.

Image segmentation

Stefano Ferrari

Università degli Studi di Milano
stefano.ferrari@unimi.it

Methods for Image Processing

academic year 2018–2019

Segmentation by thresholding

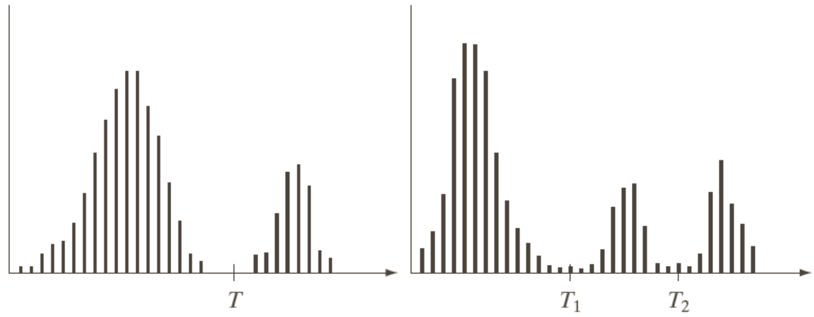
- Thresholding is the simplest segmentation method.
- The pixels are partitioned depending on their intensity value.
- Global thresholding, using an appropriate threshold T:

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

- Variable thresholding, if T can change over the image.
  - Local or regional thresholding, if T depends on a neighborhood of (x, y).
  - adaptive thresholding, if T is a function of (x, y).
- Multiple thresholding:

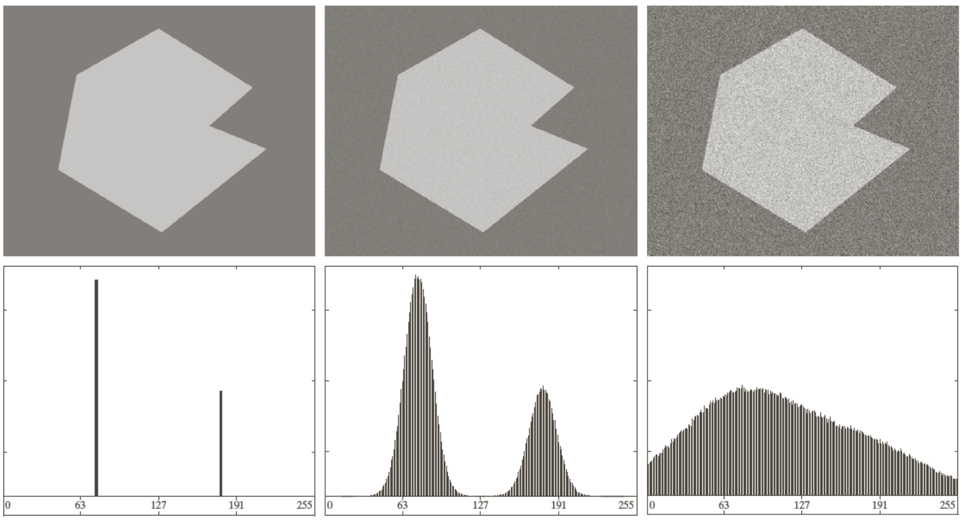
$$g(x, y) = \begin{cases} a, & \text{if } f(x, y) > T_2 \\ b, & \text{if } T_1 < f(x, y) \leq T_2 \\ c, & \text{if } f(x, y) \leq T_1 \end{cases}$$

Choosing the thresholds



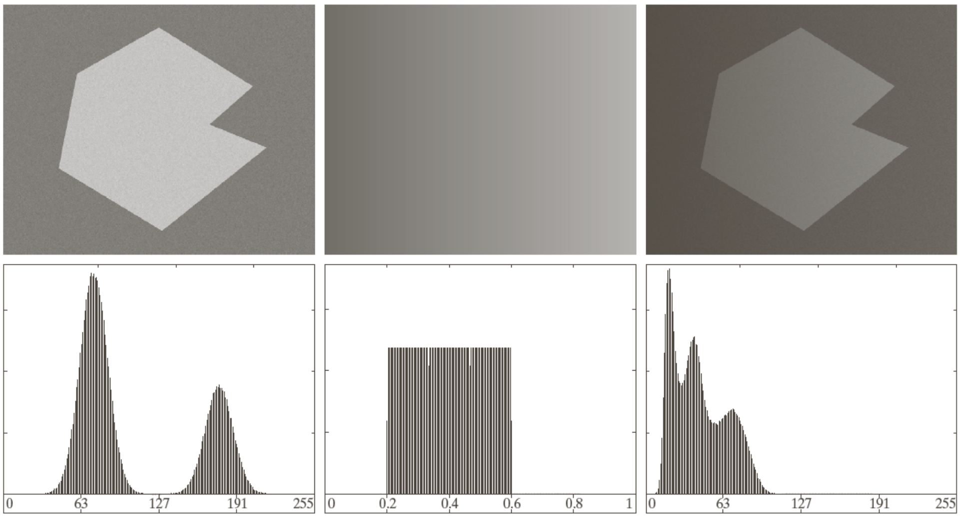
- Peaks and valleys of the image histogram can help in choosing the appropriate value for the threshold(s).
- Some factors affects the suitability of the histogram for guiding the choice of the threshold:
  - the separation between peaks;
  - the noise content in the image;
  - the relative size of objects and background;
  - the uniformity of the illumination;
  - the uniformity of the reflectance.

Noise role in thresholding



No noise   10% noise   50% noise

Illumination and reflection role in thresholding



A $\quad$ B $\quad$ $A \cdot B$

Global thresholding

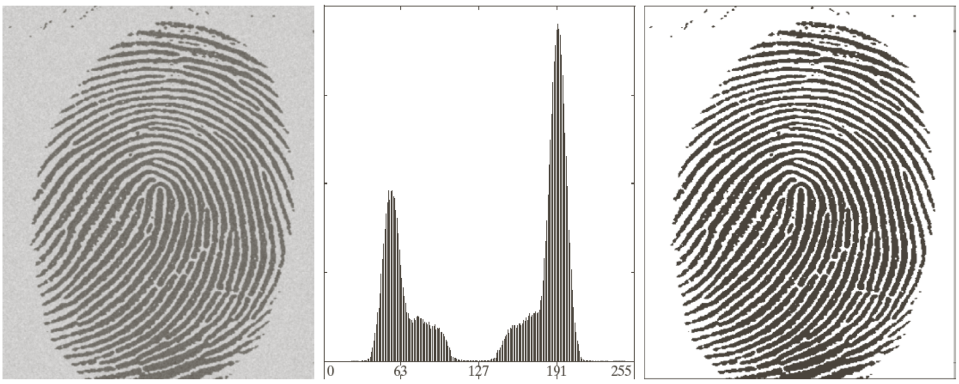
A simple algorithm:

1. Initial estimate of T
2. Segmentation using T:
   - G_1, pixels brighter than T;
   - G_2, pixels darker than (or equal to) T.
3. Computation of the average intensities m_1 and m_2 of G_1 and G_2.
4. New threshold value:

$$T_{\text{new}} = \frac{m_1 + m_2}{2}$$

5. If $|T - T_{\text{new}}| > \Delta T$, back to step 2, otherwise stop.

Global thresholding: an example



Otsu's method

- Otsu's method is aimed in finding the optimal value for the global threshold.
- It is based on the interclass variance maximization.
  - Well thresholded classes have well discriminated intensity values.
- $M \times N$ image histogram:
  - L intensity levels, $[0, \ldots, L-1]$;
  - n_i #pixels of intensity i:

$$MN = \sum_{i=0}^{L-1} n_i$$

- Normalized histogram:

$$p_i = \frac{n_i}{MN}$$

$$\sum_{i=0}^{L-1} p_i = 1, \qquad p_i \geq 0$$

Otsu's method (2)

- Using k, $0 < k < L-1$, as threshold, $T = k$:
  - two classes: C_1 (pixels in $[0, k]$) and C_2 (pixels in $[k+1, L-1]$)
  - $P_1 = P(C_1) = \sum_{i=0}^{k} p_i$, probability of the class C_1
    - $P_2 = P(C_2) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1$, probability of the class C_2
  - m_1, mean intensity of the pixels in C_1:

$$\begin{aligned} m_1 &= \sum_{i=0}^{k} i \cdot P(i|C_1) \\ &= \sum_{i=0}^{k} i \frac{P(C_1|i)P(i)}{P(C_1)} \\ &= \frac{1}{P_1} \sum_{i=0}^{k} i \cdot p_i \end{aligned}$$

where $P(C_1|i) = 1$, $P(i) = p_i$ e $P(C_1) = P_1$.

Otsu's method (3)

- Similarly, m_2, mean intensity of the pixels in C_2:

$$m_2 = \frac{1}{P_2} \sum_{i=k+1}^{L-1} i \cdot p_i$$

- Mean global intensity, m_G:

$$m_G = \sum_{i=0}^{L-1} i \cdot p_i$$

- while the mean intensity up to the k level, m:

$$m = \sum_{i=0}^{k} i \cdot p_i$$

- Hence:

$$P_1 m_1 + P_2 m_2 = m_G$$

$$P_1 + P_2 = 1$$

.

Otsu's method (4)

- The global variance σ_G^2:

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 \cdot p_i$$

- The *between-class variance*, σ_B, can be defined as:

$$\begin{aligned} \sigma_B^2 &= P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \\ &= P_1 P_2 (m_1 - m_2)^2 \\ &= \frac{(m_G P_1 - m)^2}{P_1(1 - P_1)} \end{aligned}$$

- The *goodness* of the choice $T = k$ can be estimated as the ratio η:

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

Otsu's method (5)

- The quantities required for the computation of η, can be obtained from the histogram:
- Hence, for each value of k, $\eta(k)$ can be computed:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$$

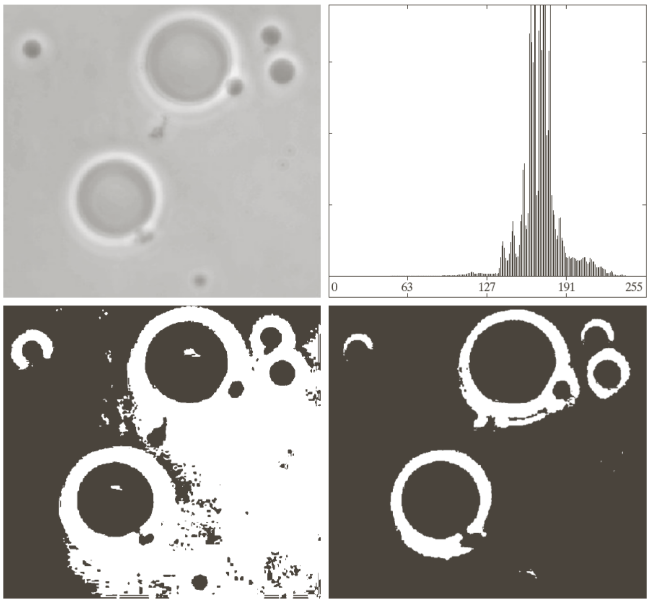
where

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

- The optimal threshold value, k^*, satisfies:

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

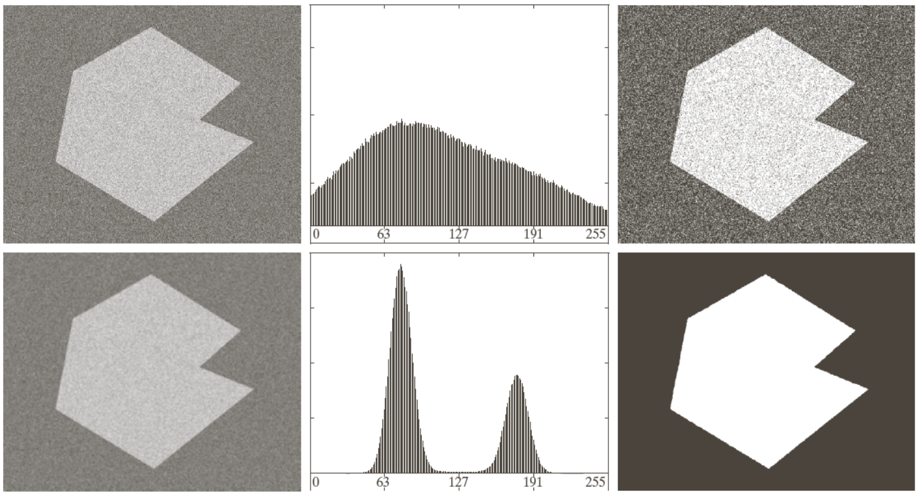
Otsu's method: an example



a	b
c	d

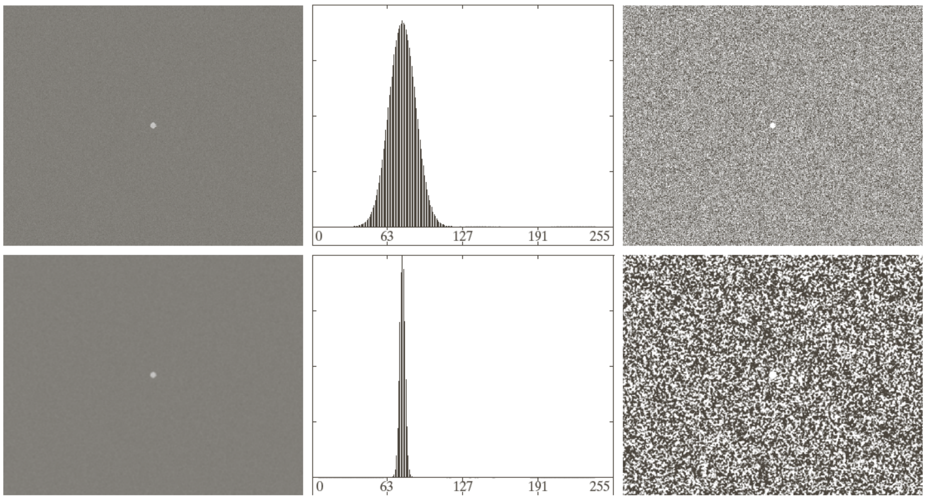
- (a) original image;
- (b) histogram of (a);
- (c) global threshold: $T = 169$, $\eta = 0.467$;
- (d) Otsu's method: $T = 181$, $\eta = 0.944$.

Smoothing



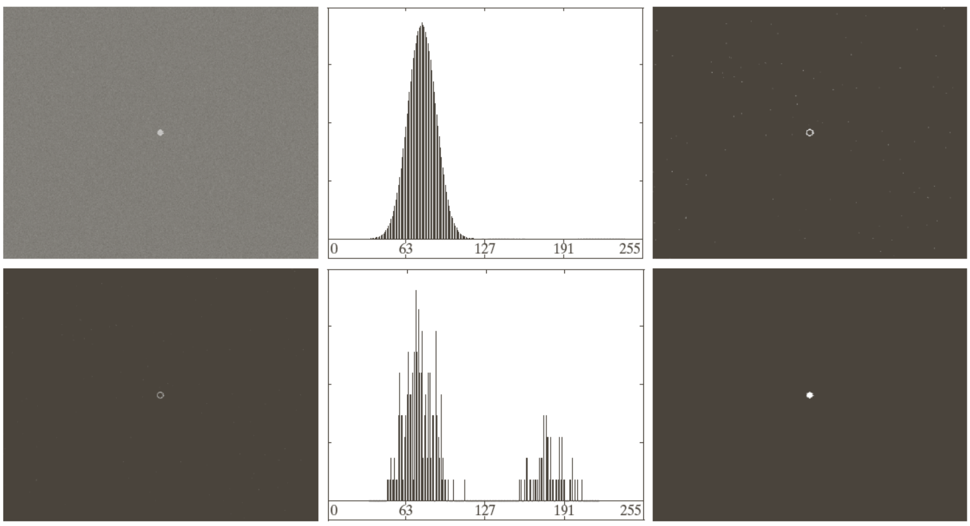
- Otsu's method may not work in presence of noise.
- Smoothing can produce a histogram with separated peaks.

Significance of the histogram



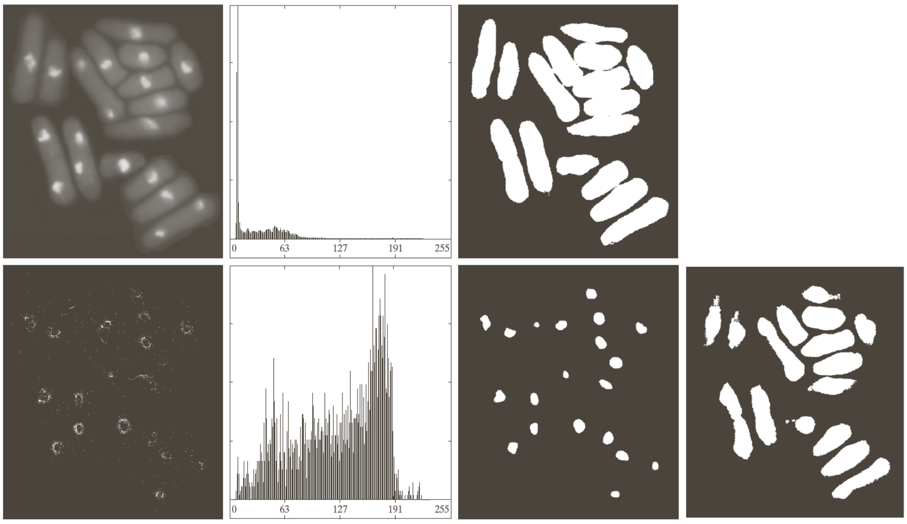
- If the distribution is not balanced, no information can be extracted from the histogram.
- Smoothing cannot help.

Selection of the border region



- Edge extraction techniques (e.g., Laplacian), can be used for selecting the region that carry the valuable information:
  - Those pixels that belong to the objects and to the background with an equal probability.

Use of edge for global thresholding (2)



- Changing the threshold of the Laplacian, several segmentations are obtained.
  - It can be useful for nested classes.

Multiple thresholds Otsu's method

- The Otsu's method can be applied also for the multiple thresholds segmentation (generally, double threshold).
- Between-class variance:

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

- The optimal thresholds k_1^* and k_2^* can be computed as:

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

- The separability degree can be measured as:

$$\eta(k_1^*,\, k_2^*) = \frac{\sigma_B^2(k_1^*,\, k_2^*)}{\sigma_G^2}$$

.

Multiple thresholds Otsu's method: an example

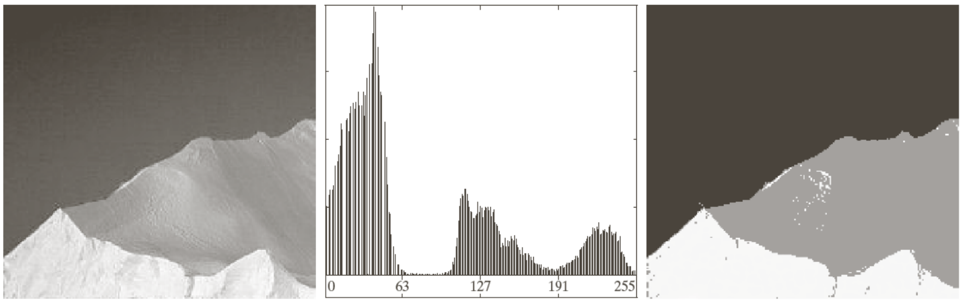
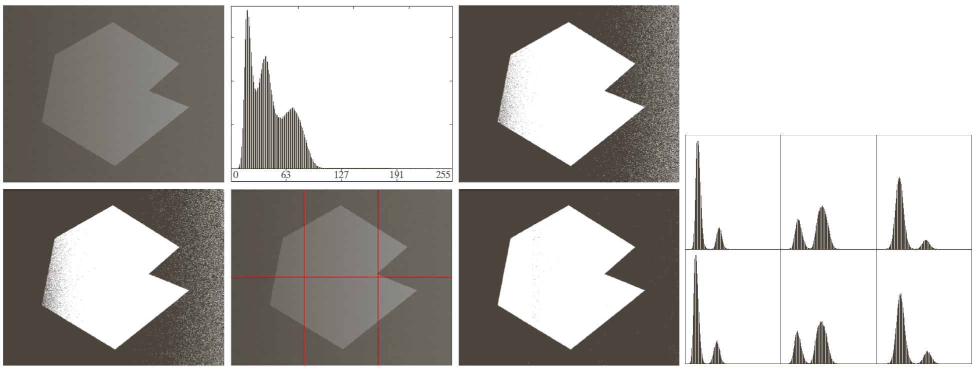


Image partitioning based thresholding

- In order to face non uniform illumination or reflectance, the image is partitioned and the thresholding is operated on each partition.
  - In each partition, the illuminance and reflectance is supposed uniform.
  - In each partition, objects and background have to be equally represented.



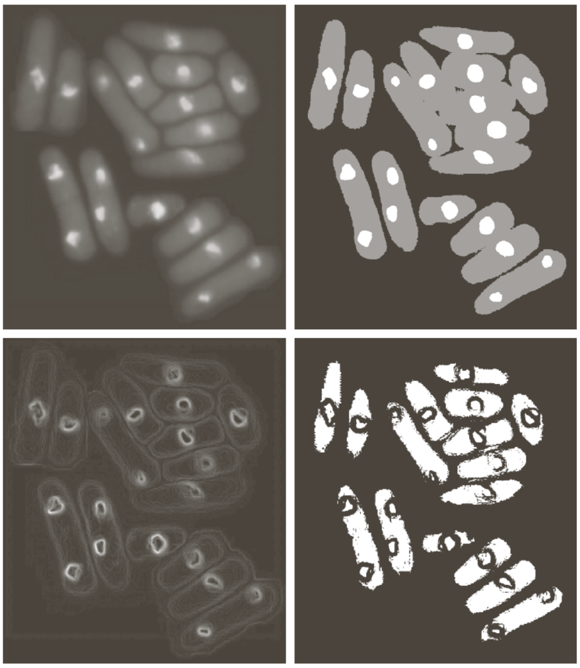
Local properties based thresholding

- Local properties (e.g., statistics) based criteria can be used for adapting the threshold.
- For example:
  - $T_{xy} = a\sigma_{xy} + bm_{xy}$
  - $T_{xy} = a\sigma_{xy} + bm_G$
- The segmentation is operated using a suitable predicate, Q_{xy}:

$$g(x, y) = \begin{cases} 1, & \text{if } Q_{xy} \\ 0, & \text{otherwise} \end{cases}$$

  where Q_{xy} can be, for instance:
  - $f(x, y) > T_{xy}$
  - $f(x, y) > a\sigma_{xy}$ AND $f(x, y) > bm_{xy}$
- This technique can be easily generalized to multiple thresholds segmentation.

Local properties based thresholding: an example



a	b
c	d

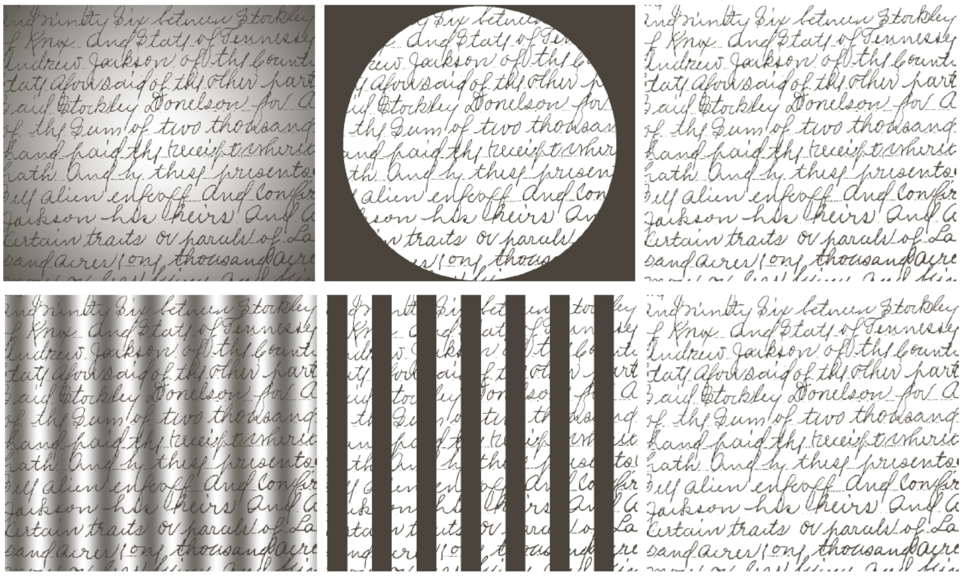
(b) segmentation of (a) with double threshold Otsu;

(c) local (3×3) standard deviation;

(d) segmentation with local thresholding.

Moving averages thresholding

- Pixels are visited following a zigzag path and the statistics are computed using only the last n visited pixels.



Growing based segmentation

- *Region growing* is a technique based on a controlled growing of some initial pixels (*seeds*).
- The selection of the *seeds* can be operated manually or using automatic procedures based on appropriate criteria.
  - A-priori knowledge can be included.
  - It is strictly application-dependent.
- The growing is controlled by the connectivity.
- The stop rule is another parameter of the algorithm.
  - It can depend on the a-priori knowledge on the problem.

Region growing: the basic algorithm

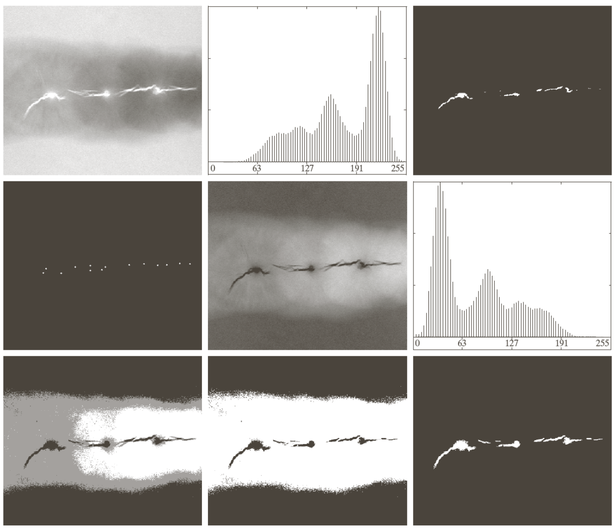
Given:

- $f(x, y)$, the image to be segmented;
- $S(x, y)$, binary image with the seeds (it is 1 only where the seeds are located);
- Q, predicate to be tested for each location (x, y).

A simple region growing algorithm (based on 8-connectivity) is the following:

1. Erode all the connected components of S until they are only one pixel wide.
2. Generate the binary image f_Q such that $f_Q(x, y) = 1$ if $Q(x, y)$ is true.
3. Create the binary image g where $g(x, y) = 1$ if $f_Q(x, y) = 1$ and (x, y) is 8-connected to a seed in S.
4. The resulting connected components in g are the segmented regions.

Region growing: an example



a	b	c
d	e	f
g	h	i

(a) f

(c) $S(x, y) := f(x, y) > 254$

(d) erosion of S

(e) $|f - S|$

(h) $|f - S| > 68$

(i) segmentation by region growing with $Q := |f - S| <= 68$

.

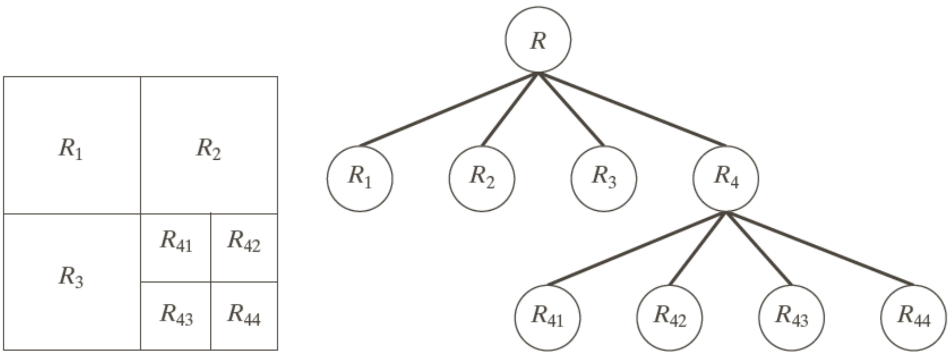
Region splitting and merging

- Iterative subdivision of the image in homogeneous regions (*splitting*).
- Joining of the adjacent homogeneous regions (*merging*).

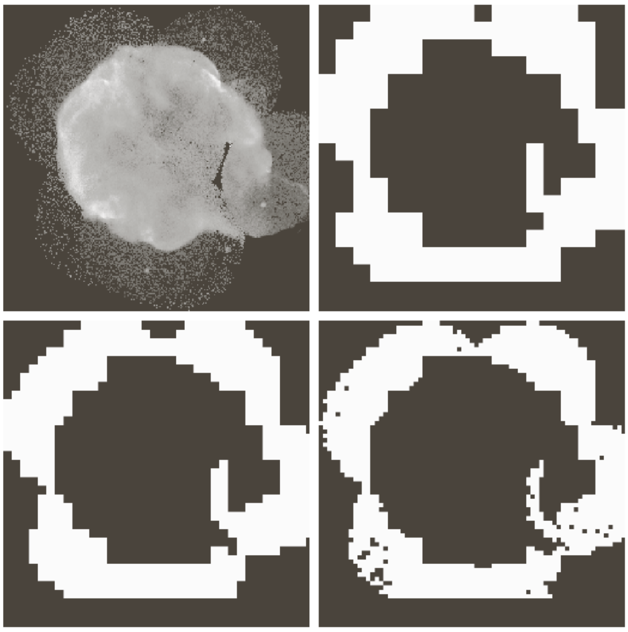
Given an image f and a predicate Q, the basic algorithm is:

1. $R_0 = f$
2. Iterative subdivision in quadrants of each region R_i for which $Q(R_i) = \text{FALSE}$.
3. If $Q(R_i) = \text{TRUE}$ for every regions, merge those adjacent regions R_i and R_j such that $Q(R_i \cup R_j) = \text{TRUE}$; otherwise, repeat step 2.
4. Repeat the step 3 until no merging is possible.

Quadtree based partitioning



Splitting and merging: an example

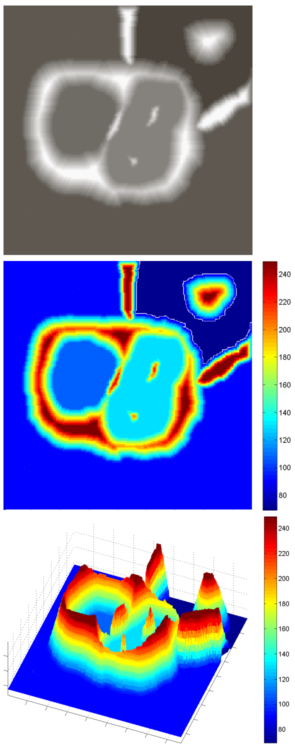


a	b
c	d

$$Q := \sigma > a \text{ AND } 0 < m < b$$

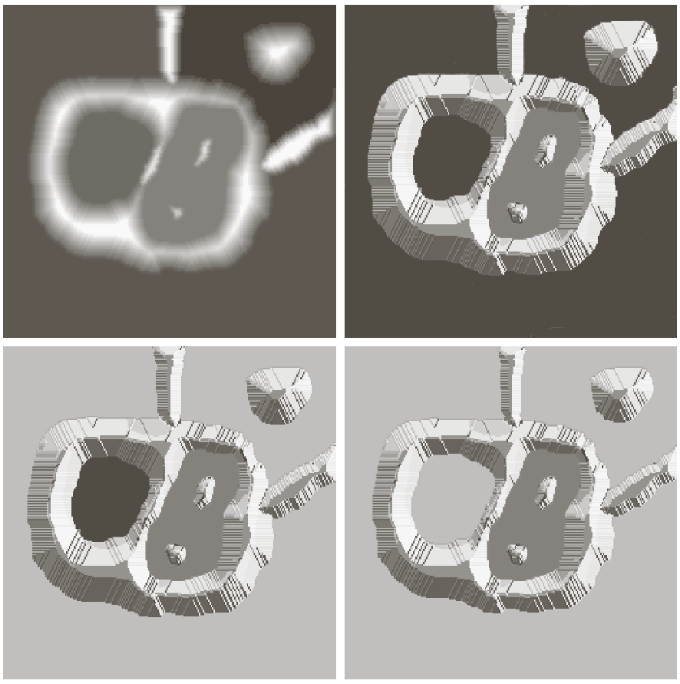
- (b) 32×32
- (c) 16×16
- (d) 8×8

Watershed

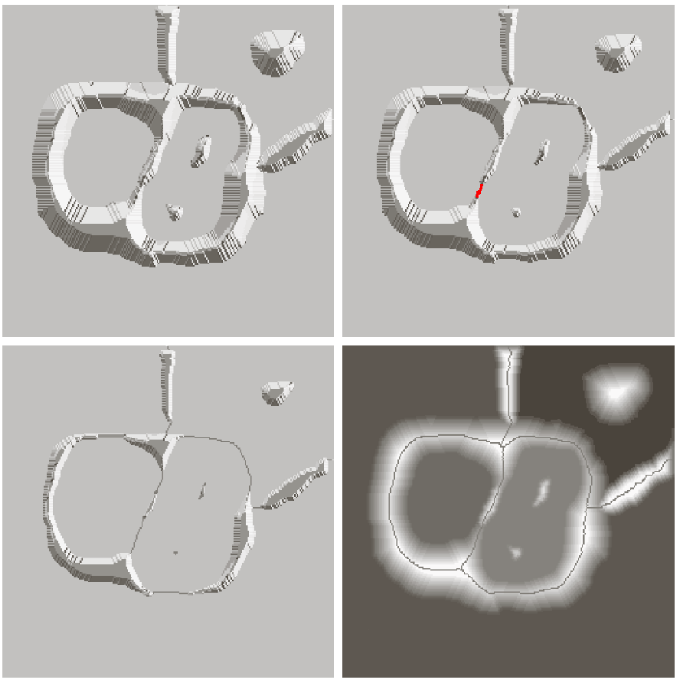


- The watershed technique is based on a topological interpretation of the image.
  - The intensity levels represent the height of the terrain that describe mountains and basins.
- For each basin, a hole in its minimum is supposed to be realized, from which, the rising underground water spills and fills the basins.
- As the water rises, the level reach the border of the basin and two or more adjacent basins tend to merge together.
- Dams are required for maintaining a separation between basins.
- These dams are the borders of the regions of the segmentation.

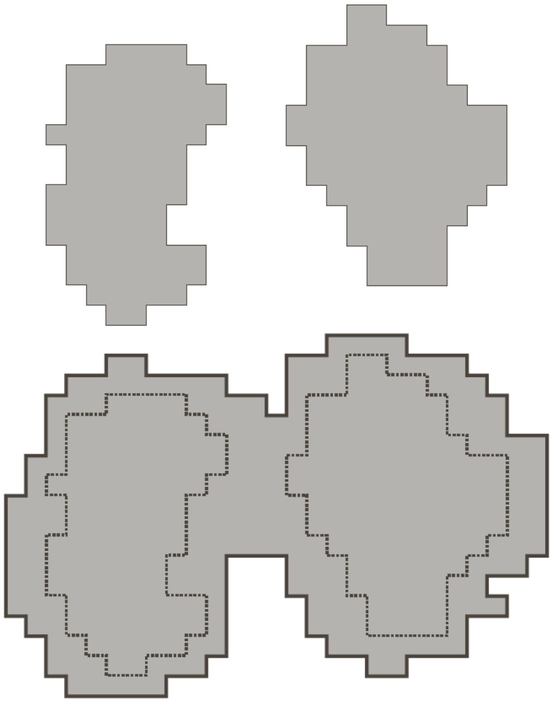
Watershed (2)



Watershed (3)

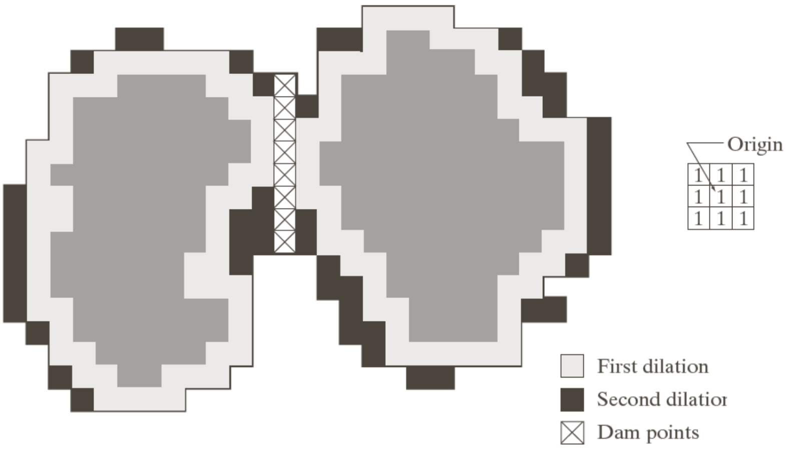


Dams construction



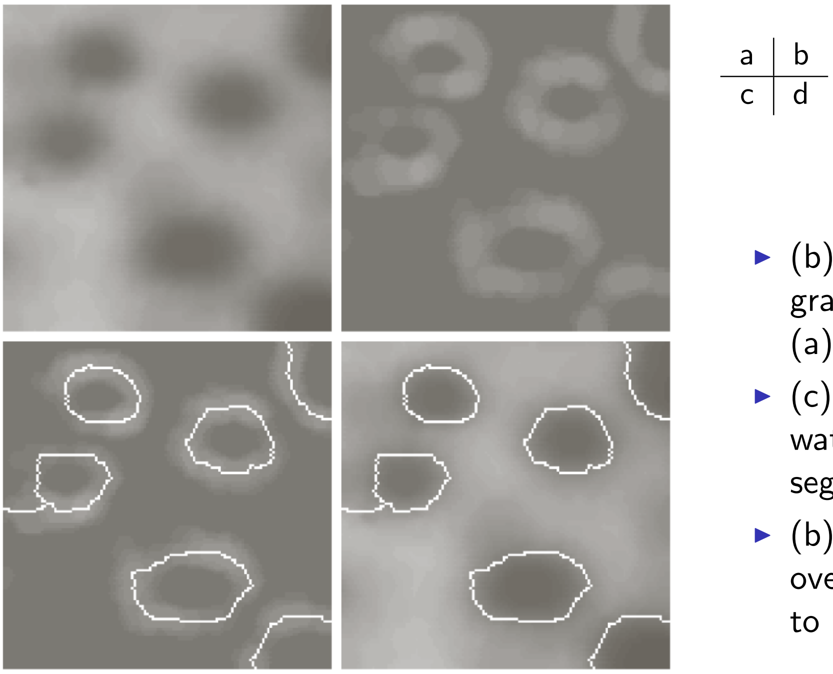
- The dams can be build using morphological dilation.
- Starting from the last step before merging, dilation can be performed until the two disjoint components become connected.

Dams construction (2)



- The dams can be realized setting the pixels at L (where the levels of intensity are in $[0,\ L-1]$).
- Generally, the watershed algorithm is applied to the gradient of the image to be segmented.

Watershed: an example



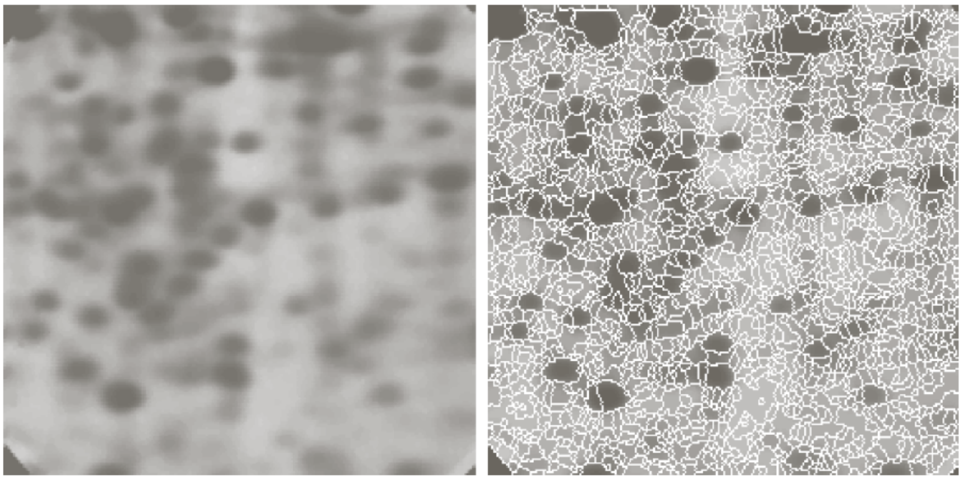
- (b) the gradient of (a)
- (c) resulting watershed segmentation
- (b) dams overimposed to (a)

Watershed with marker

- Noise as well as irrelevant details make difficult the application of the watershed technique in real images.
  - *Oversegmentation* can be produced.
- These problems can be handled limiting the flooding through markers:
  - *internal*, associated to the object of interest;
  - *external*, associated to the background (border of the objects).
- The watershed algorithm can then be applied considering the marker as the only minimum points from which starting the procedure.
- The criteria used for defining the markers incorporate the a-priori knowledge on the problem.

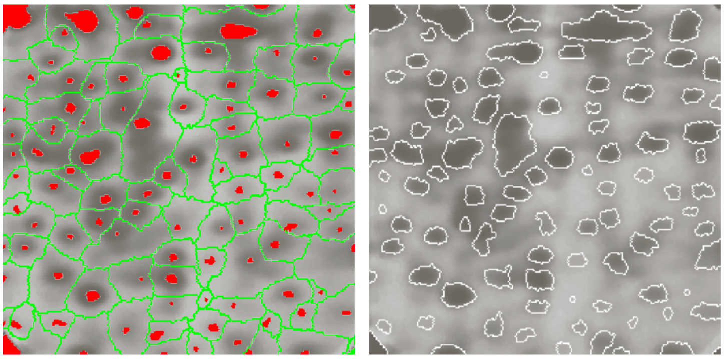
.

Watershed with markers: an example



- Oversegmentation obtained applying watershed to the image gradient.

Watershed with markers: an example (2)

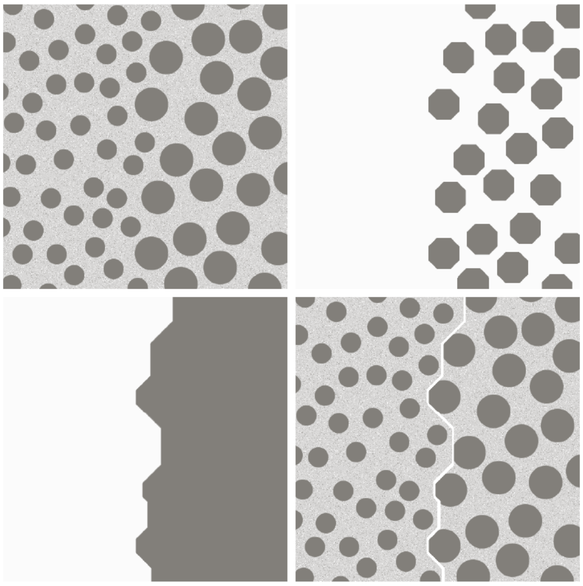


a | b

(a) Smoothing of the original image.
- Internal markers are defined as minimum points that forms connected components (in red).
- The application of the watershed starting from internal markers generates the dams (in green), that can be used as external markers.

(b) Segmentation obtained applying watershed in each region of (a).
- Other segmentation algorithms can be applied to the single regions, as well.

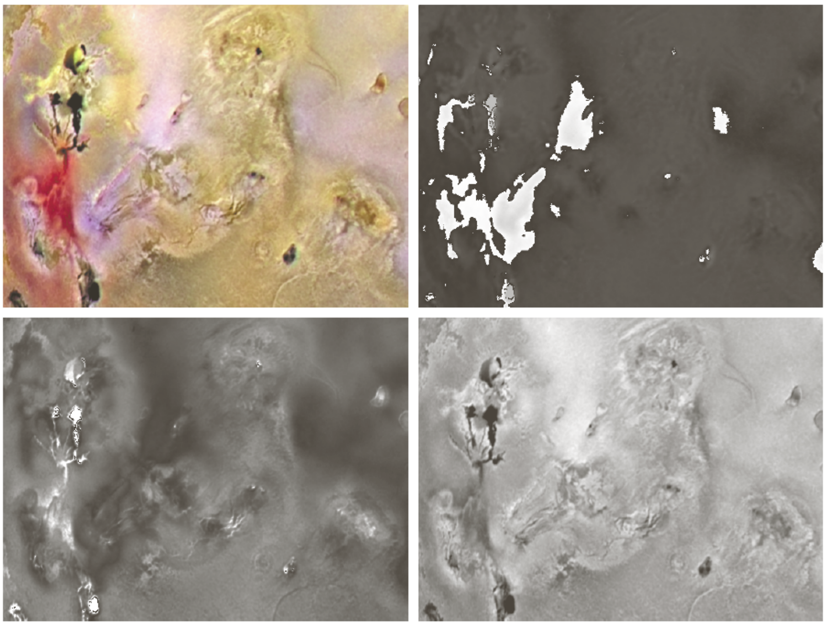
Texture based segmentation



a | b
c | d

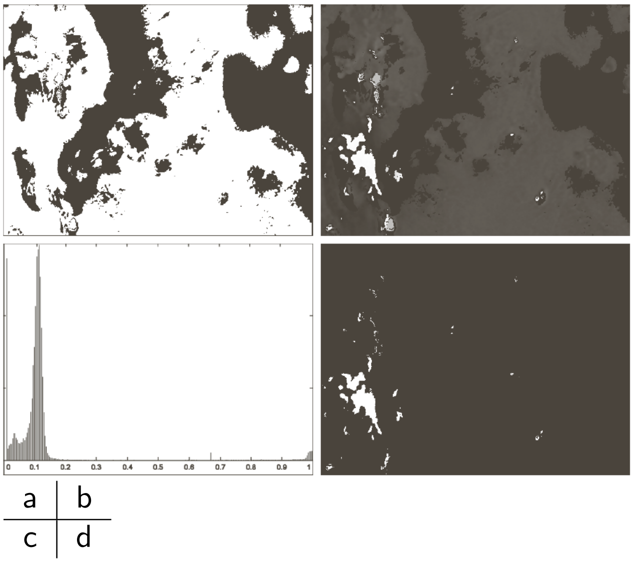
- (a) 600×600 pixels image
- (b) Closing of (a) using a disk of 30 pixels of radius.
- (c) Opening using a disk of 60 pixels of radius.
- (d) Segmentation boundary obtained as morphological gradient.

Color based segmentation



- HSI space

Color based segmentation (2)

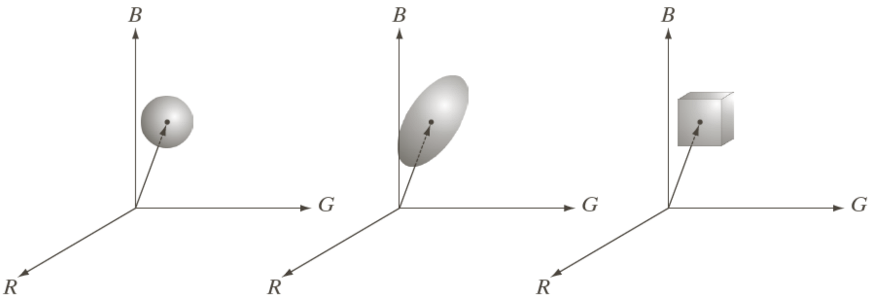


(a) Binary saturation mask (threshold at 90%);

(b) product of the mask by the hue;

(d) segmentation of (b) based on its histogram (c).

Color based segmentation (3)

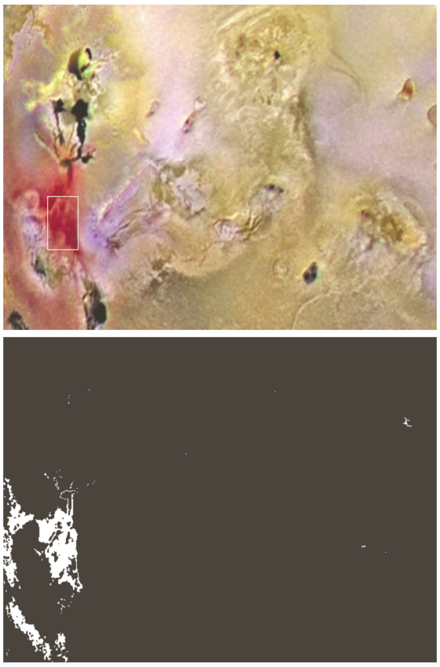


- RGB space: selection of colors similar to a
  - Criterion: $D(z, a) < D_0$

$$D(z, a) = ||z - a|| = \left((z-a)^T(z-a)\right)^{\frac{1}{2}}$$

$$D(z, a) = \left((z-a)^T C^{-1}(z-a)\right)^{\frac{1}{2}}$$

Color based segmentation (4)

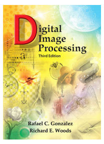


a
b

(a) Manual selection of the color of interest.
  - Average color computation, $a = [a_R \; a_G \; a_B]$
  - Standard deviation of the color of the selected pixels computation, $\sigma = [\sigma_R \; \sigma_G \; \sigma_B]$.

(b) Segmentation of the pixels that have a red channel value in the interval $[a_R - \sigma_R, \, a_R + \sigma_R]$.

Homeworks and suggested readings



DIP, Sections 10.3–10.5
- pp. 738–778

  Sections 6.7.1-6.7.2
  - pp. 443–447

  Section 9.6.3
  - pp. 675–676

GIMP
- Tools
  - Selection Tools
    - Foreground Select
    - By Color Select