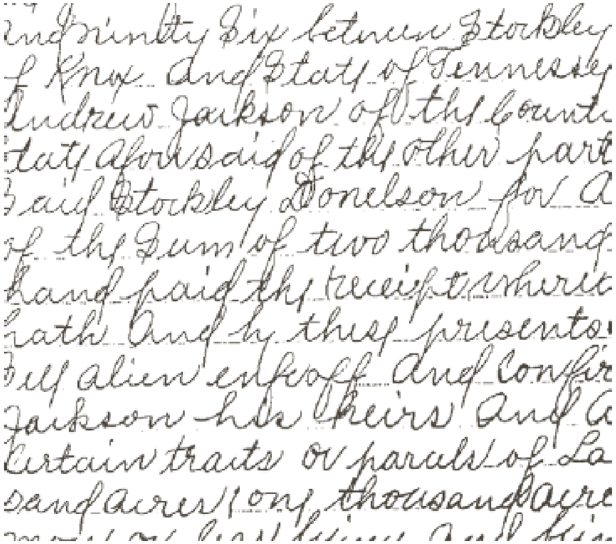
Variable Thresholding (cont.)



Shaded text images occur typically from photographic flashes

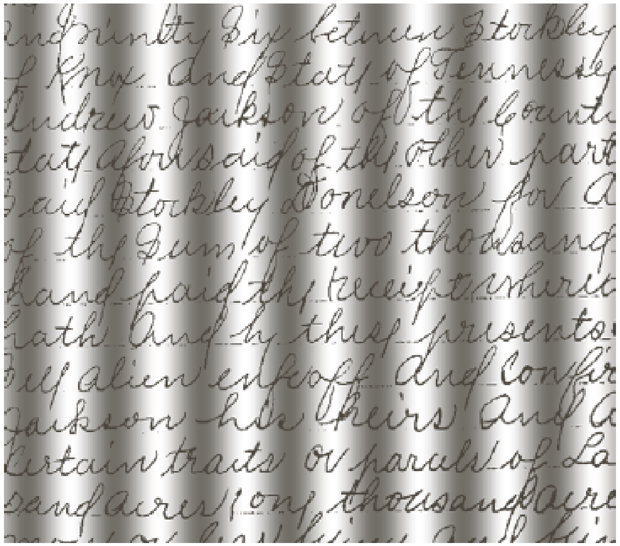


Otsu



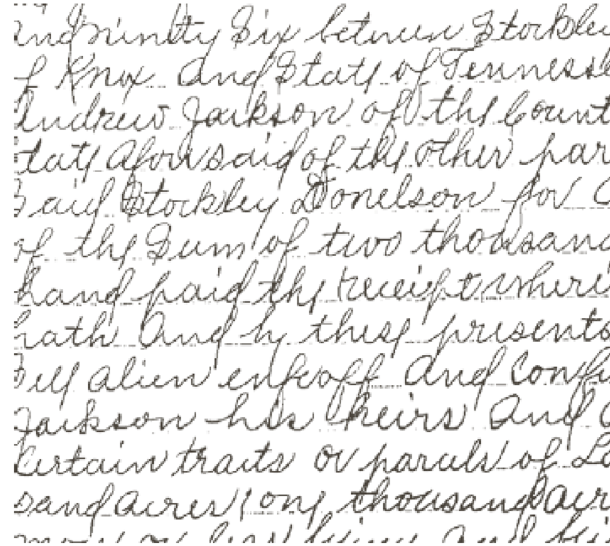
Moving averages

Variable Thresholding (cont.)



Sinusoidal variations (the power supply of the scanner not grounded properly)

Otsu



Moving averages

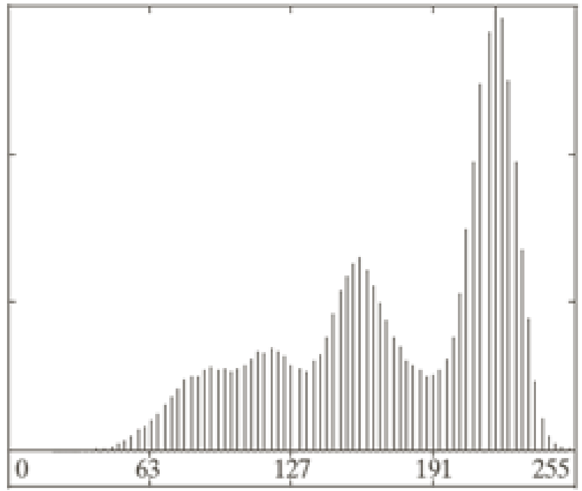
The moving average works well when the structure of interest is small with respect to image size (handwritten text).

Region Growing

- Start with seed points $S(x,y)$ and grow to larger regions satisfying a predicate.
- Needs a stopping rule.
- **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.
  - Form image $g(x,y)=1$ for all pixels in $f_q(x,y)$ that are 8-connected with to any seed point in $S(x,y)$.
  - Label each connected component in $g(x,y)$ with a different label.

Region Growing (cont.)

X-Ray image of weld with a crack we want to segment.



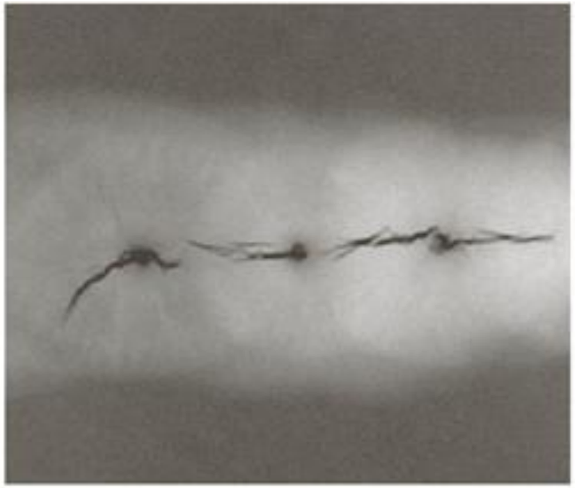
Histogram

Seed image (99.9% of max value in the initial image).
Crack pixels are missing.

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.

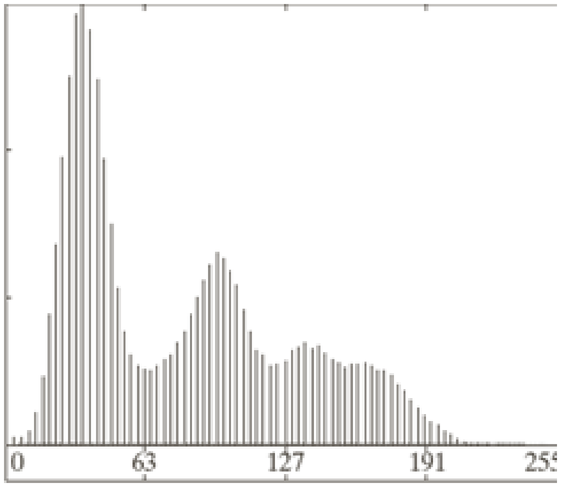
Region Growing (cont.)

Seed image eroded to 1 pixel regions.



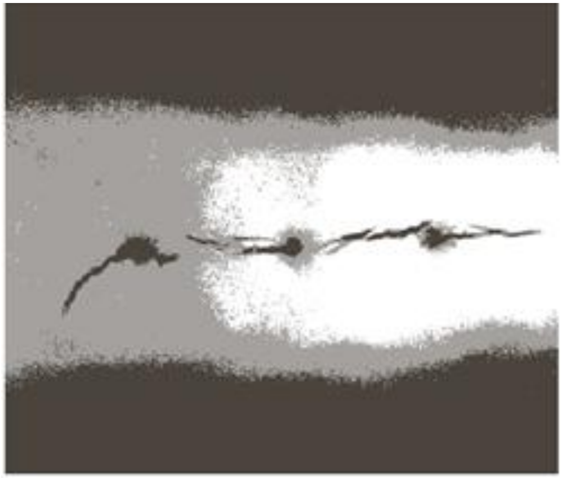
Difference between the original image and the initial seed image.

The pixels are ready to be compared to the threshold.



Histogram of the difference image. Two valleys at 68 and 126 provided by Otsu.

Region Growing (cont.)



Otsu thresholding of the difference image to 3 regions (2 thresholds).

Thresholding of the difference image with the lowest of the dual thresholds.

Notice that the background is also considered as crack.

Segmentation result by region growing.

The background is not considered as crack.
It is removed as it is not 8-connected to the seed pixels.

Region Splitting and Merging

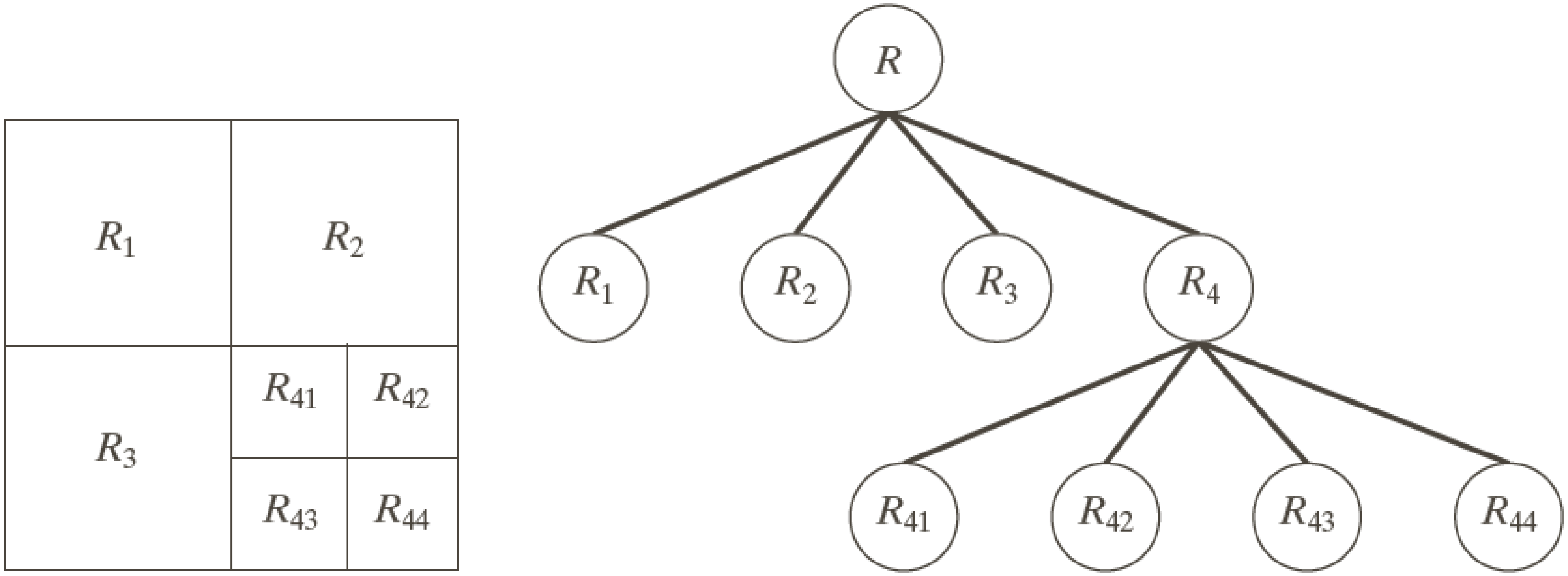
- Based on quadtrees (*quadimages*).
- The root of the tree corresponds to the image.
- Split the image to sub-images that do not satisfy a predicate Q.
- If only splitting was used, the final partition would contain adjacent regions with identical properties.
- A merging step follows merging regions satisfying the predicate Q.

Region Splitting and Merging (cont.)

- **Algorithm**
  - Split into four disjoint quadrants any region R_i for which $Q(R_i)=\mathrm{FALSE}$.
  - When no further splitting is possible, merge any adjacent regions R_i and R_k for which $Q(R_i \cup R_k)=\mathrm{TRUE}$.
  - Stop when no further merging is possible.
- A maximum quadregion size is specified beyond which no further splitting is carried out.
- Many variations heve been proposed.
  - Merge any adjacent regions if each one satisfies the predicate individually (even if their union does not satisfy it).

Region Splitting and Merging (cont.)

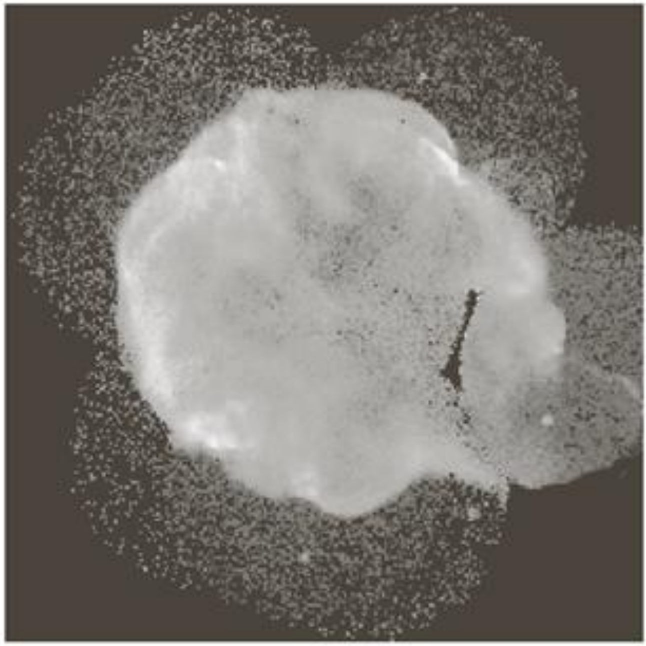
Quadregions resulted from splitting.



Merging examples:

- R_2 may be merged with R_{41}.
- R_{41} may be merged with R_{42}.

Region Splitting and Merging (cont.)



- Image of the Cygnus Loop. We want to segment the outer ring of less dense matter.
- Characteristics of the region of interest:
  - Standard deviation greater 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:

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

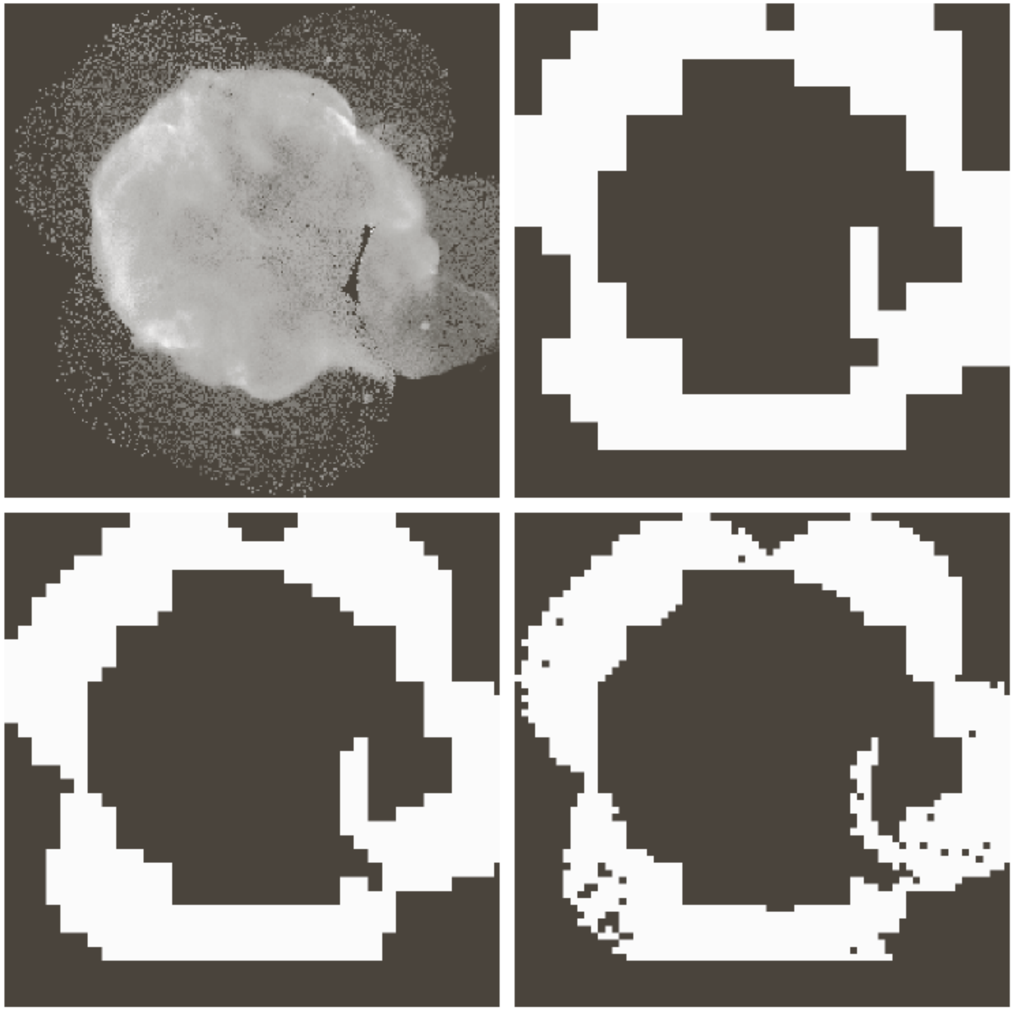
Region Splitting and Merging (cont.)

Varying the size of the smallest allowed *quadregion*.

Larger *quadregions* lead to block-like segmentation.

Smaller *quadregions* lead to small black regions.

16x16 seems to be the best result.



32x32

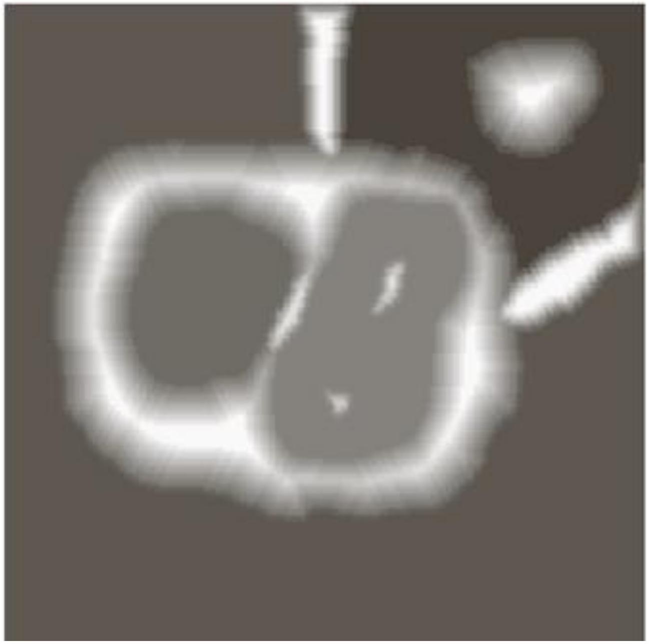
16x16

8x8

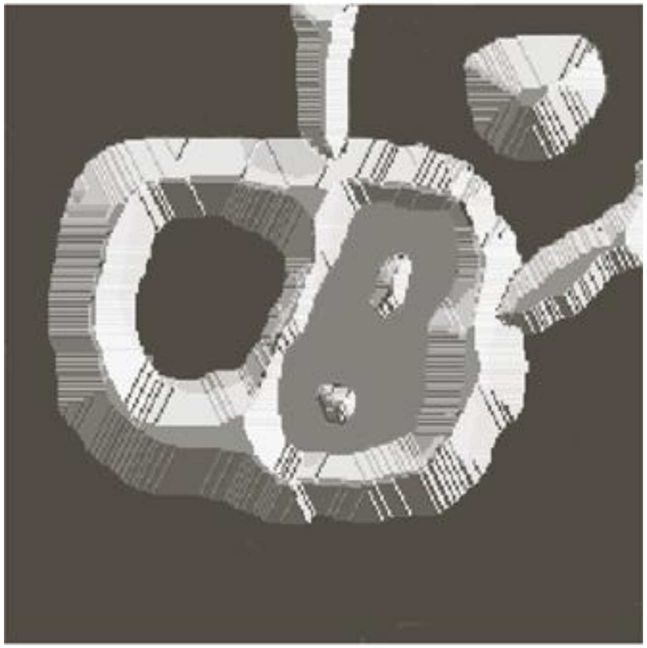
Morphological Watersheds

- Visualize an image topographically in 3D
  - The two spatial coordinates and the intensity (relief representation).
- Three types of points
  - Points belonging to a regional minimum.
  - Points at which a drop of water would fall certainly to a regional minimum (*catchment basin*).
  - Points at which the water would be equally likely to fall to more than one regional minimum (crest lines or *watershed lines*).
- Objective: find the watershed lines.

Morphological Watersheds (cont.)

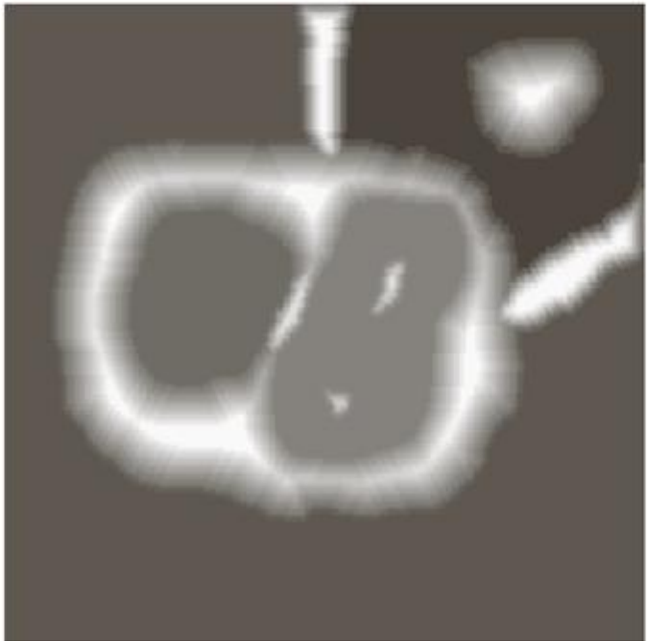


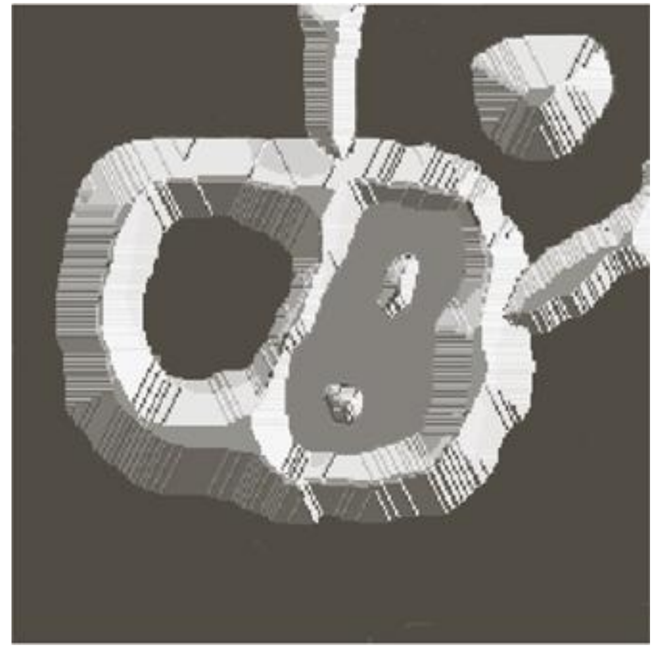
Image



- Topographic representation.
- The height is proportional to the image intensity.
- Backsides of structures are shaded for better visualization.

Morphological Watersheds (cont.)

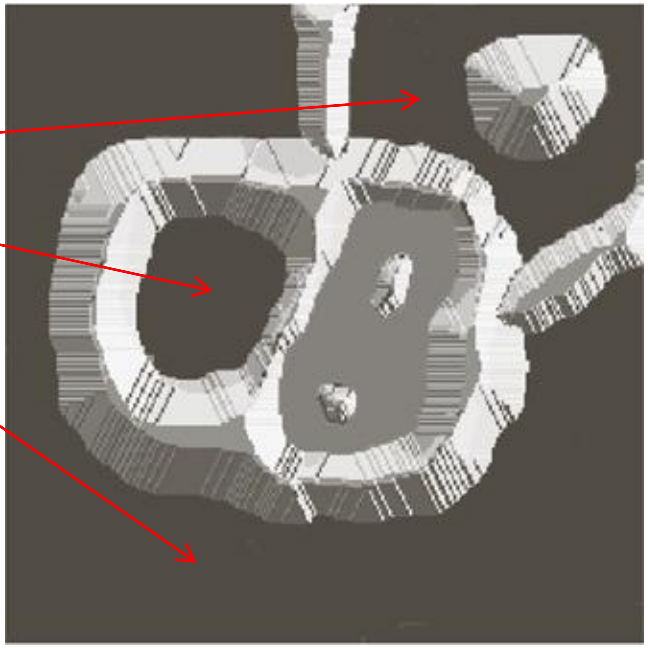




- A hole is punched in each regional minimum and the topography is flooded by water from below through the holes.
- When the rising water is about to merge in catchment basins, a dam is built to prevent merging.
- There will be a stage where only the tops of the dams will be visible.
- These continuous and connected boundaries are the result of the segmentation.

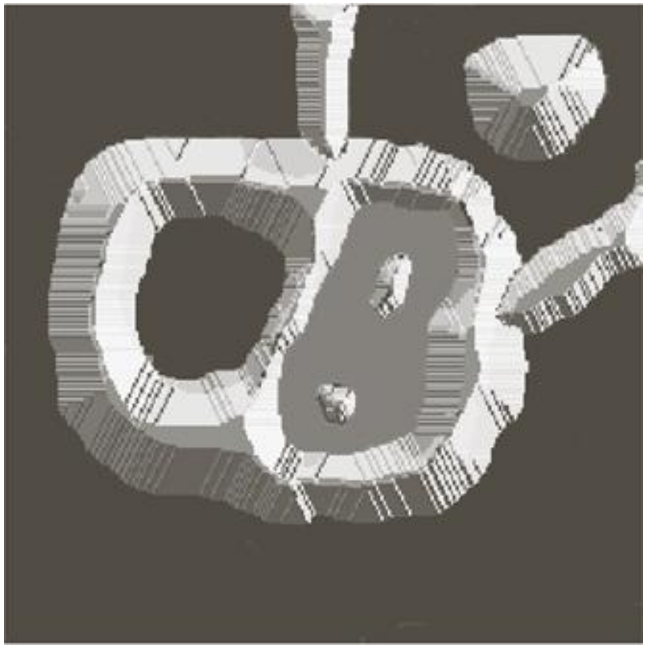
Morphological Watersheds (cont.)

Regional
minima



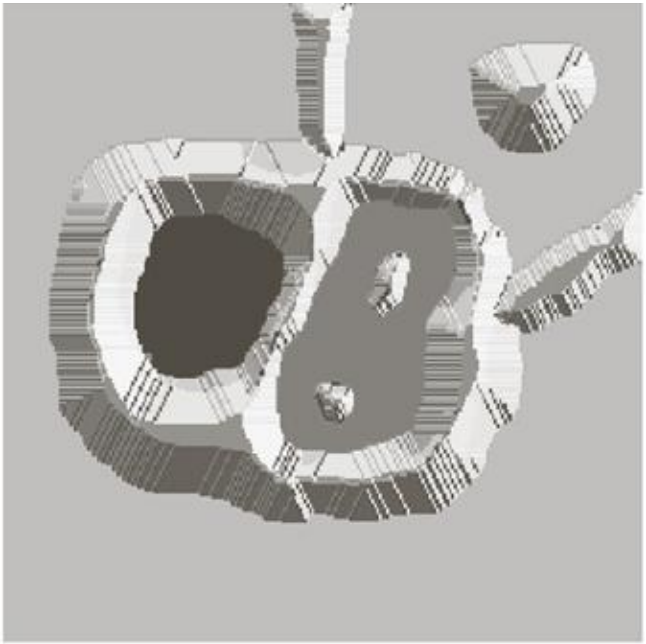
- Topographic representation of the image.
- A hole is punched in each regional minimum (dark areas) and the topography is flooded by water (at equal rate) from below through the holes.

Morphological Watersheds (cont.)



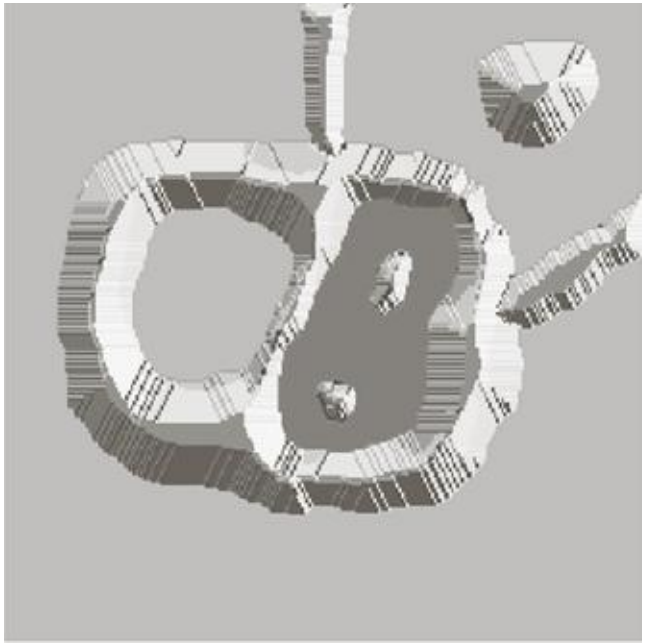
- Before flooding.
- To prevent water from spilling through the image borders, we consider that the image is surrounded by dams of height greater than the maximum image intensity.

Morphological Watersheds (cont.)



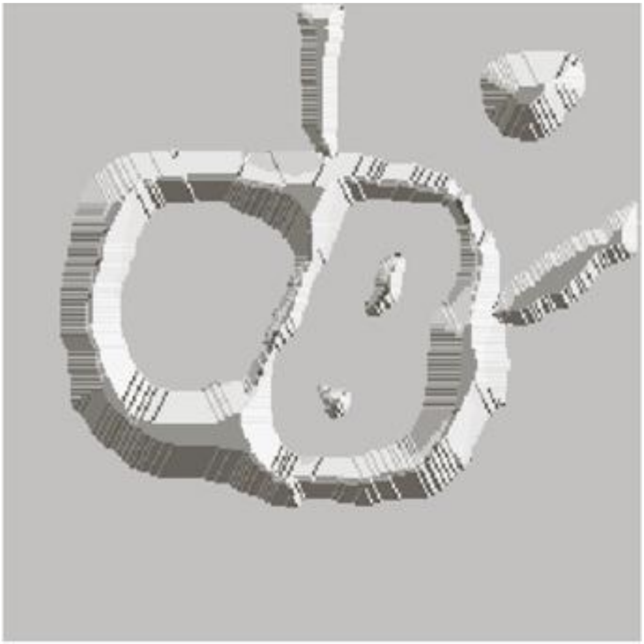
- First stage of flooding.
- The water covered areas corresponding to the dark background.

Morphological Watersheds (cont.)



- Next stages of flooding.
- The water has risen into the other catchment basin.

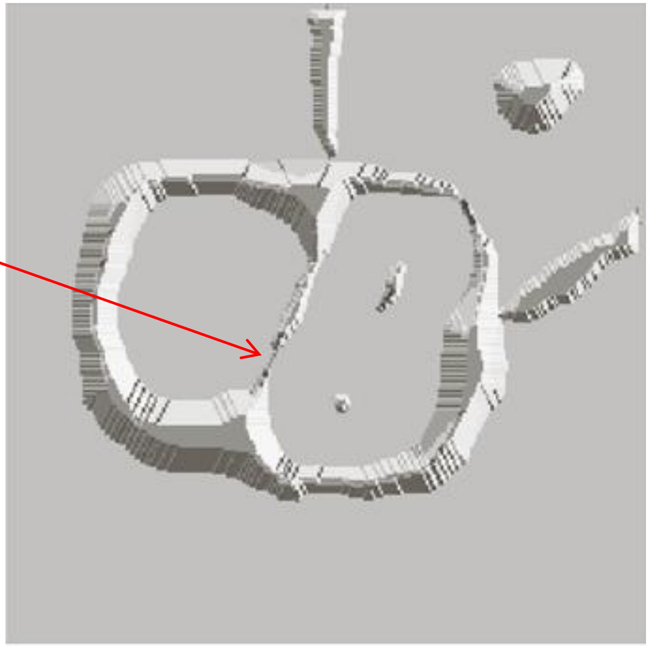
Morphological Watersheds (cont.)



- Further flooding. The water has risen into the third catchment basin.

Morphological Watersheds (cont.)

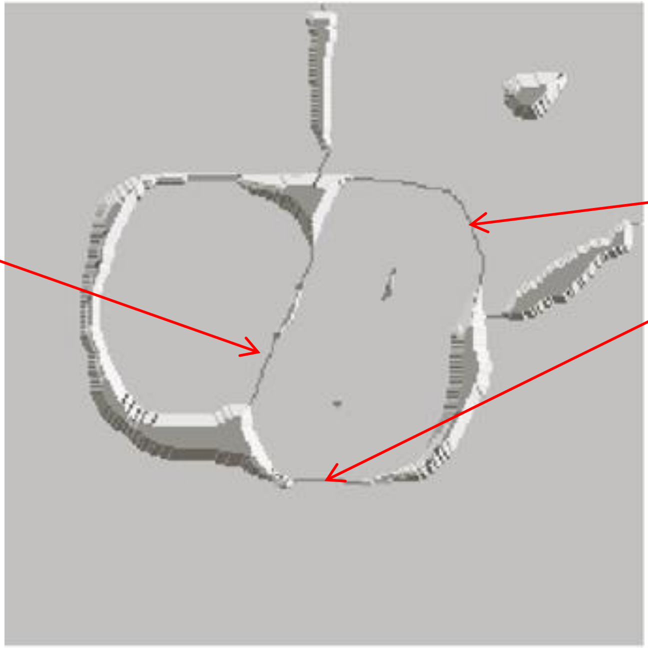
Short dam



- Further flooding.
- The water from the left basin overflowed into the right basin.
- A short dam is constructed to prevent water from merging.

Morphological Watersheds (cont.)

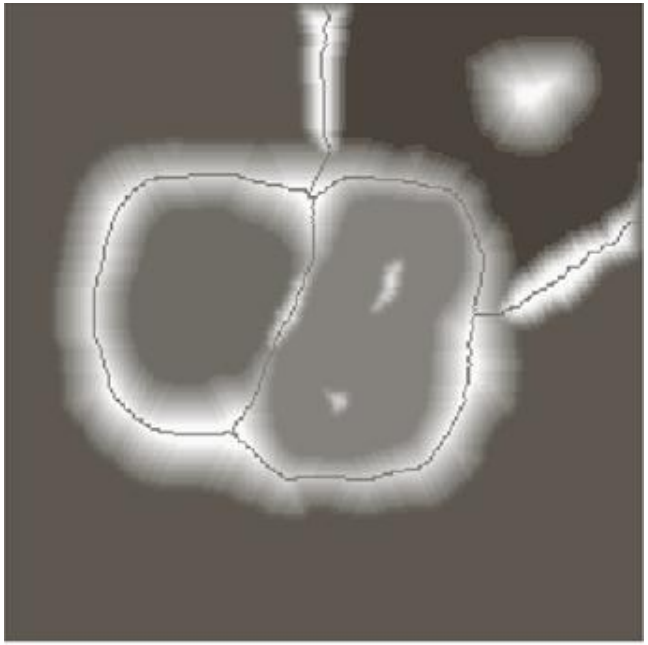
Longer dam



New dams

- Further flooding.
- The effect is more pronounced.
- The first dam is now longer.
- New dams are created.

Morphological Watersheds (cont.)

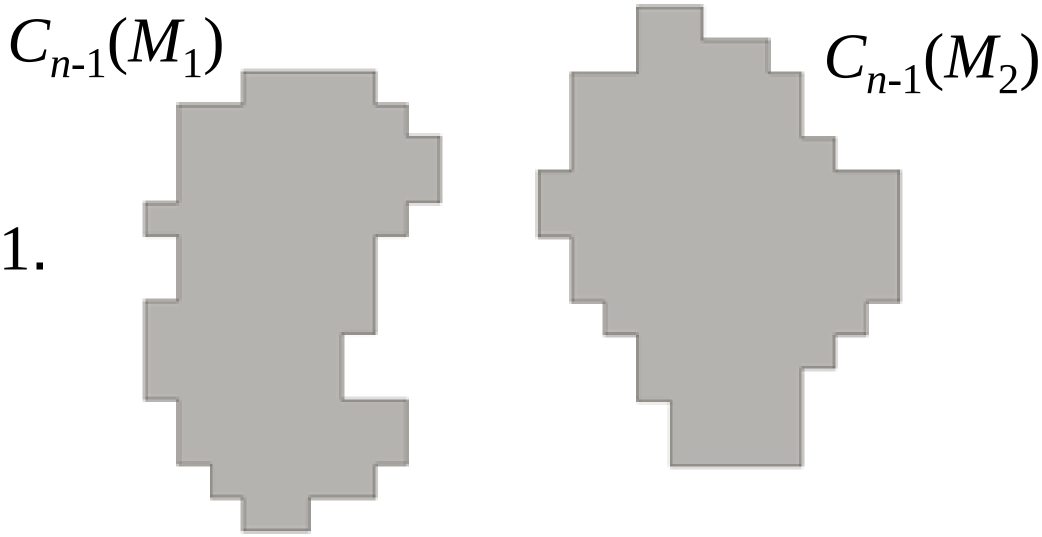


Final watershed lines superimposed on the image.

- The process continues until the maximum level of flooding is reached.
- The final dams correspond to the watershed lines which is the result of the segmentation.
- Important: continuous segment boundaries.

Morphological Watersheds (cont.)

- Dams are constructed by morphological dilation.



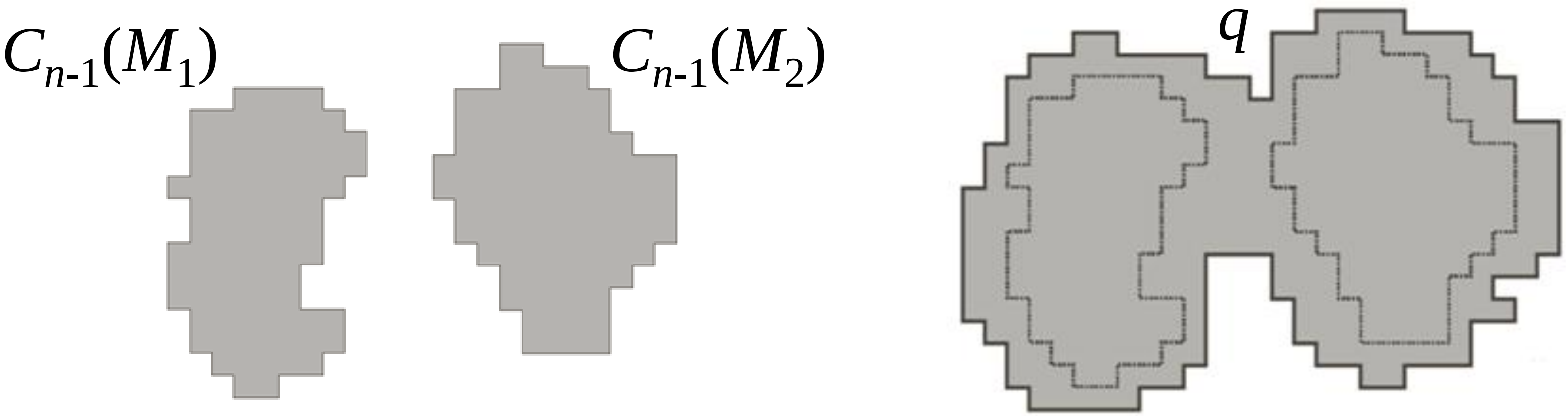
Flooding step n-1.

Regional minima: M_1 and M_2.
Catchment basins associated: $C_{n\text{-}1}(M_1)$ and $C_{n\text{-}1}(M_2)$.

$$C[n-1] = C_{n-1}(M_1) \cup C_{n-1}(M_2)$$

$C[n\text{-}1]$ has two connected components.

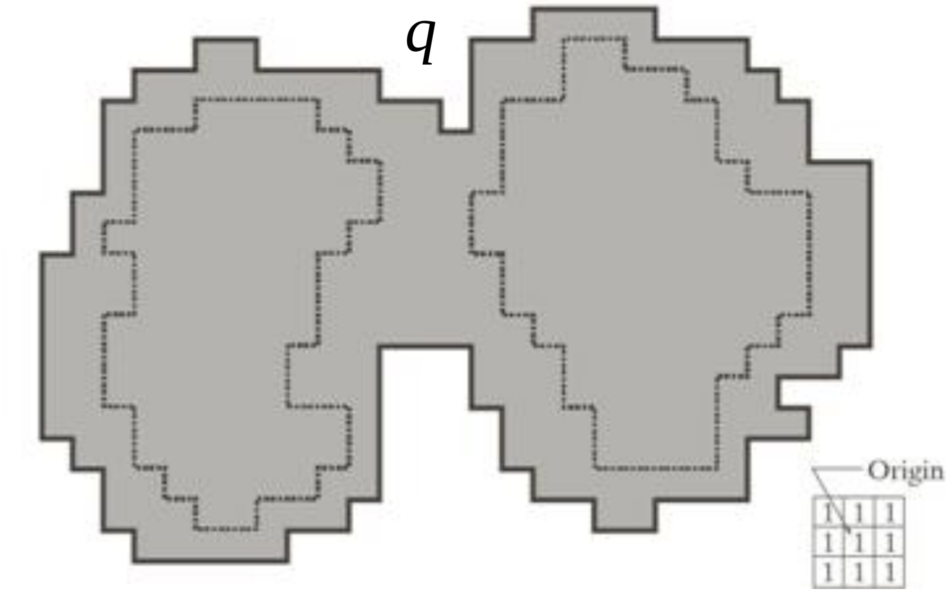
Morphological Watersheds (cont.)



Flooding step n-1.   Flooding step n.

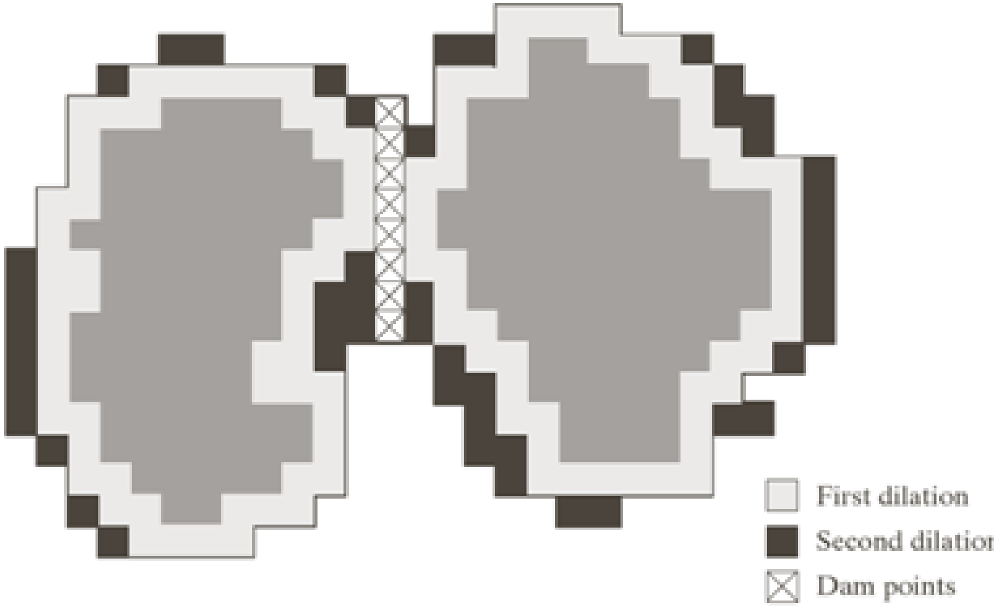
- If we continue flooding, then we will have one connected component.
- This indicates that a dam must be constructed.
- Let q be the merged connected component if we perform flooding a step n.

Morphological Watersheds (cont.)



- 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.

Morphological Watersheds (cont.)

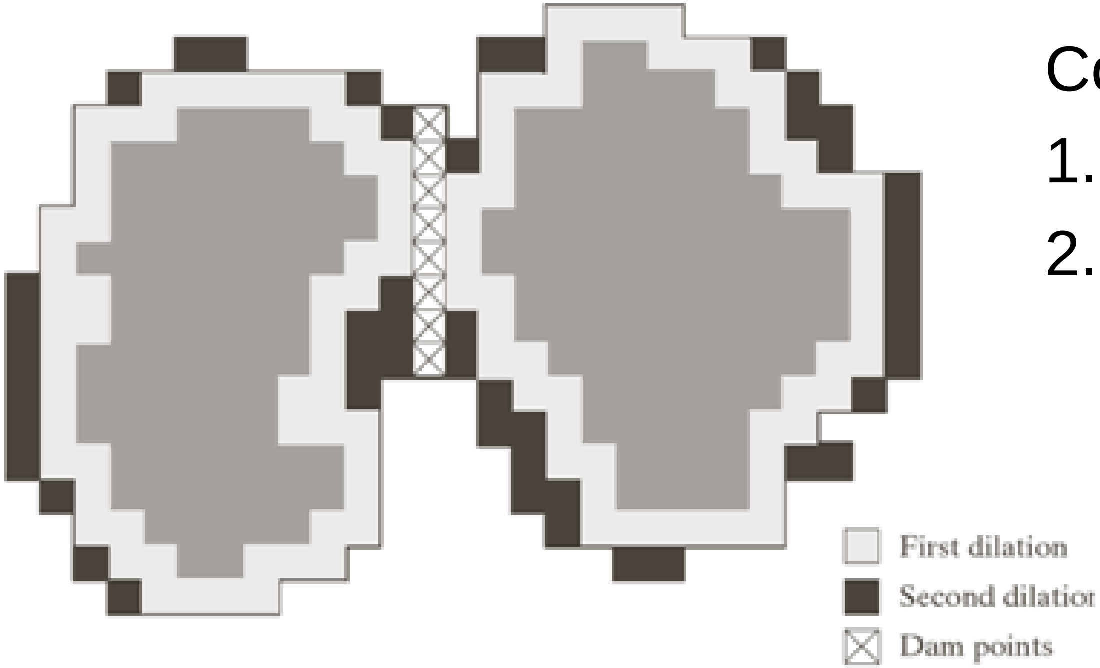


Conditions

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).

Morphological Watersheds (cont.)



Conditions

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

- The only points in q that satisfied both conditions form the 1-pixel thick path.
- This is the dam at step n of the flooding process.
- The points should satisfy both conditions.

Morphological Watersheds (cont.)

- 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.

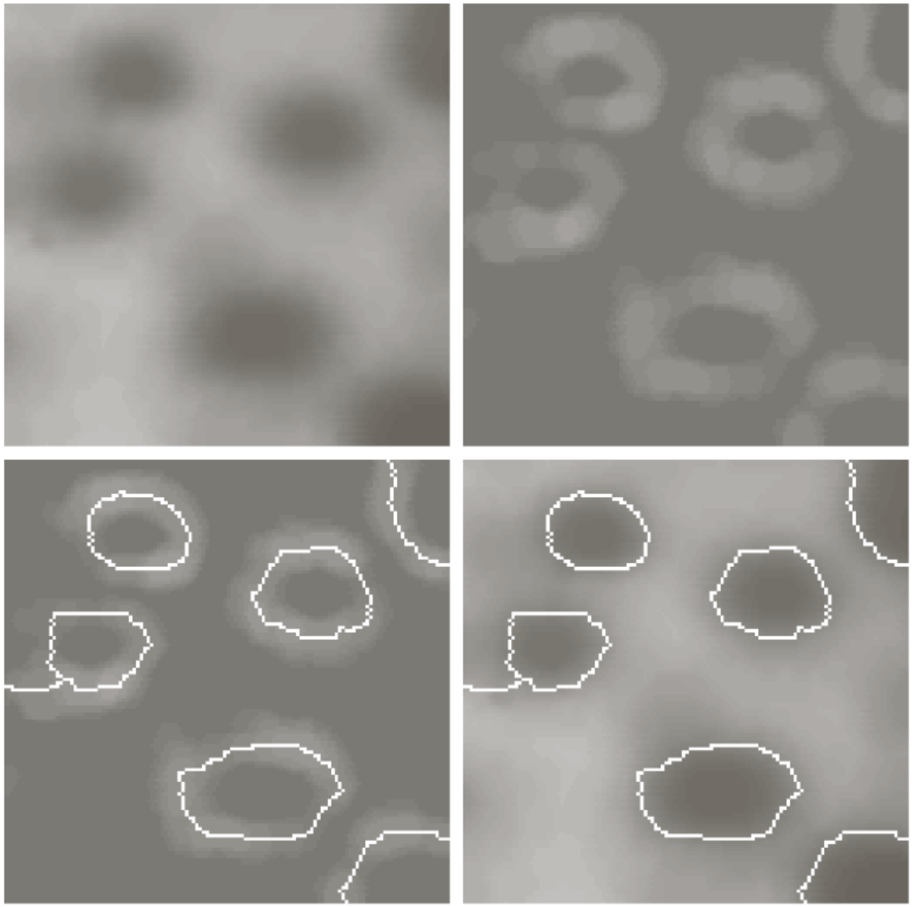
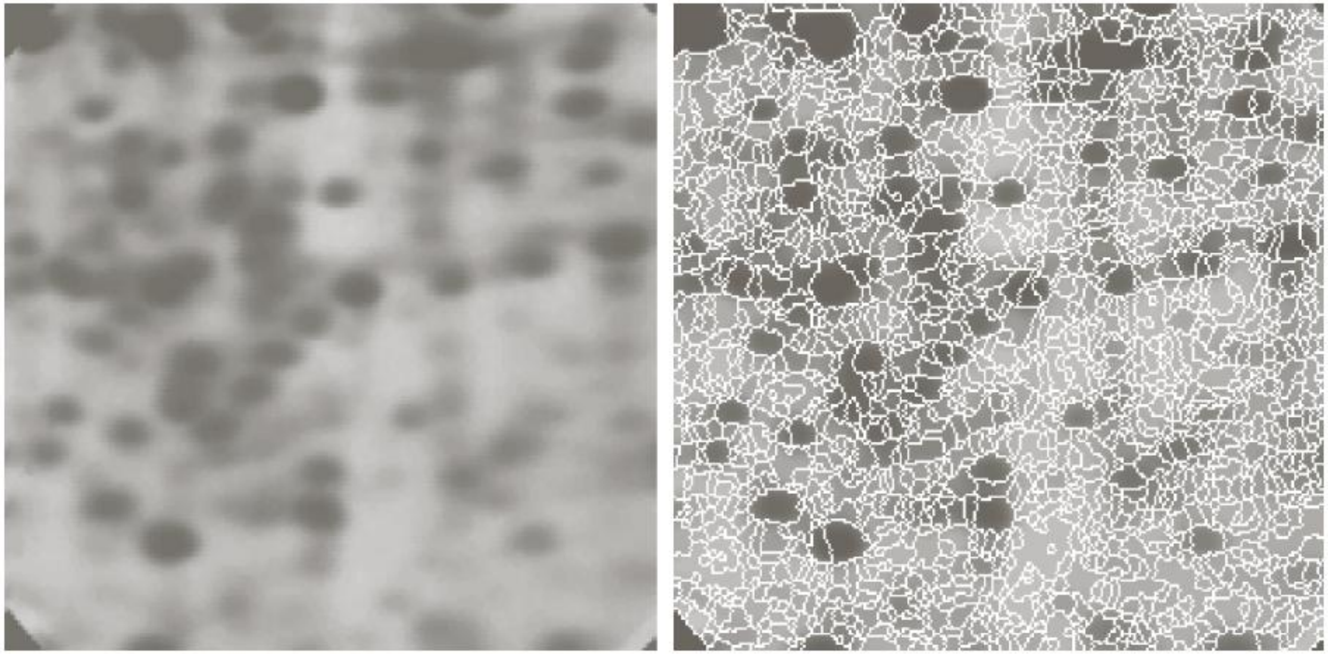


Image — Gradient magnitude

Watersheds — Watersheds on the image

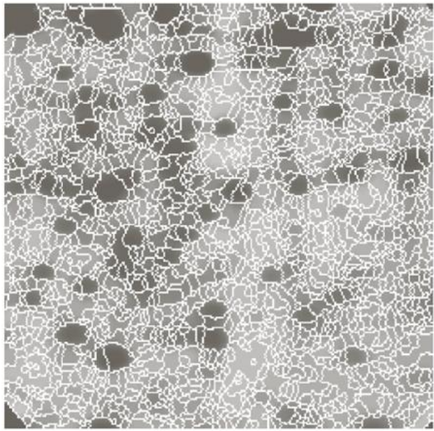
Morphological Watersheds (cont.)



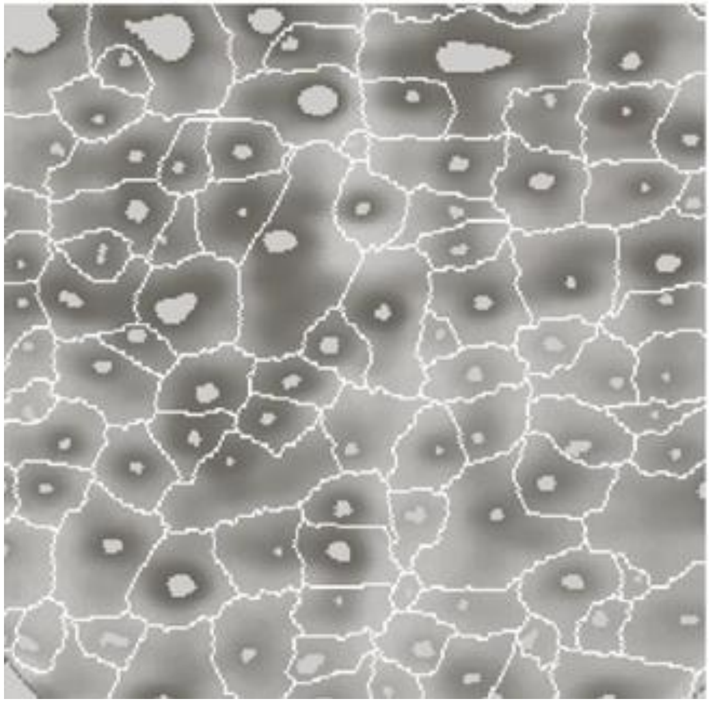
- Noise and local minima lead generally to *oversegmentation*.
- The result is not useful.
- Solution: limit the number of allowable regions by additional knowledge.

Morphological Watersheds (cont.)

- **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.

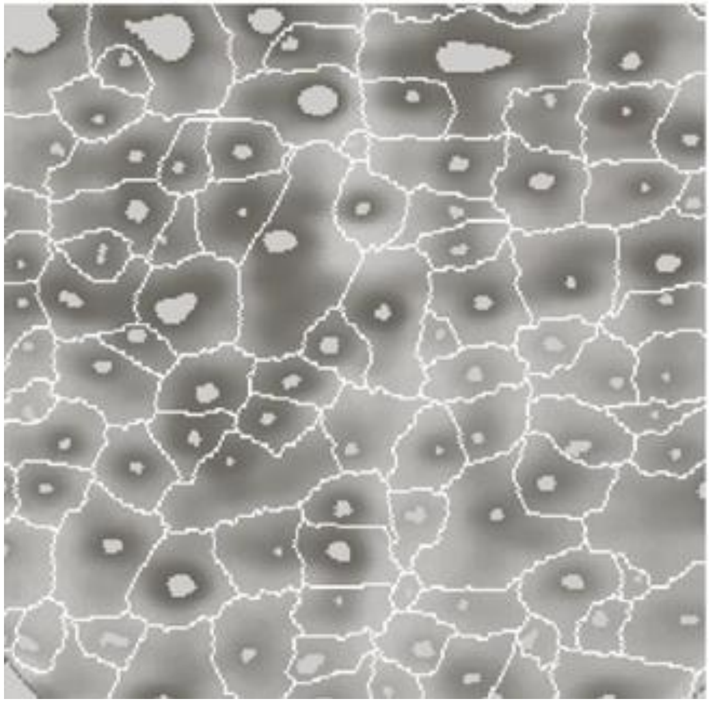


Morphological Watersheds (cont.)



- After smoothing, the internal markers are shown in light gray.
- The watershed algorithm is applied and the internal markers are the only allowable regional minima.
- The resulting watersheds are the external markers (shown in white).

Morphological Watersheds (cont.)



- Each region defined by the external marker has a single internal marker and part of the background.
- The problem is to segment each of these regions into two segments: a single object and background.
- The algorithms we saw in this lecture may be used (including watersheds applied to each individual region).

Morphological Watersheds (cont.)

Final segmentation.

Morphological Watersheds (cont.)

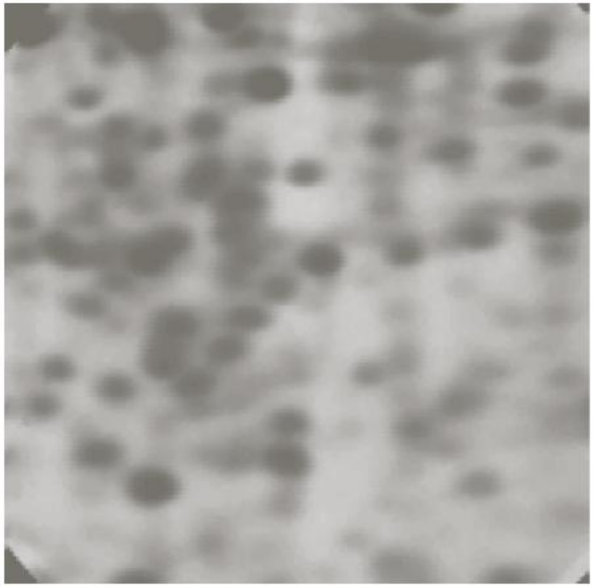
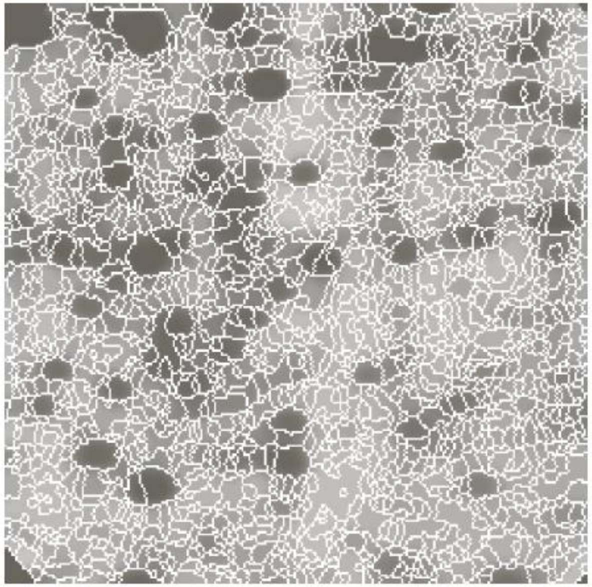
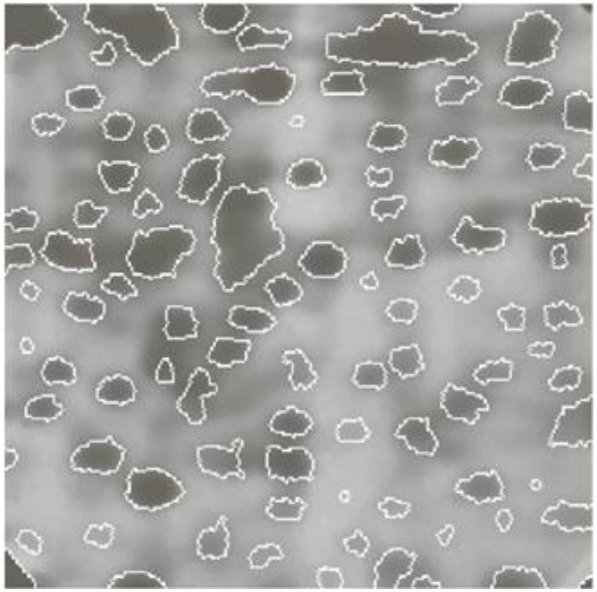
		

Image	Watersheds	Watersheds with markers

https://blog.csdn.net/fengye2two/article/details/79116105

Morphological Watersheds (cont.)

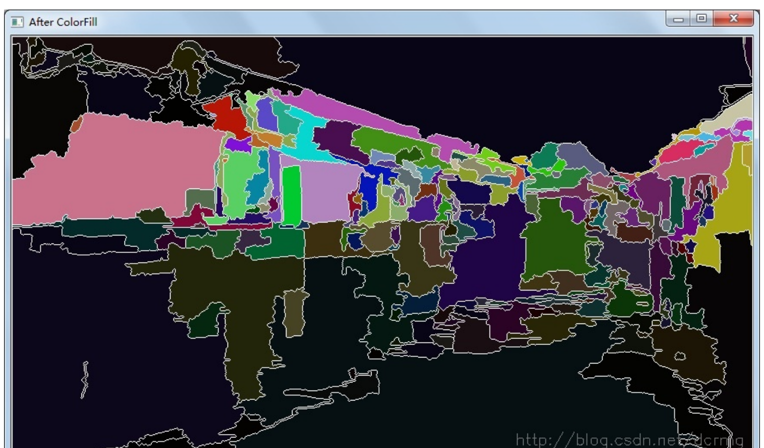
Image

Watersheds

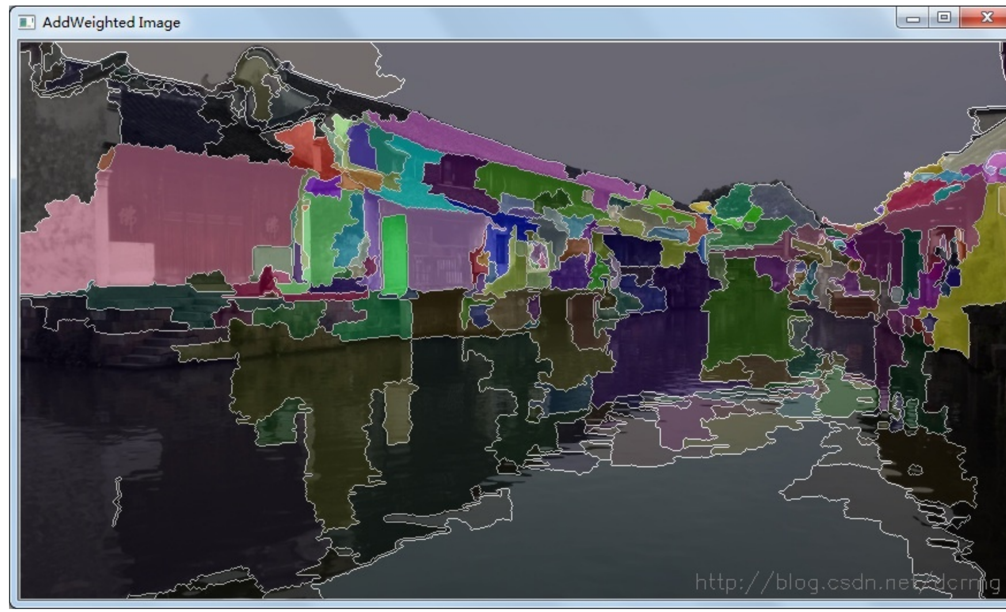
Watersheds segmentation merged with original image

Morphological Watersheds (cont.)

Image



Watersheds



Watersheds segmentation merged with original image

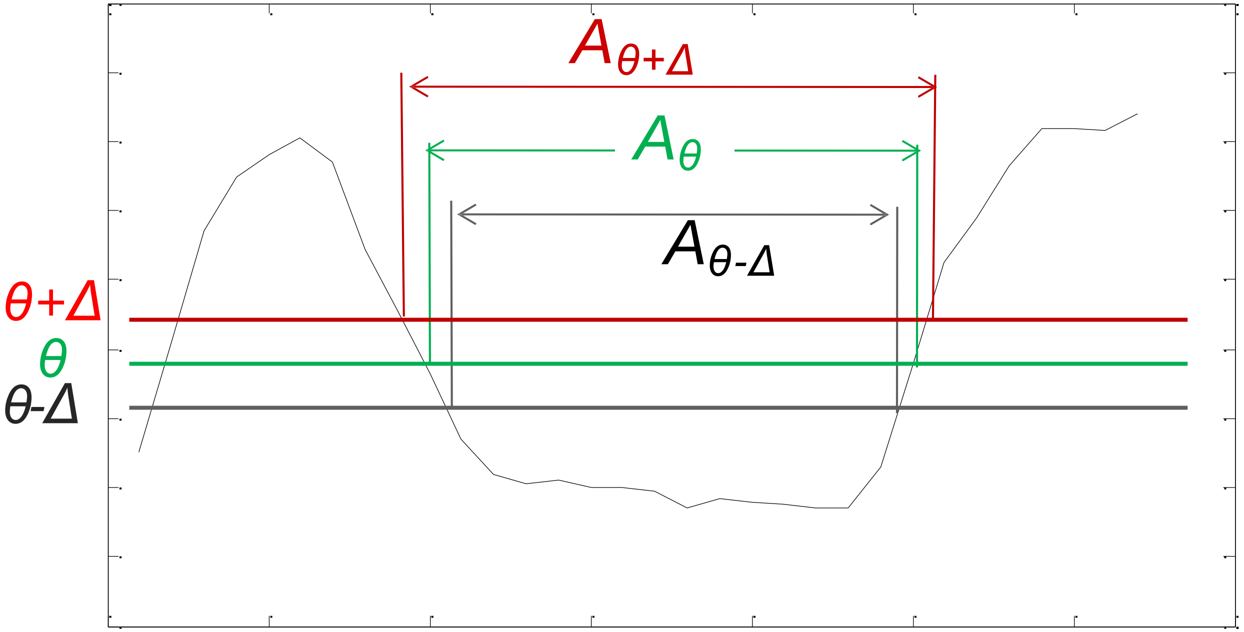
Maximally stable extremal regions

- Extremal region: any connected region in an image with all pixel values above (or below) a threshold
- Observations:
  - Nested extremal regions result when the threshold is successively raised (or lowered).
  - The nested extremal regions form a "component tree."
- Key idea: choose thresholds θ such that the resulting bright (or dark) extremal regions are nearly constant when these thresholds are perturbed by +/-Δ

→ ***"maximally stable" extremal regions (MSER)***

[Matas, Chum, Urba, Pajdla, 2002]

MSERs: illustration



$$\text{Local minimum of } \left| \frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_{\theta}} \right| \rightarrow \text{ MSER}$$

[Matas, Chum, Urba, Pajdla, 2002]

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

Image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 8$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

Image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 7$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

Image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 6$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

Image

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	1	1	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 5$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$$f[x,y]$$

Image

0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	0	0	0	0	1	0	0	1
0	1	1	0	0	0	1	1	1	1	0	1	0	0	1
0	1	1	0	0	0	1	1	1	1	0	1	0	0	0
0	1	1	0	0	0	1	1	1	1	0	1	1	1	1
0	0	0	0	0	0	1	1	1	1	0	1	1	1	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	1	1	1	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$f[x,y] > 4$$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

Image

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	1	1	1	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 3$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$$f[x,y]$$

Image

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$f[x,y] > 2$$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

Image

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	0	0	0	1	1	0	0	0	0	1	1	1	1
0	1	1	1	0	1	1	0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	0	1	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1	1	0	0	0	1
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$f[x,y] > 1$

Level Set

Level sets of an image

1	1	1	1	1	1	1	1	1	1	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	3	3	3	3	1	5	4	4	8
1	7	6	4	2	2	5	5	5	5	1	5	4	4	8
1	6	6	4	2	2	5	5	5	6	1	5	4	4	4
1	6	6	4	2	2	6	6	6	6	1	5	5	5	5
1	4	4	4	2	2	6	6	6	6	1	5	5	5	5
1	1	1	1	1	2	6	1	1	1	1	2	2	2	2
1	8	8	5	1	2	6	1	7	7	1	2	2	2	2
1	8	8	5	1	1	1	1	7	7	1	1	1	1	2
1	8	8	5	5	5	3	3	7	7	1	1	1	1	2
1	8	8	5	5	3	3	3	7	7	7	1	1	1	2
1	8	8	5	5	3	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$f[x,y]$

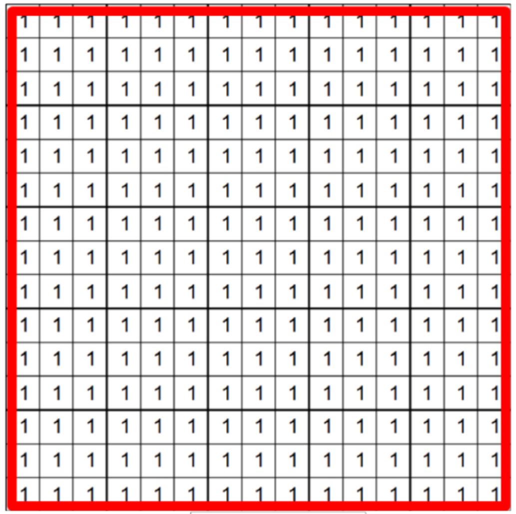
Image

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

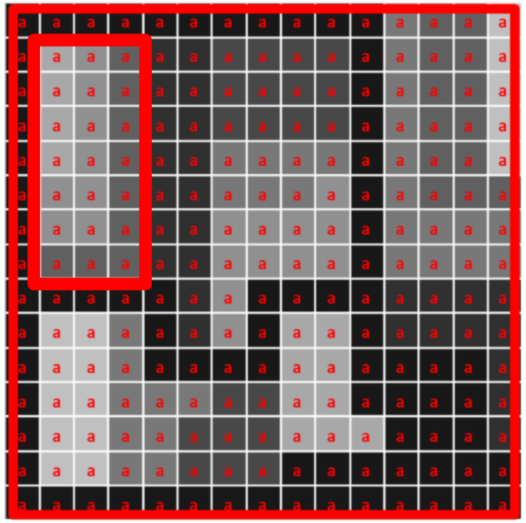
$f[x,y] > 0$

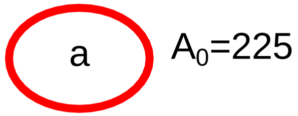
Level Set

Component tree of an image



$f[x,y] > 0$





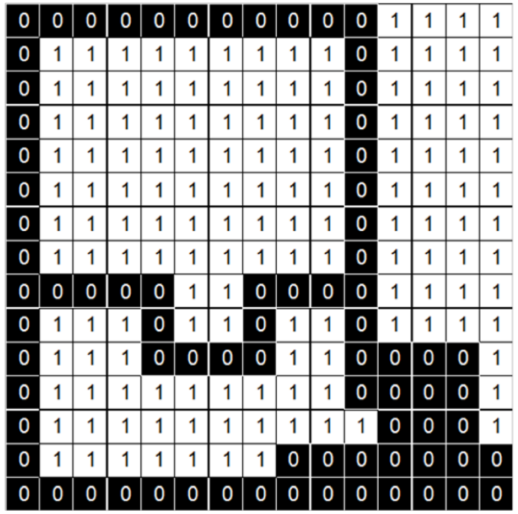
Local minimum of sequence

$$\left| \frac{A_{\theta-\Delta} - A_{\theta+\Delta}}{A_\theta} \right| \rightarrow \text{MSER}$$

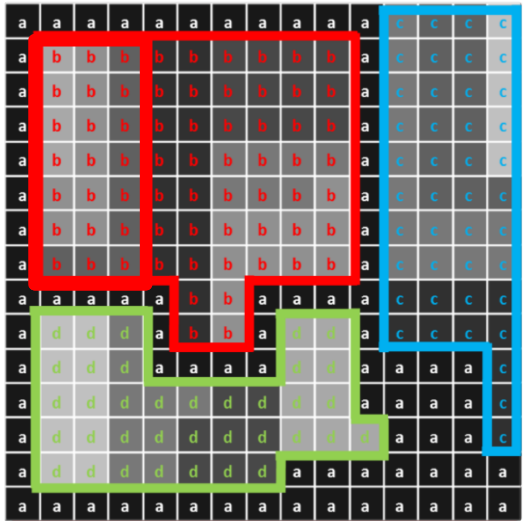
$$\theta = \Delta,\ \ \Delta + 1,\ \ldots \ \rightarrow \ \text{MSERs}$$

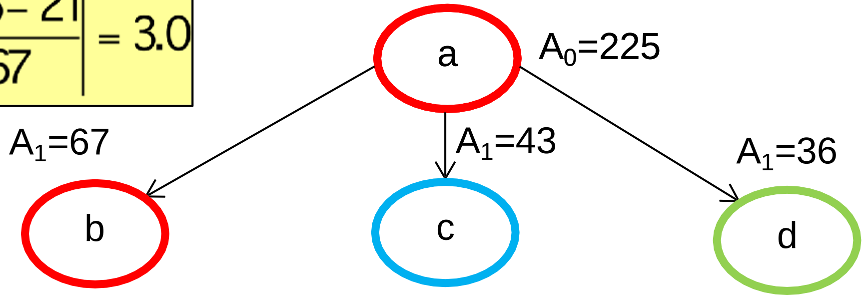
Component tree of an image

$$\left|\frac{225-21}{67}\right| = 3.0$$



$f[x,y] > 1$





Local minimum of sequence

$$\left|\frac{A_{\theta-\Delta}-A_{\theta+\Delta}}{A_\theta}\right| \to \text{MSER}$$

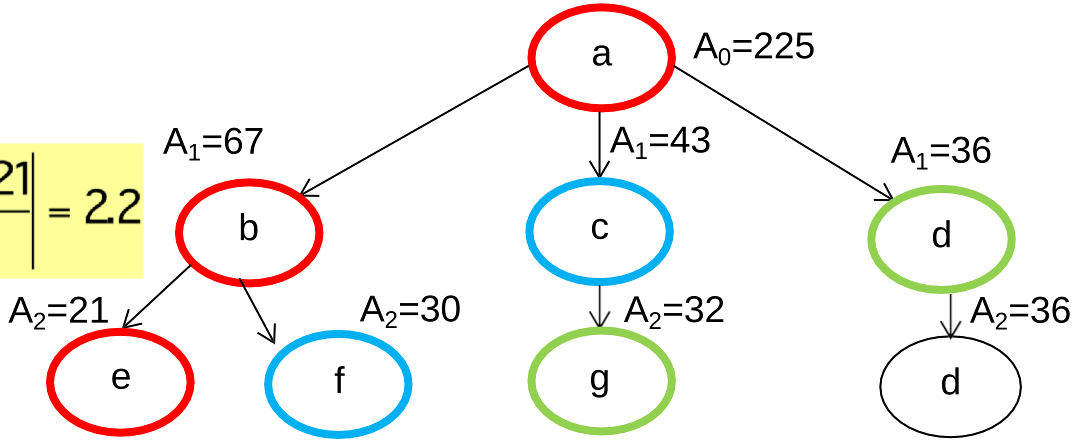
$$\theta = \Delta,\ \Delta+1,\ \ldots \ \to \ \text{MSERs}$$

Component tree of an image

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	[illegible]	[illegible]
0	1	1	1	0	0	1	1	1	1	0	1	1	[illegible]	[illegible]
0	1	1	1	0	0	1	1	1	1	0	1	1	[illegible]	[illegible]
0	0	0	0	0	0	1	0	0	0	0	0	0	[illegible]	[illegible]
0	1	1	1	0	0	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$\left|\frac{67-21}{21}\right| = 2.2$$

$f[x,y] > 2$



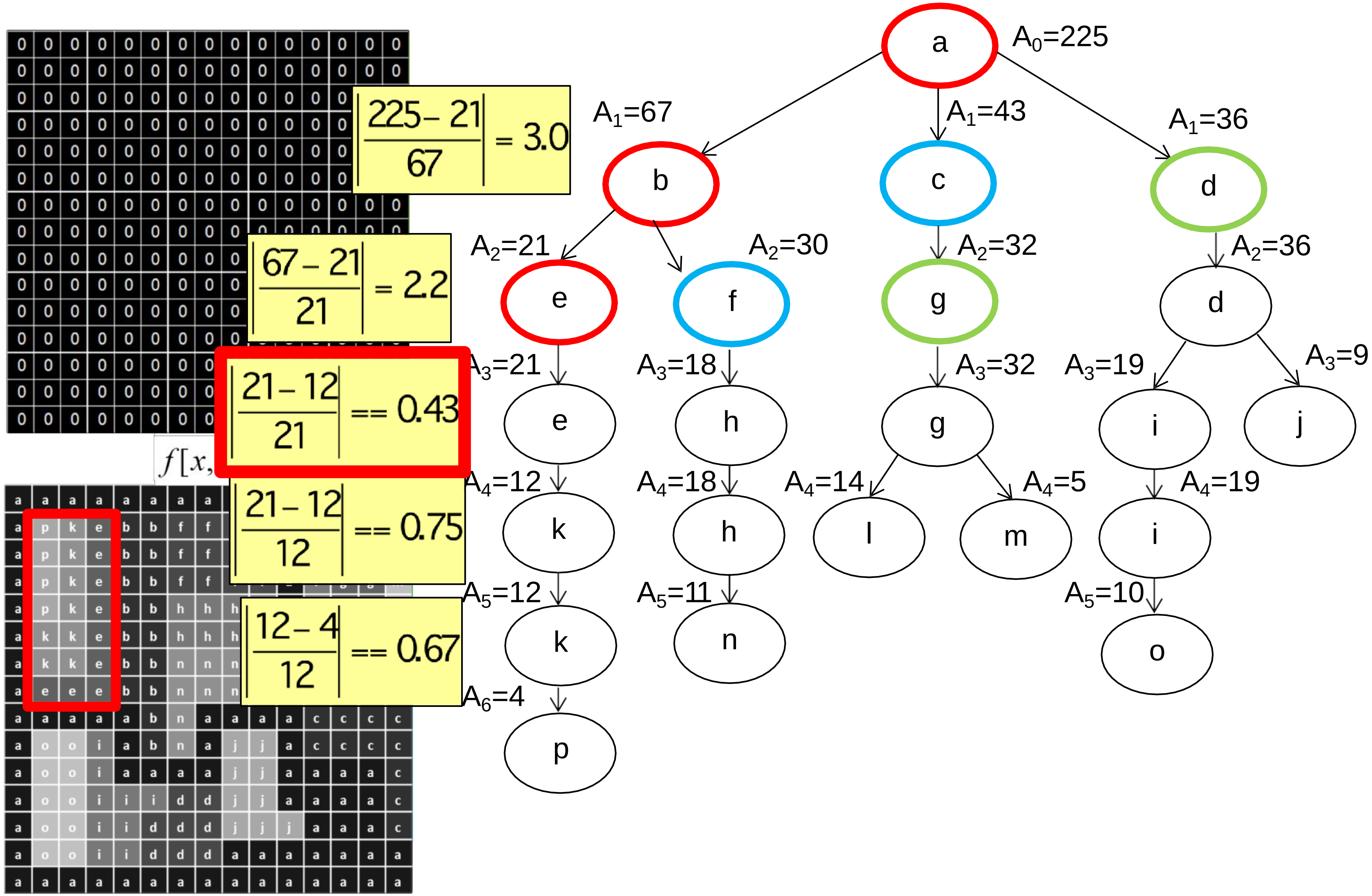
Local minimum of sequence

$$\left|\frac{A_{\theta-\Delta}-A_{\theta+\Delta}}{A_\theta}\right| \rightarrow \text{MSER}$$

$$\theta = \Delta,\ \Delta+1,\ \ldots \rightarrow \text{MSERs}$$

a	a	a	a	a	a	a	a	a	a	a	g	g	g	g
a	e	e	e	b	b	f	f	f	f	a	g	g	g	g
a	e	e	e	b	b	f	f	f	f	a	g	g	g	g
a	e	e	e	b	b	f	f	f	f	a	g	g	g	g
a	e	e	e	b	b	f	f	f	f	a	g	g	g	g
a	e	e	e	b	b	f	f	f	f	a	g	g	g	g
a	e	e	e	b	b	f	f	f	f	a	g	g	g	g
a	e	e	e	b	b	f	f	f	f	a	g	g	g	g
a	a	a	a	a	b	f	a	a	a	a	c	c	c	c
a	d	d	d	a	b	f	a	d	d	a	c	c	c	c
a	d	d	d	a	a	a	a	d	d	a	a	a	a	c
a	d	d	d	d	d	d	d	d	d	a	a	a	a	c
a	d	d	d	d	d	d	d	d	d	d	a	a	a	c
a	d	d	d	d	d	d	d	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a

Component tree of an image



Local minimum of sequence

$$\left|\frac{A_{\theta-\Delta}-A_{\theta+\Delta}}{A_\theta}\right| \rightarrow \text{MSER}$$

MSER: examples

Dark MSERs, Δ=15

Original image

Bright MSERs, Δ=15

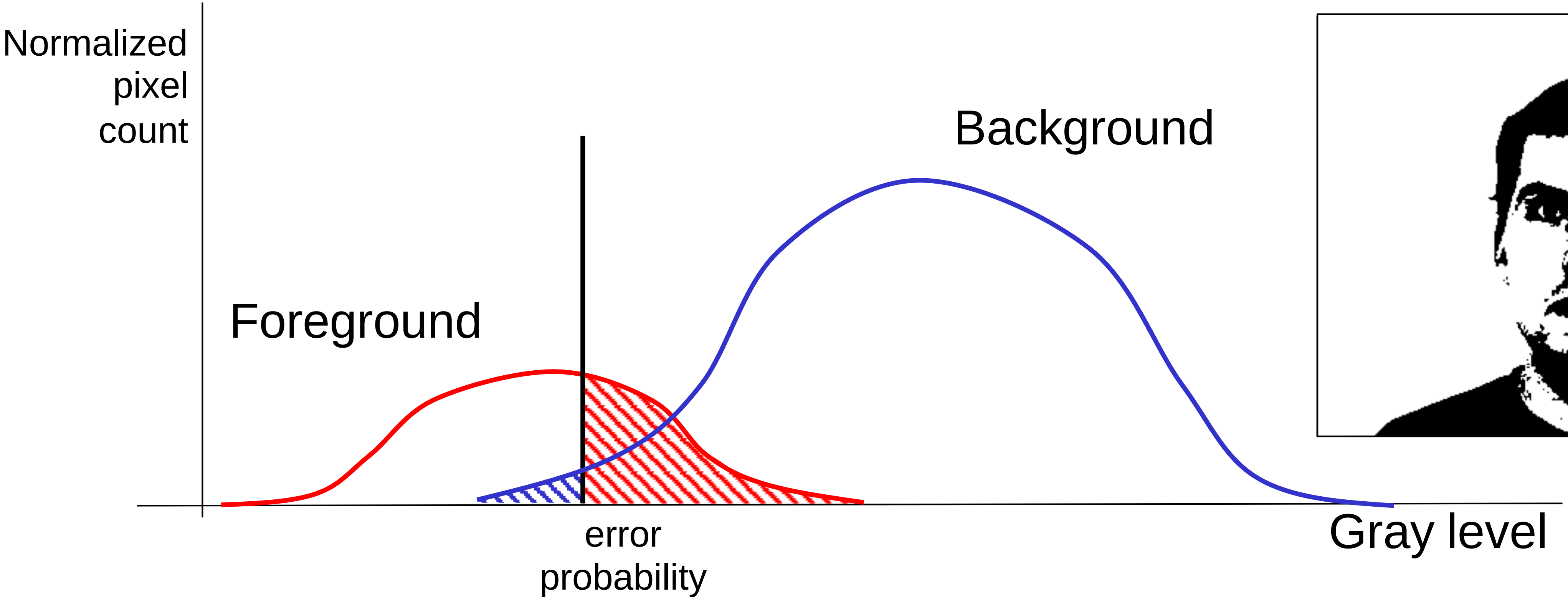
MSER: examples

Dark MSERs, Δ=15

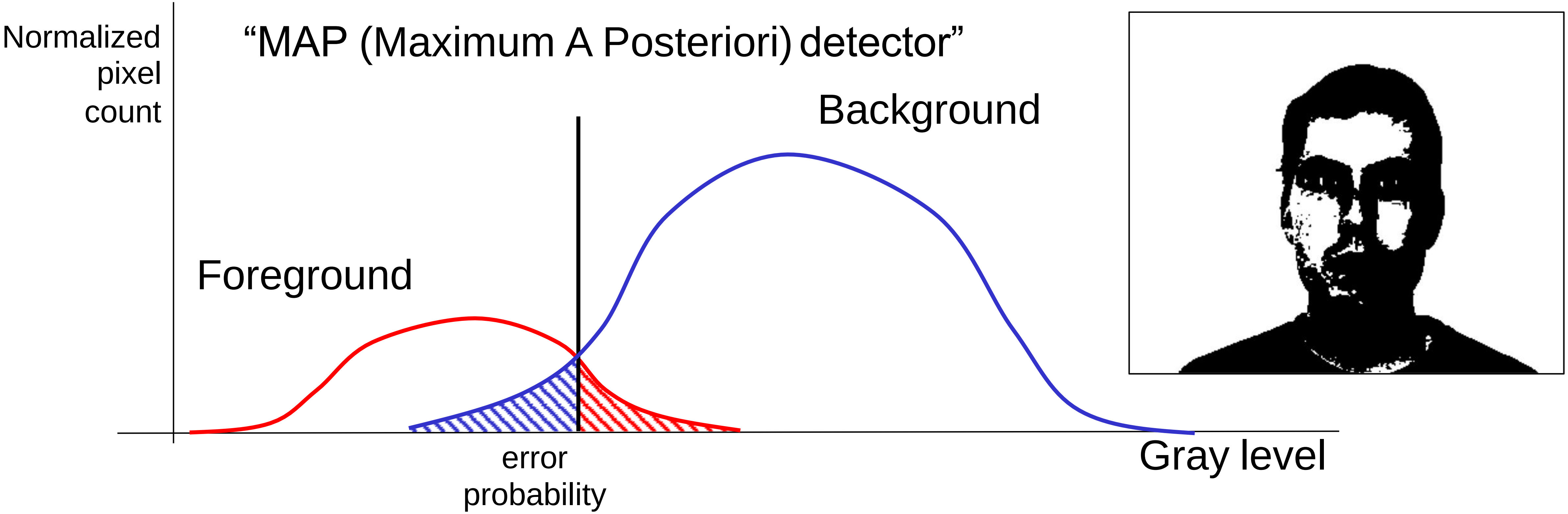
Original image

Bright MSERs, Δ=15

Supervised thresholding



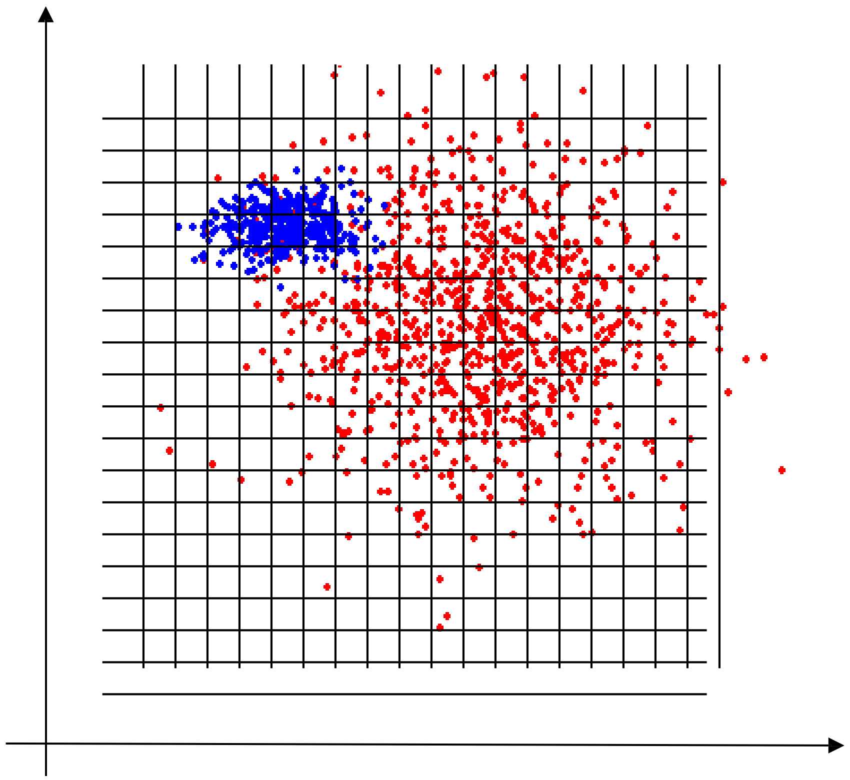
Supervised thresholding



If errors BG➔FG and FG➔BG are associated with different costs:
“Bayes minimum risk detector” is optimal.

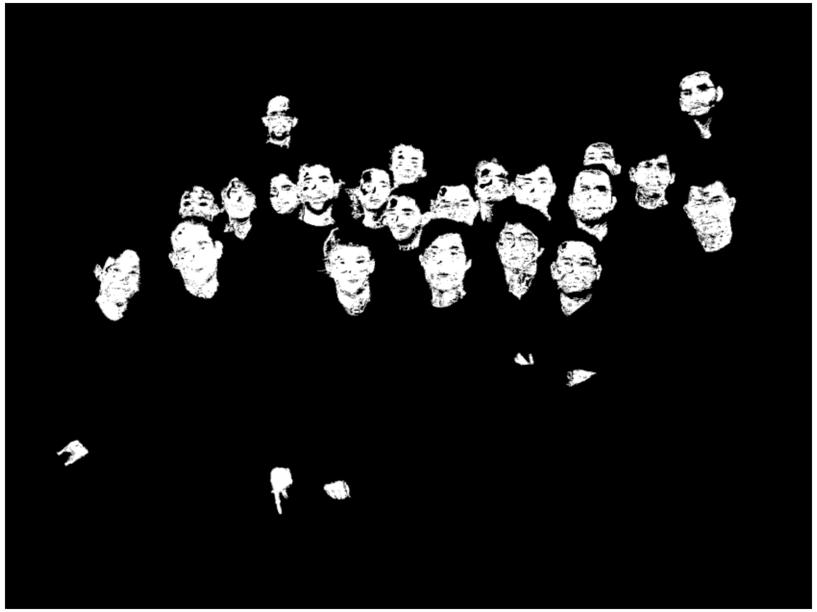
Multidimensional MAP detector

- Training
  - Provide labelled set of training data
  - Subdivide n-dimensional space into small bins
  - Count frequency of occurrence for each bin and class in training set, label bin with most probable class
  - (Propagate class labels to empty bins)
- For test data: identify bin, look up the most probable class

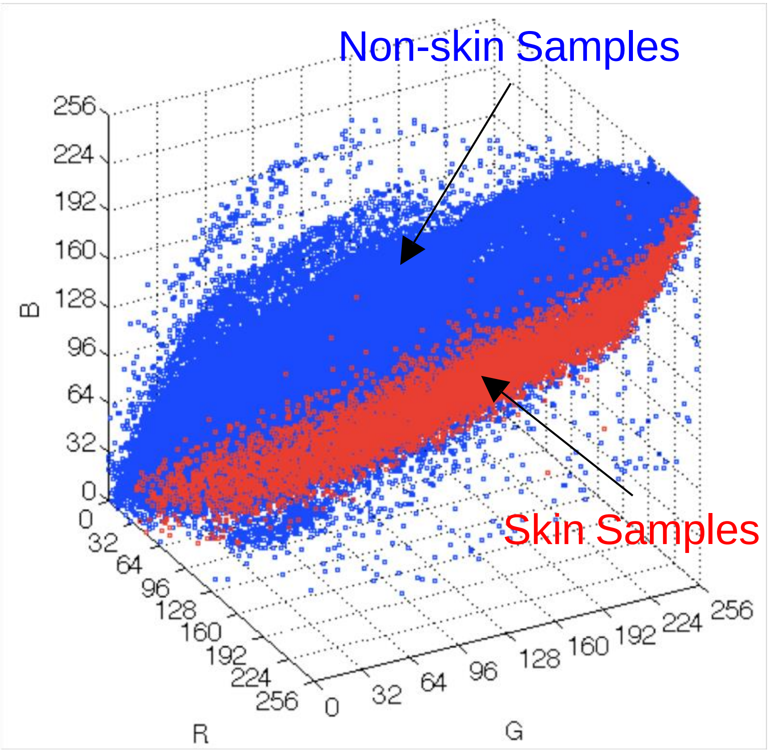


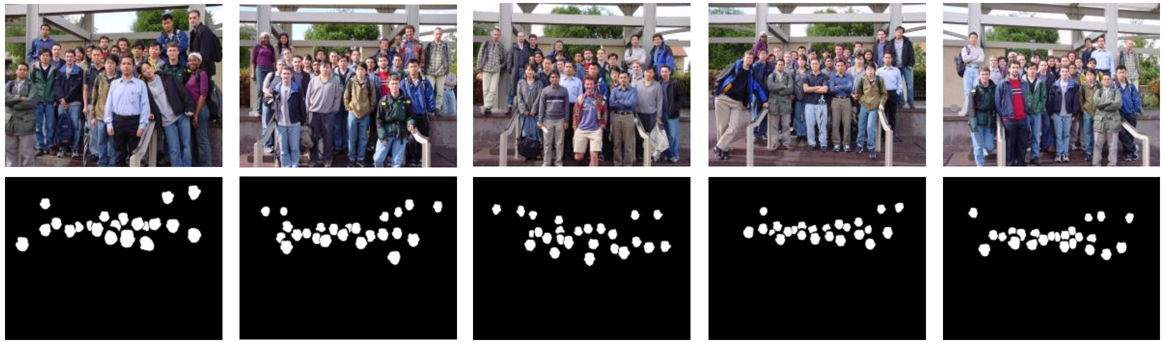
MAP detector in RGB-space

Original image



Skin color detector



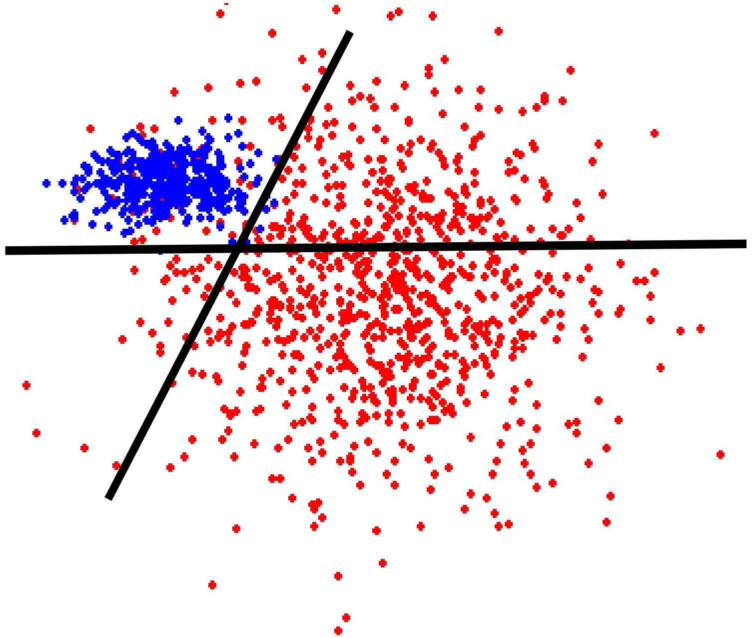


Five training images

Linear discriminant function

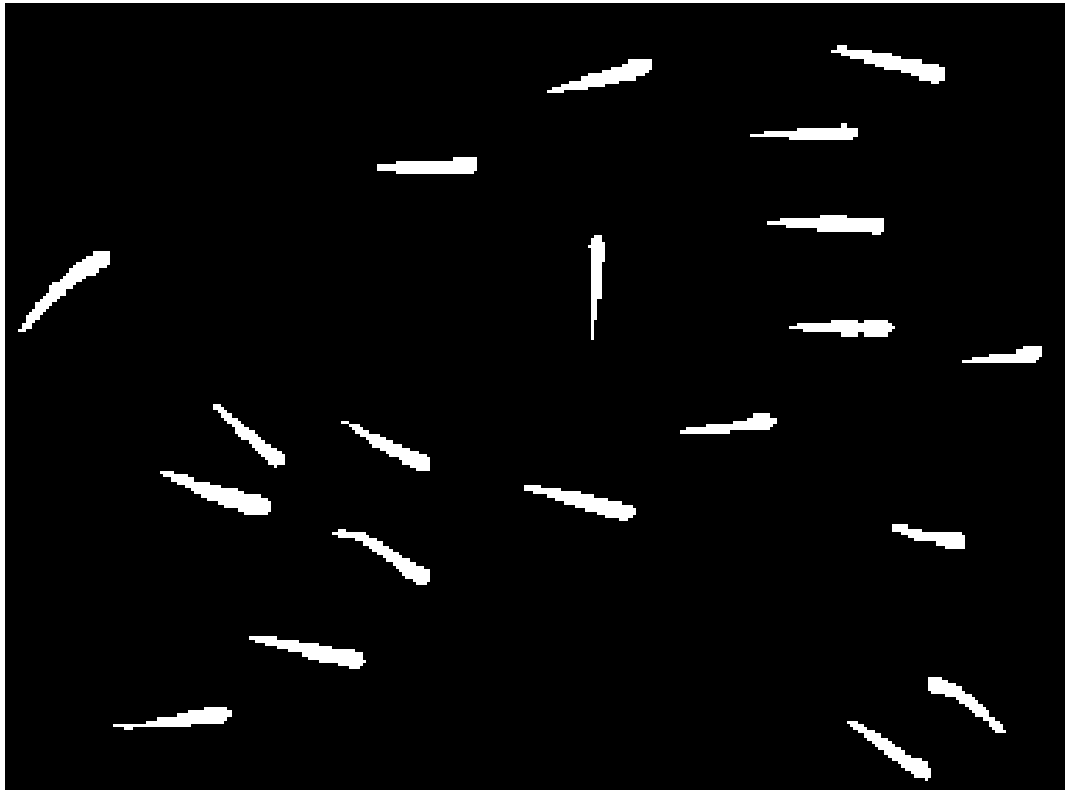
- To segment image with n components f_i, $i=1,2,\ldots,n$ into two classes, perform test

$$\sum_i w_i f_i + w_0 \geq 0 \quad ?$$

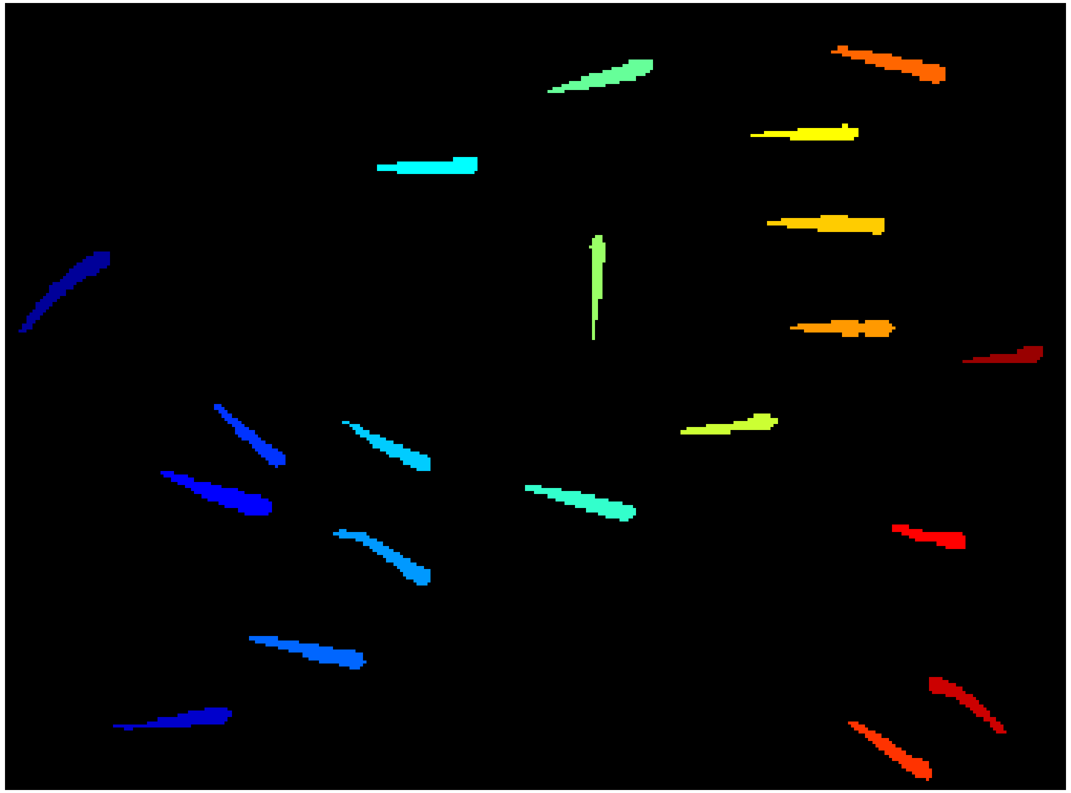


- Categories are separated by hyperplane in n-space
- Numerous techniques to determine weights w_i, $i=0,1,2,\ldots,n$, see, e.g., [Duda, Hart, Stork, 2001]
- Can be extended to the intersection of several linear discriminant functions
- Can be extended to multiple classes

Example: region labeling



Thresholded image



20 labeled regions

Region counting algorithm

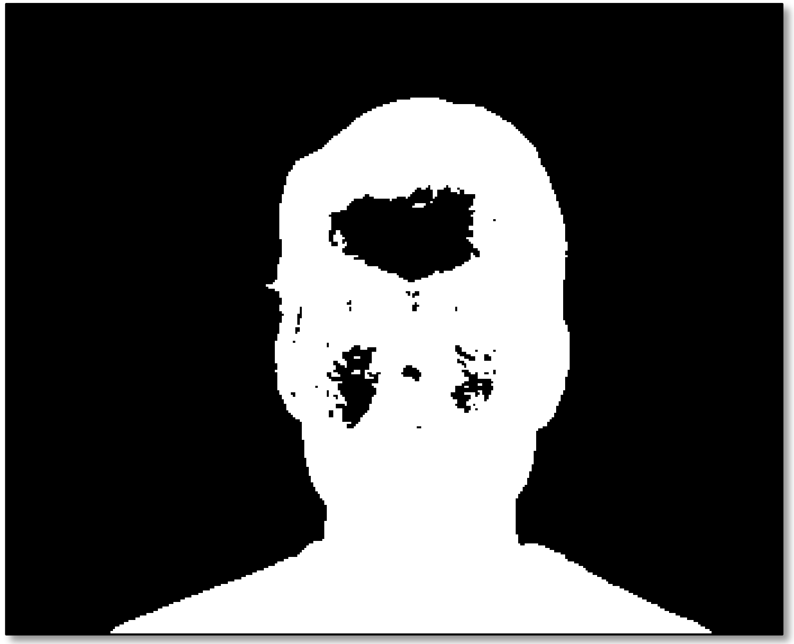
- Measures the size of each region
- Initialize $counter[label]=0$ for all $label$
- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$, increment $counter[label[x,y]]$

Small region removal

- Loop through all pixels $f[x,y]$, left to right, top to bottom
  - If $f[x,y]=0$, do nothing.
  - If $f[x,y]=1$ and $counter[label[x,y]]<S$, set $f[x,y]=0$
- Removes all regions smaller than S pixels

Hole filling as dual to small region removal

Mask with holes



After NOT operation, (background) region labeling, small region removal, and second NOT operation