Image enhancement and segmentation

Image and signal processing

Samia Bouchafa-Bruneau

M1 E3A - UEVE/Upsay

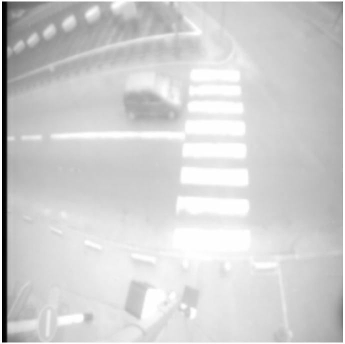
- **Shooting problems**
  - Camera shift
  - Variations of illumination
- **Optic problems**
  - Spatial distortions
  - Focus defects
- **Sensing problems**
  - Electronic noise
- **Discretization problems**
  - Discretization error
  - Coding and quantization error

Consequences: homogeneous regions do not exist, edges are not continous, borders between regions are not easily visible, etc.

Image basic Statistics

Level Contrast

- Standard deviation
  $$C = \sqrt{\frac{\sum_x \sum_y \left[I(x,y) - \bar{I} \right]}{M \times N}}$$
- Variations between min and max
  $C = \max(I(x,y)) - \min(I(x,y))$ ou $C = \frac{\max(I(x,y)) - \min(I(x,y))}{\max(I(x,y)) + \min(I(x,y))}$

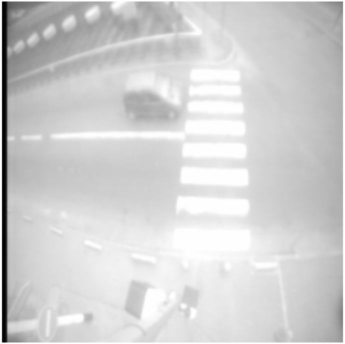


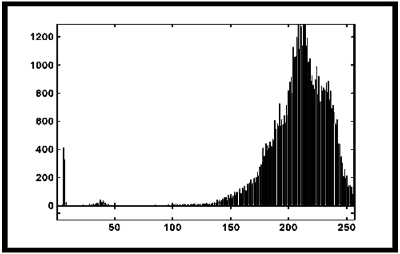
- Histogram: diagram that gives the statistical distribution of gray level in the image
- In statistics: Probability density function

Consider n pixels with random value x

- Distribution Function of the variable x : $P(x \preceq a) = F(a)$
- $P(a, b) = F(b) - F(a)$
- Histogram: $P(x) = \lim \frac{P(\Delta x)}{\Delta x} = F'(x)$
- Cumulative Histogram: $\int_0^{x \max} F'(x)dx = F(x \max)$

- Example of histogram:





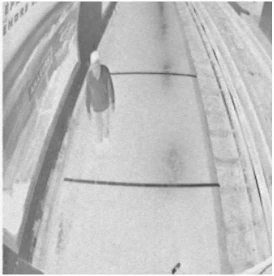
- **Histogram descriptor**
  - Mean: $\overline{x} = \frac{\sum x_k h(x_k)}{\sum h(x_k)}$ with $h(x_k) = P(x_k) * N$
  - Variance : $v = \frac{\sum x_k^2 h(x_k)}{\sum h(x_k)} - \overline{x}$
  - Amount of information provided by a level:$I_k = -\log_2 \frac{h(x_k)}{N}$
  - Entropy: $E = -\frac{\sum h(x_k)}{N} \log_2 \frac{h(x_k)}{N}$

- **Principle:**

$$u \in [0, L] \to v \in [0, L]$$
$$v = f(u)$$

$\longrightarrow$ New gray level values could be stored in a Look up table (LUT)

- Example: image negative function: $v = L - u$



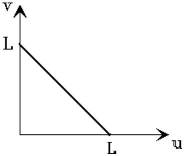
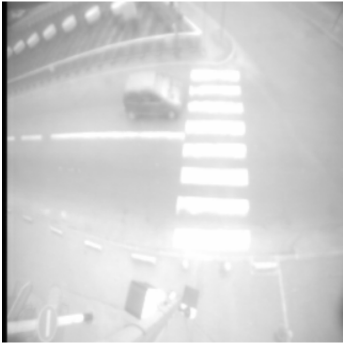
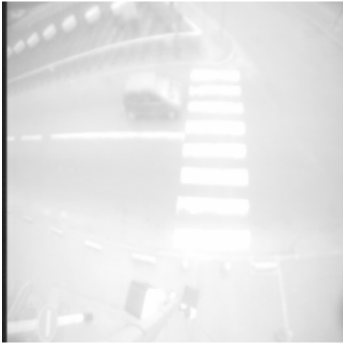


Image Enhancement
Point operation - Contrast enhancement

$$f(u) = \left\{ \begin{array}{ll} u^{1-\beta} & \text{if } \beta > 0 \\ u^{\frac{1}{1+\beta}} & \text{else} \end{array} \right._{u \in [0,1]}$$



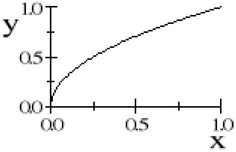


Lower contrast: $(\beta = 0.5)$

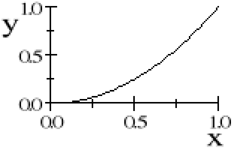
Image Enhancement

Point operation - Contrast enhancement

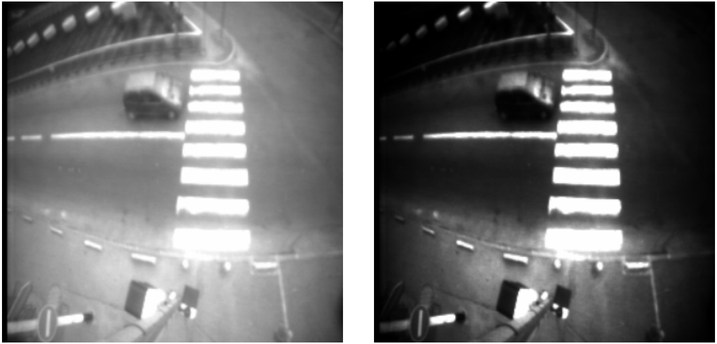
$$f(u) = \left[u^2\right] \rightarrow \beta = -0.5$$



$$f(u) = \left[u^{0.5}\right] \rightarrow \beta = 0.5$$



- **Example**

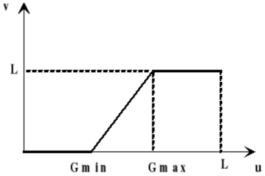


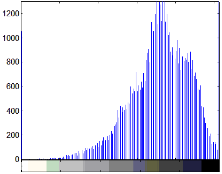
$\beta = -0.5$ and $\beta = -0.9$

Image Enhancement

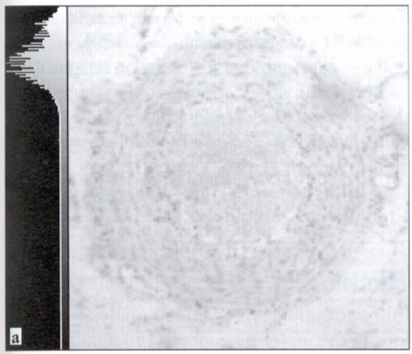
Point operation - Contrast stretching

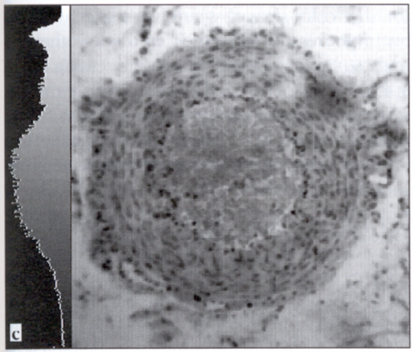
$$f(u) = L(u - G\min)/(G\max - G\min)$$





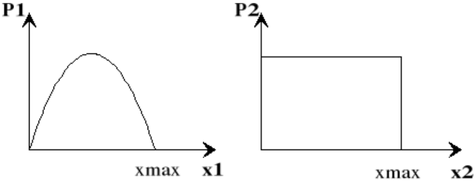
- **Example**





- **Example**

- **Objective: contrast enhancement** $\rightarrow$ constant (flat) histogram



$$f \xrightarrow{T} g$$
$$x_2 = T(x_1) \Longrightarrow dx_2 = T' dx_1$$
$$P_1(x_1) \xrightarrow{T} P_2(x_2)$$

Yet,
$\int_0^{x_{\max}} P_1 dx_1 = \int_0^{x_{\max}} P_2 dx_2 =$
$\int_0^{x_{\max}} C dx_2 = 1 \Longrightarrow [Cx]_0^{x_{\max}} = C x_{\max} = 1 \Longrightarrow C = \frac{1}{x_{\max}}$

$$P_1 dx_1 = P_2 dx_2$$

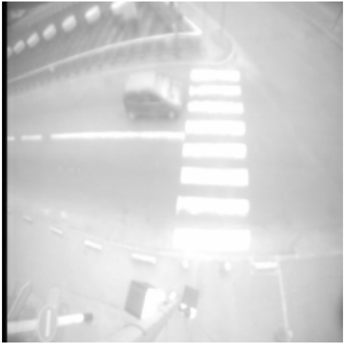
$\Longrightarrow P_2 = P_1 \frac{dx_1}{dx_2} = P_1 \frac{dx_1}{T' dx_1} = \frac{P_1}{T'} = \frac{P_1}{dT/dx_1}$.
Yat, $P_2 = C = \frac{1}{x_{\max}}$
Then

$$\frac{dT}{dx_1} = P_1 x_{\max}$$

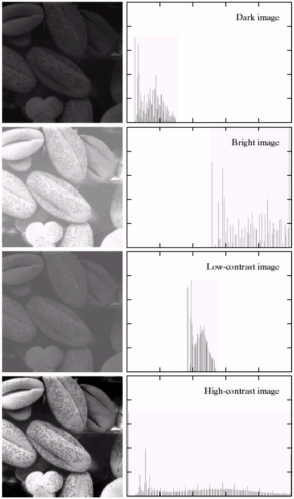
$$\Longrightarrow T = x_{\max} \int_0^{x_1} P_1(m) dm$$

Image Enhancement

Point operation - histogram equalization



Point operation - histogram equalization



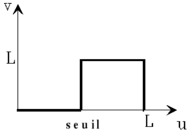
- **Objective:** change the histogram $P_r(r)$ to $P_z(z)$
- Let: $s = T(r) = (L-1)\int_0^r P_r(w)dw \rightarrow$ see histogram equalization
- Suppose now that we define: $G(z) = (L-1)\int_0^z P_z(t)dt = s$
- $G(z) = T(r) \Rightarrow z = G^{-1}\left[T(r)\right] = G^{-1}(s)$
- $T(r)$, $P_r(r)$ and $P_z(z)$ are all known or calculable.

- **Algorithm:**
  - Obtain $P_r(r)$ from the original image. Deduce s_k values (cumulative histogram).
  - Use the chosen density probability function $P_z(z)$ to obtain $G(z)$ (cumulative histogram). Store these values in a table.
  - For each s_k value, find z_q so that $G(z_q)$ be near s_k. Store in a correspondence table s_k and z_q.

- **Example:**

r_k	$P_r(r_k)$	s_k	z_q	$P_z(z_q)$	$G(z_q)$	$G(z_q) * 7$	
0	0.19	1	0	0	0	0	$\to 0$
1	0.25	3	1	0	0	0	$\to 0$
2	0.21	5	2	0	0	0	$\to 0$
3	0.16	6	3	0.15	0.15	1.05	$\to 1$
4	0.08	6	4	0.20	0.35	2.45	$\to 2$
5	0.06	7	5	0.30	0.65	4.55	$\to 5$
6	0.03	7	6	0.20	0.85	5.95	$\to 6$
7	0.02	7	7	0.15	1	7	$\to 7$

$$f(u) = \begin{cases} L \text{ if } u > T \\ 0 \text{ else} \end{cases}$$



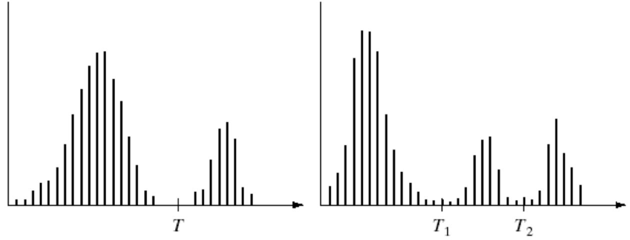
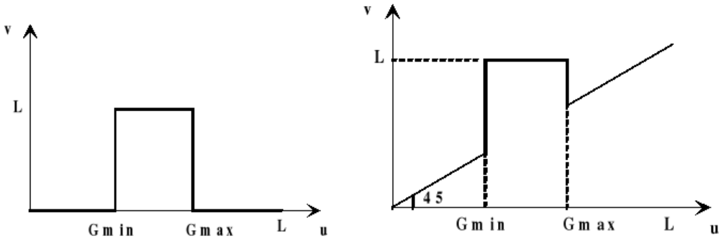


FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

The approach is well suited when the image is composed of a background and a single object (visually well separated from the background)

Image Enhancement

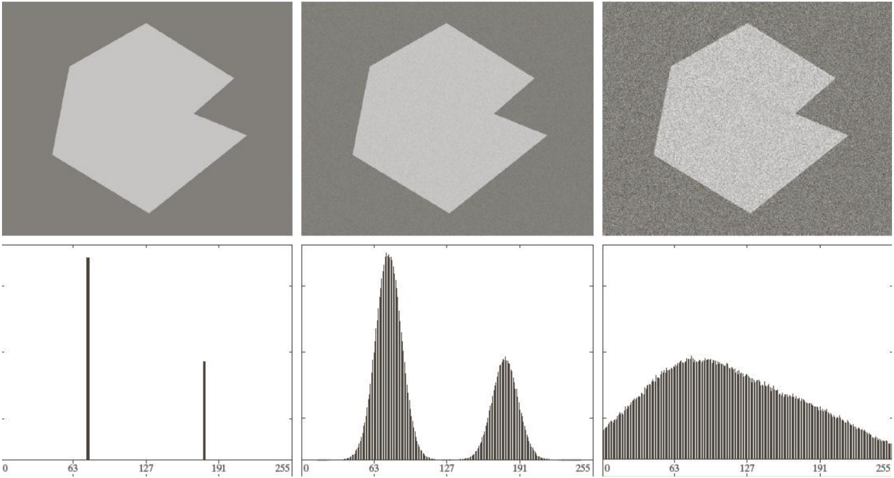
Point operation - Thresholding (others possibilities)



- Two thresholds make it possible to highlight 3 areas: the background and 2 objects
- It is possible to keep the background unchanged in order to highlight a single object.

What are the effects of noise on thresholding?

- Without noise, the histogram of an image composed of two "regions" is a couple of Dirac .
- If the image is noisy (Gaussian noise), each Dirac becomes a Gaussian but the quality of the separation between the Gaussians depends on their standard deviation.


0
63
127
191
255

Global thresholding (basic method) :

1. Choose an estimated value of the global threshold T (often the average of the gray levels of the image)
2. Segment the image using T. This segmentation produces two groups of pixels: those $> T$ and those $<= T$.
3. Calculate the averages m_1 and m_2 of the two obtained regions.
4. Calculate the new threshold: $T = \frac{1}{2}(m_1 + m_2)$
5. Repeat steps 2-3-4 until the difference between the thresholds of two successive iterations is less than a threshold δT

Image Enhancement

Point operation - Adaptive Thresholding

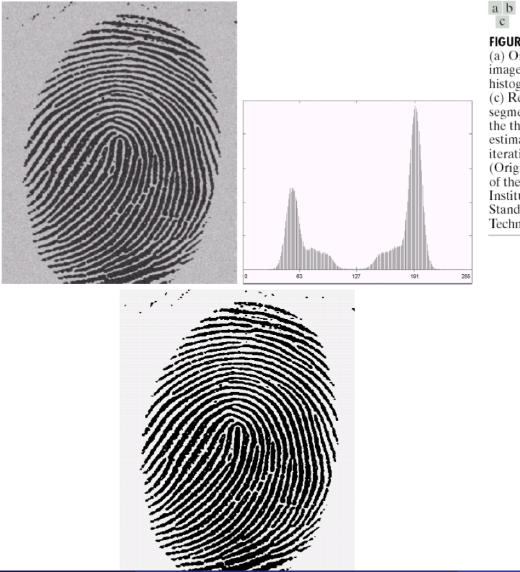


FIGURE 10.29 (a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

Principe

- Reduce the problem of thresholding to a problem of classification
- Maximize inter-class variance
- Use only the histogram of the image (1D).
- Let $0, 1, 2, ..., L-1$ be the L intensities of an image of size $M \times N$
- Let n_i be the number of pixels of intensity i. The total number of pixels is $MN = n_0 + n_1 + ... + n_{L-1}$.
- We consider the normalized histogram of components $p_i = \frac{n_i}{MN}$ with $\sum_{i=0}^{L-1} p_i = 1$

- Chose a threshold $T(k) = k, 0 < k < L-1$ that allows to seperate into two classes C_1 (pixels in $[0,k]$) and C_2 (pixels in $[k+1, L-1]$)
- $P_1(k) = \sum_{i=0}^{k} p_i$: probability for a pixel to be in C_1
- $P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$: probability for a pixel to be in C_2
- Average of pixels in C_1: $m_1(k) = \sum_{i=0}^{k} iP(i/C_1)$ where $P(i/C_1)$ is the probability of a level i knowing it belongs to C_1.
- Using Baye's reule: $P(A/B) = \frac{P(B/A)P(A)}{P(B}$, then: $m_1(k) = \sum_{i=0}^{k} iP(C_1/i)P(i)/P(C_1) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i$ because $P(C_1/i) = 1$ (we consider only values in C_1) and $P(C_1) = P_1(k)$ because it correspond to the probability of C_1. $P(i) = p_i$ is simply the ième probability value in the histogram, noted p_i

Image Enhancement

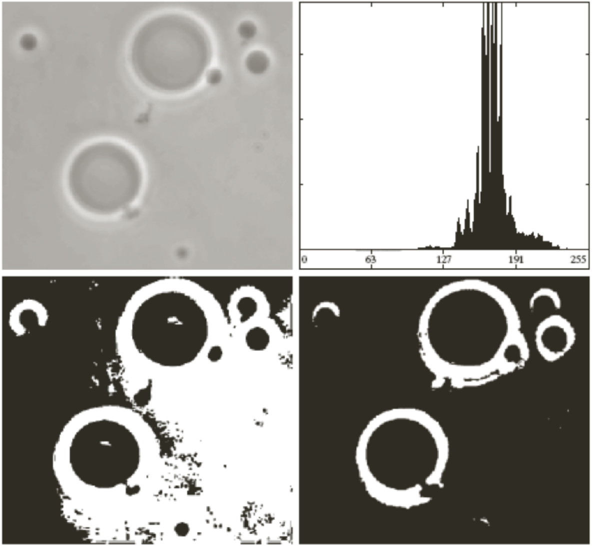
Point operation - Adaptive Thresholding: Otsu method [1979]

- In the same way: $m_2(k) = \sum_{i=k+1}^{L-1} iP(i/C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$
- Consider the average until k: $m(k) = \sum_{i=0}^{k} ip_i$ and the global average: $m_G = \sum_{i=0}^{L-1} ip_i$
- Let : $P_1 m_1 + P_2 m_2 = m_G$ with $P_1 + P_2 = 1$
- To measure the quality of a threshold at the level k, we use: $\eta = \frac{\sigma_B^2}{\sigma_G^2}$
- The global variance is: $\sigma_G^2 = \sum_0^{L-1} (i - m_G)^2 p_i$
- The inter-class variance is defined as : $\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$

- The objective is to maximize $\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$ by varying k and then retaining the one that maximizes the inter-class variance.
- $\eta(k^*)$ is a quality indicator for the value of k obtained denoted by k^*. This indicator, between 0 and 1, is close to 0 for a constant image and close to 1 for a two-mode image.

Otsu Algorithm

1. Calculate the normalized histogram of the image. Each value is called p_i
2. Calculate the cumulated values $P_1(k) = \sum_0^k p_i$ and $P_2 = 1 - P_1$
3. Calculate the cumulated averages $m_1 = \sum_0^k ip_i$ and $m_2 = \sum_{k+1}^{L-1} ip_i$
4. Calculate the global mean $m_G = \sum_0^{L-1} ip_i$
5. Calculate the inter-class variance $\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$ for all k.
6. Keep k^*, the one that maximizes the inter-class variance. If the maximum is not unique, calculate the average of all candidate k.
7. Calculate $\eta(k^*)$ to the evaluate the obtained threshold.



a b
c d

FIGURE 10.14 (a) Original image. (b) Histogram (high values were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Segmentation

- **Threholding: a first step toward segmentation...**


Exemple de segmentation

Region Definition

Labeling = Partitioning

- **Attribute (predicate)** [Horowitz 75]
  - A logical proposition whose value depends on the argument.
- **Basic segmentation attribute:**
  - The region R_i is homogeneous.

Region Definition

- **Several possible attributes:** $P(R_i) = TRUE \Longleftrightarrow$
  - ∗ Gray level values between $n - \epsilon$ and $n + \epsilon$
  - ∗ Gray level average between $\mu - \epsilon$ and $\mu + \epsilon$
  - ∗ $\max_{R_i} [f(x,y)] - \min_{R_i} [f(x,y)] < \epsilon$ $\rightarrow$ Contrast
  - ∗ $\sqrt{\frac{1}{N} \sum_{R_i} [f(x,y) - m]^2} < \epsilon$ with $N = card(R_i)$ and $m = \frac{1}{N} \sum_{R_i} f(x,y)$ $\rightarrow$ Std
  - ∗ Region R_i is different from neighbors: mean differences, inter-median distance, mean contrast, etc.

Region Definition

- **Definition of a partition:**
  - It is a set of sub-sets R_i, called regions, of the image that verify:

$$\begin{cases} \forall i,j\ R_i\ \cap R_j = \emptyset \text{ no pixel belongs to more than one region} \\ \cup_i R_i\ = image \text{ every pixel belongs to a region} \\ \forall i\ \ R_i \neq \emptyset \end{cases}$$

  - There are a very large number of partitions that verify the predicate. Criteria for choosing a partition:
    - Minimize the cardinal of the partition
    - Minimize the size of the smallest region
    - Maximize a distance between regions
  - Often, we are looking for a partition such as:
    - $\forall i, j$, i and j adjacent $\Rightarrow P(R_i\ \cup R_j) = FALSE$ (Coherence of region attributes)
    - p **connected** with $p\prime\ \forall\ p, p' \in\ R_i$.

Region segmentation

- **Several categories:**
  - Region merging
  - Region growing
  - Region splitting
  - Split & Merge

- **Principle:**
  - Explore image starting from small regions
  - Regions grow if they satisfy a homogeneity criterion

Region segmentation

Region merging

- **Blob coloring**
  - Browsing the image with a 4-connectivity operator or 8-connectivity operator whose shape depends on the direction of the scal.

	b
c	a

ou

b	c	d
e	a	

  - Labeling when entering a new region and propagation of the label in the neighborhood when the criteria of similarity is true between neighbors.

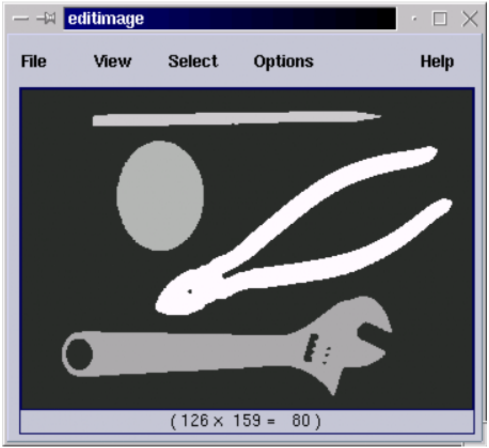
- **Blob coloring (binary case)**
  - 1st pass:

```
CurrentLabel = 0
For pixel a:
IF a≠0 THEN
    IF b≠0 AND c==0 THEN a=b
    ELSE IF b==0 ET c≠0 THEN a=c
    ELSE IF b≠0 AND c≠0 AND b==c THEN a=c
    ELSE IF b≠0 AND c≠0 AND b≠c THEN tab_eq[b]=c
    ELSE      a = CurrentLabel
                CurrentLabel = CurrentLabel+1
    END IF
END IF
```

  - 2nd Pass:

    Equivalence updates.

- **Example**



- **Blob coloring with gray level images**
  - Propagation of the label under condition of similarity.
  - Process all points without distinction of object/background.

- **Principle:**
  - We start from a seed point and extend it by adding points which satisfy the criterion of homogeneity
  - The seed point can be chosen either manually or automatically, avoiding areas of strong contrast (gradients) $\Rightarrow$ boosting methods.

- **Principle:**

Region segmentation

Region growing - algorithm

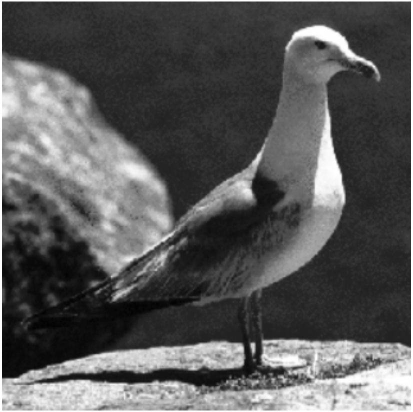
```
void GrowingMain(double *in_image, double *out_image, int w, int h, int sx, int sy, double
thresh) {
 int x,y;
 double in_val, out_val;
 double nb, sum;
 sum = nb = 0.;
     /* Mean*/
     for (x=sx-5;x¡=sx+5;x++)
     for (y=sy-5;y¡==sy+5;y++)
     if ((x¿0) && (x¡h) && (y¿0) && (y¡w)) {
           nb++;
           sum+= in_image[PIXEL(x,y)];
     }
     Grow(in_image,out_image,w,h,sx,sy,thresh,sum,nb);
}
```

Segmentation en régions

Approches par croissance - algorithme

```
void Grow(double *in_image,double *out_image,int w, int h,int x, int y, double thresh, double
sum, double nb) {
 double diff, mean;
 if (out_image[PIXEL(x,y)]==0.)  {
 mean = sum / nb;
 diff = abs(in_image[PIXEL(x,y)] - mean);
 if (diff
< thresh)  {
     sum += in_image[PIXEL(x,y)];
     nb++;
    out_image[PIXEL(x,y)] = 1.;
    if (x>0) Grow(in_image,out_image,w,h,x-1,y,thresh,sum,nb);
    if (y>0) Grow(in_image,out_image,w,h,x,y-1,thresh,sum,nb);
    if (x<h) Grow(in_image,out_image,w,h,x+1,y,thresh,sum,nb);
    if (y<w) Grow(in_image,out_image,w,h,x,y+1,thresh,sum,nb);
  }
```

- **Example**



- **How to choose the seed**
  - Randomly
  - By Histogram analysis
  - Edges then filling $\rightarrow$ center of mass, skeleton, etc.

Region Segmentation

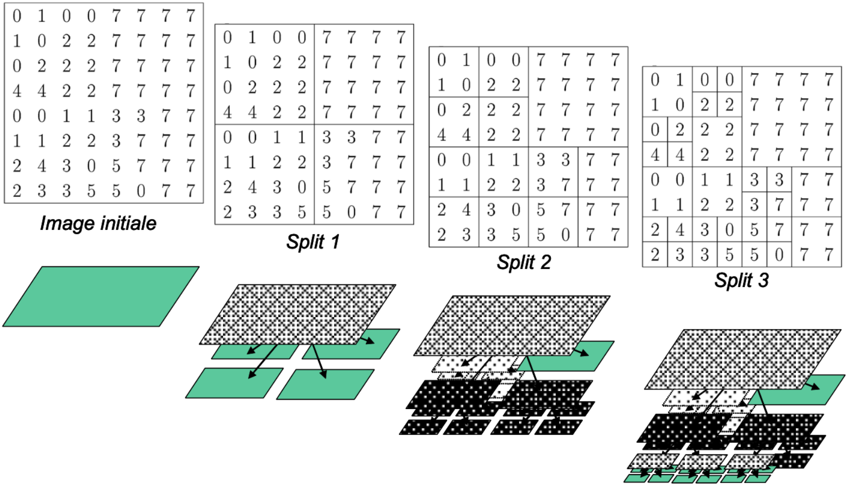
- **Constat:**
  - Segmentation by division provides a hierarchical structure which allows to establish close relationships between regions, but who can split the same region into several distinct sets.
  - Merge segmentation produces a minimum number of connected regions, but provides these in a horizontal structure that does not indicate no close relationship.
- **Proposition:**
  - Collect, from the coarse division obtained by division, the different adjacent blocks of the image
    - Division and collection algorithm, also called algorithm "Split and Merge"

- **Principle of split approaches**
  - Definition of an homogeneity criterion
  - Test of the validity of the criterion on the image
  - If the criterion is valid, the image is segmented $\rightarrow$ the algorithm stops
  - Otherwise, the image is split into smaller areas and the method is reapplied on each of the zones

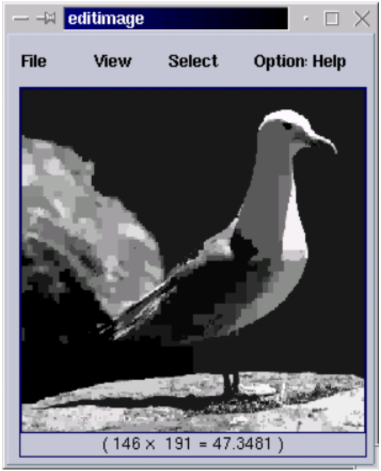
Region Segmentation

Region split - example

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7



- **Example**



- Merge:
  - Each node of the Region Adjacency Graph is examined.
  - If one of the neighbors of this node is at a lower distance at a merge threshold, the two nodes merge into the RAG.
  - When no more node can merge with one of its neighbors, STOP.