

Master Mathematical Analysis and Applications
Course M1 - S2

Computer vision

Segmentation and contours detection

Week 2-4

Mohammed Hachama

hachamam@gmail.com

<http://hachama.github.io/home/>

University of Khemis Miliana

-2020-

Plan

1. Introduction
2. Détection de contours
3. Segmentation de régions

Qu'est ce que la segmentation ?

- Partitionnement d'une image en un nombre minimal de régions R_i connexes (en un sens à préciser)
- Chaque région doit satisfaire une propriété $P(R_i)$



Dualité contour/région

Qu'est ce que la segmentation ?

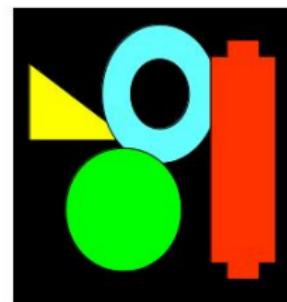
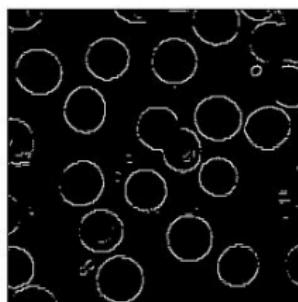
Dualité contour/région

Segmentation

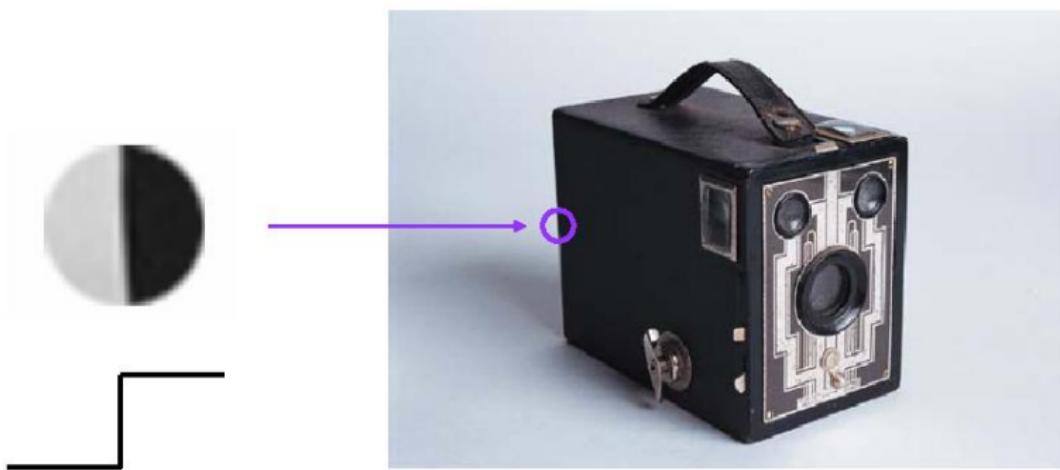


Recherche de frontières
(approches « contours »)

Recherche de régions
(approches « régions »)



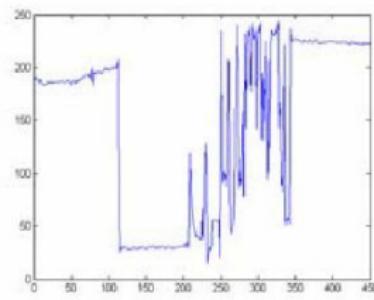
Qu'est-ce qu'un contour ?



- C'est une frontière qui sépare deux objets dans une image.
 - Une discontinuité de l'image.
 - Variation brusque et forte de l'intensité.

Qu'est-ce qu'un contour ?

- Un contour est une variation brusque et forte de l'intensité.

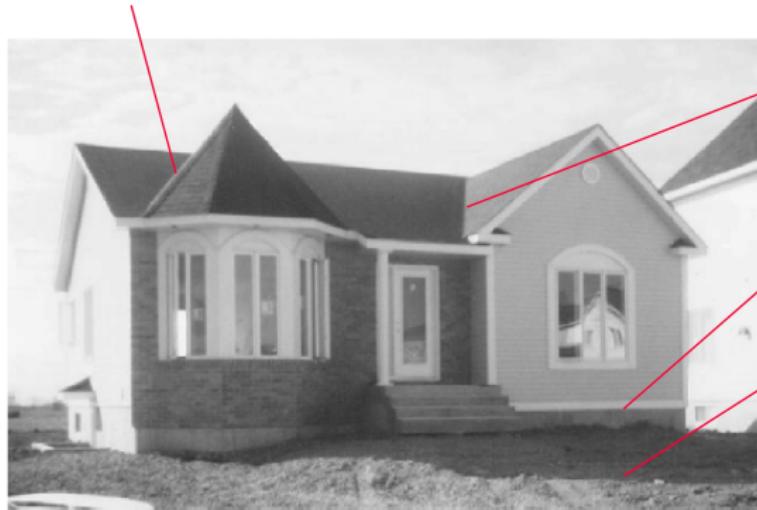


Qu'est-ce qu'un contour ?

- Un contour est une variation brusque et forte de l'intensité.

Exemples de détection des discontinuités

de profondeur



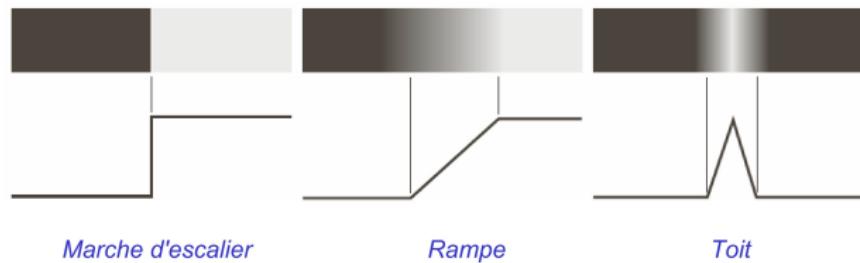
d'orientation
de surface

de réflectance

d'illumination

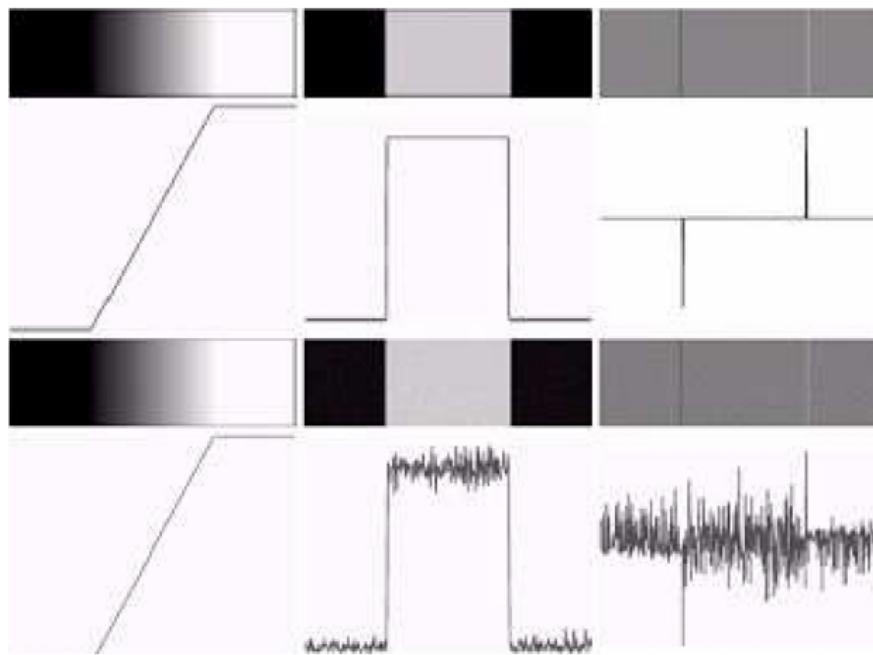
Qu'est-ce qu'un contour ?

- Types de contours



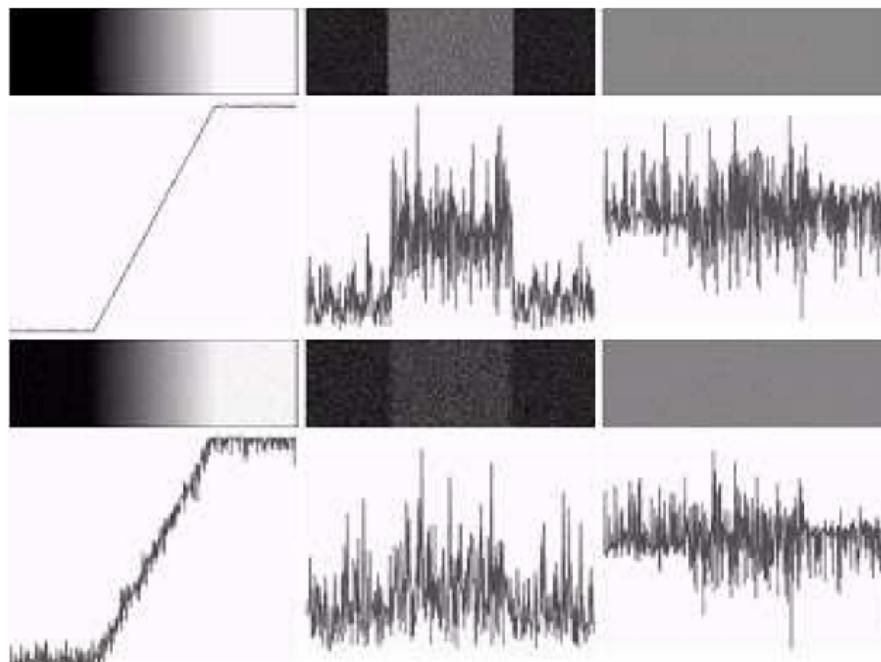
Qu'est-ce qu'un contour ?

- Types de contours



Qu'est-ce qu'un contour ?

- Types de contours



DéTECTEURS SIMPLES

- Détection de points : Application du masque suivant et seuillage de l'image résultat

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



DéTECTEURS SIMPLES

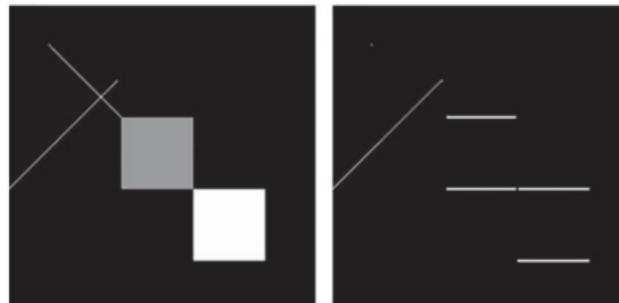
- Détection de lignes : Filtres définissant 4 directions E-W, NW-SE, N-S, NE-SW

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

DéTECTEURS SIMPLES

- Détection de lignes : Exemple



Méthodes dérivatives : Gradient

- Dérivée d'une image et contours

Image 1D $f(x)$



1ère dérivée $f'(x)$



$|f'(x)|$

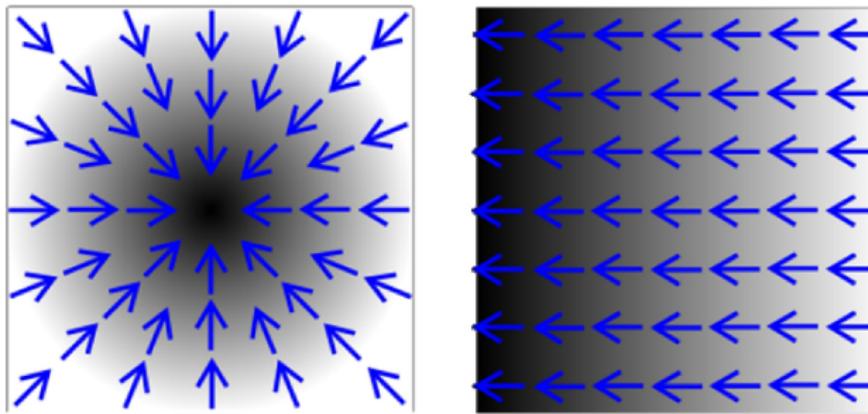


Pixels contours:
 $|f'(x)| > \text{Seuil}$



Méthodes dérivatives : Gradient

- Dérivée d'une image et contours



Les lignes bleues représentent le gradient de couleur du plus clair vers le plus foncé

- Norme : $G = G_x^2 + G_y^2 \sim \sqrt{G_x^2 + G_y^2}$
- Direction : $\theta = \arctan(G_y / G_x)$

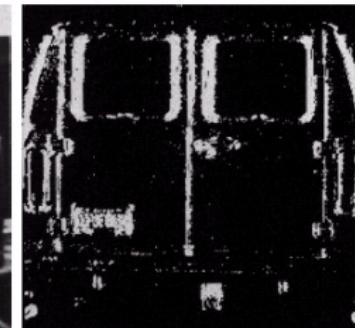
Méthodes dérivatives : Gradient

- Exemple de détection de contours

a
b
c d

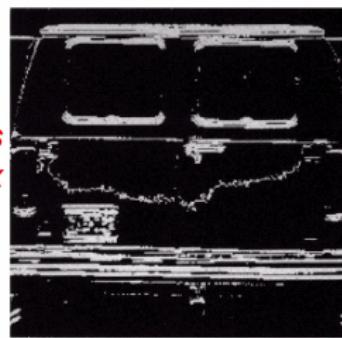
FIGURE 10.16

(a) Input image.
(b) G_y component of the gradient.
(c) G_x component of the gradient.
(d) Result of edge linking. (Courtesy of Perceptics Corporation.)

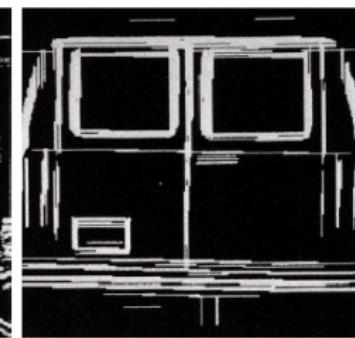


$$\frac{\partial f}{\partial x}$$

Contours
verticaux



Contours
horizontaux



$$\frac{\partial f}{\partial y} + \frac{\partial f}{\partial x}$$

Norme

Méthodes dérivatives : Gradient

- Approximations simples de la dérivée discrète

-1	1
----	---

-1
1

-1	0	1
----	---	---

-1
0
1

- Filtres de Roberts : dérivées directionnelles

1	0
0	-1

0	1
-1	0

Méthodes dérivatives : Gradient

- Filtres de Prewitt : lissage de l'image + dérivée de l'image

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

- Moyenneur + Dérivée

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} * (-1 \ 0 \ 1)$$

Détection des contours moins sensible au bruit.

Méthodes dérivatives : Gradient

- Filtres de Sobel

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

- Gaussienne + Dérivée

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} * (-1 \ 0 \ 1)$$

Détection des contours moins sensible au bruit

Méthodes dérivatives : Gradient

- Exemple de détection de contours



Détection avec Sobel sans seuillage



Seuillage avec
 $S=10\% \text{ Imax}$



Seuillage avec
 $S=25\% \text{ Imax}$

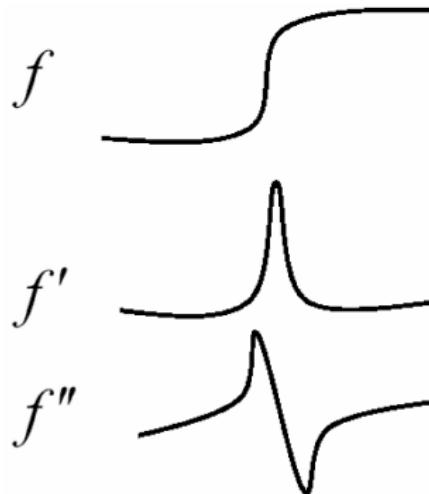
Méthodes dérivatives : Laplacien

- Détection des contours

Les contours correspondent :

Aux maxima de la première dérivée

Aux passages par zéros de la deuxième dérivée



Méthodes dérivatives : Laplacien

- Plusieurs approximations discrètes du Laplacien existent.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

- Isotropique : Symétrique par rapport à la rotation.

Méthodes dérivatives : Laplacien

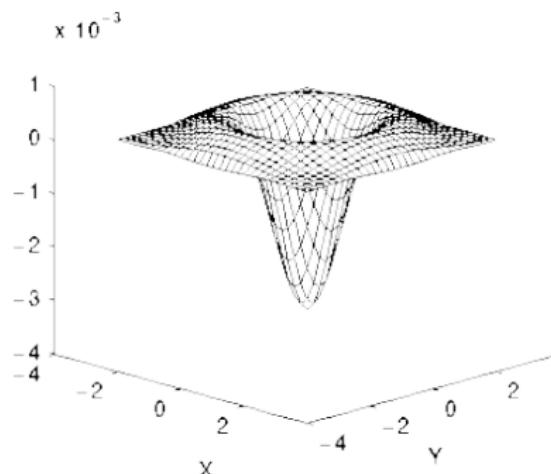
- Le Laplacien, comme tous les détecteurs de contours, est très sensible aux bruits. On préfère lisser l'image avant de détecter les contours. Pour ce faire, on combine un lissage Gaussien avec le Laplacien. La Gaussienne lisse l'image et rend les contours flous, mais conserve leurs positions. Le Laplacien donne les passages par zéro aux contours
- Laplacien de Gaussienne (LoG)

$$\text{LoG} * I = \Delta * G * I = \Delta G * I$$

$$\text{LoG}(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Méthodes dérivatives : Laplacien

- Filtre Laplacien



La fonction LoG à la forme d'un chapeau mexicain inversé.

On choisit souvent comme taille de filtre gaussien $5 * \sigma$.

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

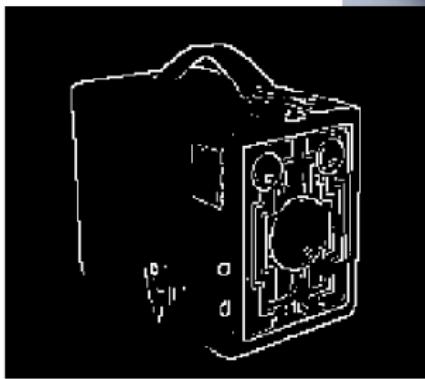
Approximation discrète
de LoG avec $\sigma=1.4$

Méthodes dérivatives : Laplacien

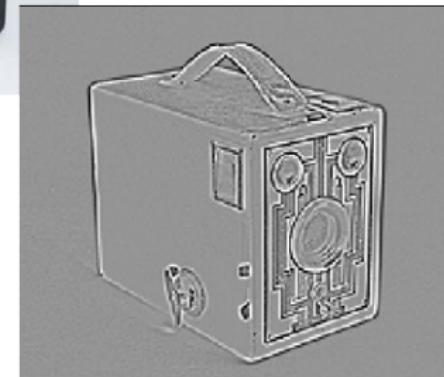
- Contours = passage par zéro du Laplacien



Gradient



Laplacien



Méthodes dérivatives : Laplacien

- Comparaison des détecteurs des contours

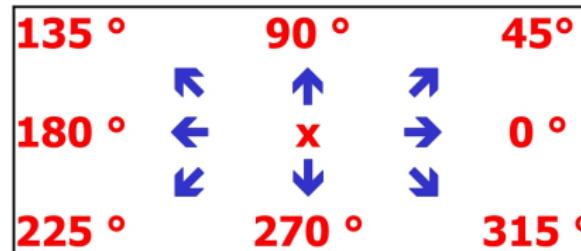
Aucun opérateur n'est parfait pour détecter les contours. En pratique, on obtient des contours incomplets il y a des pixels superflus il y a des manques il y a des erreurs dans la position et l'orientation des pixels contours Chacun semble avoir sa préférence pour une méthode ou une autre. Un opérateur de détection de contour n'est qu'une première étape dans la chaîne de segmentation.

Méthode de Canny

- Filtre optimal pour la détection des contours
 - Filtre en plusieurs étapes (pas seulement une convolution)
- Étant donnés
 - un modèle de contour (marche)
 - un modèle de bruit (blanc gaussien)
- Caractériser les performances en termes de :
 - détection : surtout pour les contours faibles
 - localisation : contour détecté proche du contour réel
 - réponse unique (un contour = une seule réponse)

Méthode de Canny

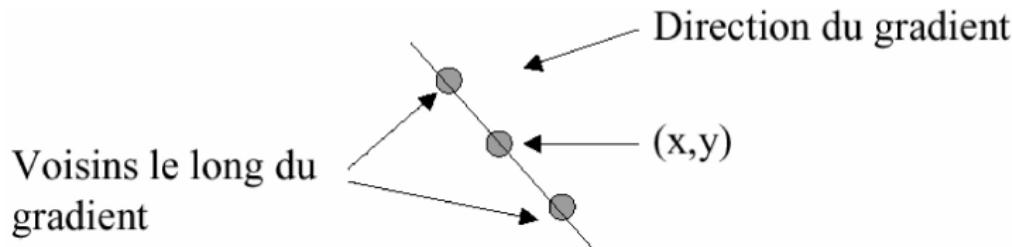
- 1. Appliquer un filtre Gaussien sur l'image
 - Filtre passe-bas pour enlever le bruit
- 2. Calculer l'intensité du gradient dans l'image
 - Filtre de Sobel en X et Y
 - Calcul de la norme $|G| = |G_x| + |G_y|$
- 3. Calculer les directions du gradient dans l'image
 - Direction du gradient $\theta = \arctan(G_y/G_x)$
 - Arrondi des directions par multiples de 45 degrés



Méthode de Canny

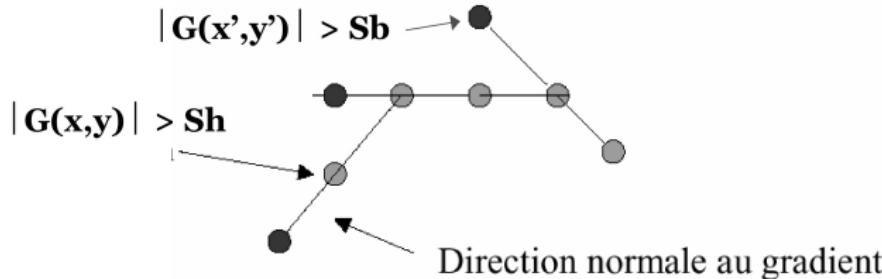
- 4. Suppression des non-maxima

- Si la norme du gradient en un pixel est inférieure à la norme du gradient d'un de ses 2 voisins le long de la direction du gradient, alors mettre la norme pour le pixel à zéro.



Méthode de Canny

- 5. Seuillage des contours (hystérésis)
 - Utilise deux seuils : un seuil haut (Sh) et un seuil bas (Sb)
 - Pour chaque pixel
 - Si $\|\nabla I(x, y)\| < Sb$, alors le pixel est mis à zéro (non-contour).
 - Si $\|\nabla I(x, y)\| > Sh$, alors le pixel est contour.
 - Si $Sb \leq \|\nabla I(x, y)\| \leq Sh$, alors le pixel est contour s'il est connecté à un autre pixel déjà accepté comme contour.



Méthode de Canny

- Exemple

Image
originale



Sobel



Suppression des
non-maxima



Seuillage



Quel filtre choisir pour les contours ?

- Aucun opérateur n'est parfait pour détecter les contours
- En pratique, on obtient des contours incomplets
 - il y a des pixels superflus
 - il y a des manques
 - il y a des erreurs de position et d'orientation des pixels contours
- Chacun semble avoir sa préférence pour une méthode ou une autre
- Un opérateur de détection de contour n'est qu'une première étape dans la chaîne de segmentation

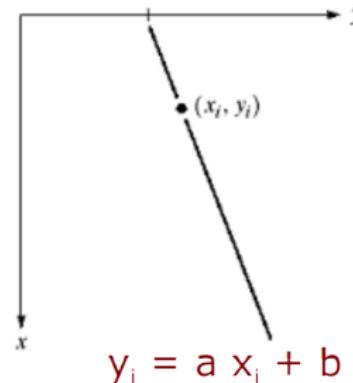
Quel filtre choisir pour les contours ?

- Il existe des approches globales pour les contours
 - On ne recherche pas seulement des pixels contours
 - On cherche le contour au complet
- On cherche plusieurs pixels correspondant à un contour
 - Comment définir le contour ? Problème ?
- Différentes techniques
 - Ici : Transformée de Hough

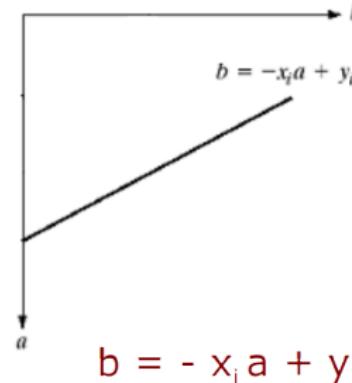
Méthode de Hough

- Approche globale : représentation des droites par des points
- Plan $x - y$: $y_i = ax_i + b$
 - Une inf. de lignes passent par (x_i, y_i) et une seule pour (a, b)
- Plan paramétrique $a - b$: $b = -x_i a + y_i$
 - Une seule ligne pour (x_i, y_i) et une inf. passent par (a, b)

Plan x-y

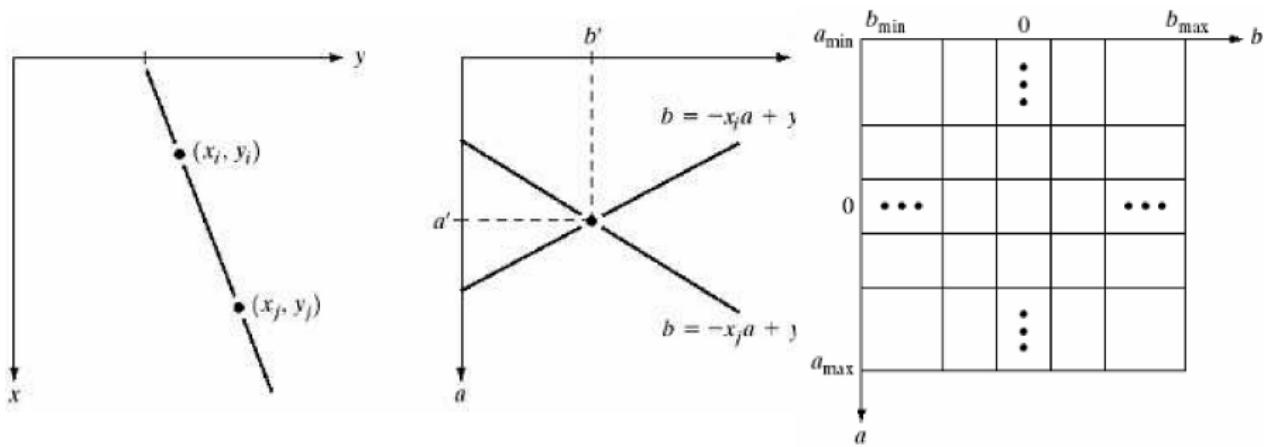


Plan paramétrique a-b



Méthode de Hough

- Principe de la transformée de Hough
 - Cellules d'accumulation - Matrice (a, b)
 - On construit une image des votes
 - chaque point permet de voter pour une droite particulière
 - Les droites recevant le plus de votes sont conservées



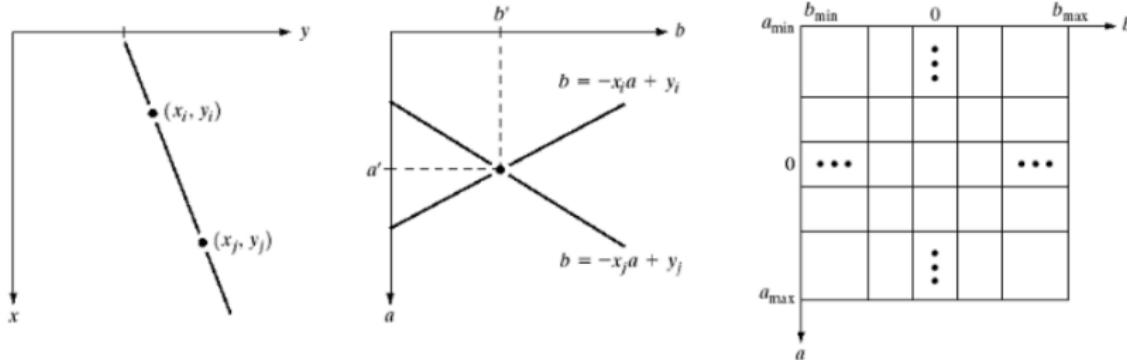
Méthode de Hough

- 1. Détection des contours (Gradient, Laplace, ...)
- 2. Pour chaque (x_i, y_i) du plan $x - y$ (retenu par le détecteur) nous calculons pour toutes les valeurs de a la valeur de b résultante : On obtient une droite dans le plan $a - b$
 - Le nombre de valeurs dans chaque cellule a' , b' est le nombre de points du plan $x-y$ qui existent sur chacune des lignes $b = -x a + y$ du plan paramétrique $a-b$.
- 3. Les points de croisement des droites dans le plan $a-b$ indiquent les vraies droites existantes dans le plan $x-y$

Méthode de Hough

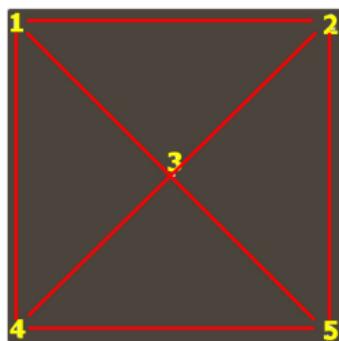
- Problème : la pente approche l'infinie pour des lignes qui approchent la verticale !
 - Solution : représentation sous forme polaire

$$\rho = x \cos \theta + y \sin \theta$$

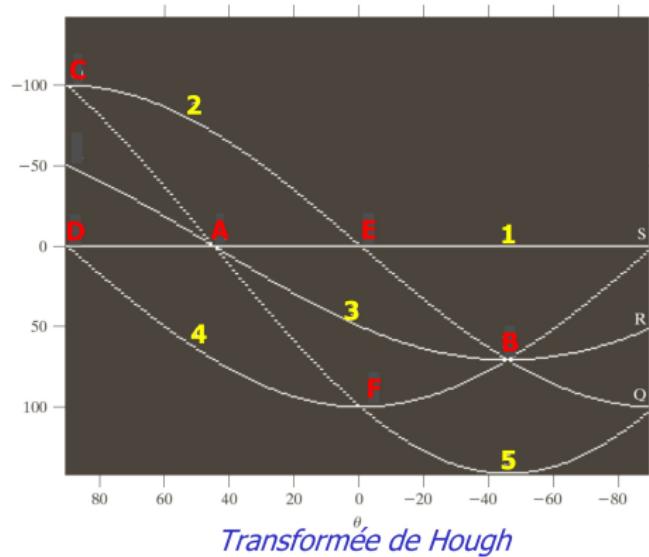


Méthode de Hough

- Application

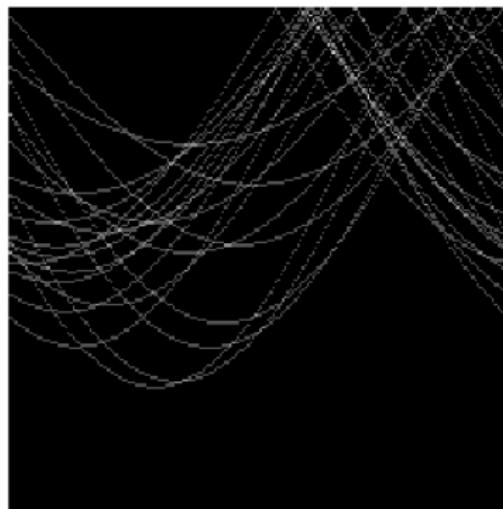
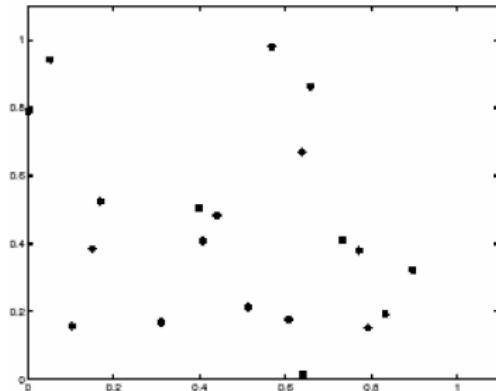


*Image contenant
5 points*



Méthode de Hough

- Application

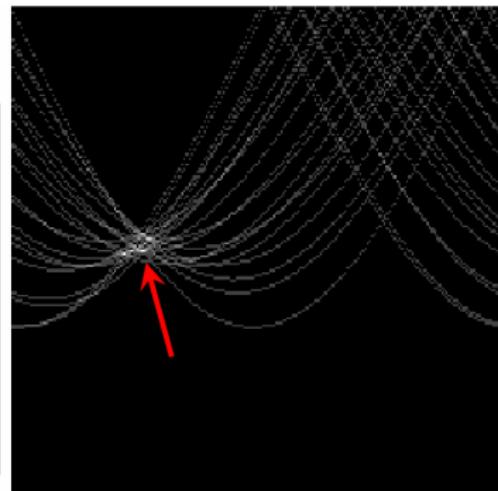
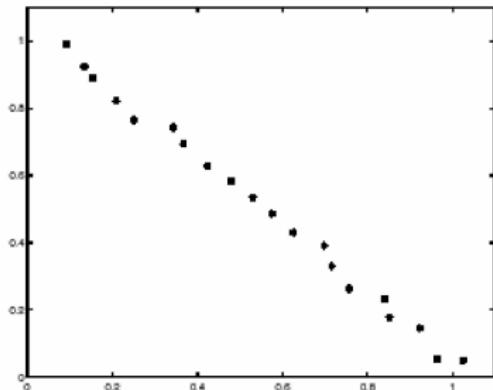


La transformée de points aléatoires ne donne rien de précis

Méthode de Hough

- Application

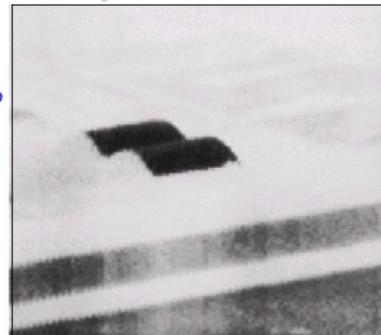
La transformée de points alignés permet de retrouver la droite



Méthode de Hough

- Application

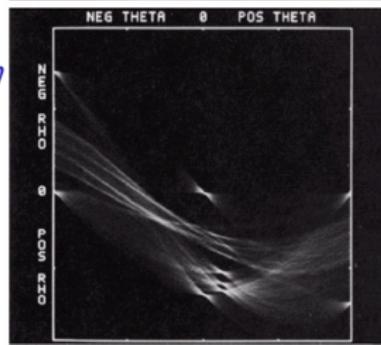
Image



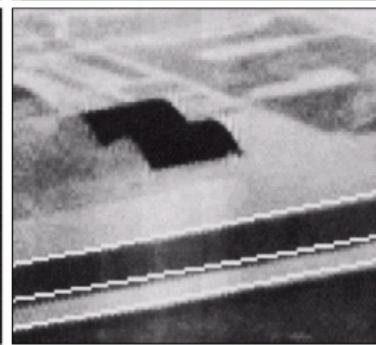
Gradient



Hough



Final



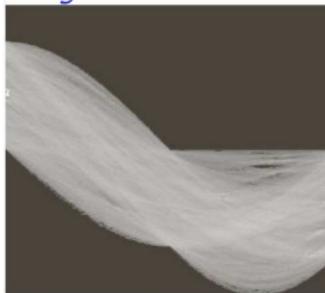
Méthode de Hough

- Application

Image



Hough



Canny



Final

Droites



Seuillage

- Le seuillage est une méthode simple et très populaire pour le traitement des images numériques
- Ce n'est pas une méthode de segmentation en régions
 - Approche pixel (pas région ni contour)
 - Mais on l'utilise souvent en segmentation (avec post-traitements)
- Le seuillage peut être
 - Global : un seuil pour toute l'image
 - Local : un seuil pour une portion de l'image
 - Adaptatif : un seuil s'ajustant selon les parties de l'image

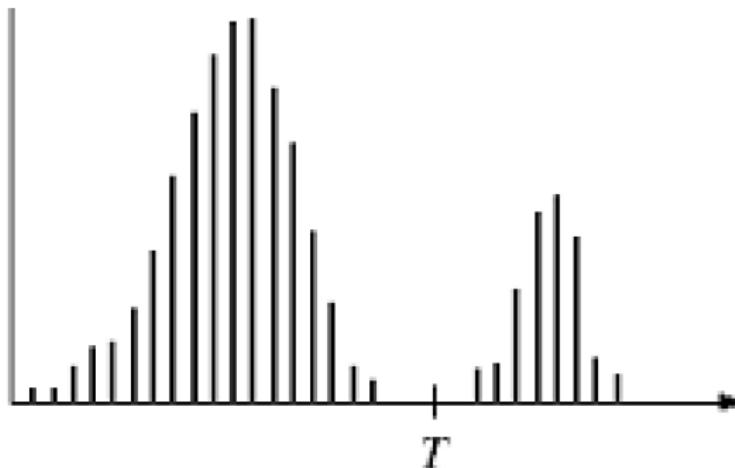
Seuillage global

Principe de base

- Seuillage de base (2 classes) :
 - Si $\text{valeur(pixel)} \geq \text{seuil}$ alors $\text{valeur(pixel)} = 1$
 - Si $\text{valeur(pixel)} < \text{seuil}$ alors $\text{valeur(pixel)} = 0$
- Le résultat du seuillage est une image binaire
 - 0 ou 1 (qu'on transforme parfois en 0 :255 pour l'affichage)
- Problème : choix du seuil !

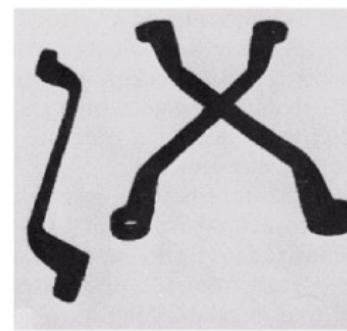
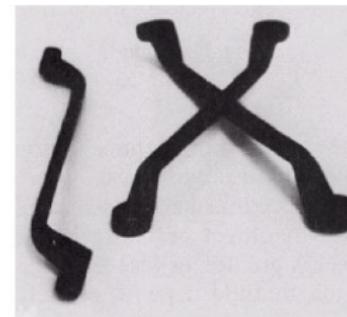
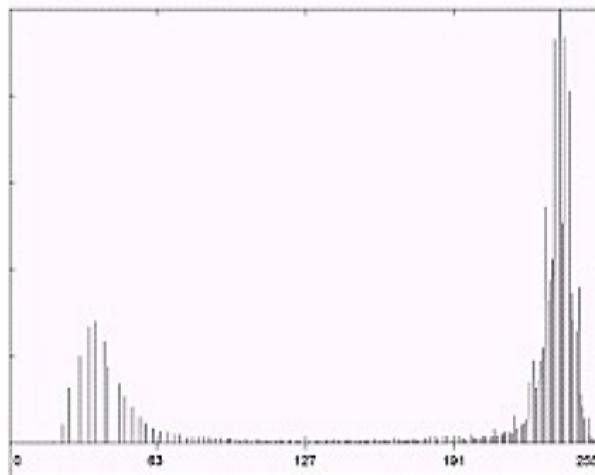
Seuillage global

- Choix du seuil : Seuillage de l'histogramme



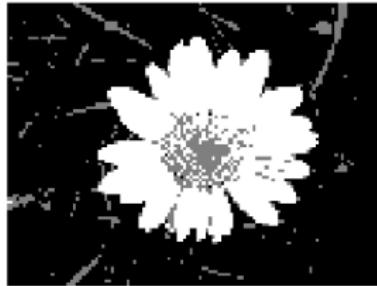
Seuillage global

- Choix du seuil : Seuillage d'histogramme simple



Seuillage global

- Choix du seuil : Seuillage de l'histogramme
- Avantages
 - Universel, temps réel, simplicité
 - Fonctionne bien sur des histogrammes multi-modaux
- Inconvénients
 - Connaître le nombre de classes.
 - Apparition de faux éléments (aucune prise en compte de la composante spatiale).
 - Nombre de modes souvent nombre de classes attendu.



2 seuils pour 3 classes

Seuillage global

- Choix du seuil : Seuillage de l'histogramme
- Comment trouver le bon seuil (T) ?
 - Une valeur obtenue par tests
 - La valeur moyenne des tons de gris
 - La valeur médiane entre le ton maximum et le ton minimum
 - Une valeur qui balance les deux sections de l'histogramme
 - Il existe des algorithmes automatiques pour trouver le seuil : Moyennage, Otsu, Kittler, ...

Seuillage global

- Méthode de moyennage

- ① Choix d'un seuil initial $S = S_0$.
- ② Repartir les pixels en deux classes en utilisant S .
- ③ Mettre à jour le seuil S comme la moyenne des deux moyennes des niveaux de gris des deux classes $S' = \frac{m_1+m_2}{2}$.
- ④ Si $|S - S'| < \delta$, arrêter. $S = S'$. Sinon, $S = S'$ et retourner à l'étape 2.

$$\begin{bmatrix} 185 & 182 & 45 & 2 \\ 188 & 140 & 10 & 5 \\ 189 & 74 & 2 & 7 \\ 164 & 21 & 5 & 6 \end{bmatrix} \begin{bmatrix} 185 & 182 \\ 188 & 140 \\ 189 & \\ 164 & \end{bmatrix} \begin{bmatrix} 45 & 2 \\ 10 & 5 \\ 74 & \\ 21 & 6 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Img. init. ($S_0 = 76$), Segm./ $S_1 = \frac{174.66+17.7}{2} = 96.18$, Segm. finale

Seuillage global

- Méthode d'Otsu

On balaie toutes les valeurs de seuil possible T

Pour chaque seuil T :

- On calcule les moyennes et les variances de chaque classe
- On s'intéresse à la variance intra-classes

Moyennes : μ_1 et μ_2

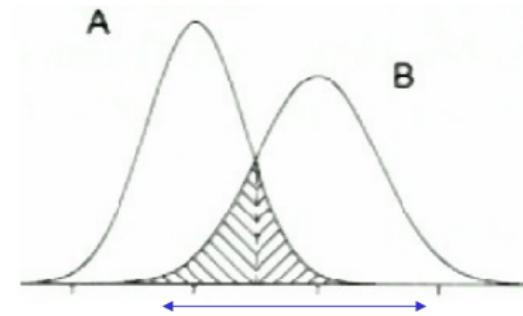
Variances : σ_1^2 et σ_2^2

Variance Intra-classes :

$$\sigma_w^2 = P_1 \cdot \sigma_1^2 + P_2 \cdot \sigma_2^2$$

Le seuil optimal est celui qui donne σ_w minimum

Basé sur le fait que les classes sont bien définies et regroupées



$$\sigma_1^2 = \frac{1}{T} \sum_{i=0}^{T-1} (h(i) - \mu_1)^2$$

$$\sigma_2^2 = \frac{1}{256-T} \sum_{i=T}^{255} (h(i) - \mu_2)^2$$

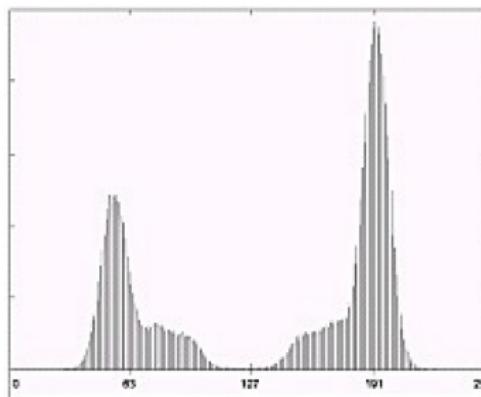
$$\mu_1 = \frac{1}{T} \sum_{i=0}^{T-1} h(i) \quad P_1 = \frac{1}{N \cdot M} \sum_{i=0}^{T-1} h(i)$$

$$\mu_2 = \frac{1}{256-T} \sum_{i=T}^{255} h(i) \quad P_2 = \frac{1}{N \cdot M} \sum_{i=T}^{255} h(i)$$

Seuillage global

- Méthode d'Otsu

Seuil trouvé par l'algorithme : $T = 125$

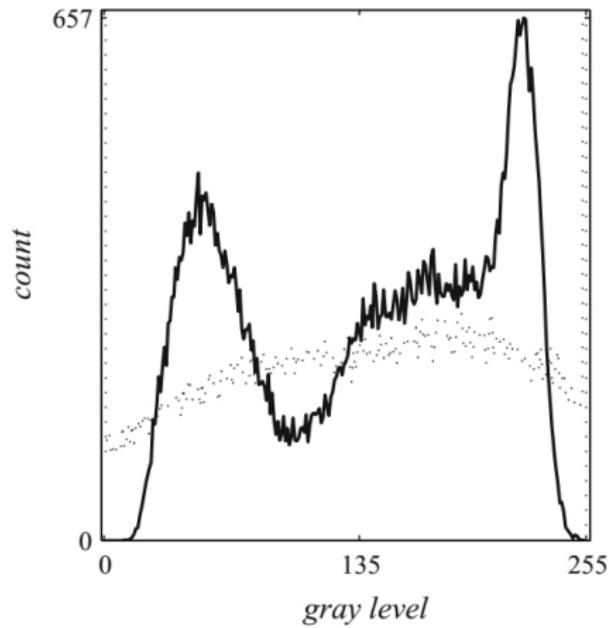


- Exemple : un et deux seuils



Seuillage global

- Exemple : seuillage d'images bruitées



En pointillé : Histo. original - Ligne pleine : Histo. après debruitage.

Seuillage global

- Exemple : seuillage d'images bruitées

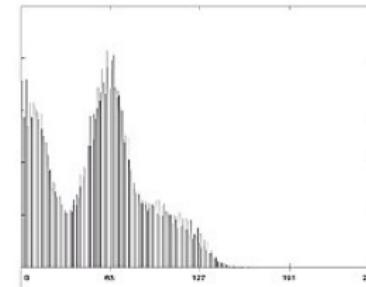
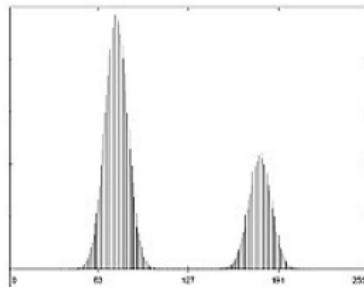
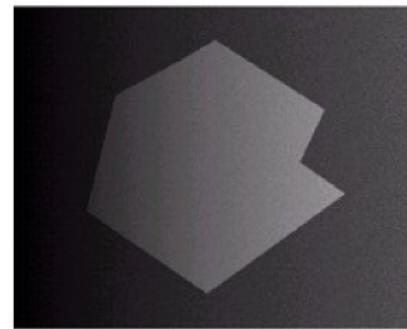
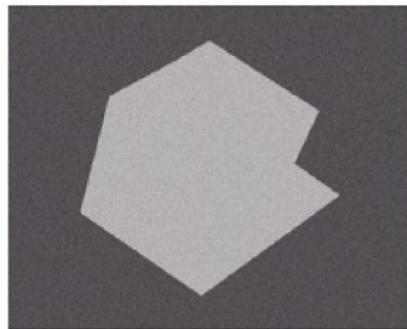


Segmentation avant et après débruitage.

Seuillage local ou adaptatif

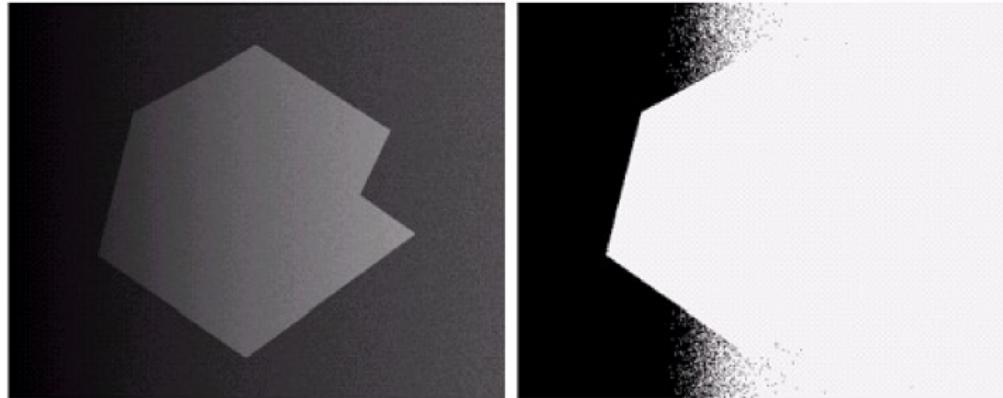
- Seuillage globale : problèmes

Problème d'éclairage ?



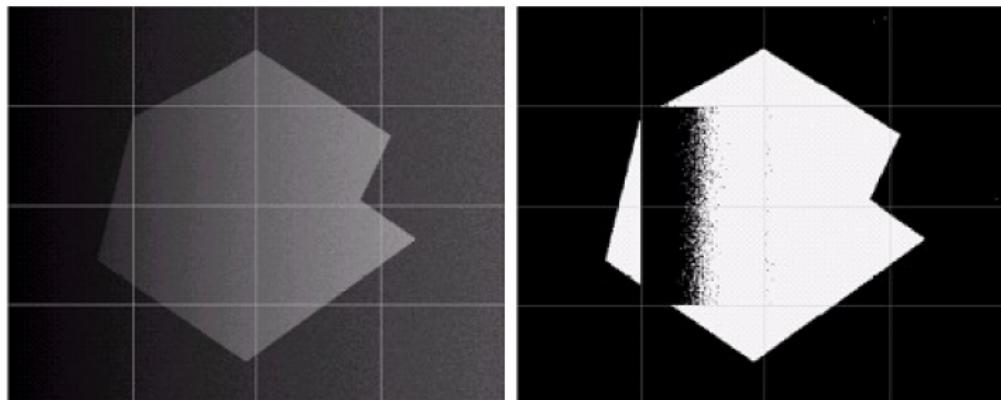
Seuillage local ou adaptatif

- Seuillage globale : problèmes



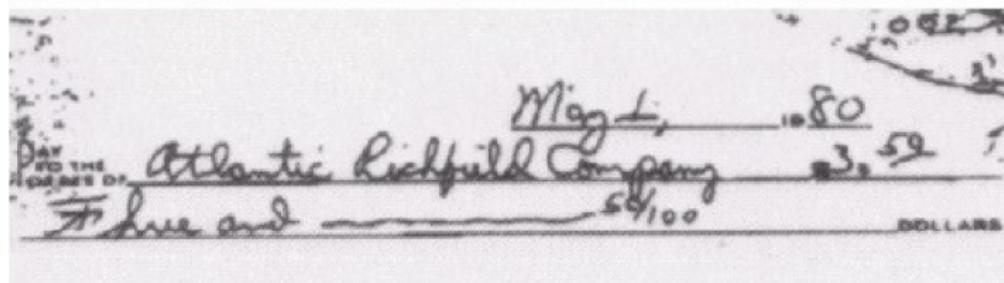
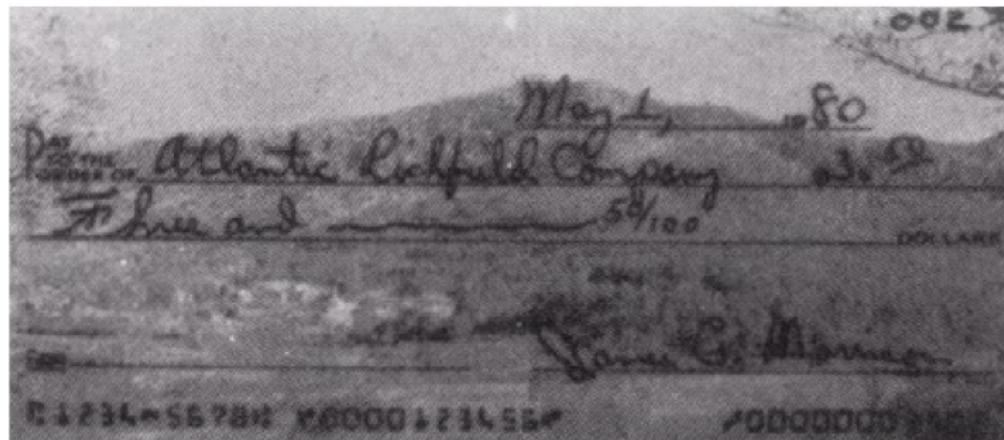
Seuillage local ou adaptatif

- On divise l'image en sous-images
- On seuille chaque sous-image indépendamment
- Les 4 sous-images de coins ne sont pas traitées car $\text{var.} < 100$



Seuillage local ou adaptatif

- Exemple



Croissance de régions

- Faire croître une région en commençant par une petite région (un pixel par exemple)
- Critère d'ajout d'un pixel à une région :
 - La différence entre l'intensité du nouveau pixel et la moyenne des intensités des pixels de la région ne dépasse pas un seuil prédéfini δ .

Croissance de régions

- Exemple 1

Région de départ

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

$$|59 - 53| > 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

$$|53 - 53| \leq 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

$$|53 - 53| \leq 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

$$|58 - 53| > 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

$$|58 - 53| > 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

$$|55 - 53| \leq 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53,5

Croissance de régions

- Exemple 1

$$|55 - 53,5| \leq 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53,5

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53,8

Croissance de régions

- Exemple 1

$$|51 - 53,8| > 2$$

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Moyenne de la région : 53,8

Croissance de régions

- Exemple 1

Point de départ de la nouvelle régi



53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Croissance de régions

- Exemple 1

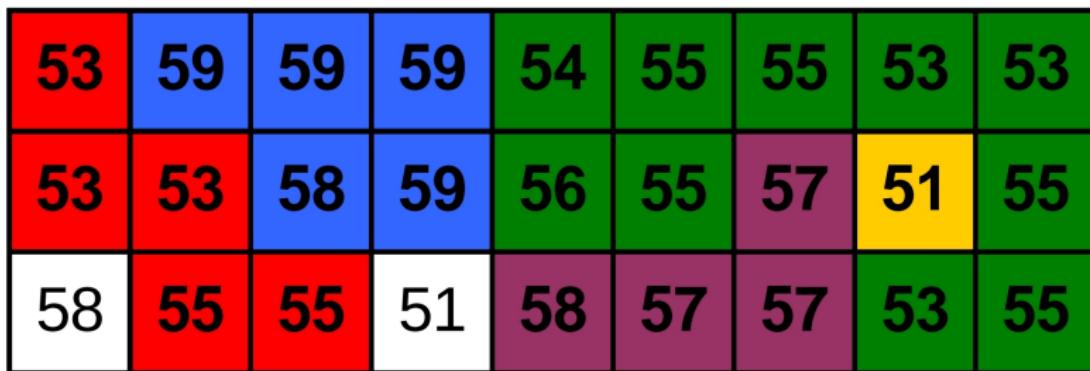


Image en niveaux de gris

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

Croissance de régions

- Exemple 1

53	59	59	59	54	55	55	53	53
53	53	58	59	56	55	57	51	55
58	55	55	51	58	57	57	53	55

Image en niveaux de gris

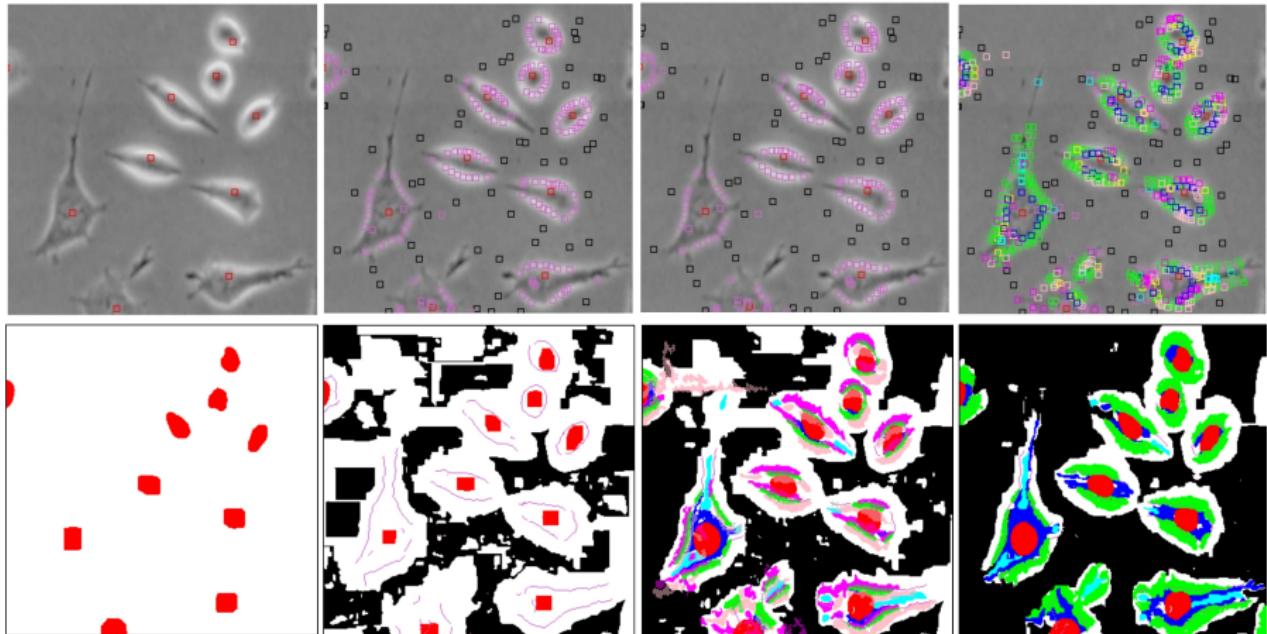
Croissance de régions

- Exemple 2



Croissance de régions

- Exemple 2



Division-fusion (Split-and-Merge)

- Étape de division (split) : Diviser récursivement tout bloc non-homogène selon un prédictat défini : variance, max-min, ...
 - La division d'un bloc donne 4 sous-blocs.
 - Les attributs de chaque sous-bloc sont recalculés.
- Étape de fusion (merge) : Regrouper les blocs adjacents représentant des régions homogènes selon un prédictat défini.
- Exemple. Critère d'homogénéité : Une région est homogène si tous ses pixels ont la même intensité

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	54	52
56	57	57	57	57	59	59	52	52
52	55	55	55	55	59	59	52	52
52	55	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	54	52
56	57	57	57	57	57	59	52	52
52	55	55	55	55	59	59	52	52
52	55	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de Fusion

51	51	54	54	54	54	54	54	52
56	57	57	57	57	57	59	52	52
52	55	55	55	55	59	59	52	52
52	55	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

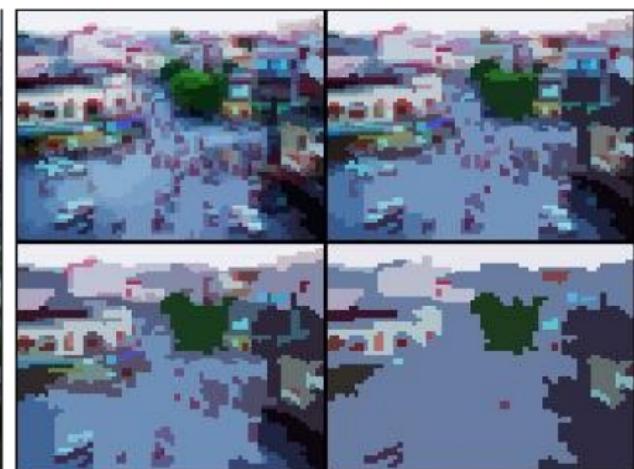
Division-fusion (Split-and-Merge)

- Exemple : Étape de division

51	51	54	54	54	54	54	52
56	57	57	57	57	59	52	52
52	55	55	55	59	59	52	52
52	55	55	55	55	56	52	52

Division-fusion (Split-and-Merge)

- Exemples : Différents seuils



Division-fusion (Split-and-Merge)

- Exemples : Différents seuils



Ligne de partage des eaux (Watershed)

- Limite du seuillage basique et la détection des contours



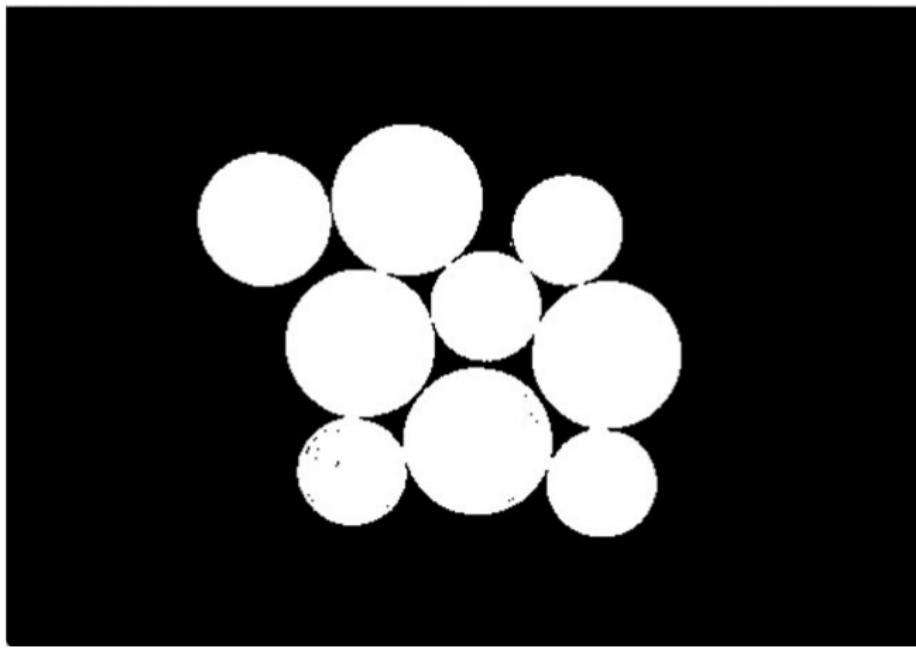
Ligne de partage des eaux (Watershed)

- Limite du seuillage basique et la détection des contours



Ligne de partage des eaux (Watershed)

- Limite du seuillage basique et la détection des contours



Ligne de partage des eaux (Watershed)

- Limite du seuillage basique et la détection des contours



Ligne de partage des eaux (Watershed)

- Algorithme de partage des eaux (objets qui se chevauchent).



Ligne de partage des eaux (Watershed)

- Visualisation de l'image en 3D : ton de gris en z

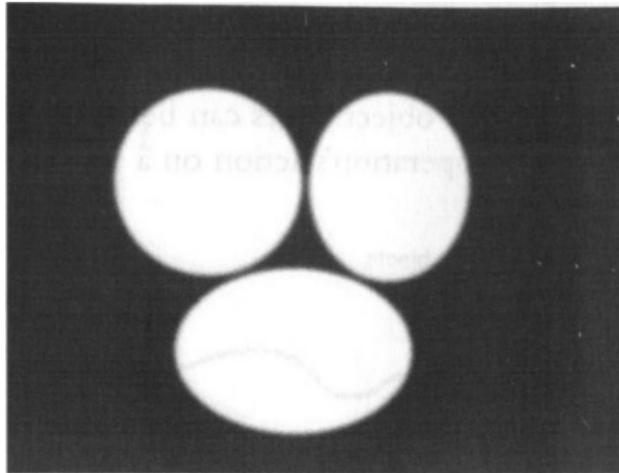
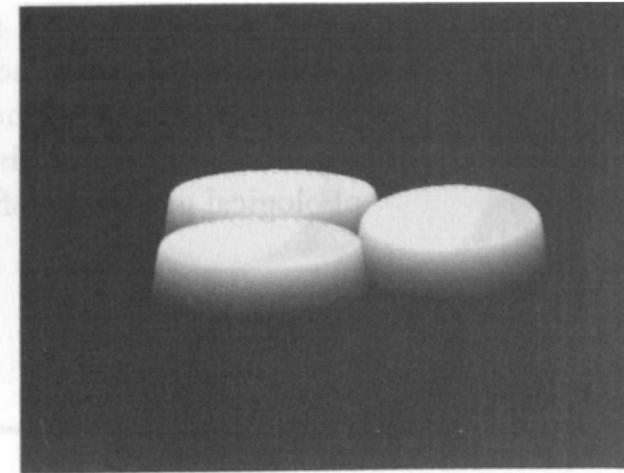


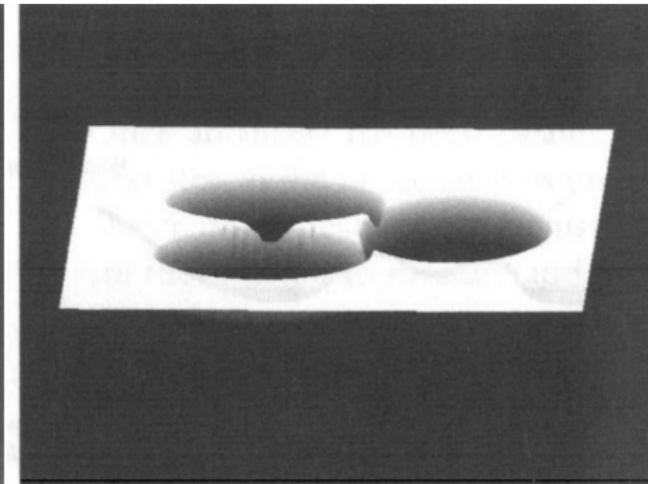
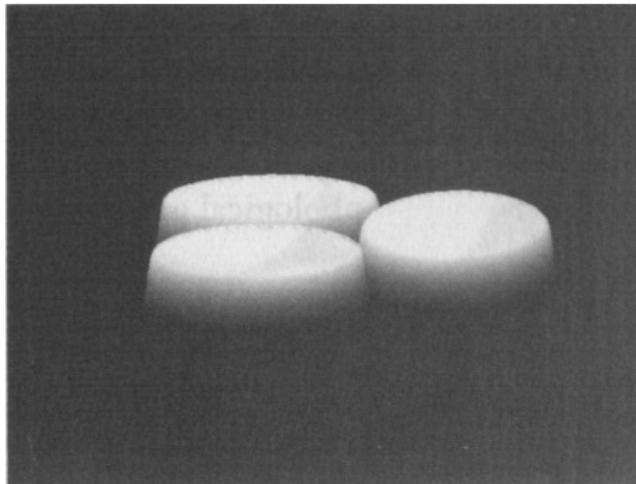
Image 2D



Visualisation en 3D

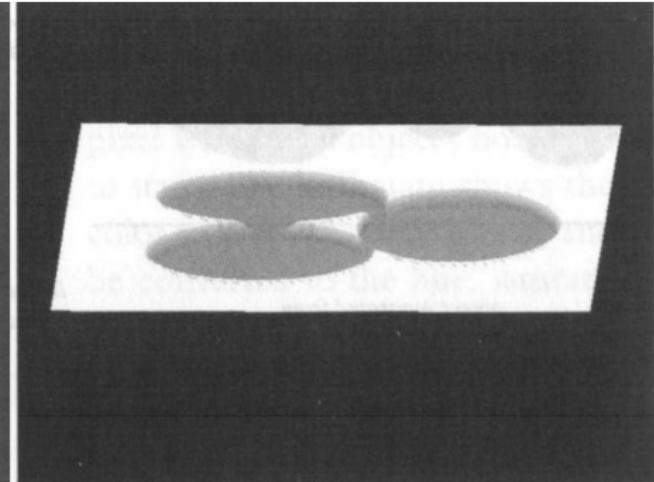
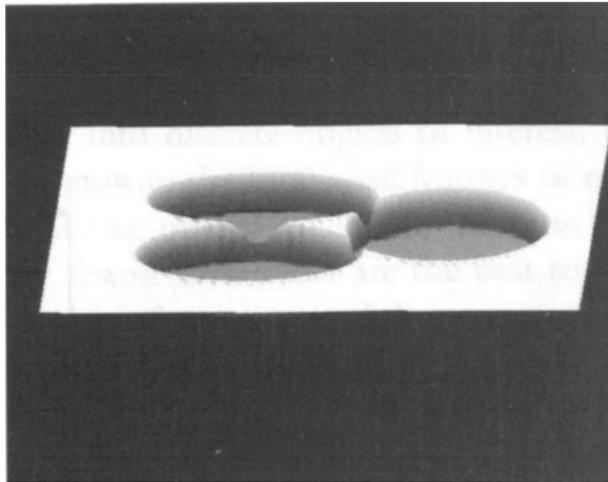
Ligne de partage des eaux (Watershed)

- "Complémenter" les valeurs pour créer des zones inondables



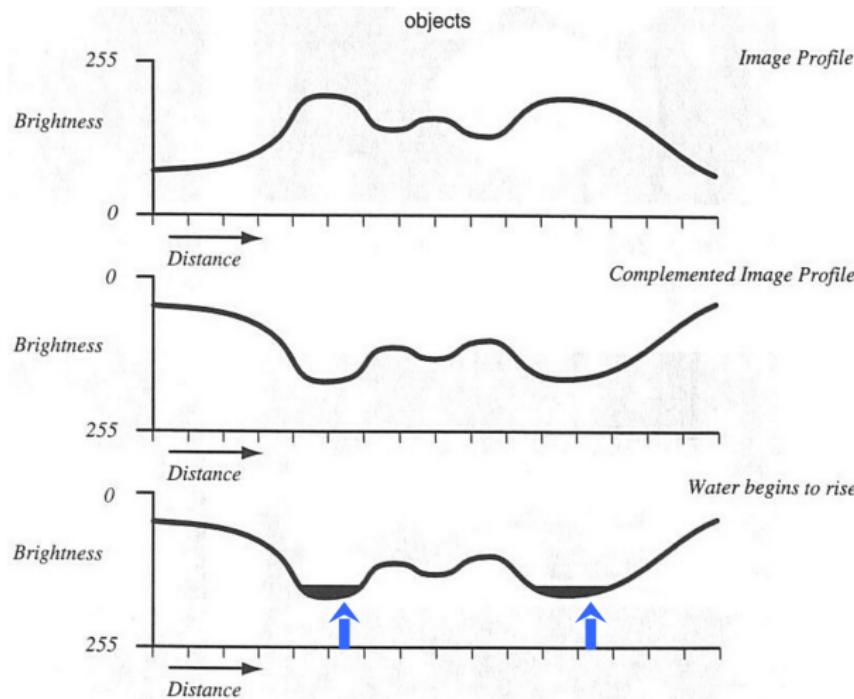
Ligne de partage des eaux (Watershed)

- "Infiltrer" les cavités des zones inondables



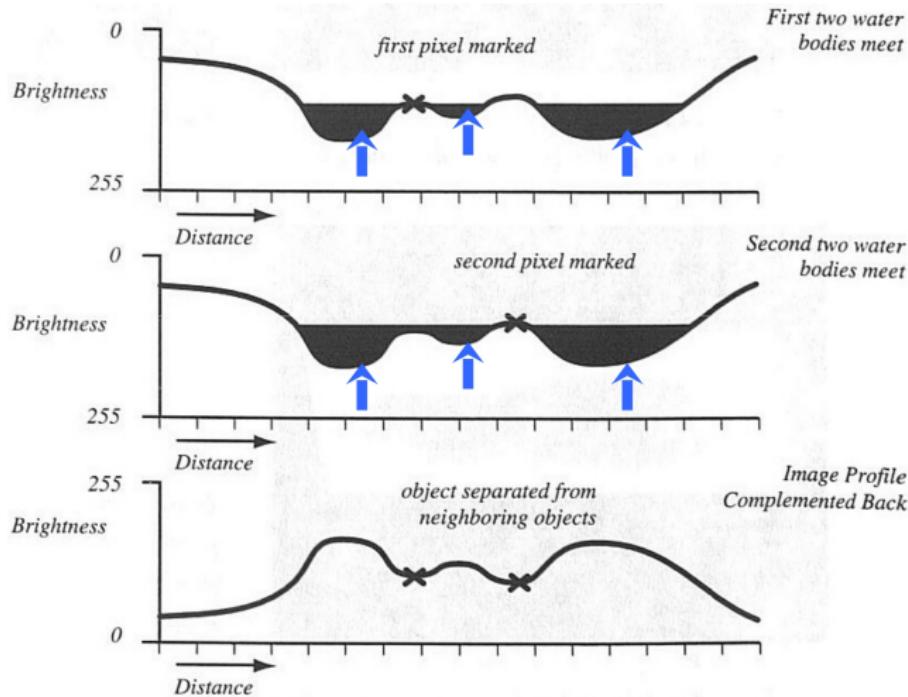
Ligne de partage des eaux (Watershed)

- En 1D



Ligne de partage des eaux (Watershed)

- En 1D

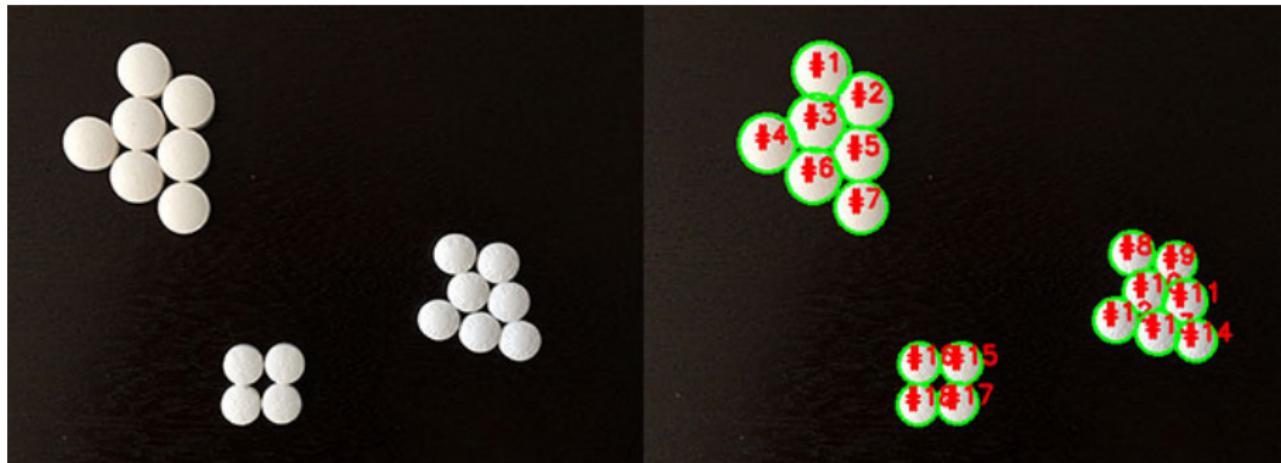


Ligne de partage des eaux (Watershed)

- 1. Calculer la transformation de la distance euclidienne : pour chaque pixel, calculer la distance au plus proche 0 (pixel du fond, background pixel)
- 2. Déterminer les pics (i.e., maxima locaux) de la matrice des distance. Une distance minimale (exp. 20 pixels) entre les pics est respectée.
- 3. Appliquer l'algorithme de ligne de partage des eaux. Le résultat est une image de segmentation (labels).

Ligne de partage des eaux (Watershed)

- Exemples



Ligne de partage des eaux (Watershed)

- Exemples



Ligne de partage des eaux (Watershed)

- Transformation de distance
 - Image binaire : distance minimale de chaque pixel à la composante "objet"

$$x(m, n) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Ligne de partage des eaux (Watershed)

- Transformation de distance
 - Image binaire : distance minimale de chaque pixel à la composante "objet"

$$DS(m, n) = \begin{bmatrix} 4.4721 & 3.6056 & 2.8284 & 2.2361 & 1.4142 & 1.0000 & 1.4142 & 2.2361 \\ 4.1231 & 3.1623 & 2.2361 & 1.4142 & 1.0000 & 0 & 1.0000 & 2.0000 \\ 4.0000 & 3.0000 & 2.0000 & 1.0000 & 0 & 0 & 1.0000 & 1.4142 \\ 3.6056 & 2.8284 & 2.2361 & 1.4142 & 1.0000 & 0 & 0 & 1.0000 \\ 3.1623 & 2.2361 & 1.4142 & 1.0000 & 1.0000 & 0 & 0 & 1.0000 \\ 3.0000 & 2.0000 & 1.0000 & 0 & 0 & 0 & 1.0000 & 1.4142 \\ 3.1623 & 2.2361 & 1.4142 & 1.0000 & 1.0000 & 0 & 1.0000 & 2.0000 \\ 3.6056 & 2.8284 & 2.2361 & 2.0000 & 1.4142 & 1.0000 & 1.4142 & 2.2361 \end{bmatrix}$$

- Temps de calculs élevés.

Ligne de partage des eaux (Watershed)

Algorithme rapide pour calculer des distances approximatives

- 1. L'image initiale est étendue par zéro-padding pour application d'un filtre 3×3 . Ensuite, 0 est remplacé par ∞ et 1 est remplacé par 0, ce qui donne une matrice initiale de distance

$$DS(m, n) = \begin{bmatrix} \infty & \infty \\ \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty \end{bmatrix}$$

Ligne de partage des eaux (Watershed)

Algorithme rapide pour calculer des distances approximatives

- 2. Un masque direct est appliqué : du coin supérieur gauche, de gauche droite et haut vers le bas. Pour chaque pixel, on calcule la somme du masque avec l'image et on la valeur du pixel est mise à jour avec le minimum de l'imagette obtenue.

$$\begin{bmatrix} \infty & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} \infty & \infty \\ \infty & 0 \end{bmatrix} = \begin{bmatrix} \infty & \infty \\ \infty & 0 \end{bmatrix}$$

$$\begin{bmatrix} \infty & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} \infty & \infty \\ 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty \\ 1 & \infty \end{bmatrix}$$

$$\begin{bmatrix} \infty & 1 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} \infty & \infty \\ 1 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty \\ 2 & \infty \end{bmatrix}$$

Ligne de partage des eaux (Watershed)

Algorithme rapide pour calculer des distances approximatives

- Résultat de l'application du premier filtre direct

$$DS(m, n) = \begin{bmatrix} \infty & \infty \\ \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 0 & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 & 0 & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 2 & 0 & 0 & 1 & \infty \\ \infty & \infty & \infty & \infty & 0 & 0 & 0 & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & 1 & 1 & 0 & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & 2 & 2 & 1 & 2 & 3 & \infty \\ \infty & \infty \end{bmatrix}$$

Ligne de partage des eaux (Watershed)

Algorithme rapide pour calculer des distances approximatives

- 3. Un masque inverse est appliqué comme précédemment : de droite à gauche et d'en bas vers le haut, avec le filtre $\begin{bmatrix} 0 & 1 \\ 1 & \infty \end{bmatrix}$

$$DS(m, n) = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 & 2 & 3 \\ 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 \\ 4 & 3 & 2 & 1 & 0 & 0 & 1 & 2 \\ 5 & 4 & 3 & 2 & 1 & 0 & 0 & 1 \\ 4 & 3 & 2 & 1 & 1 & 0 & 0 & 1 \\ 3 & 2 & 1 & 0 & 0 & 0 & 1 & 2 \\ 4 & 3 & 2 & 1 & 1 & 0 & 1 & 2 \\ 5 & 4 & 3 & 2 & 2 & 1 & 2 & 3 \end{bmatrix}$$

Ligne de partage des eaux (Watershed)

Algorithme rapide pour calculer des distances approximatives

- Autre choix des filtres et résultats

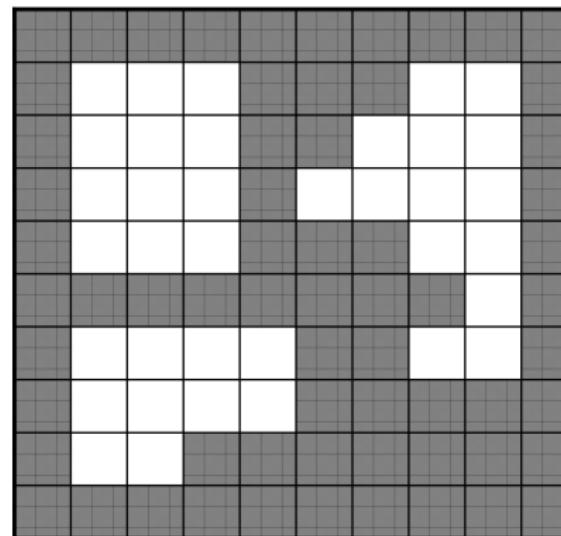
$$h_f(m, n) = \begin{bmatrix} 4 & 3 & 4 \\ 3 & 0 & \infty \\ \infty & \infty & \infty \end{bmatrix} \quad h_b(m, n) = \begin{bmatrix} \infty & \infty & \infty \\ \infty & 0 & 3 \\ 4 & 3 & 4 \end{bmatrix}$$

$$DS(m, n) = \begin{bmatrix} 14 & 11 & 8 & 7 & 4 & 3 & 4 & 7 \\ 13 & 10 & 7 & 4 & 3 & 0 & 3 & 6 \\ 12 & 9 & 6 & 3 & 0 & 0 & 3 & 4 \\ 11 & 8 & 7 & 4 & 3 & 0 & 0 & 3 \\ 10 & 7 & 4 & 3 & 3 & 0 & 0 & 3 \\ 9 & 6 & 3 & 0 & 0 & 0 & 3 & 4 \\ 10 & 7 & 4 & 3 & 3 & 0 & 3 & 6 \\ 11 & 8 & 7 & 6 & 4 & 3 & 4 & 7 \end{bmatrix}$$

- On divise toutes les valeurs par 3. La 1ère valeur est $14/3 = 4.6667 \simeq 4.4721$.

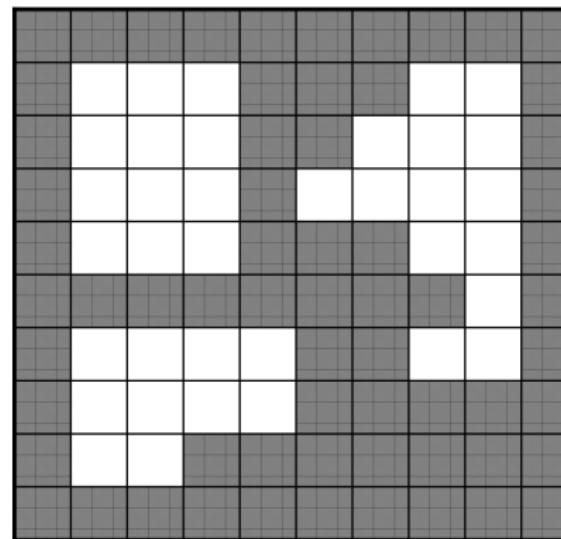
Étiquetages de composantes connexes

- Affecter un numéro unique (étiquette) pour chaque région
- Effectuer deux parcours de l'image



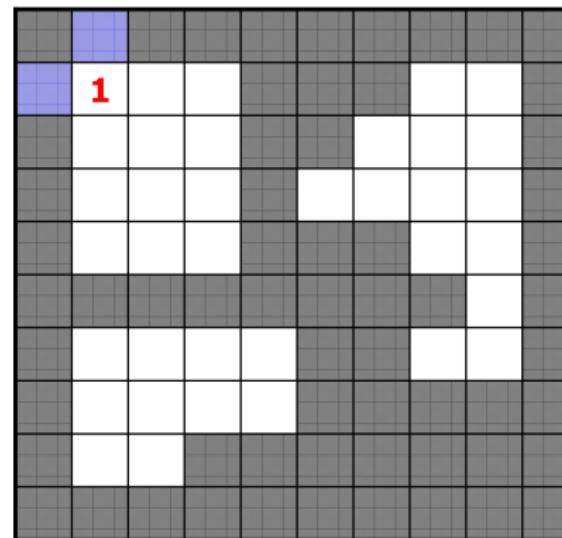
Étiquetages de composantes connexes

- Premier parcours : de gauche à droite et de haut vers le bas
- On affecte à chaque pixel d'une région la plus petite étiquette parmi ses voisins haut et gauche ou bien une nouvelle étiquette.



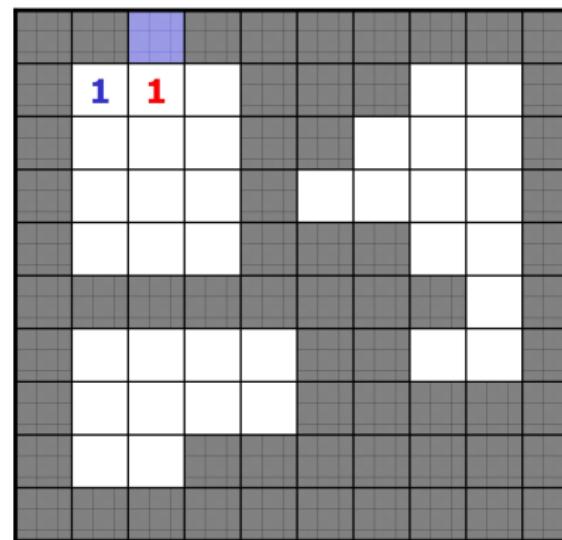
Étiquetages de composantes connexes

- Premier parcours : de gauche à droite et de haut vers le bas
- On affecte à chaque pixel d'une région la plus petite étiquette parmi ses voisins haut et gauche ou bien une nouvelle étiquette.



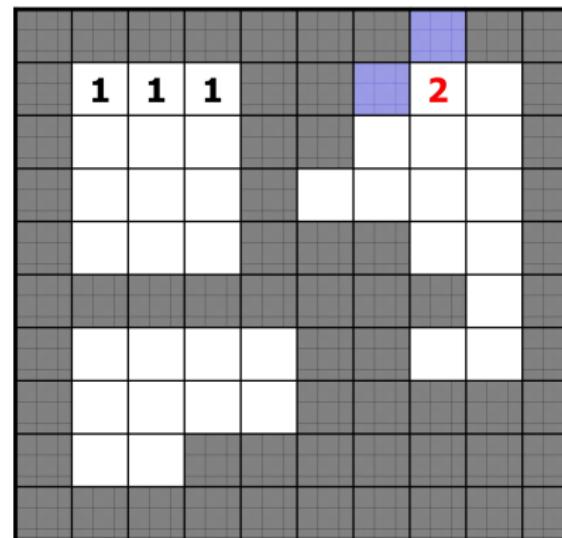
Étiquetages de composantes connexes

- Premier parcours : de gauche à droite et de haut vers le bas
- On affecte à chaque pixel d'une région la plus petite étiquette parmi ses voisins haut et gauche ou bien une nouvelle étiquette.



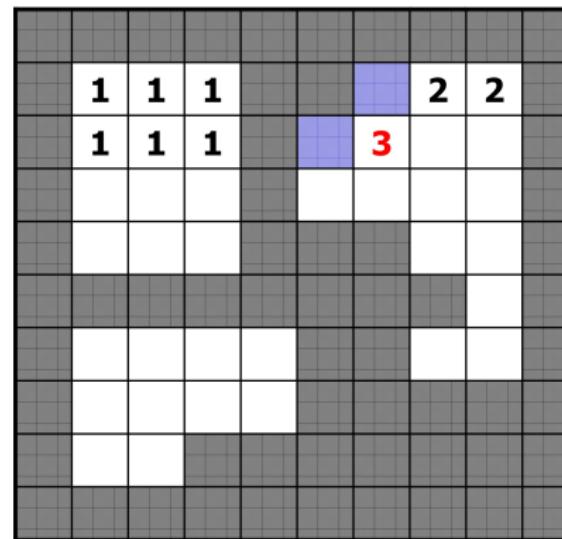
Étiquetages de composantes connexes

- Premier parcours : de gauche à droite et de haut vers le bas
- On affecte à chaque pixel d'une région la plus petite étiquette parmi ses voisins haut et gauche ou bien une nouvelle étiquette.



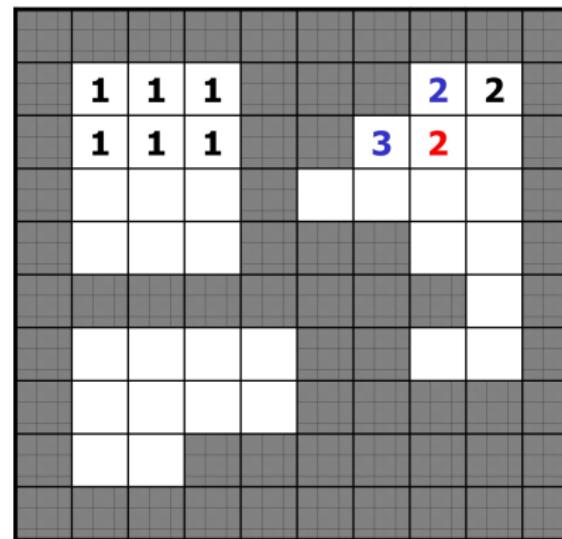
Étiquetages de composantes connexes

- Premier parcours : de gauche à droite et de haut vers le bas
- On affecte à chaque pixel d'une région la plus petite étiquette parmi ses voisins haut et gauche ou bien une nouvelle étiquette.



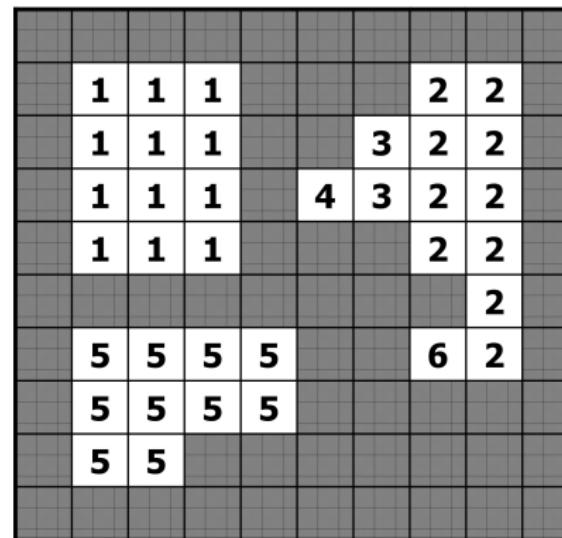
Étiquetages de composantes connexes

- Premier parcours : de gauche à droite et de haut vers le bas
- On affecte à chaque pixel d'une région la plus petite étiquette parmi ses voisins haut et gauche ou bien une nouvelle étiquette.



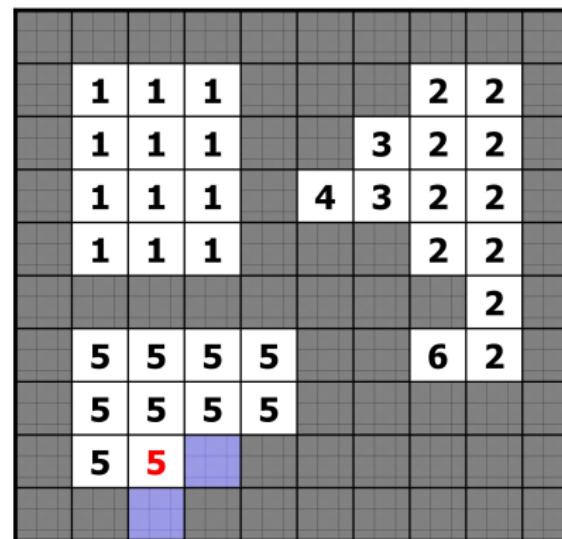
Étiquetages de composantes connexes

- Premier parcours : de gauche à droite et de haut vers le bas
- On affecte à chaque pixel d'une région la plus petite étiquette parmi ses voisins haut et gauche ou bien une nouvelle étiquette.



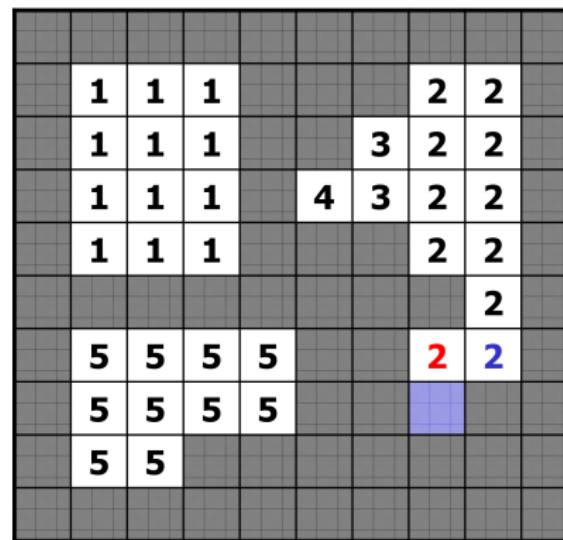
Étiquetages de composantes connexes

- Deuxième parcours : de droite à gauche et d'en bas vers le haut
- On affecte à chaque pixel d'une région la plus petite étiquette parmi la sienne et celles ses voisins bas et droite.



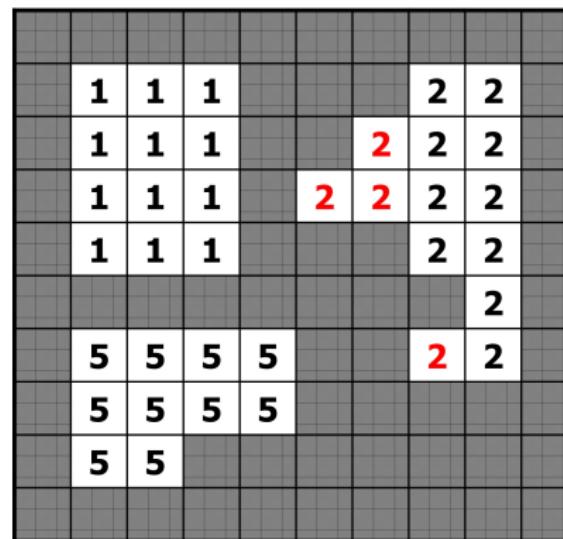
Étiquetages de composantes connexes

- Deuxième parcours : de droite à gauche et d'en bas vers le haut
- On affecte à chaque pixel d'une région la plus petite étiquette parmi la sienne et celles ses voisins bas et droite.



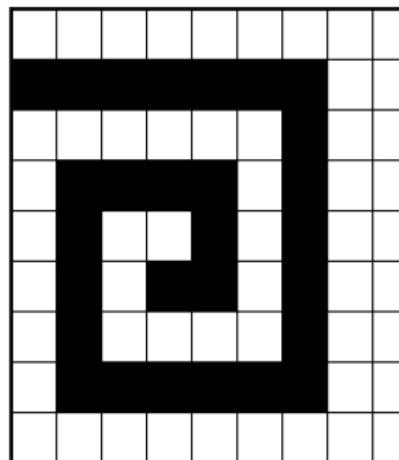
Étiquetages de composantes connexes

- Deuxième parcours : de droite à gauche et d'en bas vers le haut
- On affecte à chaque pixel d'une région la plus petite étiquette parmi la sienne et celles ses voisins bas et droite.



Étiquetages de composantes connexes

- En deux parcours, nous avons terminé l'étiquetage des régions. Parfois, il faut plus de deux parcours. Exemple : région en spirale !
- On continue les parcours, dans un sens puis l'autre, jusqu'à ce qu'il n'y ait plus de changement d'étiquettes



Étiquetages de composantes connexes

- Il est possible de ne faire qu'un seul parcours
 - Gestion d'une table d'équivalence d'étiquettes
 - Mise à jour récursive des étiquettes lorsque 2 étiquettes se "rencontrent"
- Pour étiqueter des contours, on peut utiliser le même algorithme (avec la 8-connexité)

Classification

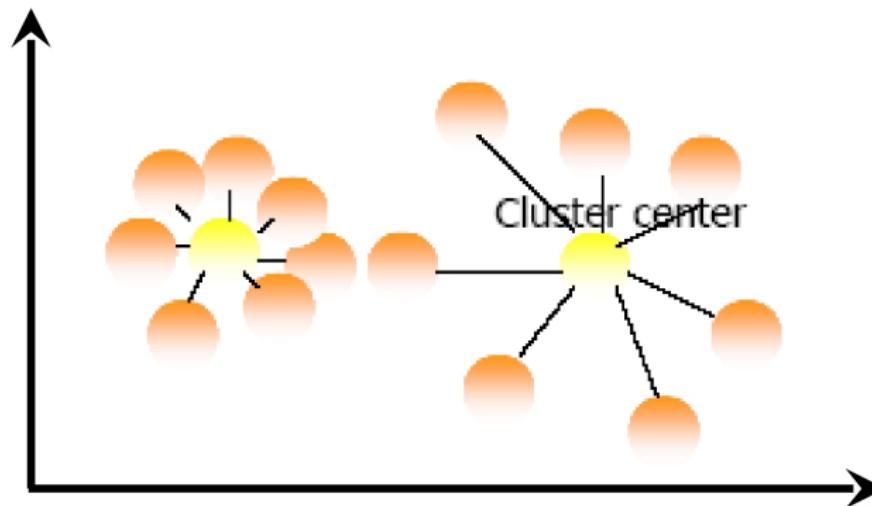
- Classification =répartition d'un ensemble d'objets (pixels) en différents groupes (classes) en fonctions de leurs caractéristiques ou features (Niveau de gris, couleur, gradient, statistique locale, ...)
- Types de classification
 - Supervisée : les caractéristiques des classes sont connues a priori. Exp, : distance minimale, k-nearest neighbors, statistiques (distributions de probabilité des modèles), ...
 - Non-supervisée (regroupement - clustering-) : La classification est faites sur les données et à partir des données directement.

Classification

- The k-Nearest Neighbors Classifier
 - Les classes sont définies par quelques éléments dits de référence.
 - Pour un nouveau objet (pixel), on détermine les k plus proches objets de référence.
 - Chaque nouvel objet est associé à la classe de la majorité de ses voisins.
- The Minimum-Distance-to-Mean Classifier
 - Les classes sont définies par leurs centres.
 - Chaque nouvel objet est associé à la classe de plus proche centre.

Classification : K-means

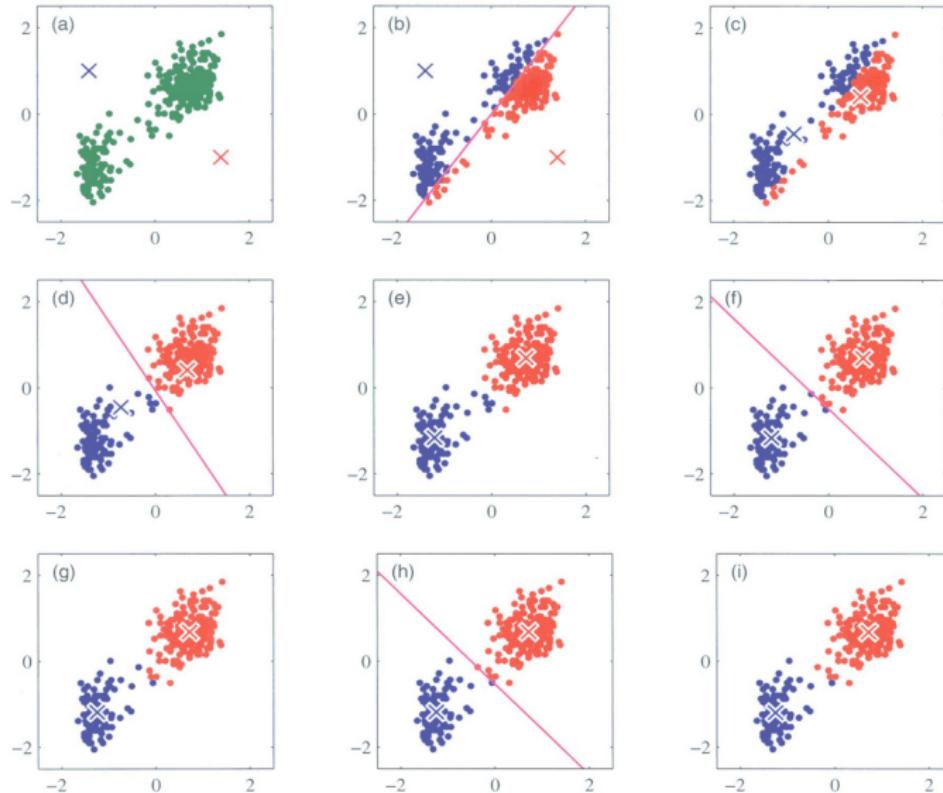
- On veut diviser les points en k groupes (clusters)
 - k est donné à l'avance (un paramètre de l'algorithme)
 - On définit le centre d'un groupe comme la moyenne des éléments (pixels) du groupe



Classification : K-means

- 1. Partitionnement des données en k sous-ensembles (non vides)
- 2. Calcul des centres des groupes de la partition courante
- 3. Les données sont affectées au groupe dont le centre leur est le plus proche
- 4. Retour à l'étape 2
- Arrêt lorsque les groupes sont \sim constants

Classification



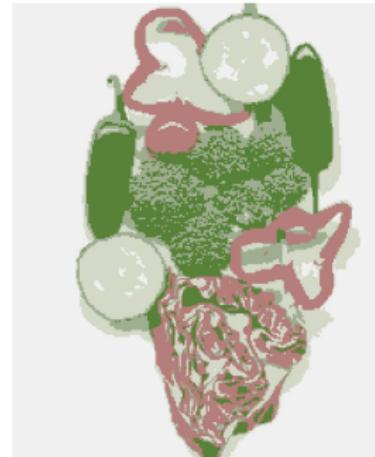
Classification



*Image
source*



*K-moyennes
sur
l'intensité*



*K-moyennes
sur la
couleur*