



# Segmentation d'images

Caroline Petitjean

# Définitions

## Définition de la segmentation

- Ensemble de zones homogènes dans l'image (homogénéité de texture, de couleur, de mouvement...).
- Partition de l'image  $I$  en sous-ensembles disjoints non-vides  $R_i$  pour  $i = 1, 2 \dots M$  appelées régions contenant des pixels connexes au sens d'un prédicat et tels que  $I = \bigcup_{i=1}^M R_i$
- Description en objets : simplification de l'image.

## Région

- Ensemble de pixels topologiquement connexes et ayant des attributs similaires : niveaux de gris, couleur, texture, mouvement.

## Remarques

- Dualité contours/régions : intérieur d'un contour fermé = région
- Segmentation des contours  $\Rightarrow$  voir cours sur les contours
- Attributs utilisés pour la segmentation très variés (niveaux de gris, couleur, texture, mouvement)  $\Rightarrow$  dans ce cours les méthodes utilisent les niveaux de gris, mais peuvent être étendues à d'autres attributs.

# Segmentation



A quoi sert la segmentation en régions ?

Important prérequis pour les étapes de mesure, de compréhension de la scène :

- mesure d'objets
- reconnaissance d'objets
- indexation : rechercher dans une base d'images, les images "ressemblantes" à une image initiale
- compression
- recalage d'images, mises en correspondance

# Segmentation

Quelques exemples d'applications

Exemple de Segmentation de visages

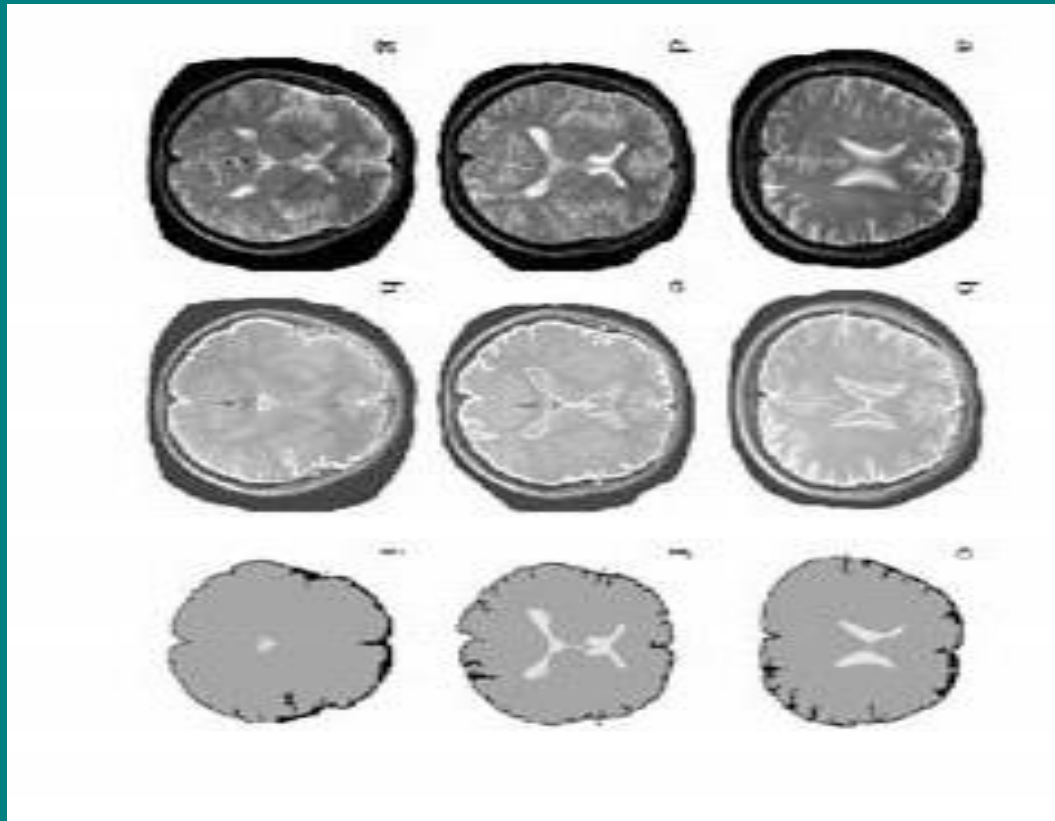
Applications en indexation, biométrie.



Source : LIRIS Laboratoire d'InfoRmatique en Images et Systèmes d'information  
UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon  
2/Ecole Centrale de Lyon

# Segmentation

Quelques exemples d'applications - Imagerie médicale IRM  
Etude des maladies neurologiques qui altèrent la distribution entre le parenchyme du cerveau (matière grise) et liquide céphalo-rachidien.

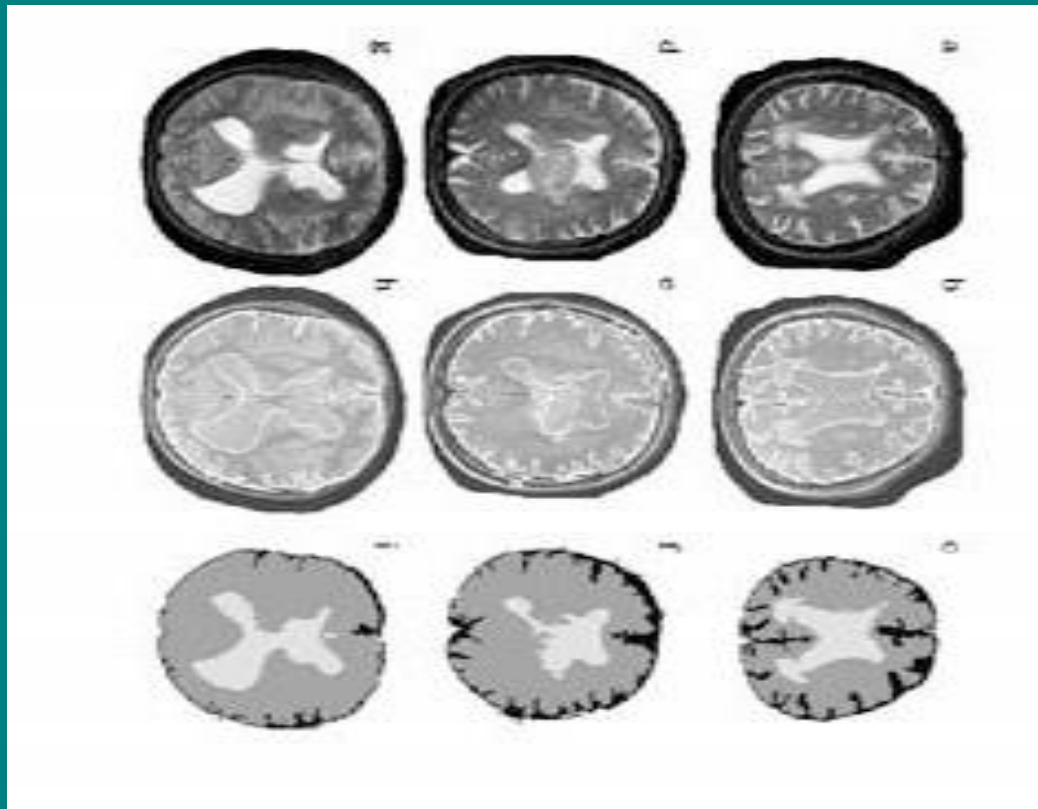


*"Segmentation of Brain Parenchyma and Cerebrospinal Fluid in Multispectral Magnetic Resonance*  
(Arvid Lundervold and Geir Storvik)

Published in IEEE Transactions on Medical Imaging, Vol. 14, No. 2, June 1995, pp. 339-349.

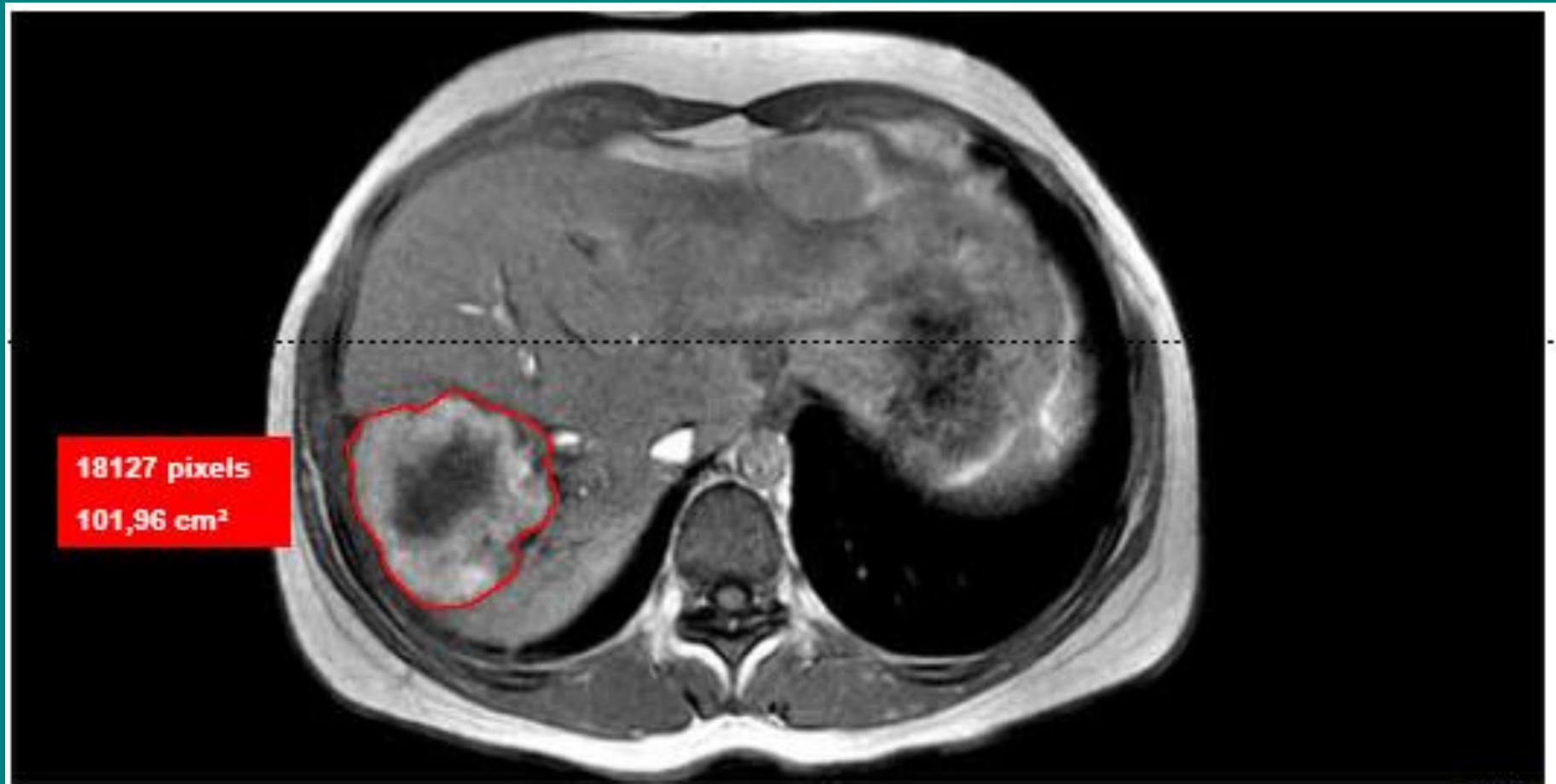
# Segmentation

Quelques exemples d'applications - Imagerie médicale IRM : cas pathogènes  
Etude des maladies neurologiques qui altèrent la distribution entre le parenchyme du cerveau (matière grise) et liquide céphalo-rachidien.



▪ *Segmentation of Brain Parenchyma and Cerebrospinal Fluid in Multispectral Magnetic Resonance*  
(Arvid Lundervold and Geir Storvik)  
Published in IEEE Transactions on Medical Imaging, Vol. 14, No. 2, June 1995, pp. 339-349.

# Segmentation d'image



GEMS

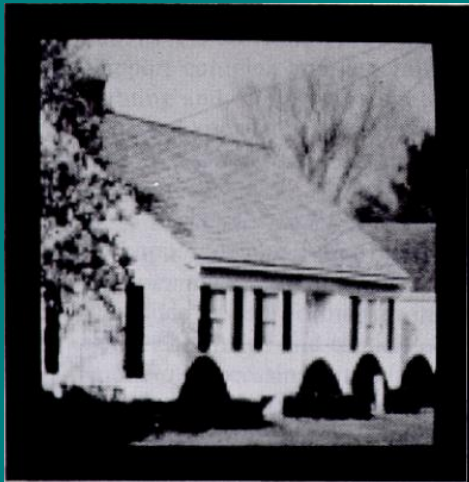
Image IRM d'une tumeur du foie



# Segmentation



Recherche de frontières  
(approches « contours »)



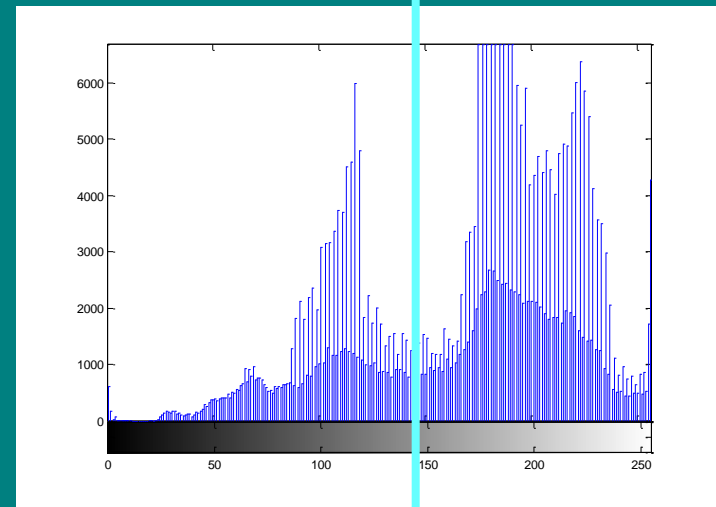
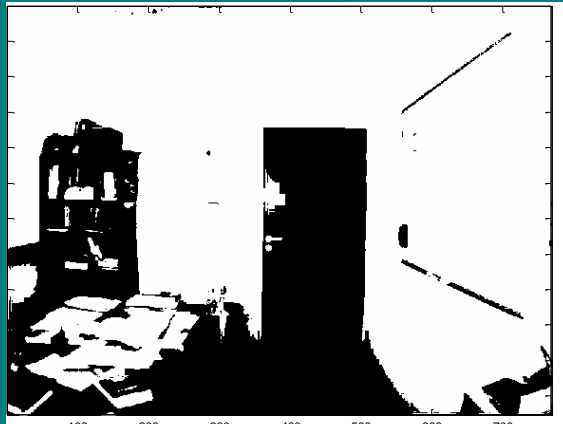
Recherche de régions  
(approches « régions »)





# Seuillage

- Seuillage (🇬🇧 thresholding)



**$T = 150$**

# Opérations entre images

Images aérienne de la ville de Cincinnati



# Opérations entre images

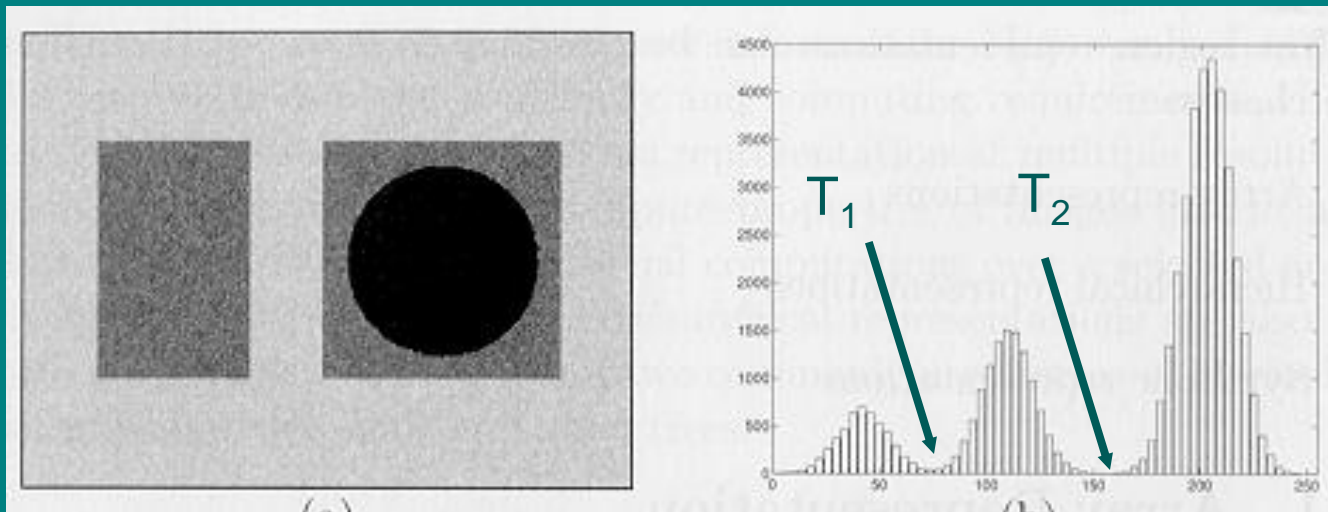
Images aérienne de la ville de Cincinnati



# Thresholding Using Image Histogram

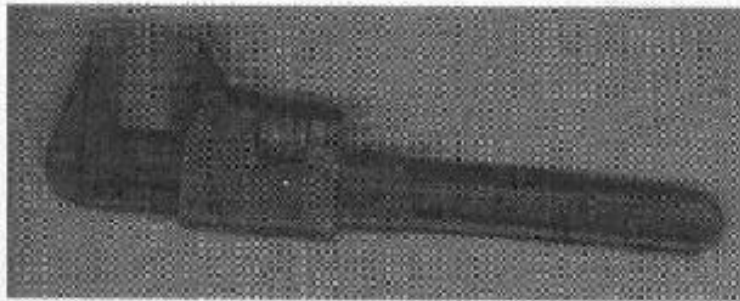
- Multiple thresholds are possible

If  $f(x, y) < T_1$  then  $f(x, y) = 255$   
else if  $T_1 < f(x, y) < T_2$  then  $f(x, y) = 128$   
else  $f(x, y) = 0$

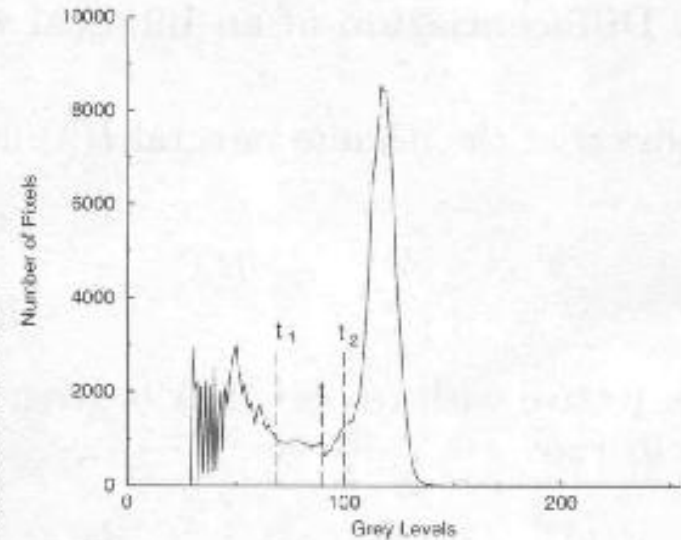


# Seuillage

- Variante (qui prend en compte la position spatiale des pixels) : seuillage par hystérésis



(a) Original image



(b) Histogram of (a)

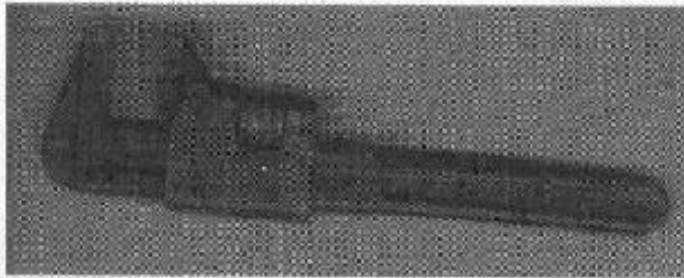
# Seuillage par hystérésis

- On introduit maintenant 2 seuil : un seuil haut et un seuil bas.
  - ✓ Si  $\text{norme} > \text{seuil haut} \rightarrow$  contour sur 1
  - ✓ Si  $\text{norme} < \text{seuil bas} \rightarrow$  pas de contour 0
  - ✓ Si  $\text{seuil bas} < \text{norme} < \text{seuil haut} \rightarrow$  contour de fermeture 2

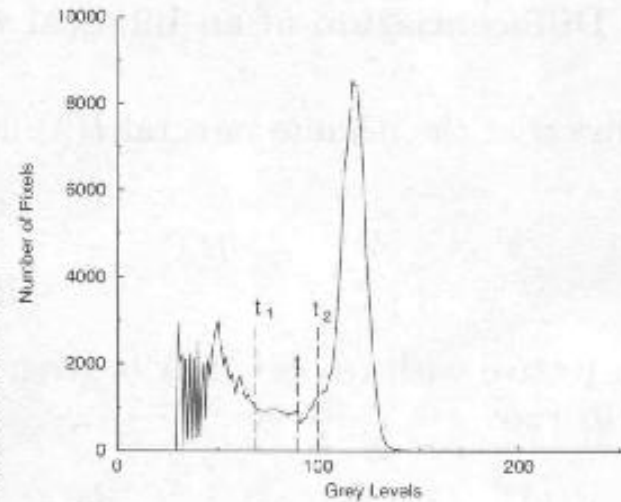
Les contours hypothétiques de fermeture sont transformés en contours sûrs s'ils sont adjacents à un contour déjà codé à 1.



# Seuillage par hystérésis



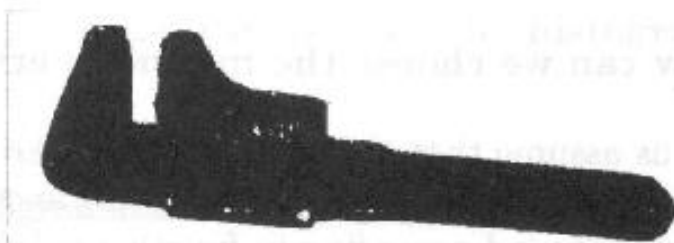
(a) Original image



(b) Histogram of (a)



single threshold

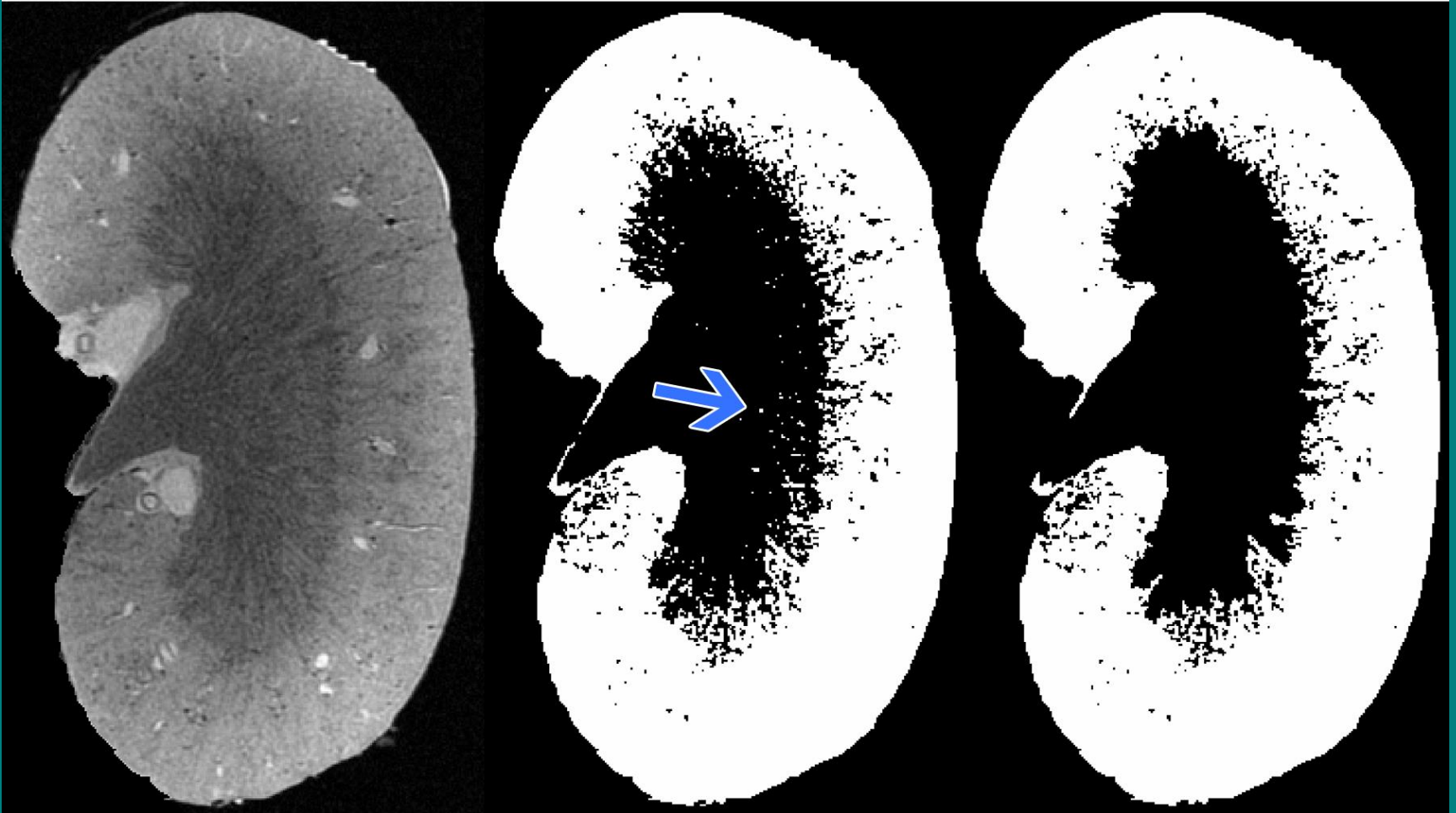


hysteresis thresholding

original

*simple  
thresholding*

*hysteresis*



# Seuillage

- Comment faire pour choisir le seuil ?
- Une méthode pour le choisir **automatiquement** : **méthode d'Otsu** (1979)
  - Choix du seuil qui minimise la **variance** intrarégions et maximise la **variance** interrégions

# Otsu's Method : Mean and Variance

- On fixe la valeur du seuil  $T$
- *L'image est séparée en 2 parties*
  - *Object vs Background*

$$q_b(T) = \sum_{i=1}^T P(i),$$

*Proportion de pixels background*

$$q_o(T) = \sum_{i=T+1}^L P(i)$$

*Proportion de pixels objet*

background ←  $T$  → object



$P(i)$  : proba ou fréquence  
normalisée du pixel  $i$

$$(q_b(T) + q_o(T) = 1)$$

# Otsu's Method : Mean and Variance

- Calcul de la variance du groupe *background* et du groupe *object*

$$\sigma_b^2(T) = \frac{\sum_{i=1}^T (i - \mu_b)^2 P(i)}{\sum_{i=1}^T P(i)} = \frac{1}{q_b(T)} \sum_{i=1}^T (i - \mu_b)^2 P(i)$$

$$\sigma_o^2(T) = \frac{\sum_{i=T+1}^L (i - \mu_o)^2 P(i)}{\sum_{i=T+1}^L P(i)} = \frac{1}{q_o(T)} \sum_{i=T+1}^L (i - \mu_o)^2 P(i)$$

# Otsu's Method

## Within-class and between-class variance

- On définit la variance intra-classe :

$$\sigma_W^2(T) = q_b(T)\sigma_b^2(T) + q_o(T)\sigma_o^2(T)$$

within-class variance

should be minimized!

- On définit la variance inter-classe :

$$\sigma_B^2(T) = q_b(T)(\mu_b(T) - \mu)^2 + q_o(T)(\mu_o(T) - \mu)^2$$

between-class variance

should be maximized!



# Otsu's Method

## Within-class and between-class variance

- On montre que la variance de l'image peut s'écrire :

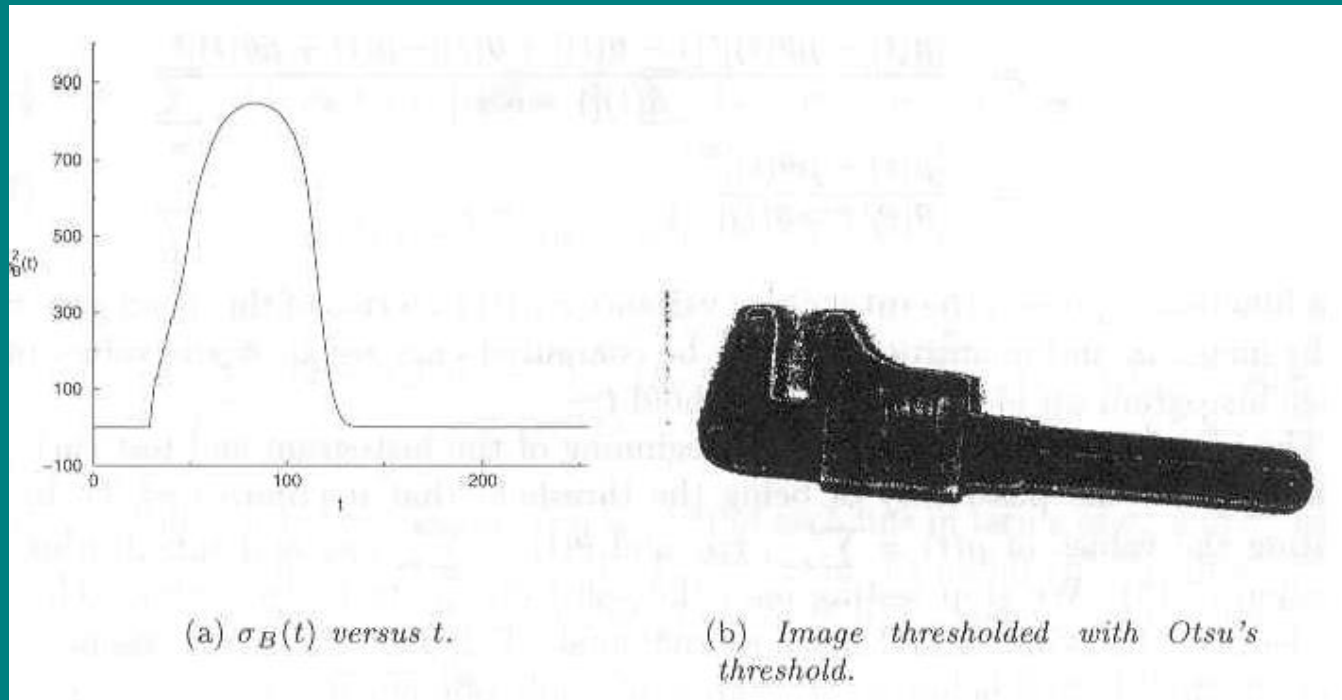
$$\sigma^2 = \underbrace{q_b(T)\sigma_b^2(T) + q_o(T)\sigma_o^2(T)}_{\sigma_W^2(T)} + \underbrace{q_b(T)(\mu_b(T) - \mu)^2 + q_o(T)(\mu_o(T) - \mu)^2}_{\sigma_B^2(T)} =$$

- *La variance de l'image est constante (indépendante de  $T$ )*
- *Minimiser la variance intra  $\sigma_W^2$*
- $\leftrightarrow$  *Maximiser la variance inter  $\sigma_B^2$*

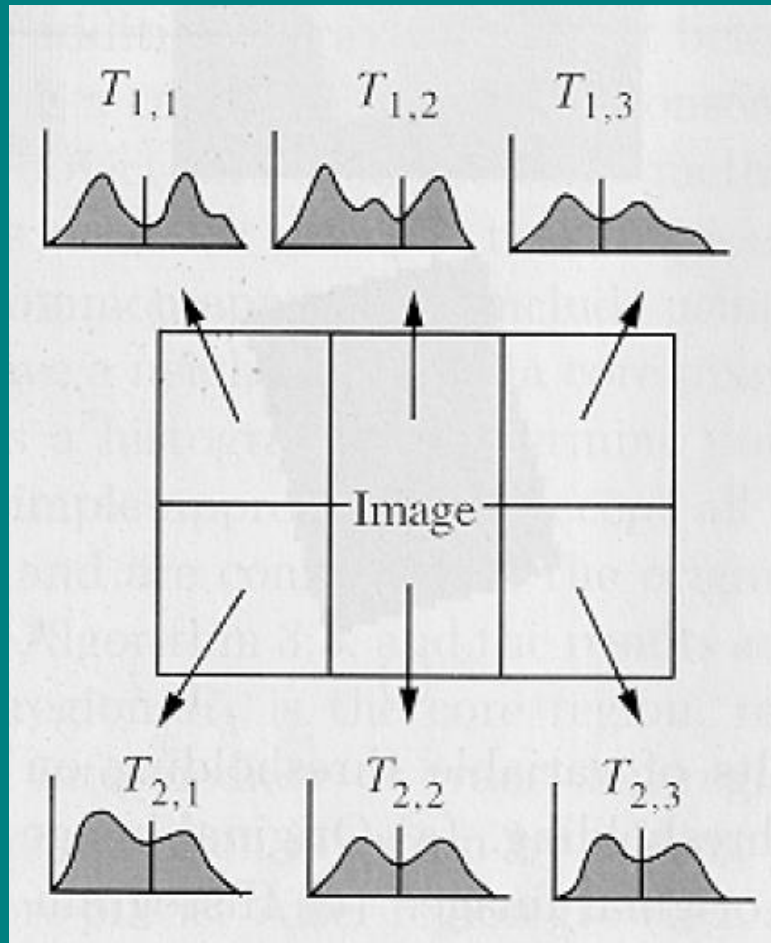
# Otsu's Method

## Determining the threshold

- Start from the beginning of the histogram and test each gray-level value for the possibility of being the threshold  $T$  that maximizes  $\sigma_B^2$

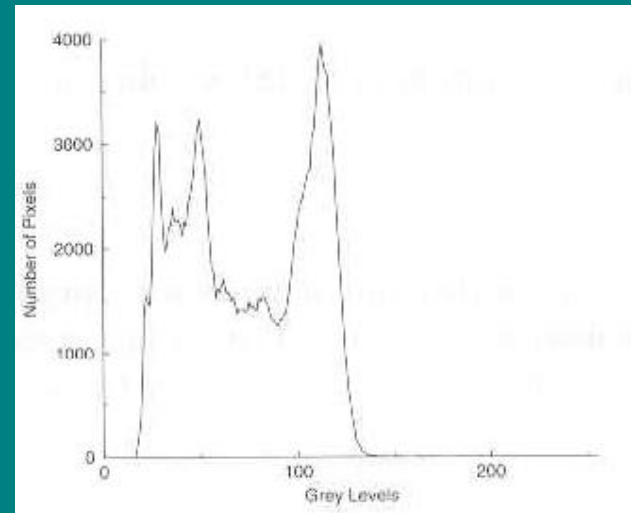


# Seuillage local



This approach might lead to subimages having simpler histogram (e.g., bimodal)

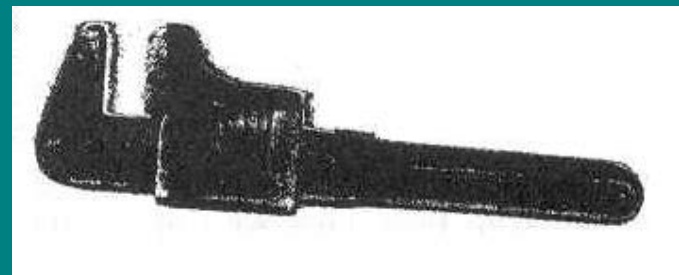
# Seuillage local



single threshold

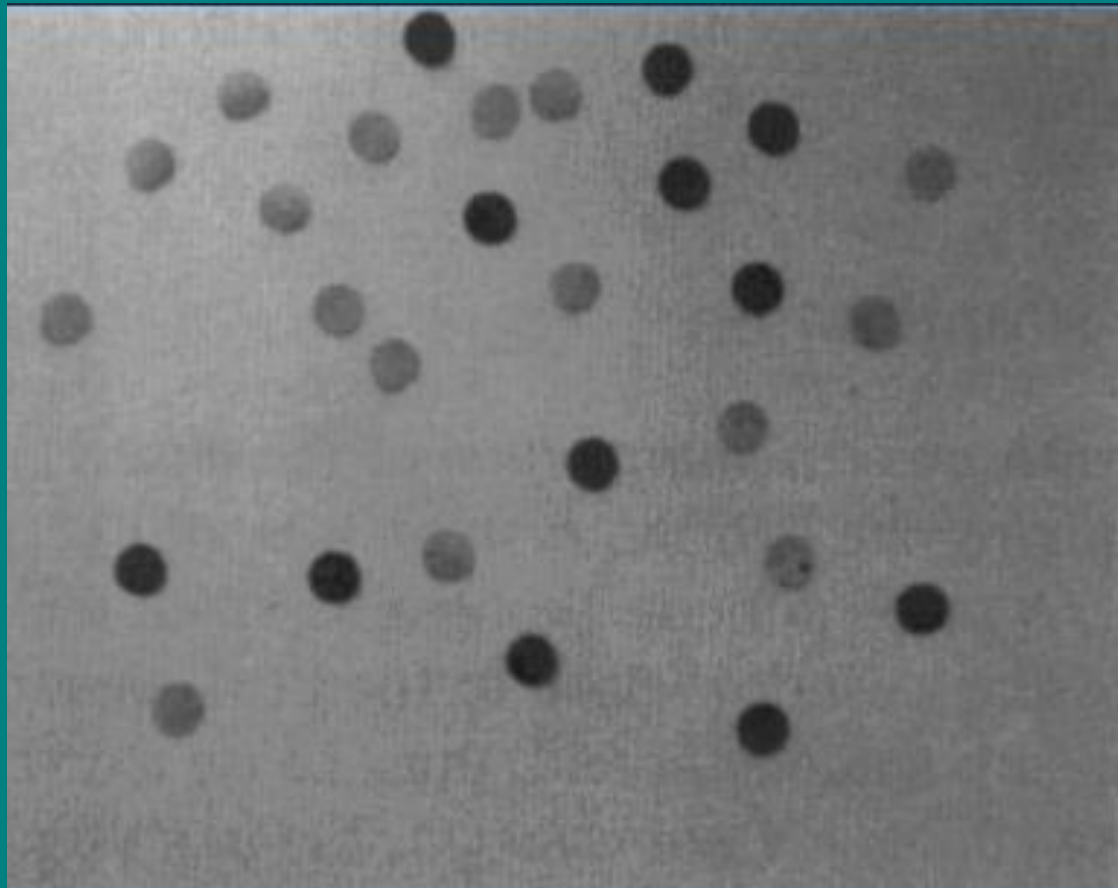


local thresholding using Otsu's method

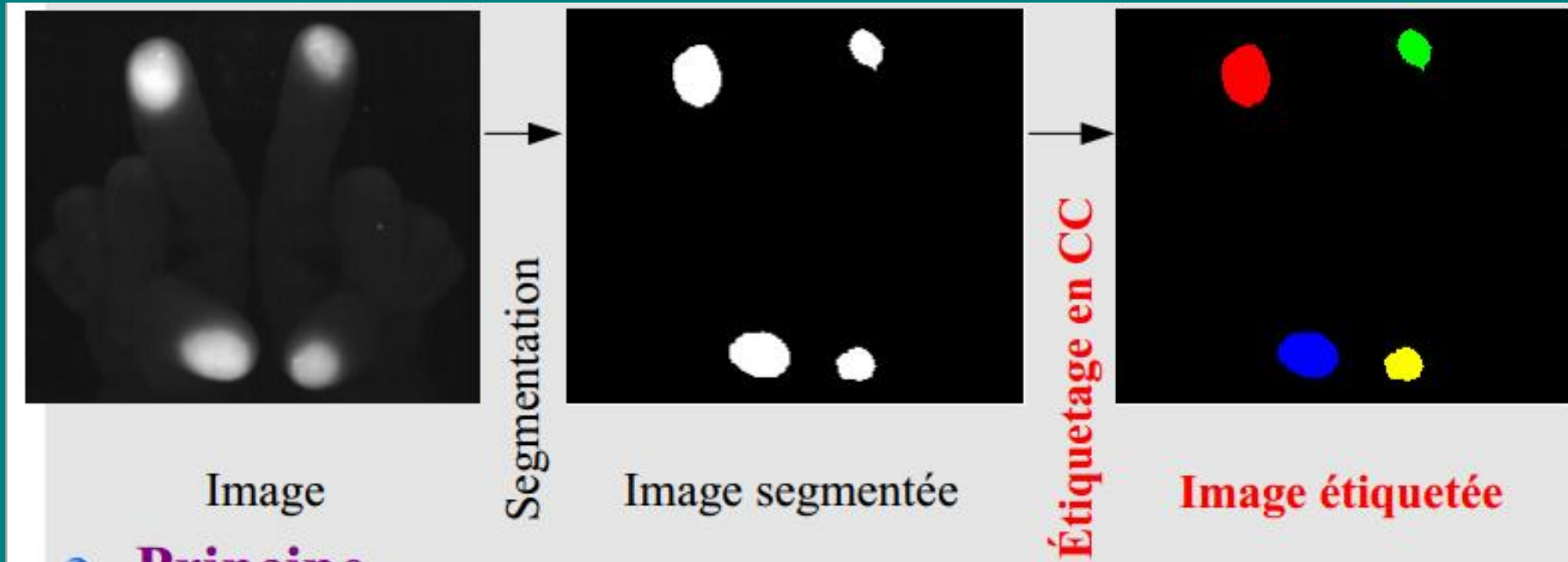


# Seuillage

- Comment compter le nombre d'éléments ?



# Etiquettage en composantes connexes

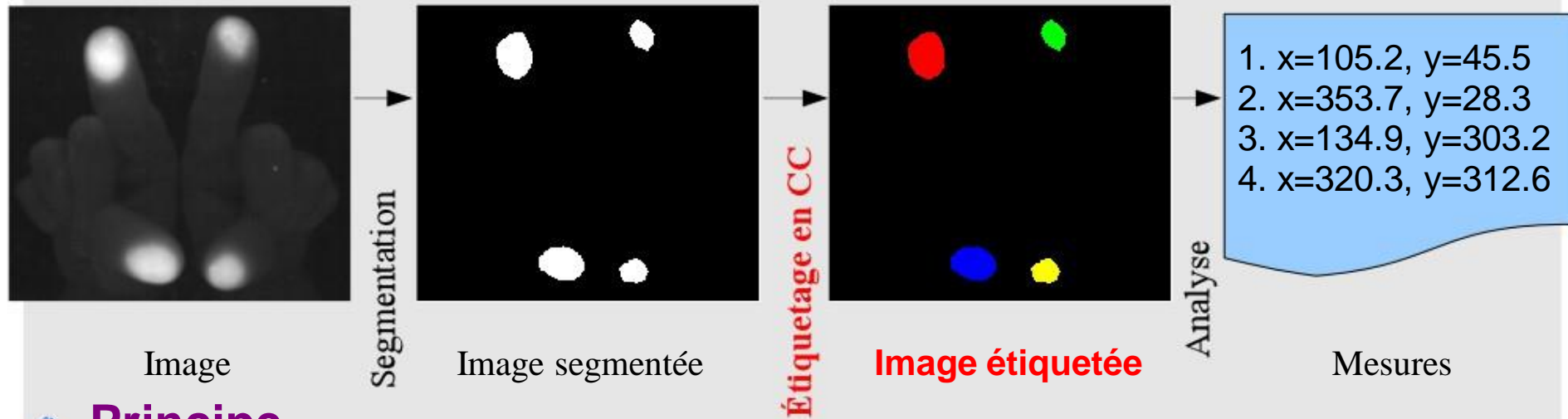


- source : cours Master informatique Univ Lille 1



# Présentation générale (1/3)

## Cadre



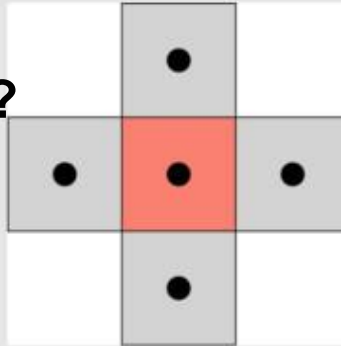
## Principe

- Partant d'une image binaire, trouver les groupes de pixels connectés, appelés composantes connexes (*connected components*) ou *blobs*
- On obtient une image dans laquelle chaque « objet » est identifié
- Cette opération s'appelle **analyse** (ou **étiquetage**) **en composantes connexes** (*connected-component analysis / labeling*, ou encore *blob extraction*)

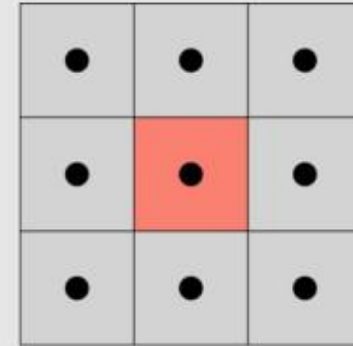
## Présentation générale (2/3)

## Connexité

→ Pixels « connectés » ?



4-connectité

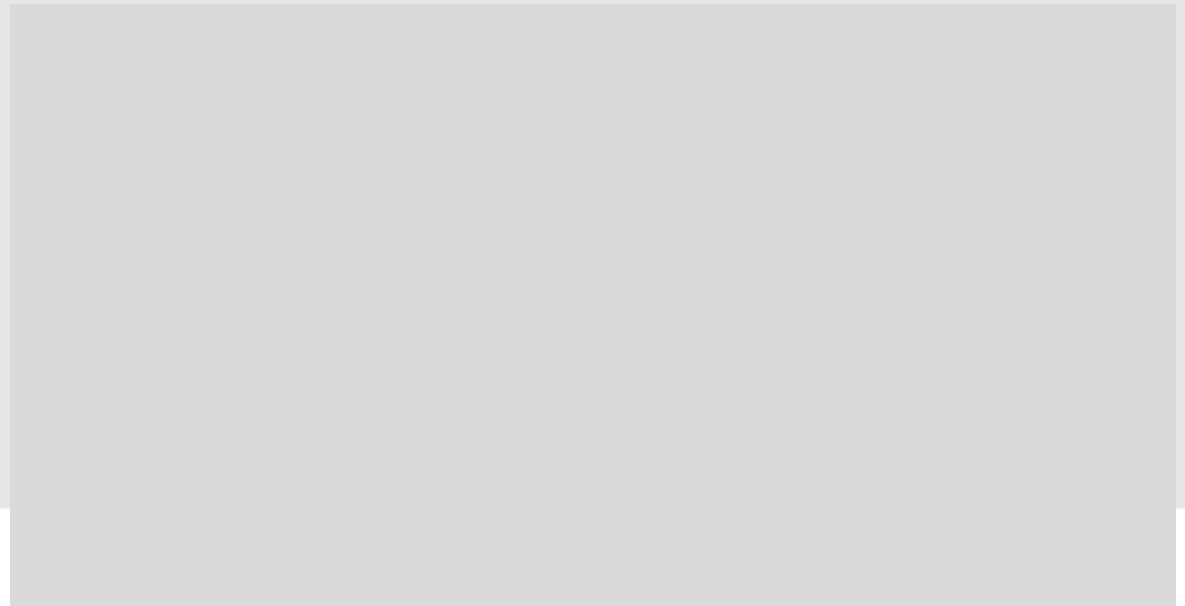


8-connectité

→ Exemples de composantes connexes

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

Image binaire



## Présentation générale (3/3)

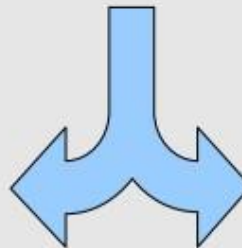
## Étiquetage

- Chaque composante connexe est identifiée de manière unique par une **étiquette (label)**

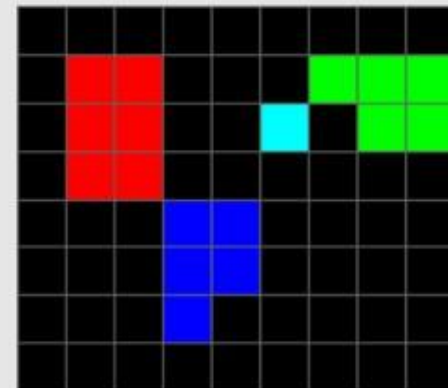
0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	1
0	1	1	0	0	1	0	1	1
0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0

- Étiquettes numériques (entiers)

0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	2	2	2
0	1	1	0	0	3	0	2	2
0	1	1	0	0	0	0	0	0
0	0	0	4	4	0	0	0	0
0	0	0	4	4	0	0	0	0
0	0	0	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0



- Étiquette de couleurs



## Approche par remplissage (1/4)

### • Principe de l'approche par remplissage (*flood fill*)

#### → E/S

- Paramètre d'entrée : image binaire **B**
- Résultat de sortie : matrice (ou *carte*) d'étiquettes **L**

#### → Algorithme

- Créer une matrice d'étiquettes **L**, de même taille que **B**, initialisées à 0
- Initialiser le compteur d'étiquettes *nbLabels* à 0
- Tant qu'il y a des pixels à 1 dans **B**
  - Trouver le prochain pixel  $P(x,y)$  à 1 dans **B**
  - Incrémenter *nbLabels* et donner à  $L(x,y)$  la valeur *nbLabels*
  - Mettre  $B(x,y)$  à 0
  - Traiter de la même manière les pixels 4-connexes avec  $P$  et à 1 dans **B**, jusqu'à ce qu'il n'y en ait plus aucun

# Approche par remplissage (2/4)

## Exemple (1/3)

- Initialisation  
 $nbLabels=0$

- Étape 1  
 $nbLabels=1$   
mettre à  $nbLabels$  l'élément  
de **L** correspondant au  
premier pixel à 1 dans **B**

- Étape 2  
mettre à 0 le pixel qui vient  
d'être étiqueté et trouver ses  
voisins 4-connexes qui sont à 1

- Étape 3  
idem avec les nouveaux pixels

**B**

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

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

**L**

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

# Approche par remplissage (3/4)

## Exemple (2/3)

- Étape 4  
idem avec les nouveaux pixels  
(il n'y a plus de voisin ici)

**B**

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

- Étape 5  
trouver un autre pixel à 1 dans **B**  
*nbLabels=2*  
mettre à *nbLabels* l'élément  
de **L** correspondant au pixel

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

- Étape 6  
mettre à 0 le pixel qui vient  
d'être étiqueté et trouver ses  
voisins 4-connexes qui sont à 1

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	0	0
0	0	0	0	0	0	0

- Étape 7  
idem avec les nouveaux pixels

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

**L**

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	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	1	1	0	0	0	0
0	1	0	2	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	1	1	0	0	0	0
0	1	0	2	2	0	0
0	1	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	0	2	2	2	0
0	1	0	0	2	0	0
0	0	0	0	0	0	0



# Approche par remplissage (4/4)

## Exemple (3/3)

- Étape 8  
idem avec les nouveaux pixels  
(il n'y a plus de voisin ici)

**B**

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

**L**

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	0	2	2	2	0
0	1	0	0	2	0	0
0	0	0	0	0	0	0

- Fin (tous les pixels de **B** sont à 0)

## Approche par double parcours (1/7)

### • Principe de l'approche par double parcours (*two-pass*)

#### → E/S

- Paramètre d'entrée : image binaire **B**
- Résultat de sortie : matrice (ou *carte*) d'étiquettes **L**

#### → Algorithme

- Premier parcours de l'image, dans le sens classique (*raster-scan order*) :  
À chaque pixel à 1 dans **B**, on affecte
  - la plus petite étiquette parmi celles de ses voisins **haut** et **gauche**  
*ou*
  - une nouvelle étiquette si aucun de ces 2 voisins n'est encore étiqueté
- Second parcours de l'image, dans le sens inverse :  
À chaque pixel précédemment étiqueté, on affecte la plus petite étiquette parmi la sienne et celles de ses voisins **bas** et **droite**.

# Approche par double parcours (2/7)

## Exemple-1<sup>er</sup> parcours (1/3)

### Initialisation

$nbLabels=0$

### Étape 1

les voisins haut et gauche du premier pixel à 1 dans **B** ne sont pas encore étiquetés

- nouvelle étiquette ( $nbLabels=1$ )

### Étape 2

le voisin gauche du pixel suivant à 1 dans **B** est déjà étiqueté à 1 • affecter cette même étiquette au pixel

### Étape 3

nouvelle étiquette ( $nbLabels=2$ )

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

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

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

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

L

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

0	0	0	0	0	0	0
0	0	1	1	0	2	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

# Approche par double parcours (3/7)

## Exemple-1<sup>er</sup> parcours (2/3)

- Étape 4  
affecter au pixel l'étiquette  
de son voisin gauche
- Étape 5  
nouvelle étiquette (*nbLabels*=3)
- Étape 6  
les voisins haut et gauche du  
pixel suivant à 1 dans **B** sont  
étiquetés différemment • affecter  
au pixel l'étiquette minimale
- Étape 7  
affecter au pixel l'étiquette  
de son voisin haut

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

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

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

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

L

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	0	0	0	0	0	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	2	2
0	3	0	0	0	0	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	2	2
0	3	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	1	1	0	2	2
0	3	1	0	0	0	2
0	0	0	0	0	0	0
0	0	0	0	0	0	0

# Approche par double parcours (4/7)

## Exemple-1<sup>er</sup> parcours (3/3)

Étape 8  
affecter au pixel l'étiquette  
de son voisin haut

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

Étape 9  
nouvelle étiquette (*nbLabels*=4)

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

Étape 10  
affecter au pixel l'étiquette  
de son voisin gauche

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

Étape 11  
les voisins haut et gauche du  
pixel suivant à 1 dans **B** sont  
étiquetés différemment • affecter  
au pixel l'étiquette minimale

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

L

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	0	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	0	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	4	0
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	4	2
0	0	0	0	0	0	0



# Approche par double parcours (5/7)

## Exemple-2<sup>ème</sup> parcours (1/2)

Image et carte initiales

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

L

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	4	2
0	0	0	0	0	0	0

1<sup>ère</sup> *étiquette modifiée*

Affecter au pixel l'étiquette de son voisin droit car elle est inférieure à la sienne

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

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	2	2
0	0	0	0	0	0	0

2<sup>ème</sup> *étiquette modifiée*

Affecter au pixel l'étiquette de son voisin droit car elle est inférieure à la sienne

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

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	2	2	2
0	0	0	0	0	0	0

(*étiquette non modifiée*)

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

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	2	2	2
0	0	0	0	0	0	0

## Approche par double parcours (6/7)

### Exemple-2<sup>ème</sup> parcours (2/2)

(étiquette non modifiée)

3<sup>ème</sup> étiquette modifiée

Affecter au pixel l'étiquette de son voisin droit car elle est inférieure à la sienne et à celle de son voisin bas

Carte à l'issue du 2<sup>ème</sup> parcours  
Aucune autre étiquette n'est plus modifiée

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

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

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

L

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	2	2	2
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	1	1	0	0	0	2
0	3	0	0	2	2	2
0	0	0	0	0	0	0

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	1	1	0	0	0	2
0	3	0	0	2	2	2
0	0	0	0	0	0	0

## Approche par double parcours (7/7)

### Utilisation d'une table d'équivalences

#### → Principe

- Lors du premier parcours, si 2 voisins portent des étiquettes différentes  $l_i$  et  $l_j$ , choisir l'une d'entre elles pour le pixel et mémoriser l'équivalence  $l_i \equiv l_j$
- Lors du second parcours, réétiqueter les pixels selon la table d'équivalences (en renumérotant éventuellement les étiquettes pour qu'elles soient consécutives)

#### → Exemple

- Étape 6 du 1<sup>er</sup> parcours  
mémoriser l'équivalence  $1 \equiv 3$
- Étape 11 du 1<sup>er</sup> parcours  
mémoriser l'équivalence  $2 \equiv 4$

**B**

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

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

**L**

0	0	0	0	0	0	0
0	0	1	1	0	2	2
0	3	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	1	1	0	2	2
0	3	1	0	0	0	2
0	3	0	0	4	4	2
0	0	0	0	0	0	0

***Le second parcours suffit ensuite à obtenir la carte finale***



# Références

## Sources des images

- Introduction to Digital Image Processing, Image Segmentation, Zhou Wang, Dept. of Electrical Engineering, the Univ. of Texas at Arlington, 2006
- Segmentation d'images, Michèle Gouiffès
- Cours de Vision artificielle, Christine Fernandez-Maloigne, Université de Poitiers
- **CS474/674 – Prof. Bebis**