



Traitement Numérique de l'Image

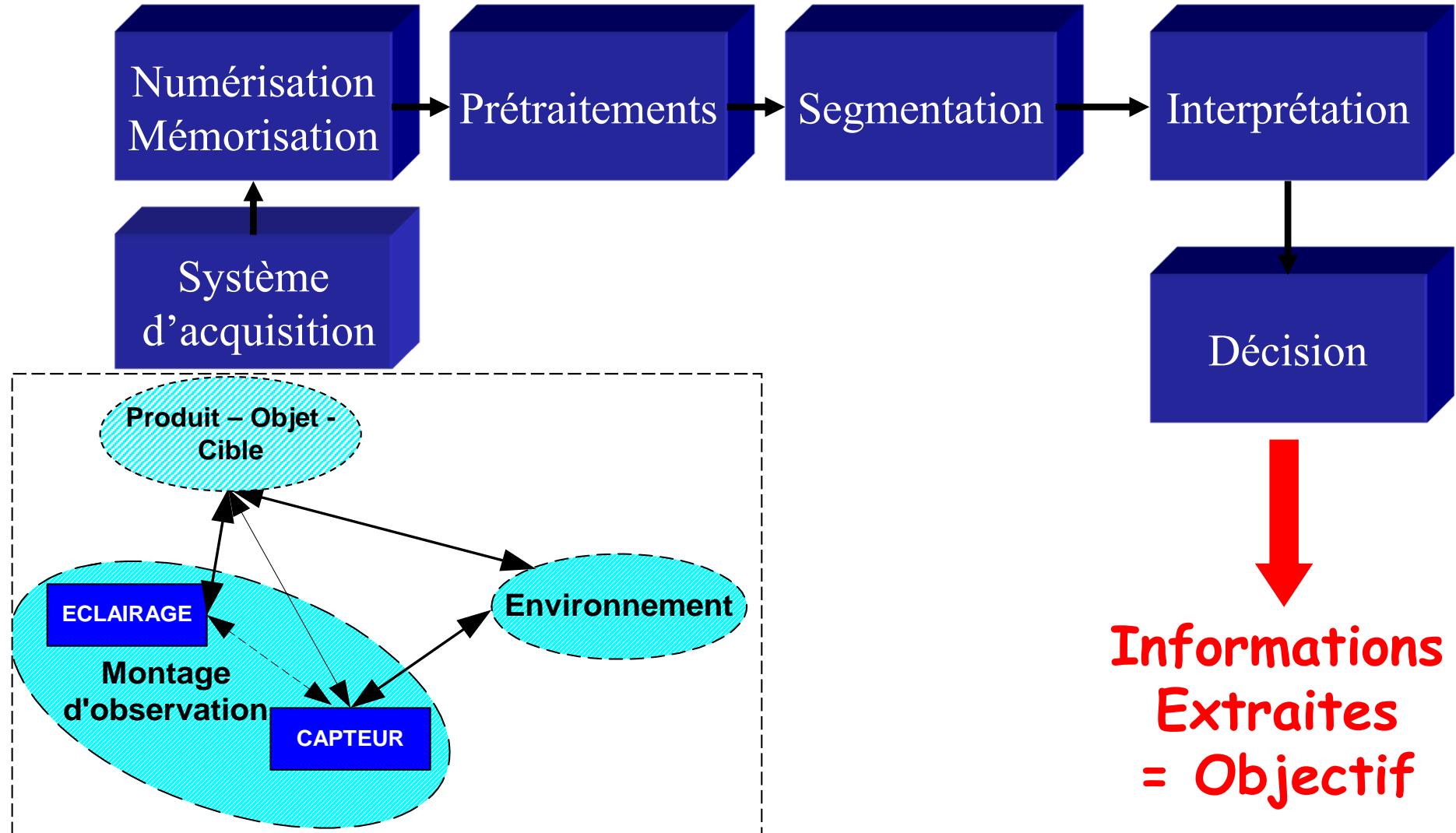
Traitements de Haut Niveau:
Segmentation - Extraction d'informations - Décision

TELECOM Nancy 2A

Vincent Bombardier
(MdC HC 61ème Section)

*Centre de Recherche en Automatique de Nancy -UMR CNRS 7039-
Département: Ingénierie des Systèmes Eco-Technique
Projet Systèmes Intelligents Ambients*

Segmentation - Décision : Modèle de vision artificielle

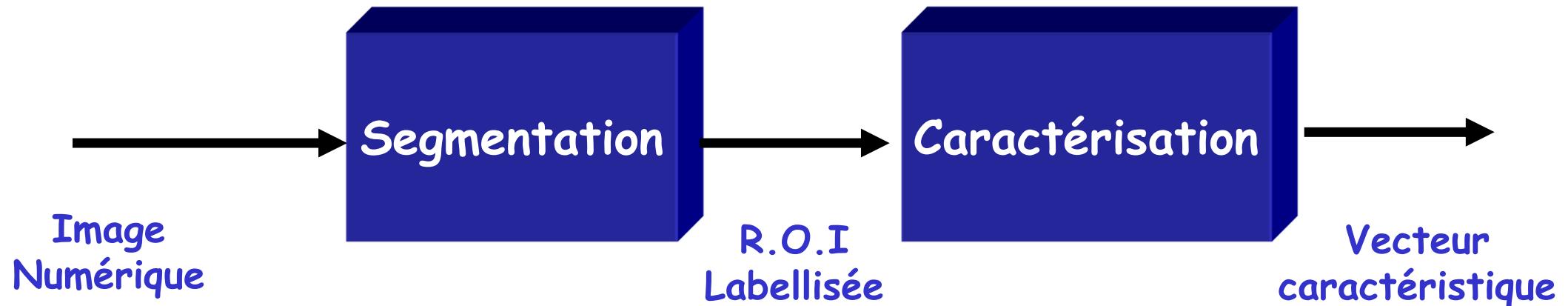


Segmentation - Décision : Définitions

➤ Etape Complexe:

↳ En fait regroupe plusieurs opérations:

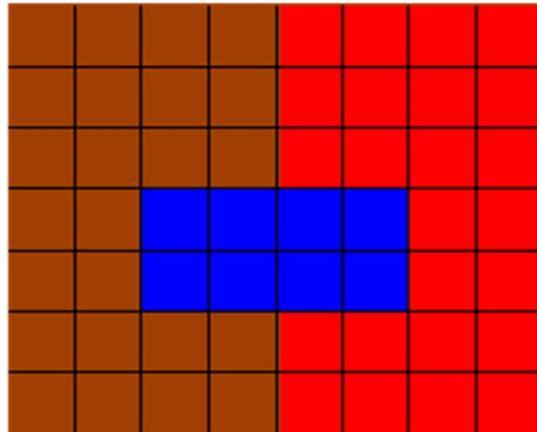
- Sélection de la région ou des objets recherchés
 - Region Of Interest
- Représentation de la segmentation
 - Labellisation :
 - » A chaque point d'un même groupe, on assigne la même valeur, qu'on appelle label ou étiquette.
 - Graphe d'adjacence
- Calcul des caractéristiques de la région sélectionnée



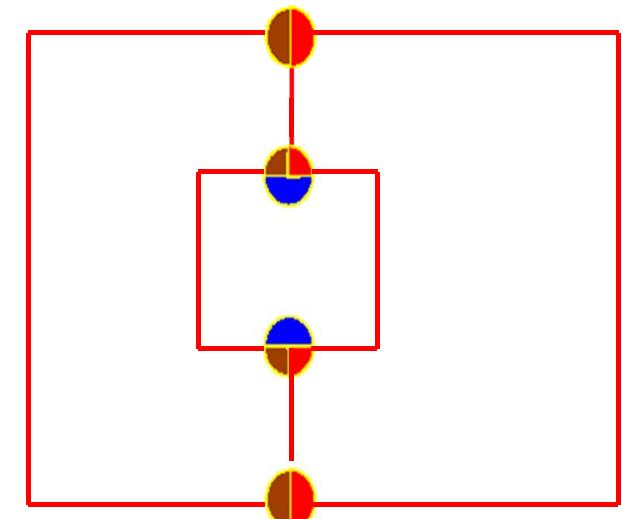
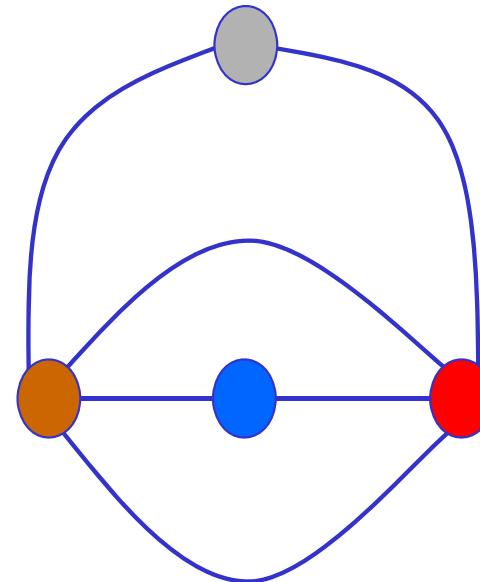
Segmentation - Décision : Définition

➤ Représentation de la segmentation:

Labels



Graphes d'adjacence

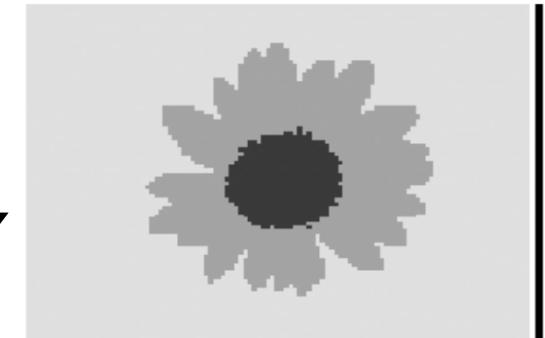


Segmentation - Décision : Modèle de vision artificielle

- Etape clé du traitement d'images.
- La segmentation vise à sous-diviser l'image en constituants distincts - objets
 - ↳ On appellera segmentation la tâche qui consiste à cataloguer chaque pixel comme appartenant à un certain groupe.
 - Création d'une partition de l'image I en s ensembles R_i de Régions non vides
 - L'union des régions forme l'image I
 - Un pixel appartient à une et une seule région ($R_i \cap R_j = \emptyset$)



Image Numérique



Partition de l'image
Numérique

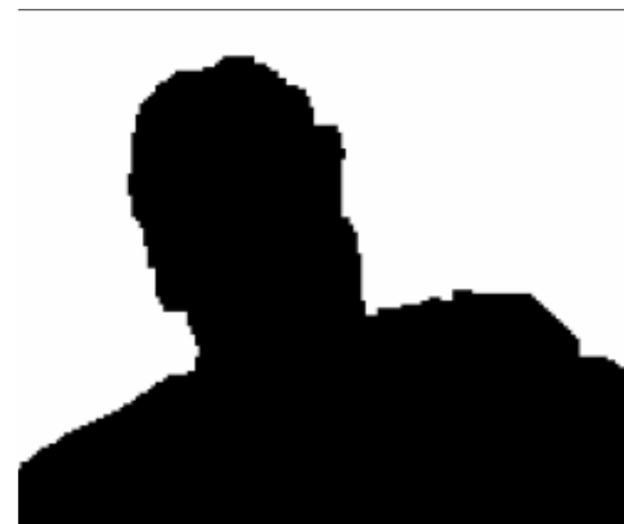
Segmentation - Décision : Objectifs de la segmentation

- Extraire (séparer) les entités d'une image
 - ↳ Pour y appliquer un traitement spécifique
 - ↳ Pour interpréter le contenu de l'image
- Dans la pratique :
 - ↳ construire une image de masques
 - ↳ Chaque masque est une composante connexe



Segmentation - Décision : Objectifs

- Les masques permettent d'interpréter le contenu de l'image
 - ↳ On traite chaque zone de l'image séparément



Segmentation - Décision : Objectifs

➤ On cherche des régions

↳ Connexes

↳ Homogènes en intensité

↳ Délimités par des contours nets

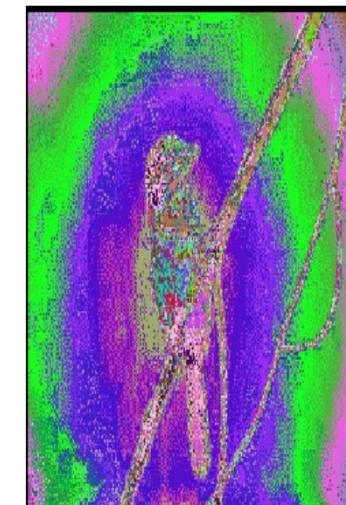
↳ Les plus grandes possible



Origine



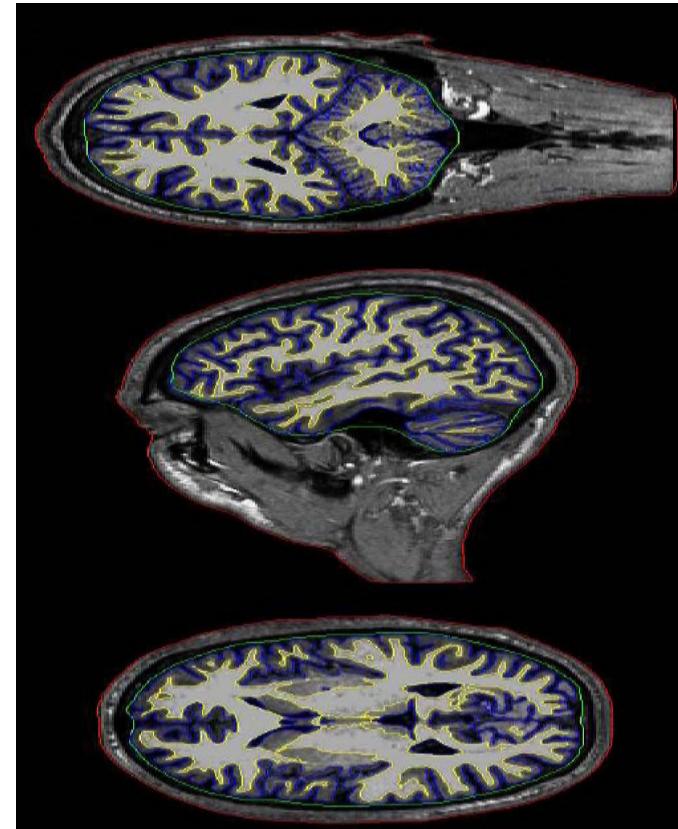
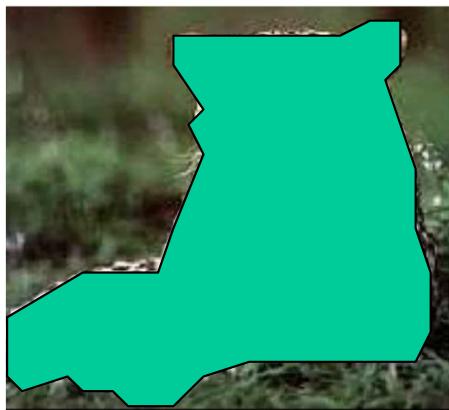
Bonne Segmentation



Mauvaise Segmentation

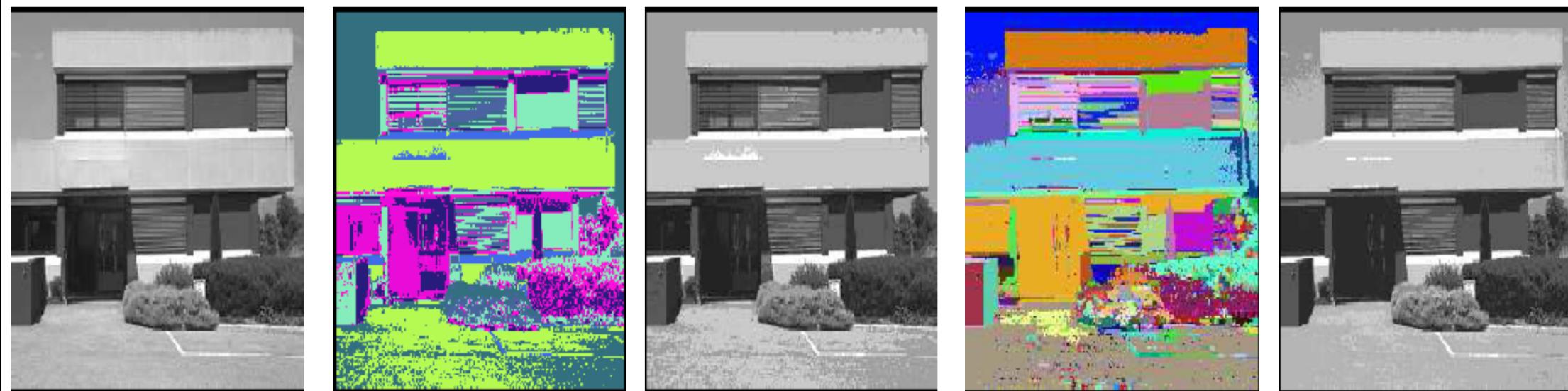
Segmentation - Décision : Difficultés

- Dans les problèmes de segmentation, il s'agit de diviser l'image selon une certaine sémantique, d'où la difficulté de donner une définition précise.
 - ↳ La segmentation est liée à la reconnaissance.



Segmentation - Décision : Segmentation vs Quantification

- La **quantification** s'intéresse à réduire le nombre d'intensités utiles, en regroupant les intensités voisines. En revanche, il n'y a aucune considération de la proximité des pixels.
- La **segmentation** vise à extraire des régions connexes d'intensité cohérente. On peut attribuer sur chaque région l'intensité moyenne sur la région ce qui donne un résultat visuellement voisin de la quantification. On ne peut pas prédire le nombre de couleurs qui seront utilisées



Segmentation - Décision : Les différentes approches

↳ Approche Contour

- Recherche de la frontière de la ROI
 - Discontinuité entre deux ensembles connexes de pixels
 - Fermeture des contours obtenus

↳ Approche Région

- Recherche d'ensembles de pixels connexes ou voisins ayant des propriétés communes
 - Similitude / homogénéité (texture, couleur, intensité)
 - Division / Fusion (split and merge)

↳ Approche par Classification de Pixels

- Extraction de classes de pixels
 - 2 classes -> Binarisation / Seuillage
 - Classification de pixels (méthodes proches Rdf..)

Segmentation - Décision : Les différentes approches

- Il existe une dualité entre régions et contours :
 - ↳ Une région est délimitée par un contour
 - ↳ Un contour sépare deux régions



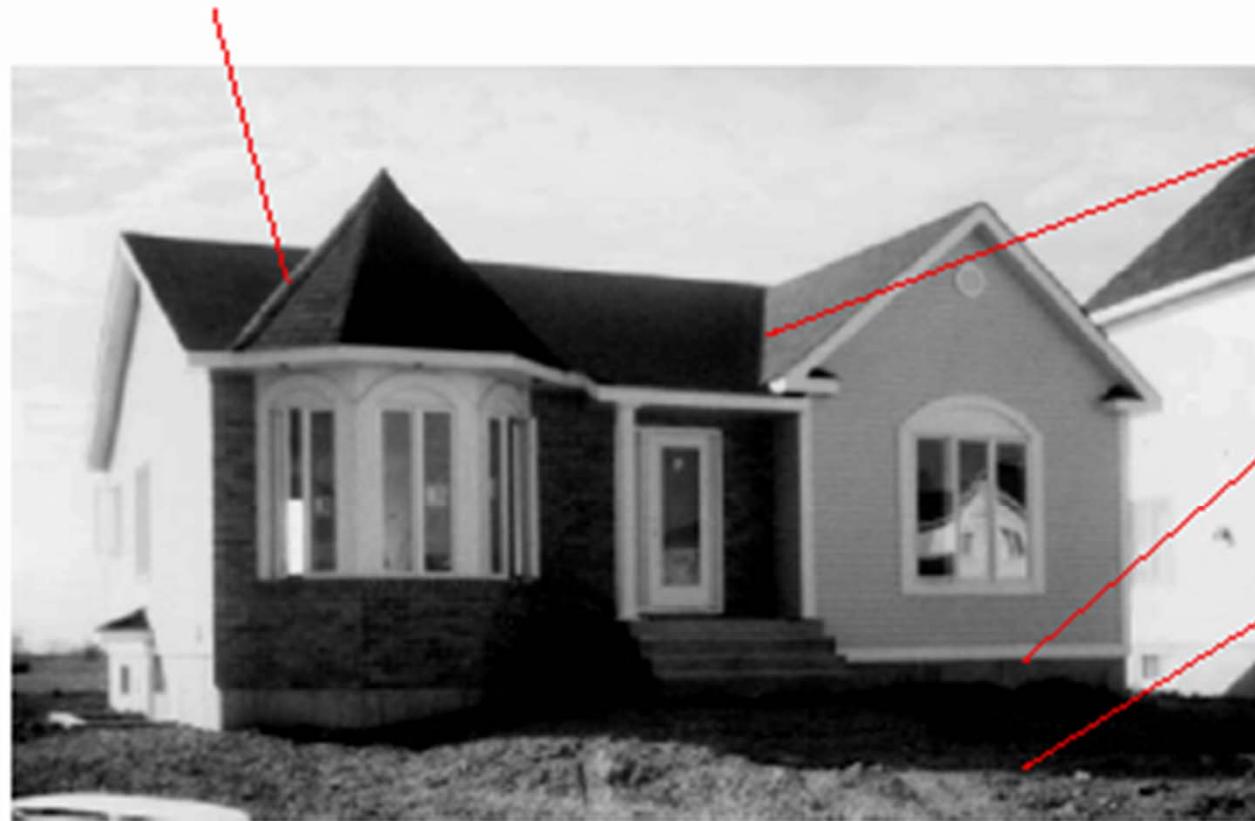
Segmentation - Décision : Approche Contours

- Par définition, un contour est la frontière qui sépare deux objets dans une image:
 - ↳ Une discontinuité de l'image
- Les opérateurs classiques détectent toutes les lignes marquant des changements d'intensité
 - ↳ Pas seulement les contours !
 - ↳ Abus de langage sur la notion de contours !
- Segmentation en contours:
 - ↳ Détections des morceaux de lignes dans l'image
 - ↳ Une segmentation sera complète si les contours sont fermés et d'épaisseur minimale

Segmentation - Décision : Approche Contours

- Détection des discontinuités : exemples

de profondeur



d'orientation
de surface

de réflectance

d'illumination

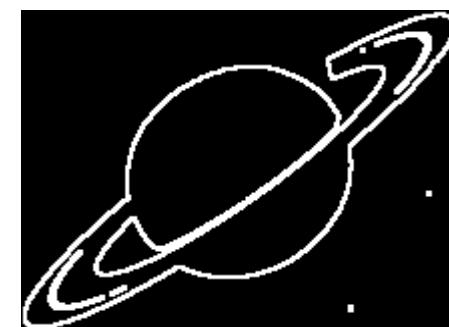
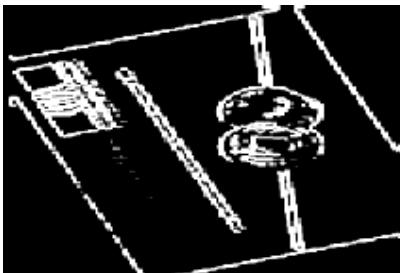
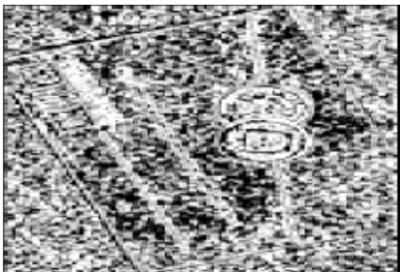
Segmentation - Décision : Approche contours

- Après l'application d'un opérateur de détection de contours (discontinuités), on doit pour obtenir une segmentation complète de l'image :
 - ↳ Choisir les pixels appartenant aux contours
 - ↳ Fermer les contours sélectionnés.



Segmentation - Décision : Approche contours

- La sélection des points de contours dépend fortement de :
 - ↳ l'opérateur de détection utilisé,
 - ↳ du contenu de l'image et des ses conditions d'acquisition
 - ↳ de l'objectif visé.



Segmentation - Décision de sélection des contours : Gradient

- La plupart de ces opérateurs (sauf Roberts) lisse l'image afin d'obtenir un meilleur résultat.
- 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
- Un opérateur de détection de contour n'est qu'une première étape dans la chaîne de segmentation.
- Il faudra utiliser d'autres techniques pour sélectionner les contours obtenus avec ces opérateurs et pour les compléter (approx polygonal)
- Deux méthodes de sélection:
 - ↳ Seuillage du gradient :
 - Simple, Sensible au bruit et aux différences de contraste
 - ↳ Recherche du max de gradient puis seuillage
 - Plus long en temps de calcul, Plus précis

Segmentation - Décision : Sélection des contours : Gradient

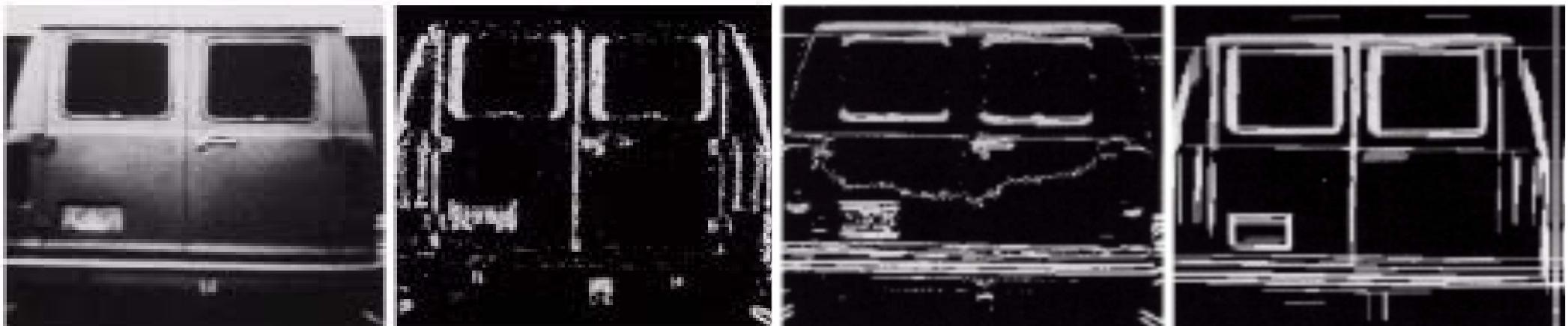
➤ Le gradient est composé de deux grandeurs

↳ Norme : intensité du gradient en chaque pixel

$$|G| = \sqrt{Gx^2 + Gy^2} \approx |Gx| + |Gy|$$

↳ Direction : la direction de gradient le plus fort pour chaque pixel

$$\theta = \arctan(Gy / Gx)$$



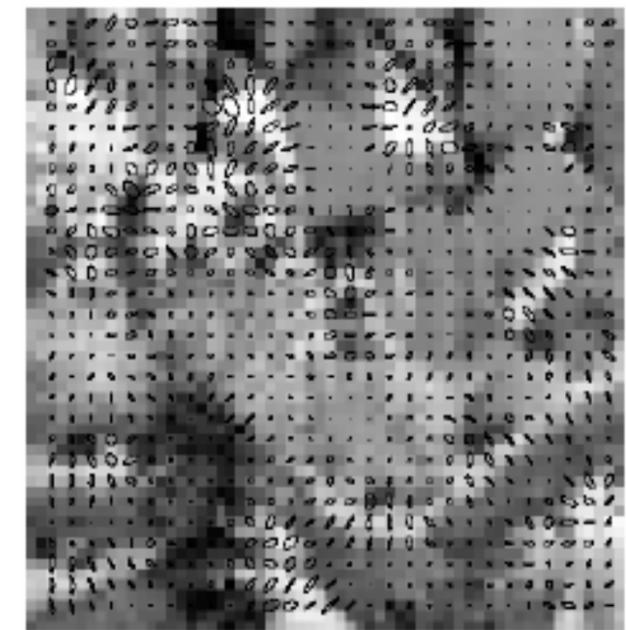
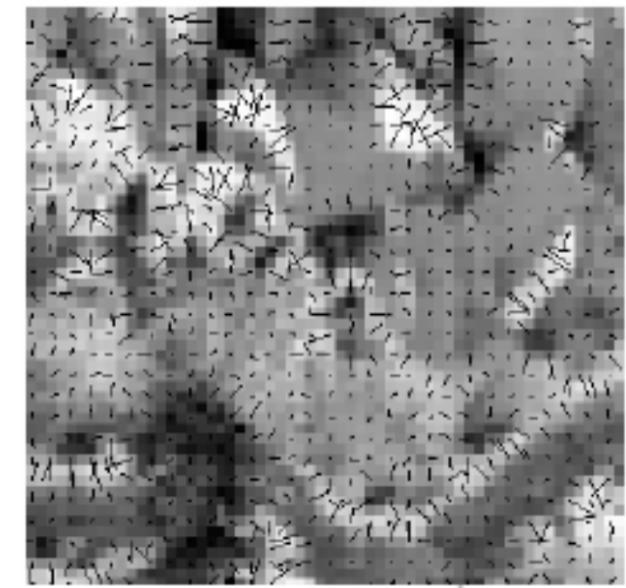
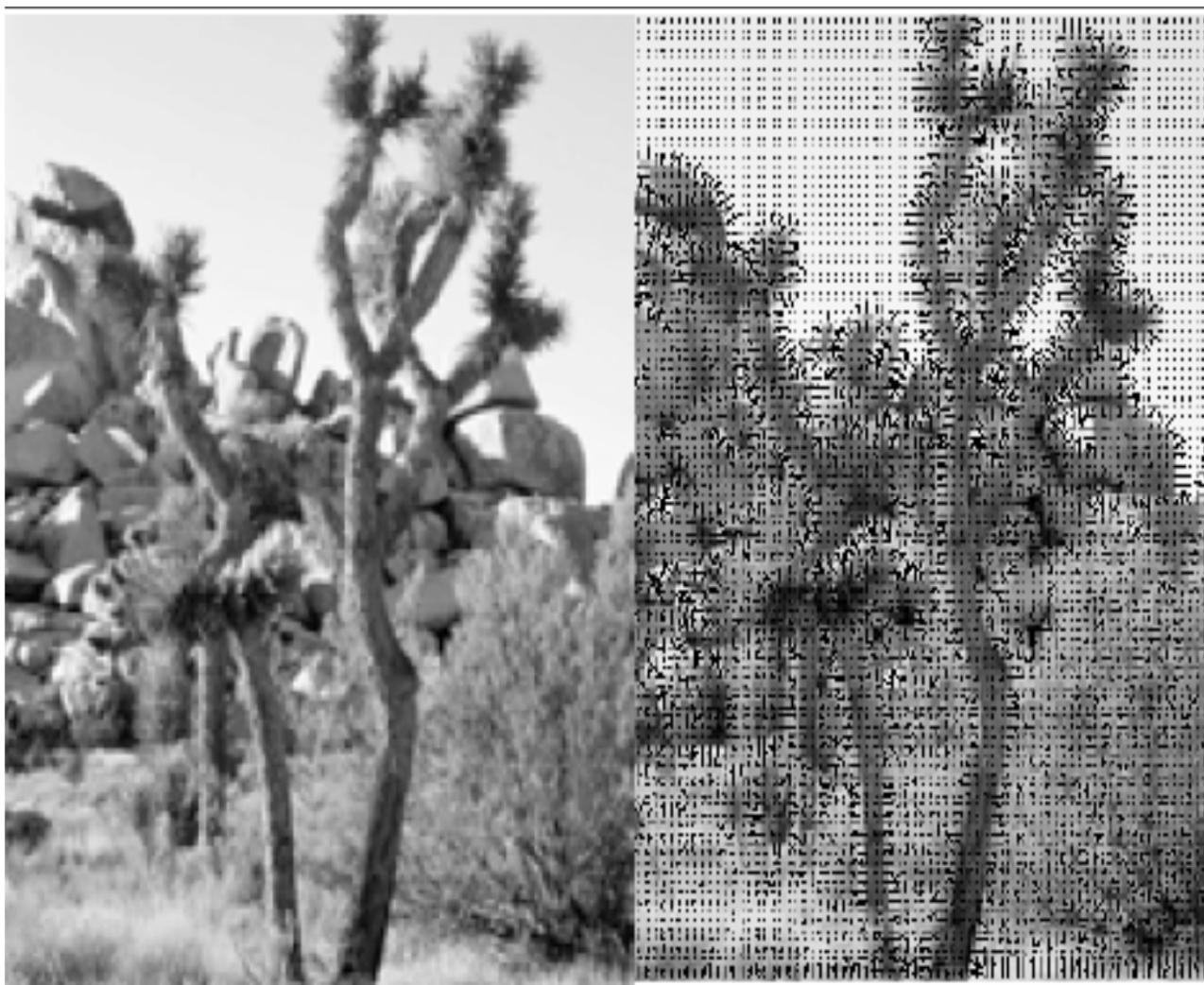
$$\frac{\partial f}{\partial x}$$

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

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

Segmentation - Décision : Sélection des contours : Gradient

- On peut tracer les deux composantes sur l'image
- Vecteur pour chaque pixel (norme + direction)



Segmentation - Décision : Sélection des contours : Gradient

➤ Seuillage du gradient :

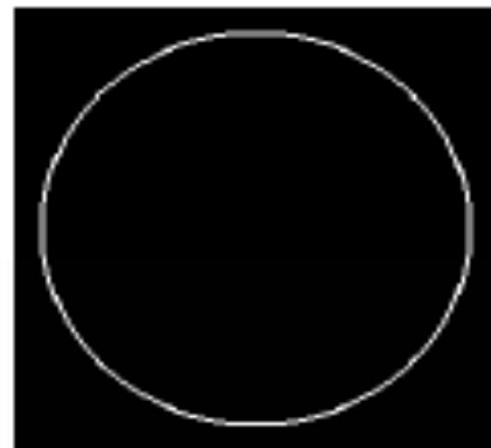
- ↳ Problème si seuillage global
- ↳ Possibilité de seuillage local, au voisinage du contours ou par suivi de contour
- ↳ Epaisseur et continuité des Contours !!



u



$|\nabla u|$



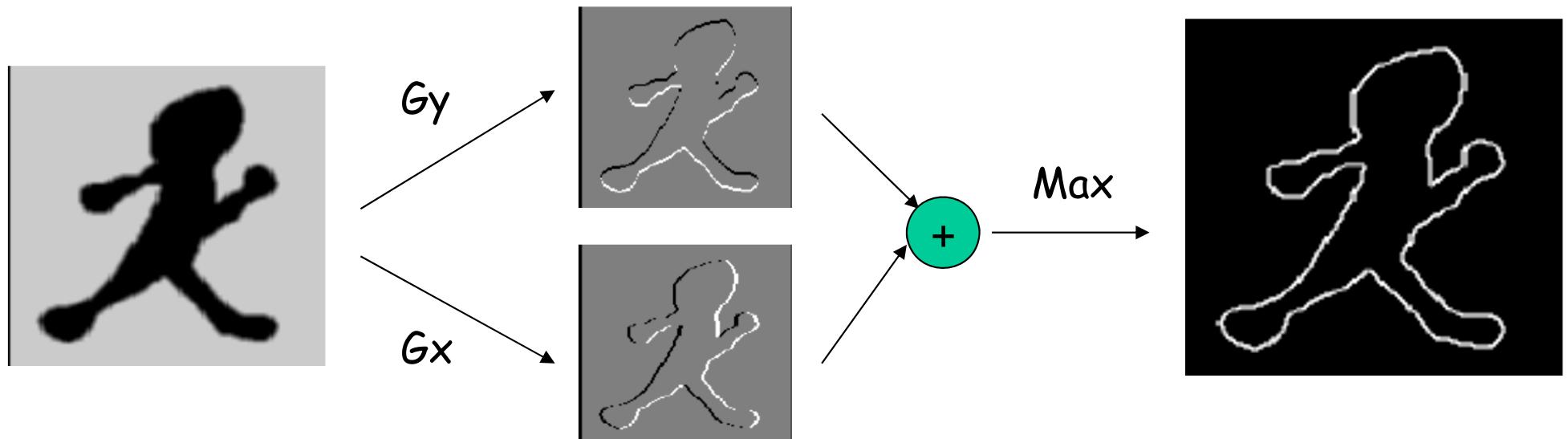
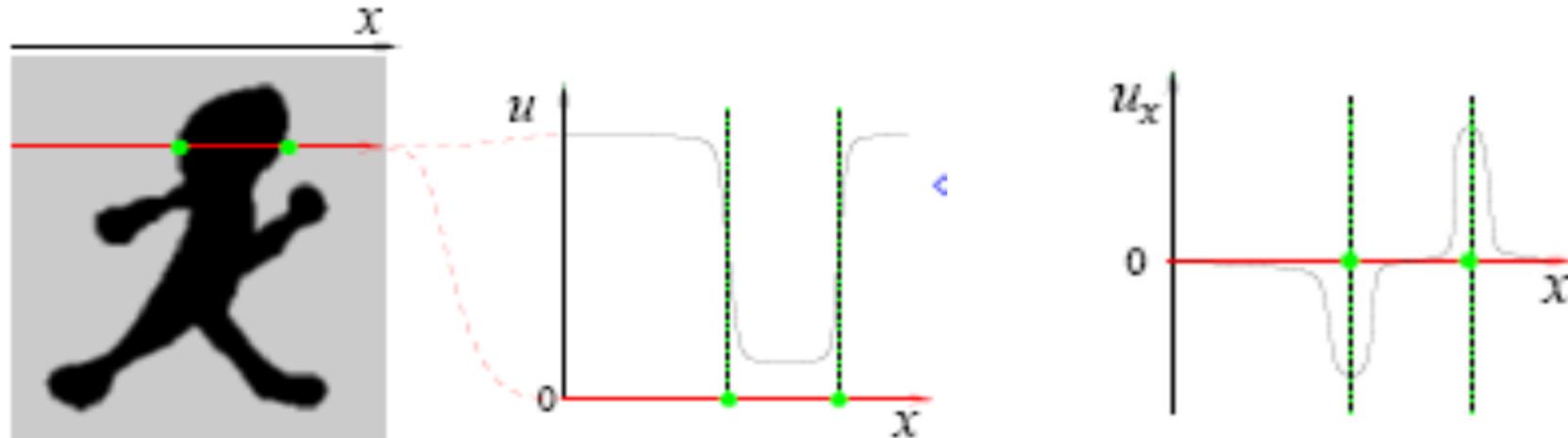
$|\nabla u| \text{ seuil}=80$



$|\nabla u| \text{ seuil}=30$

Segmentation - Décision Sélection des contours : Gradient

➤ Calcul du max du gradient :

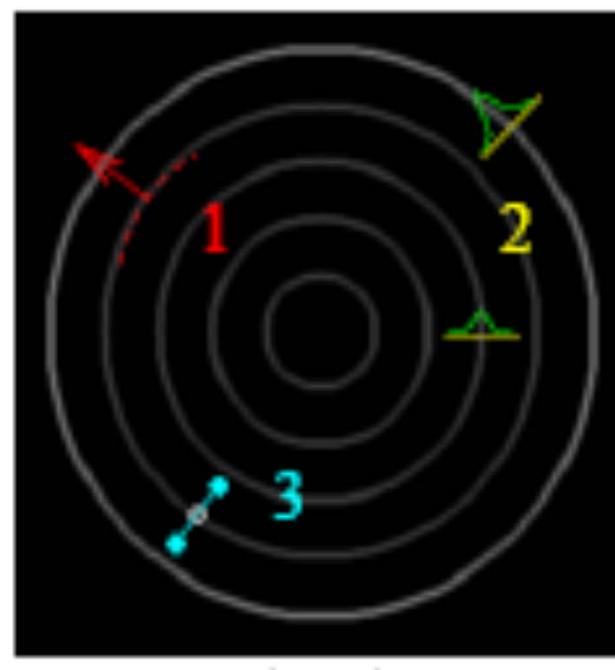


Segmentation - Décision : Sélection des contours : Gradient

- Calcul du max du gradient dans la direction perpendiculaire à l'orientation du contour :
 - ↳ La recherche est faite dans le voisinage des pixels de forts gradients



u



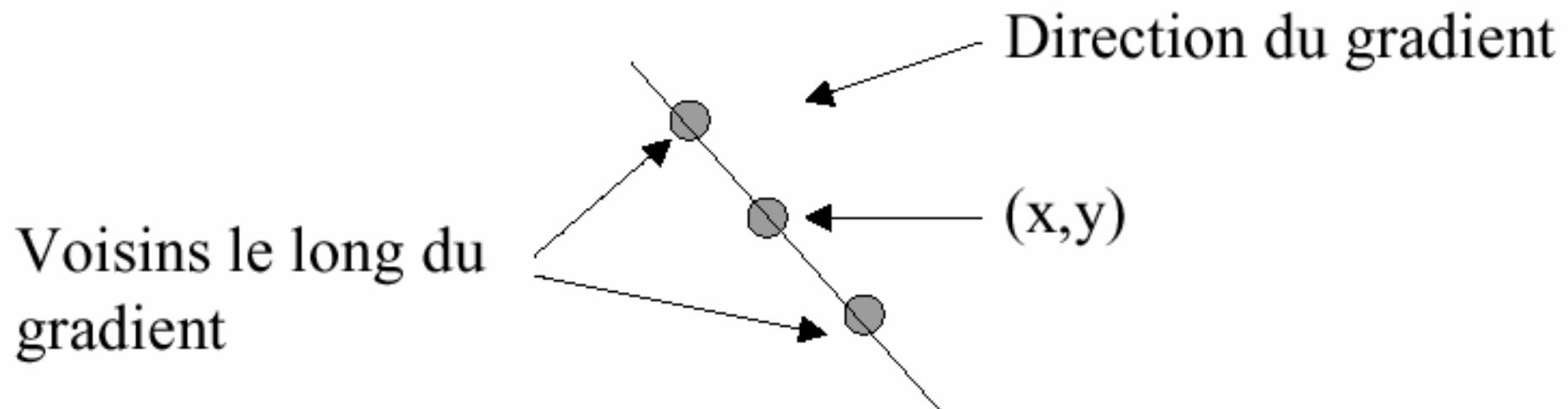
$|\nabla u|$



résultat

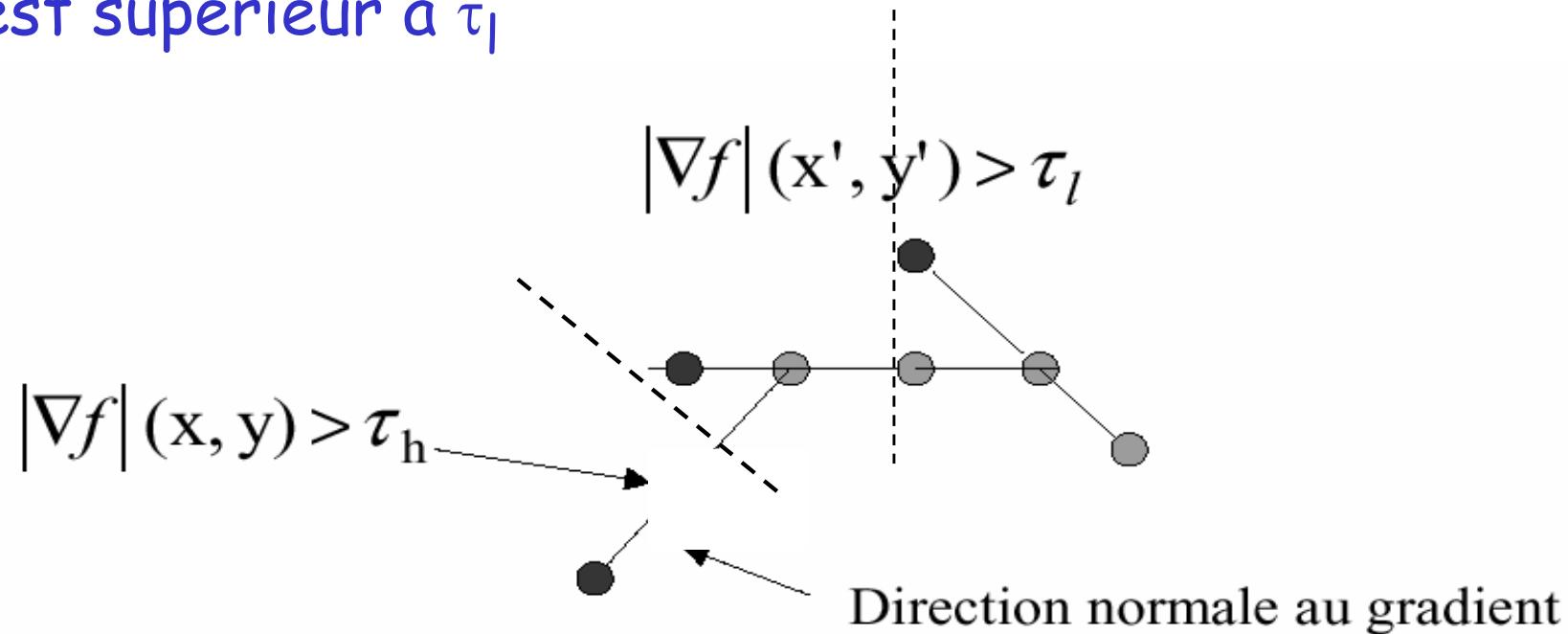
Segmentation - Décision : Sélection des contours : Gradient

- Élimination des points de contours superflus :
 - ↳ Un pixel (x,y) est supprimé si sa norme de gradient est inférieure à la norme du gradient d'un de ses deux voisins dans la direction du gradient.



Segmentation - Décision : Sélection des contours : Gradient

- Seuillage par Hystérésis: deux seuils τ_h et τ_l tq $\tau_h > \tau_l$
 - ↳ Ajouter le pixel (x,y) au contour si sa norme de gradient est supérieure à τ_h
 - ↳ Ajouter au contour les pixels (x',y') reliés au pixel (x,y) le long de la normale au gradient si leur norme de gradient est supérieur à τ_l



Segmentation - Décision : Sélection des contours : Gradient

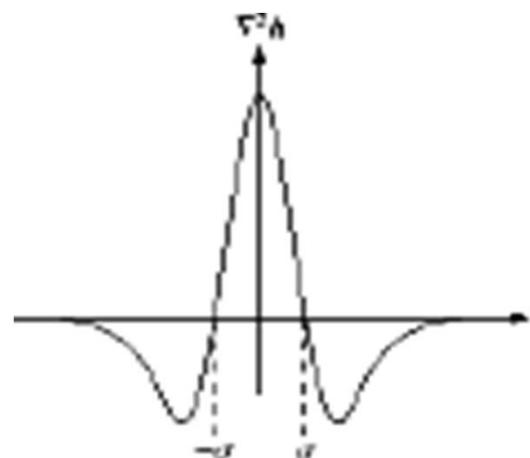


Segmentation - Décision : Sélection des contours : Laplacien

➤ Détection de contours par 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

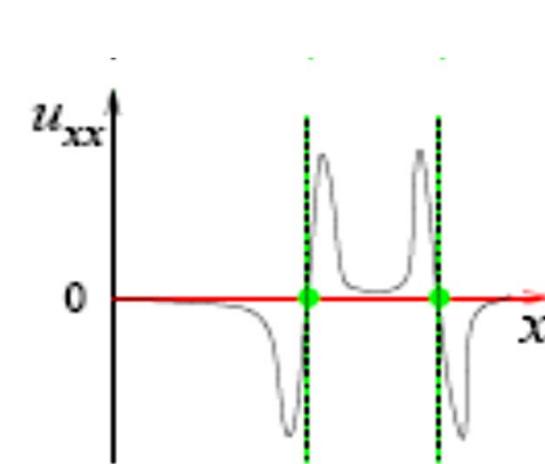
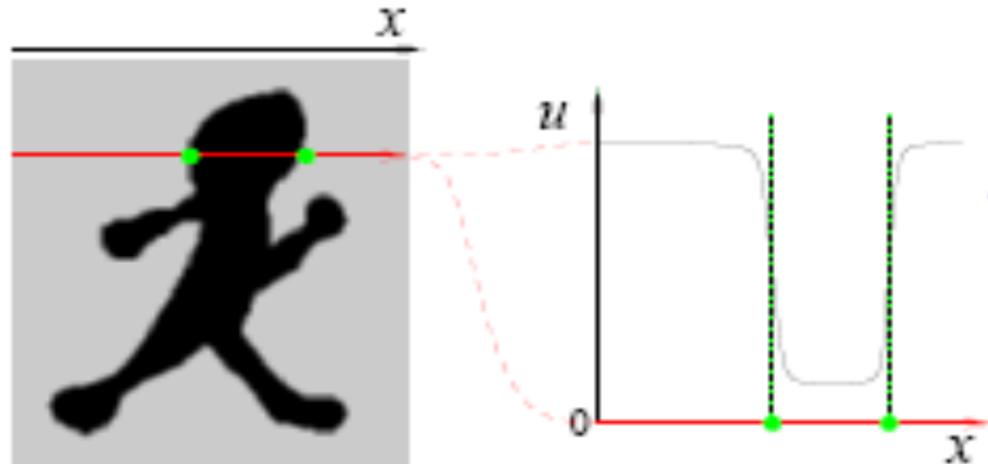
$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Segmentation - Décision Sélection des contours : Laplacien

➤ Calcul du passage par Zéro du laplacien :



ΔI



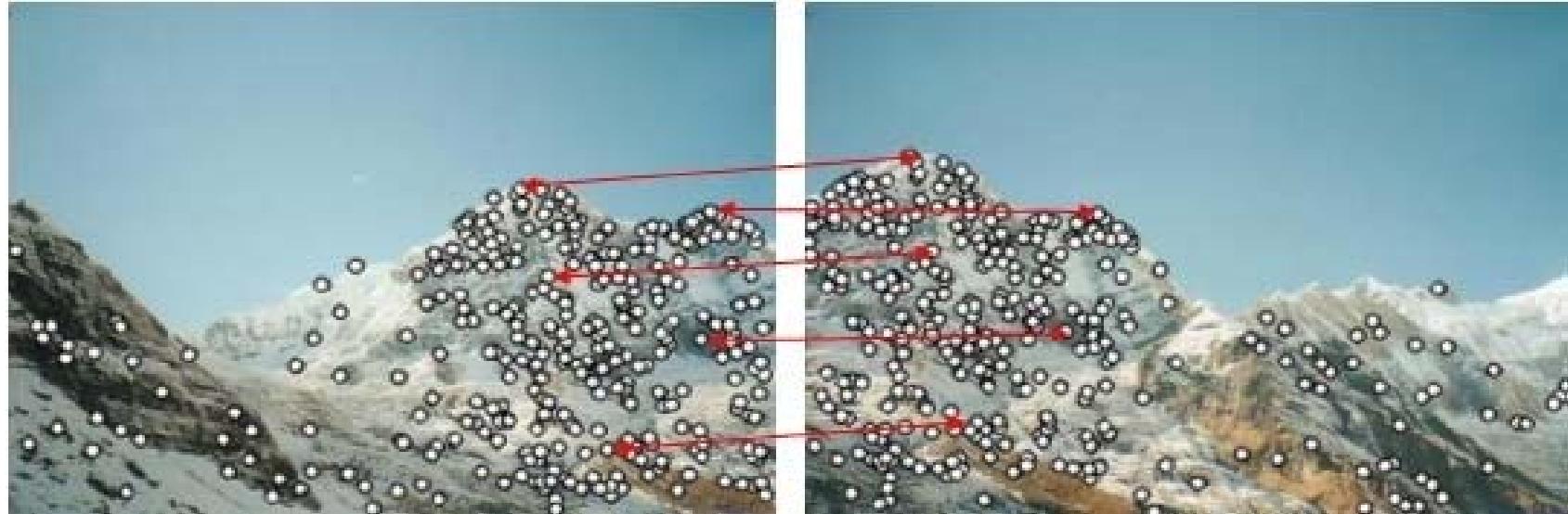
Zéro
Crossing



Segmentation - Décision DéTECTeur de coin de Harris

- Algorithme :
 - Calculer le gradient (I_x, I_y) dans toute l'image
 - Pour chaque pixel :
 - ✓ Calculer sur un voisinage du pixel la matrice suivante:
 - $$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$
 - ✓ Calculer les valeurs propres λ_1 et λ_2 (avec $\lambda_2 \leq \lambda_1$)
 - ✓ Si la valeur propre minimale $\lambda_2 >$ seuil, conserver les coordonnées du pixel dans une liste L (\rightarrow coins)
 - Trier L en ordre décroissant de λ_2
 - Balayer la liste L de haut en bas, pour chaque pixel p_i de la liste, éliminer les autres pixels qui appartiennent au voisinage de p_i
- La liste finale contient les points **saillants** pour lesquels $\lambda_2 >$ seuil et dont les voisinages ne se chevauchent pas.

Segmentation - Décision : Exemple d'utilisation : panorama

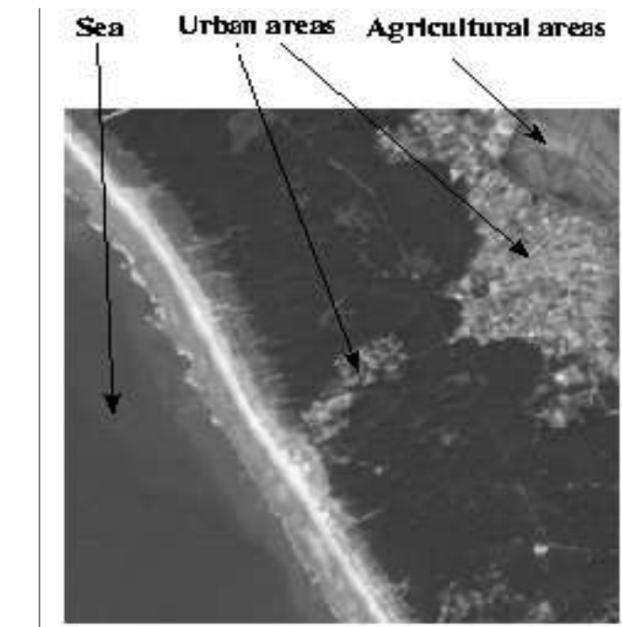
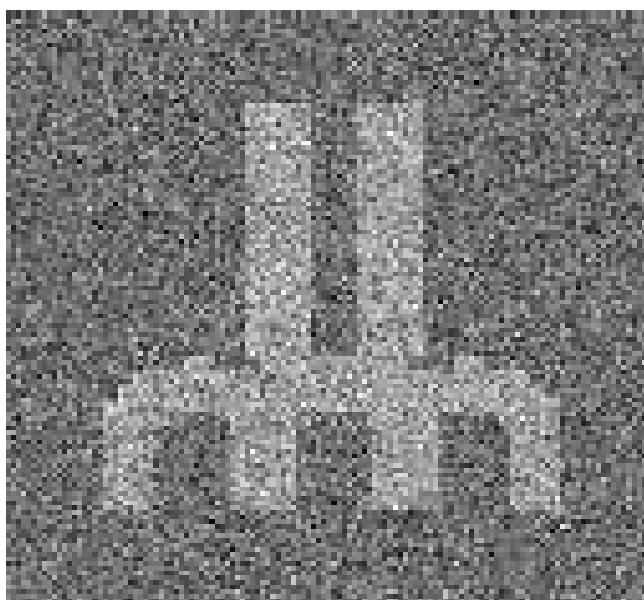


Segmentation - Décision : Approche Région

- Par définition, une région est un ensemble de pixel reliés par un critère d'homogénéité ou similitude:
 - ↳ Vérification d'un prédictat
- Il existe plusieurs méthodes de segmentation par les régions:
 - ↳ Division / Fusion (split and Merge)
 - ↳ Croissance de régions (region growing)
 - ↳ Ligne de partage des eaux (watershed line)
 - ↳ Exosquelette - Squelette par Zones d'Influence

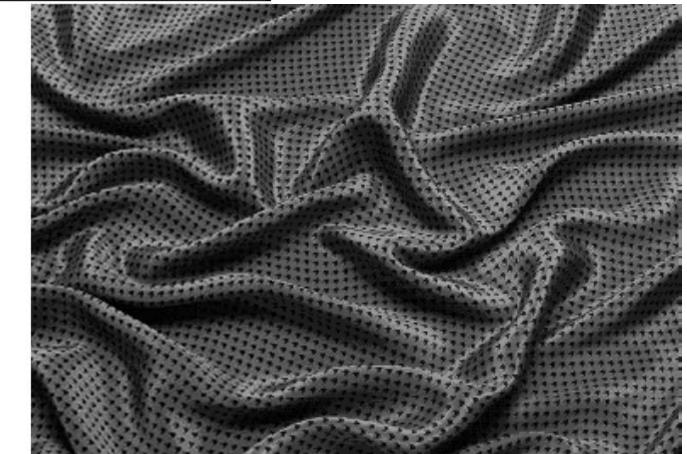
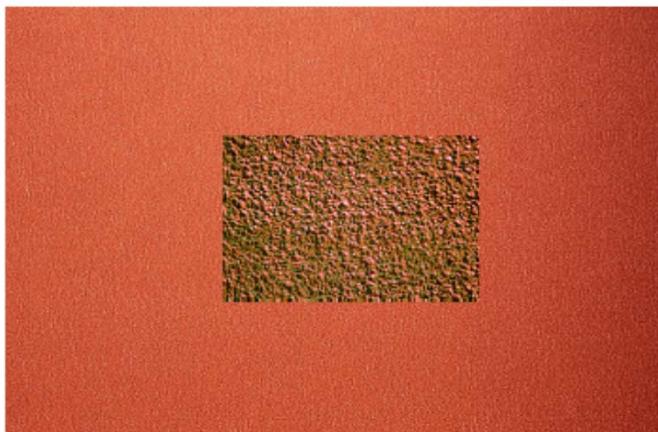
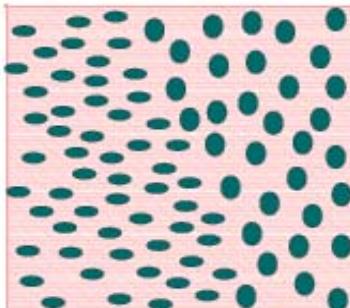
Segmentation - Décision : Similitude d'une région

- La similitude d'une région peut correspondre à différents critères:
 - ↳ Niveaux de Gris,
 - ↳ Couleurs,
 - ↳ Mouvements,
 - ↳ Textures, ...
- Il existe différents critères de regroupement de pixels



Segmentation - Décision : Similitude d'une région

- Notion de texture :
 - ↳ Région de l'image qui apparaît comme homogène pour un être humain
 - ↳ Répétition d'un motif de base dans différentes directions de l'espace
 - Caractéristiques spectrales : mur de briques, tissus, ...
 - ↳ Ne possédant pas de contours francs
 - Caractéristiques statistiques : Herbe, écorces, ...
 - ↳ Organisation de primitives (macro) ayant chacune un aspect aléatoire (micro)



Segmentation - Décision : Similitude d'une région

➤ Statistiques du 1er ordre:

$$\mu_1 = \sum_{k=0}^{255} k p_k \quad \text{Moyenne}$$

$$\mu_2 = \sigma^2 = \sum_{k=0}^{255} (k - \mu_1)^2 p_k \quad \text{Variance}$$

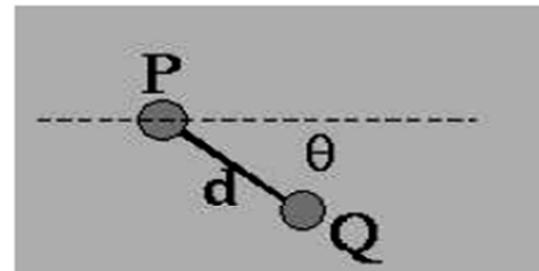
$$\mu_3 = \frac{1}{\sigma^3} \sum_{k=0}^{255} (k - \mu_1)^3 p_k \quad \text{Moment d'ordre 3}$$

$$\mu_4 = \frac{1}{\sigma^4} \sum_{k=0}^{255} (k - \mu_1)^4 p_k - 3 \quad \text{Moment d'ordre 4 (Kurtosis)}$$

p_k : Probabilité d'avoir un niveau de gris égale à k (cf. Histogramme)

Segmentation - Décision : Similitude d'une région

- Statistiques du 2nd ordre: matrice de co-occurrences



$$M(d, \theta) = [P_{i,j}]$$

$P_{i,j}$ = Probabilité[Luminance $P = i$, luminance $Q = j$]

$$\text{Énergie} = \sum_i \sum_j P_{i,j}^2$$

$$\text{Entropie} = H = - \sum_i \sum_j P_{i,j} \log P_{i,j}$$

$$\text{Probabilité max} = \max[P_{i,j}]$$

$$\text{Contraste} = C = \sum_i \sum_j |i - j|^2 P_{i,j}$$

Segmentation - Décision : Similitude d'une région

➤ Longueurs de plages : Luminance constante

$$\text{Longueur Moyenne} = \frac{1}{T} \sum_k \sum_l k q(k, l)$$

$$\text{Richesse en plages longues} = \frac{1}{T} \sum_k \sum_l k^2 q(k, l)$$

$$\text{Richesse en plages courtes} = \frac{1}{T} \sum_k \sum_l \frac{1}{k^2} q(k, l)$$

$$\text{Non-uniformité de luminance} = \frac{1}{T} \sum_k \left[\sum_l q(k, l) \right]^2$$

$$\text{Non-uniformité de longueur} = \frac{1}{T} \sum_l \left[\sum_k q(k, l) \right]^2$$

$$\text{Taux de plage} = \frac{1}{MN} \sum_k \sum_l q(k, l)$$

Avec $T = \sum_k \sum_l q(k, l)$

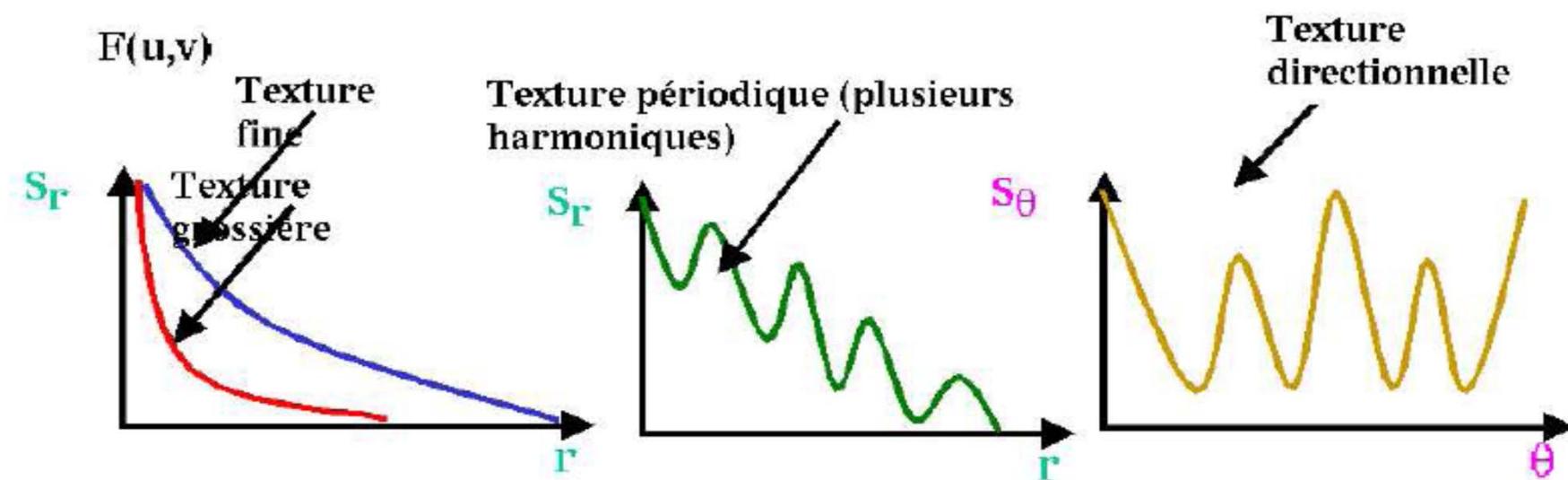
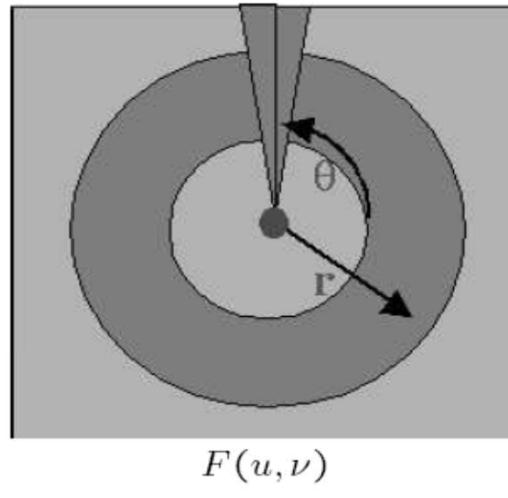
0	0	1	2
2	1	1	1
0	2	3	3
3	0	3	3

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

$\theta = 0^\circ$

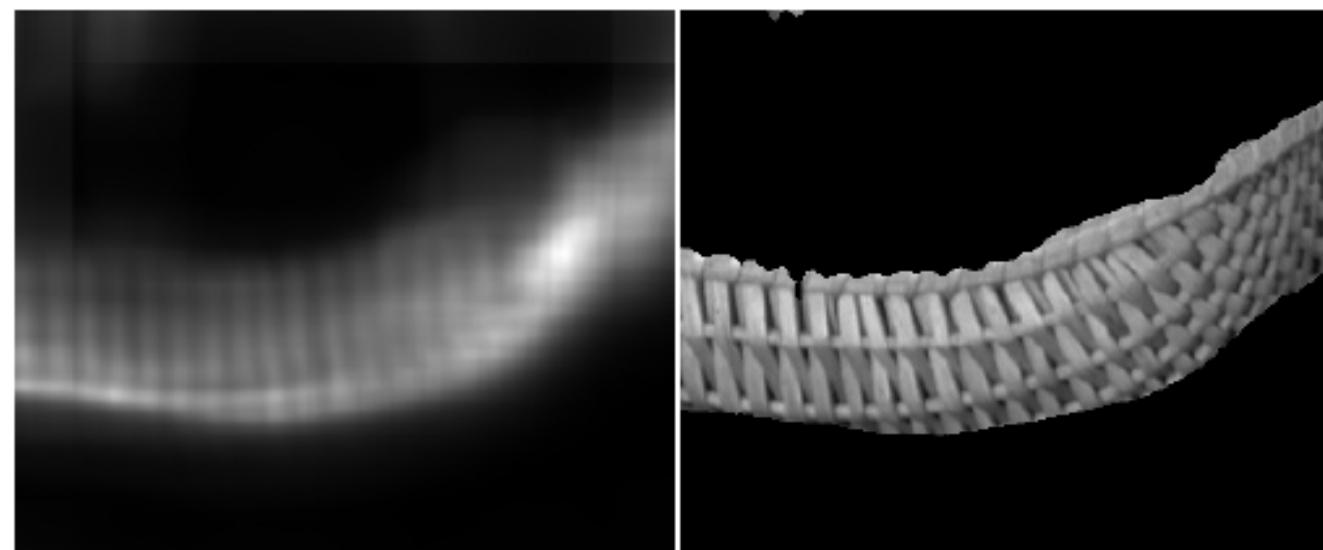
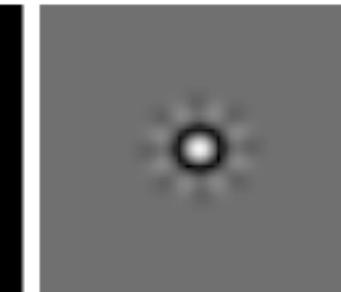
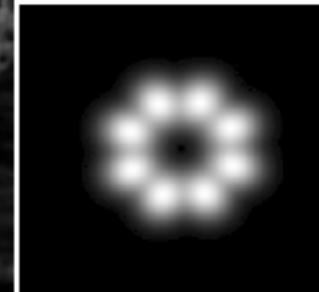
Segmentation - Décision : Similitude d'une région

➤ Caractéristiques spectrales :



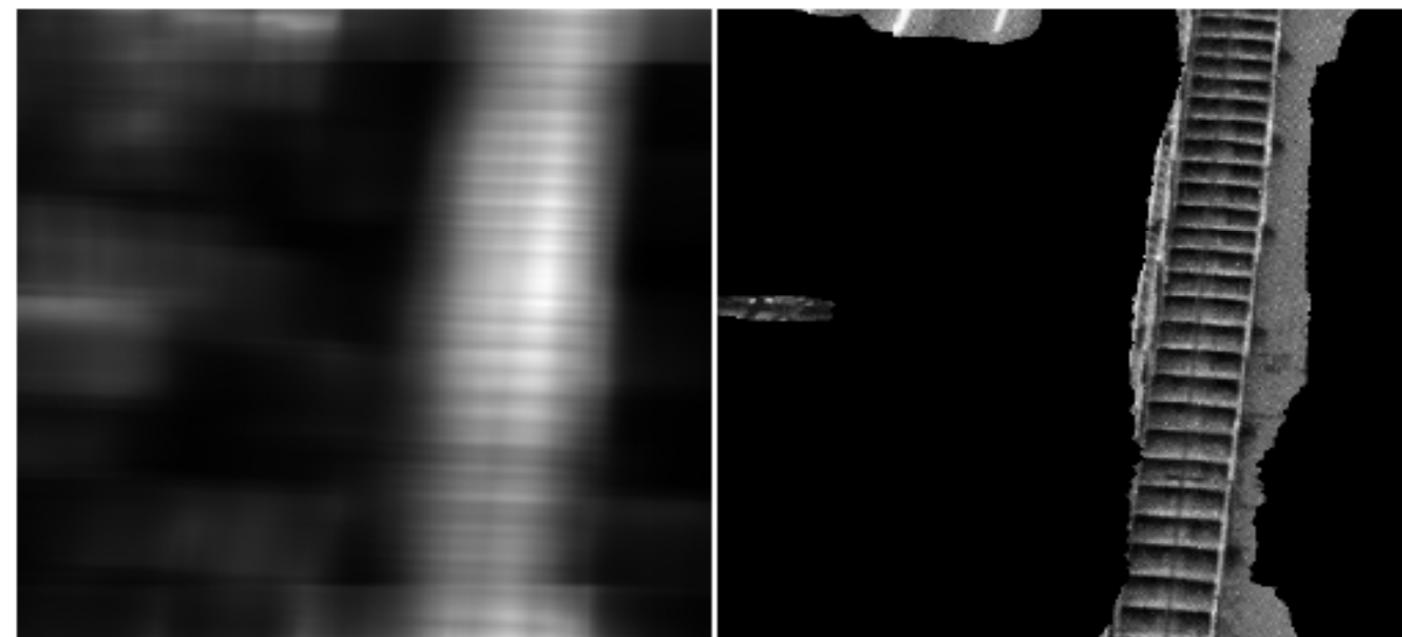
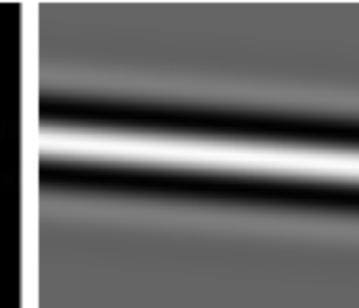
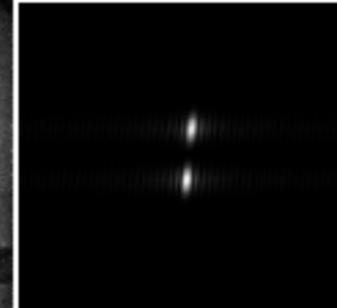
Segmentation - Décision : Similitude d'une région

➤ Exemple:



Segmentation - Décision : Similitude d'une région

➤ Exemple:



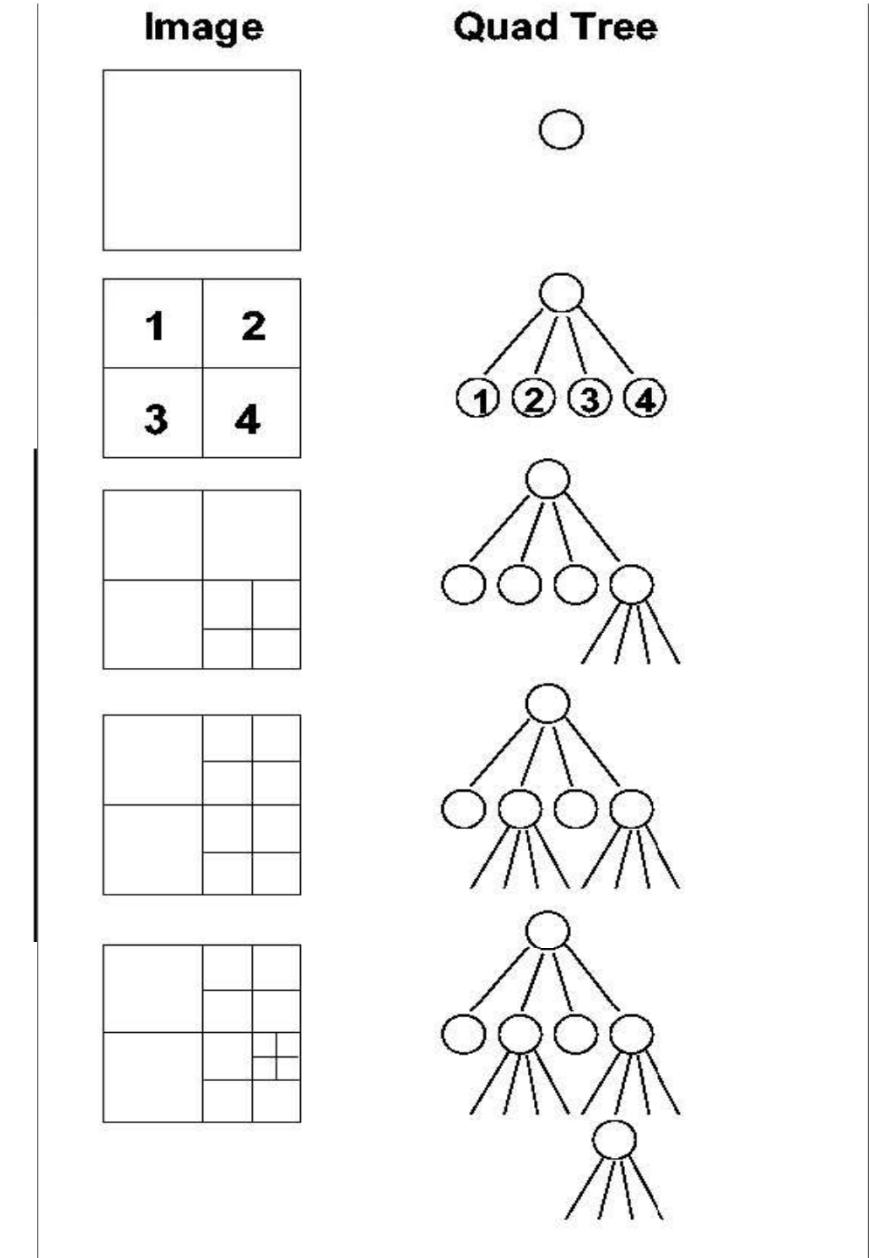
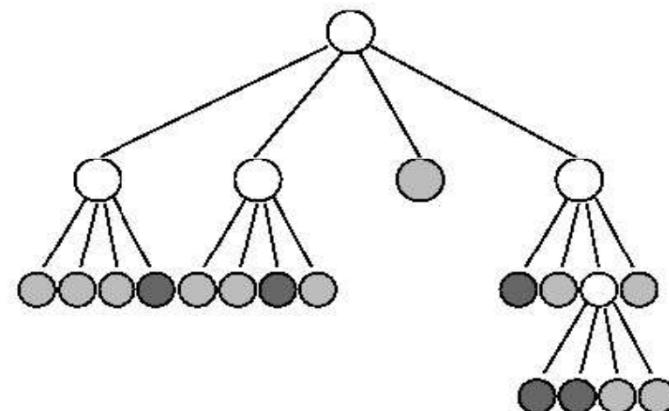
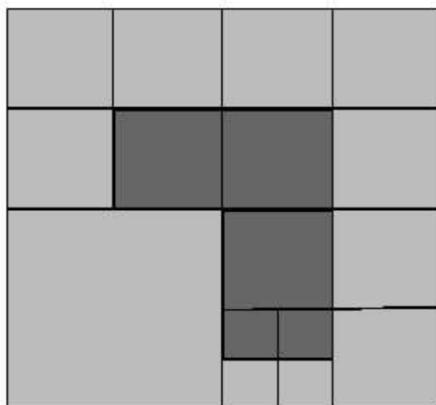
Segmentation - Décision : Approche Split and Merge

- Initialisation :
 - ↳ l'image initiale entière forme un bloc
- Étape de division :
 - ↳ Diviser récursivement tout bloc non-homogène selon un prédicat défini (variance, max-min, ...)
 - ↳ La division d'un bloc donne 4 sous-blocs
 - ↳ Les attributs de chaque sous-bloc sont recalculés
- Étape de fusion:
 - ↳ Regrouper les blocs adjacents représentant de régions homogènes selon un prédicat défini
- Post-traitement:
 - ↳ Lissage pour éliminer la forme carré des régions

Segmentation - Décision : Approche Région : Splitting

➤ Division (Split):

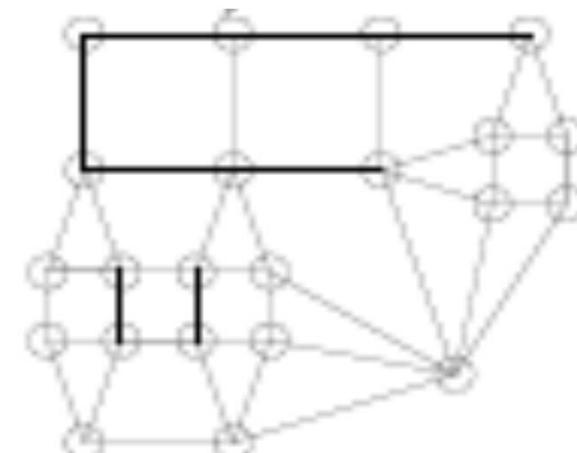
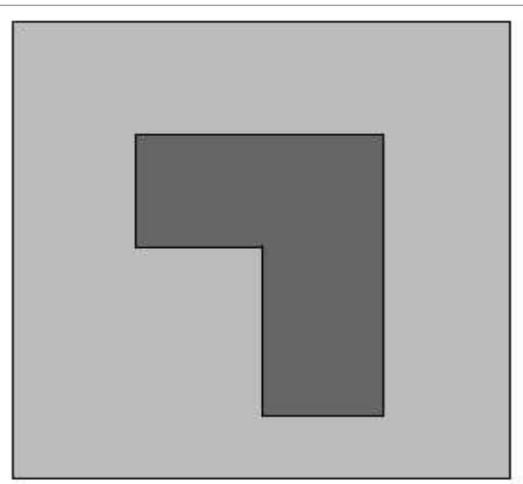
↳ selon les Quadtree:



Segmentation - Décision : Approche Région : Merging

➤ Fusion (Merge):

- ↳ Fusionne les « feuilles » du Quadtree qui sont similaires.
- ↳ Analyse d'un graphe d'adjacence de régions qui analyse une image présegmentée, constitue d'un ensemble de régions.
- ↳ L'analyse du graphe d'adjacence de régions permet de fusionner des régions d'une image sur-segmentee.
- ↳ Le procédé consiste à fusionner deux noeuds reliés par une arête à condition qu'ils respectent un critère de fusion.



Segmentation - Décision : Approche Région : Splitting

➤ Division à 100% :

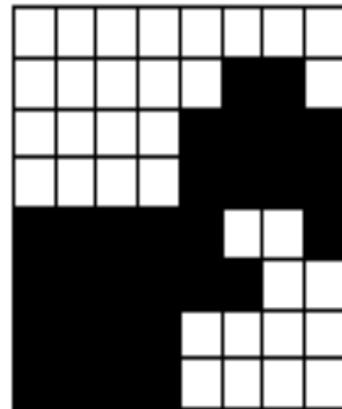
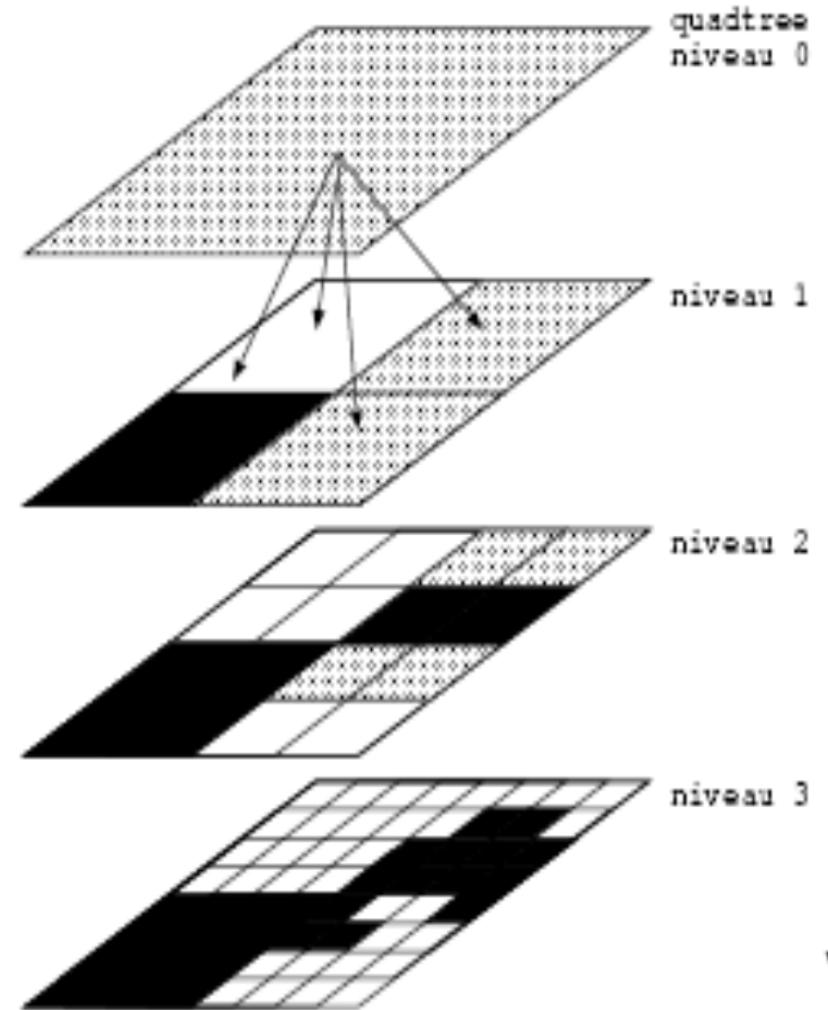
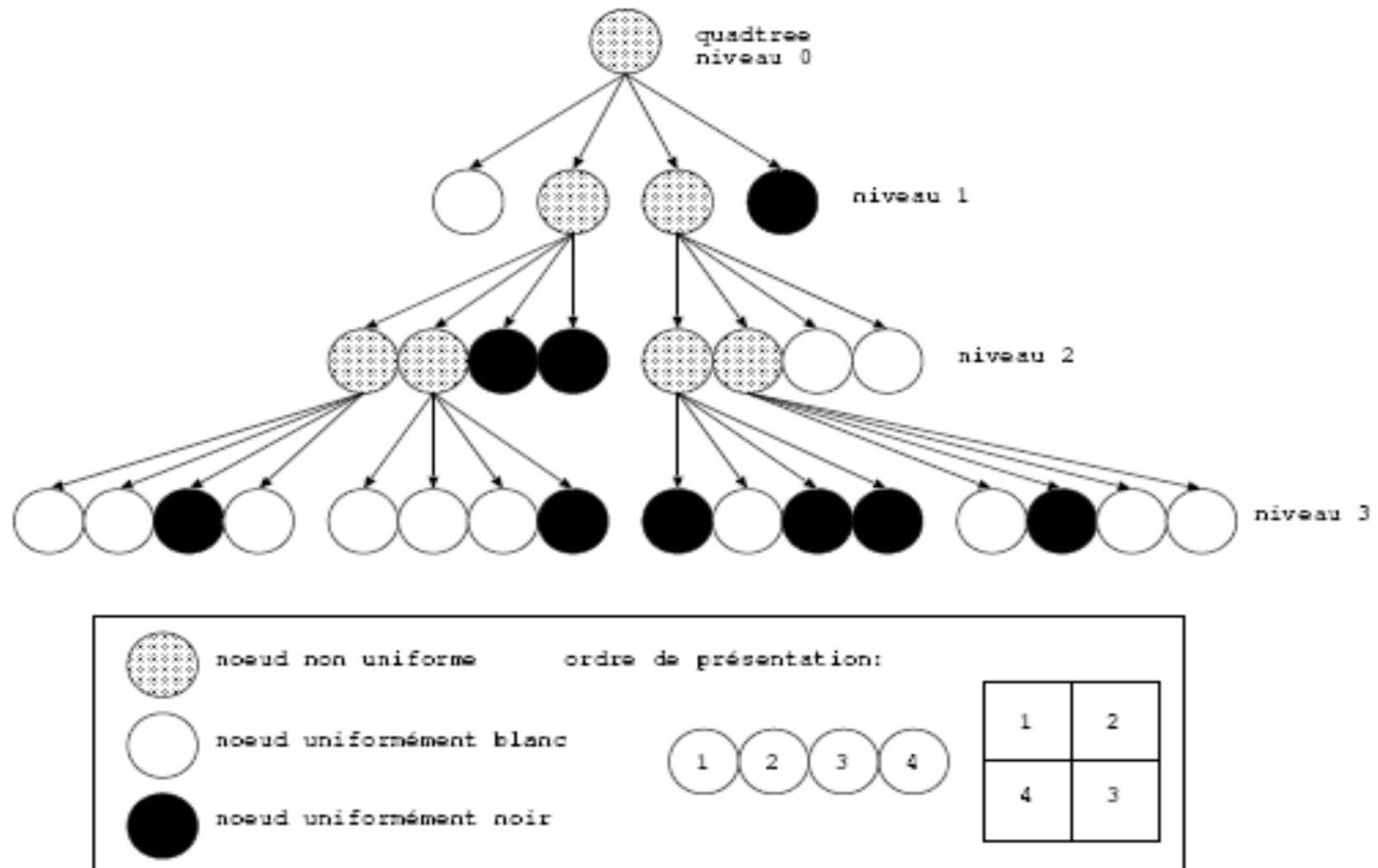


image 8x8 original (1bpp)



Segmentation - Décision : Approche Région : Merging

➤ Fusion à 100% :



Segmentation - Décision : Approche Région : split & merge

➤ Division / Fusion à 75% :

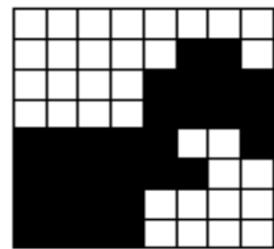
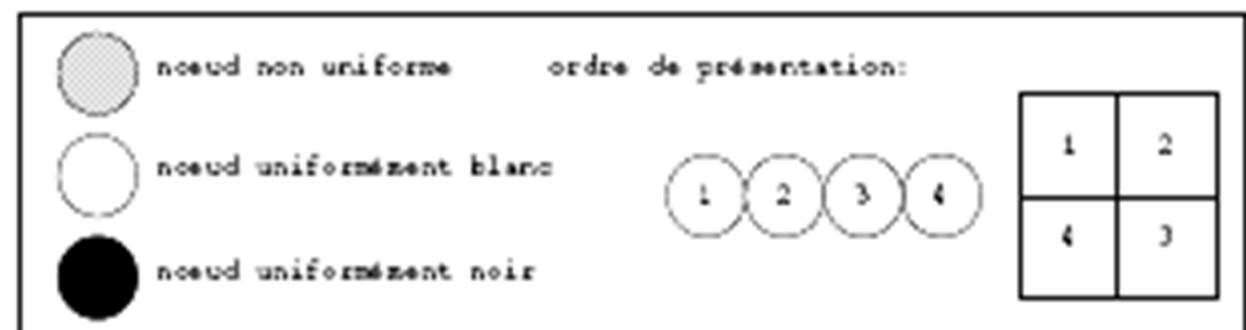
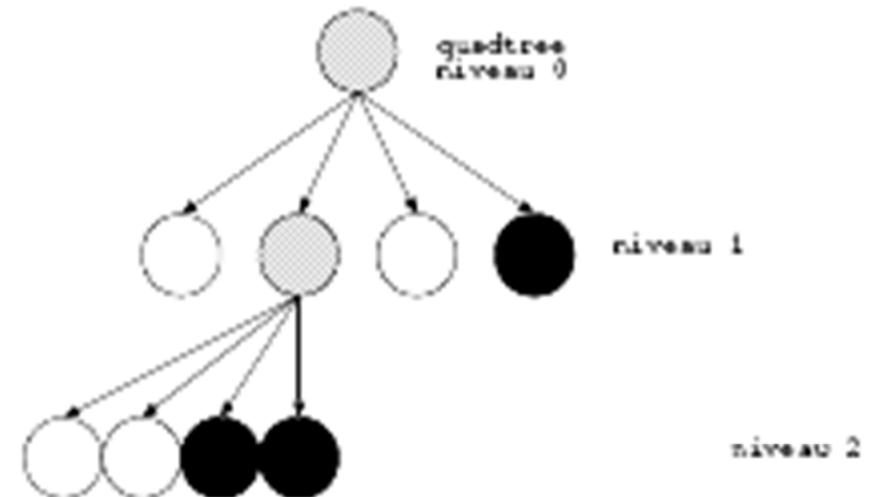
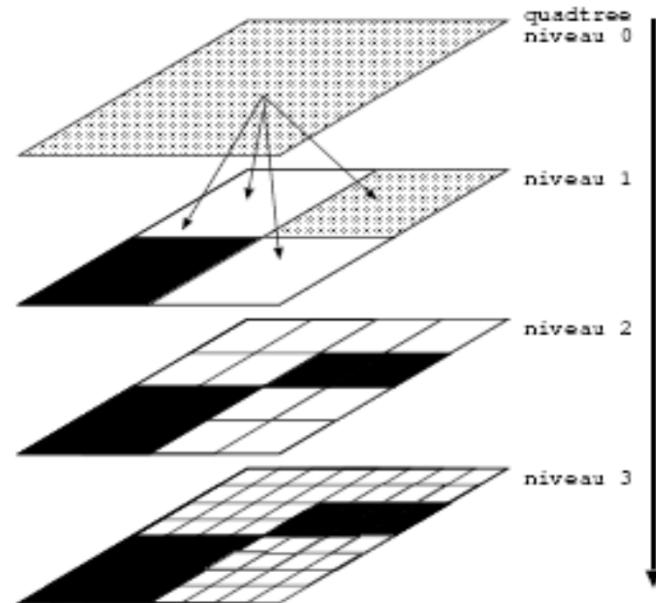
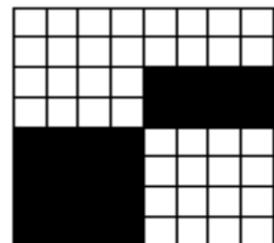
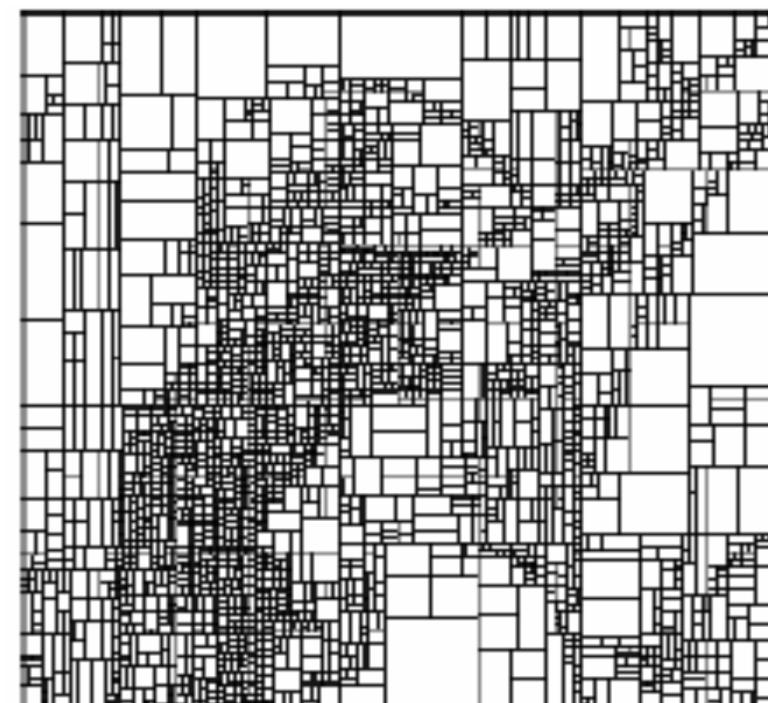


image 8x8 original (1bpp)



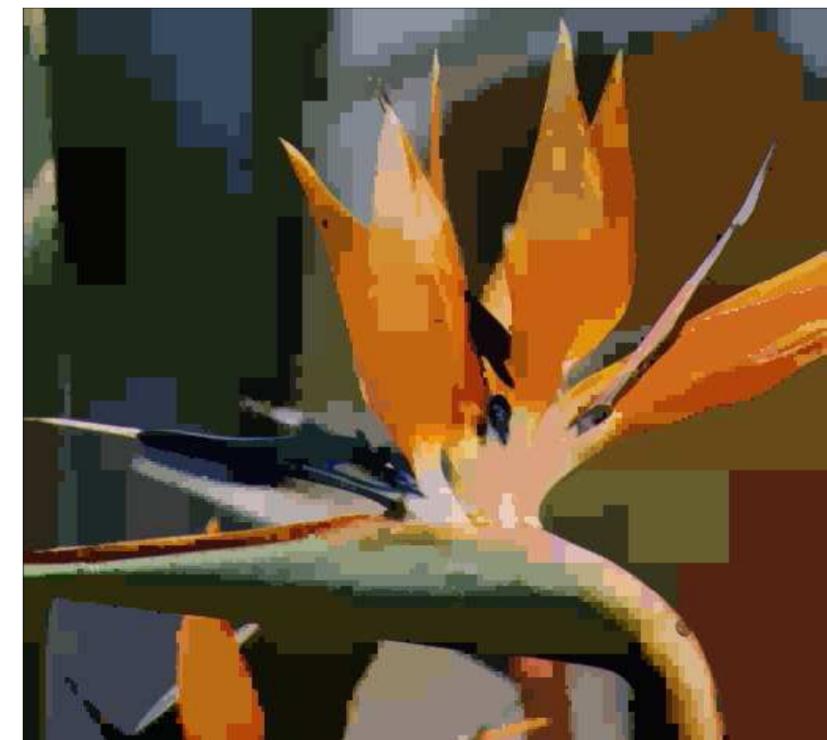
Segmentation - Décision : Approche Région : split & merge

➤ Exemple :



Segmentation - Décision : Approche Région : split & merge

➤ Exemple sur-segmentation:



Segmentation - Décision : Approche Région : Croissance

➤ Principe:

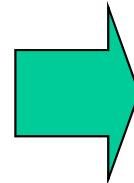
- ↳ faire croître des régions en y ajoutant successivement les pixels adjacents qui satisfont un critère d'homogénéité.
- ↳ L'étape initiale consiste à sélectionner les germes des régions qui correspondent généralement à un pixel.
- ↳ Puis, les régions sont construites en y ajoutant successivement les pixels qui leur sont connexes et qui vérifient un critère de similarité.
- ↳ La croissance s'arrête lorsque tous les pixels ont été traités.

➤ Les méthodes diffèrent de par :

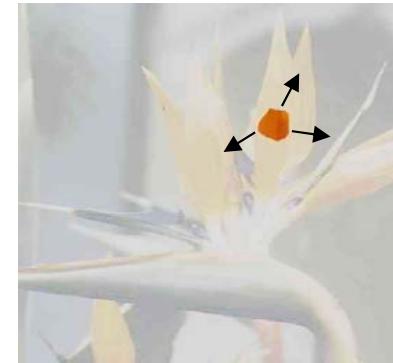
- ↳ Le choix des germes initiaux
- ↳ Le parcours de l'image selon le balayage de gauche à droite et de haut en bas ou de façon récursive

Segmentation - Décision : Approche Région : Croissance

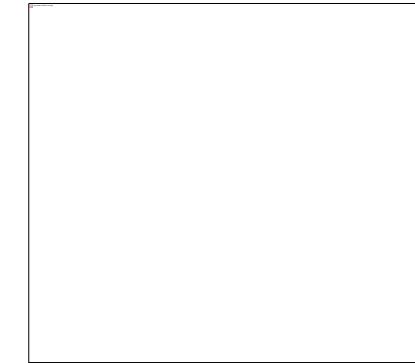
Idée: On part d'un point amorce (seed) et l'on l'étend en ajoutant les points de la frontières qui satisfont le critère d'homogénéité



amorce



croissance

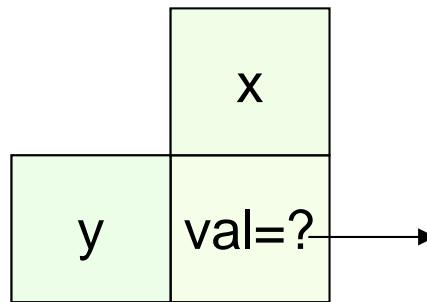


région finale

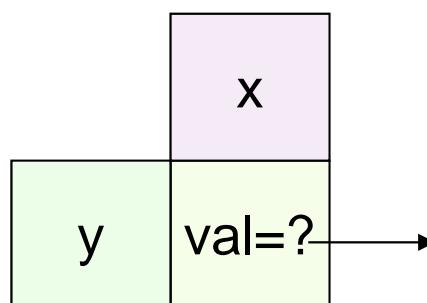
- Le point amorce peut être choisi soit par un humain, soit de manière automatique en évitant les zones de fort contraste (gradient important) => *méthode par amorce*.
- Si le critère d'homogénéité est local (comparaison de la valeur du pixel candidat et du pixel de la frontière) => *méthode linéaire*.

Segmentation - Décision : Approche Région : Croissance

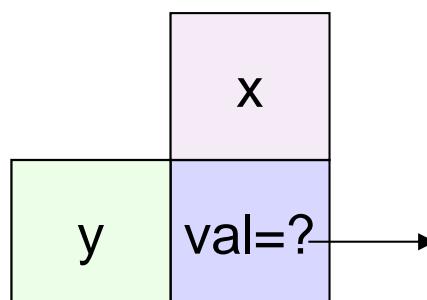
Méthode linéaire (complexité 2n)



x==y: val = x
x<>y: mettre à jour l'index
Extremité(x) → y



val = y



val = Z
(nouvel index)

Seuil S: $P_1 == P_2 \text{ ssi } \text{Diff}(\text{Col}(P_1), \text{Col}(P_2)) < S$

1	1	2	2	3
1	1	1	1	1

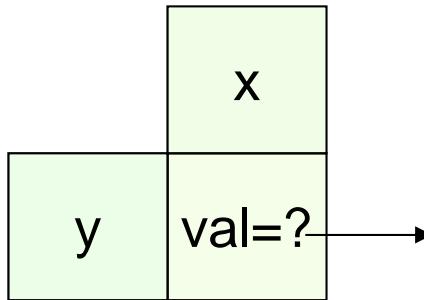
1 → 1
2 → 2
3 → 1

...

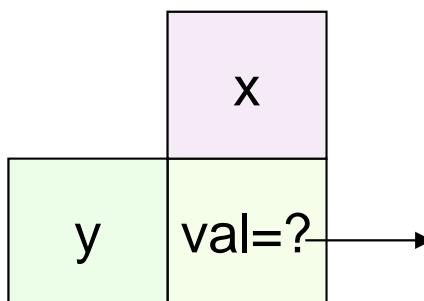
```
while (swap[x] != x)  
    x = swap[x];
```

Segmentation - Décision : Approche Région : Croissance

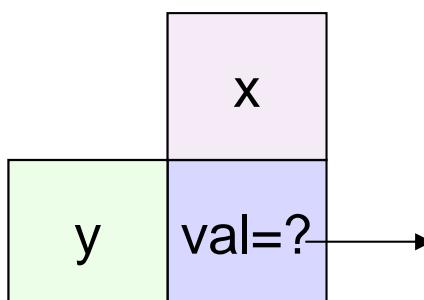
Méthode linéaire (complexité 2n)



x==y: val = x
x<>y: mettre à jour l'index
Extremité(x) → y



val = y



val = Z
(nouvel index)

Seuil S: $P_1 == P_2 \text{ ssi } \text{Diff}(\text{Col}(P_1), \text{Col}(P_2)) < S$

1	1	2	2	3
1	1	1	1	1
2	2			
			3	3
			3	3

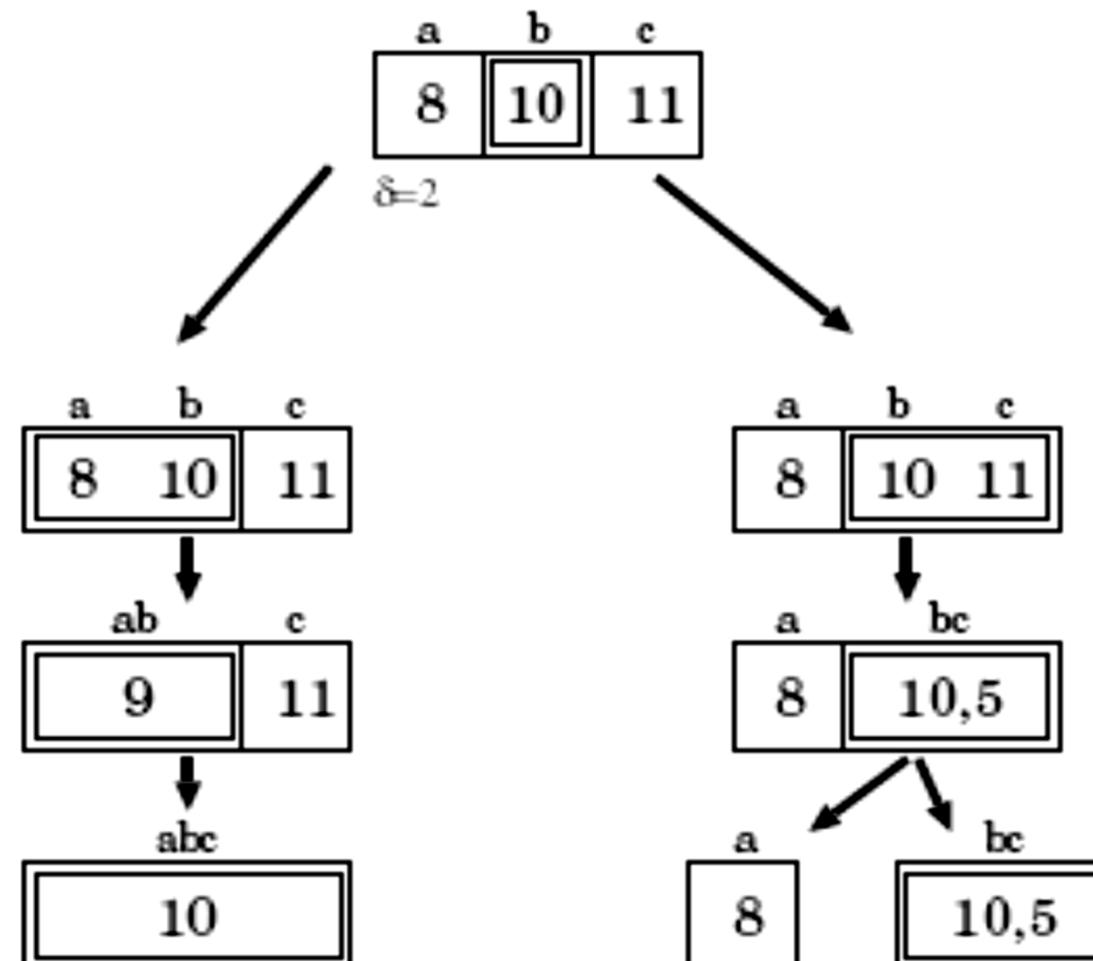
1 → 1
2 → 2
3 → 1

...

```
while (swap[x] != x)  
    x = swap[x];
```

Segmentation - Décision : Approche Région : Croissance

➤ Influence de la position du germe initial:

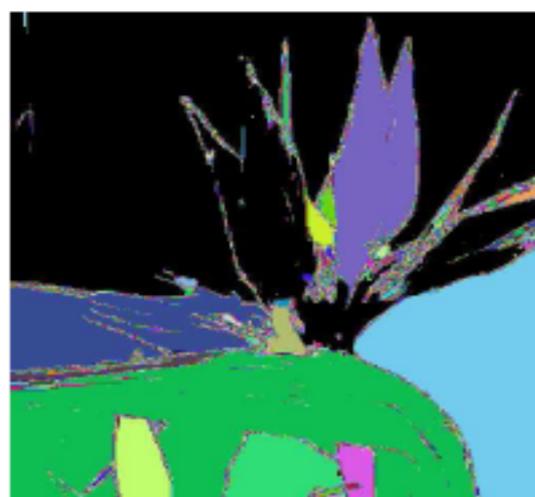


Segmentation - Décision : Approche Région : Croissance

➤ Exemples:



$S=6$



$S=7$



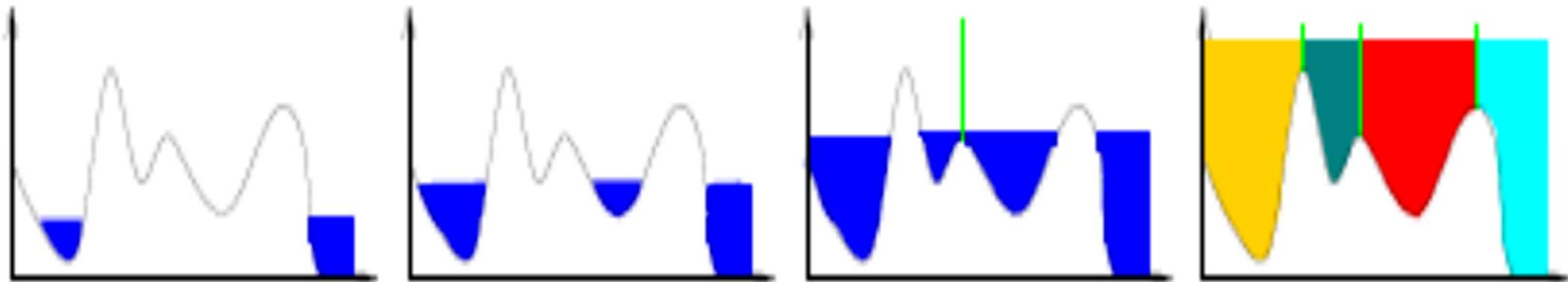
$S=3$



$S=4$

Segmentation - Décision : Approche Région : LPE

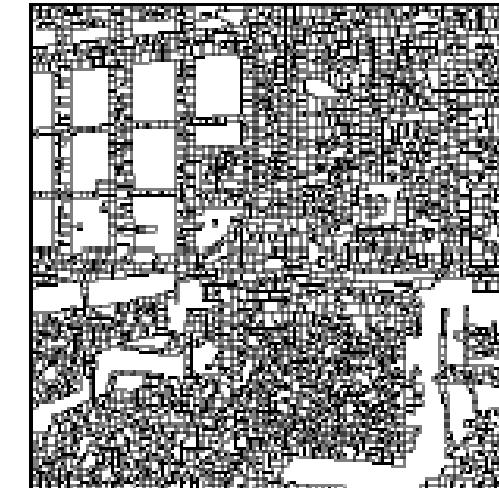
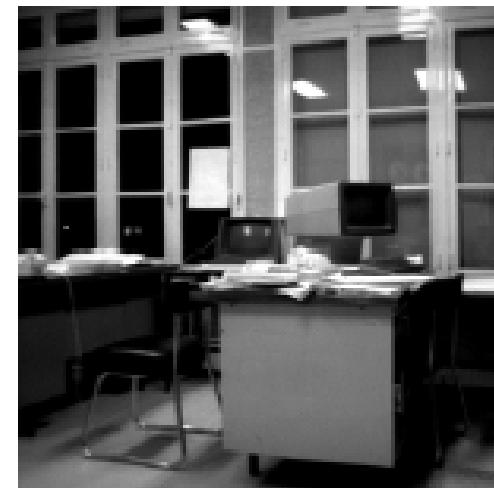
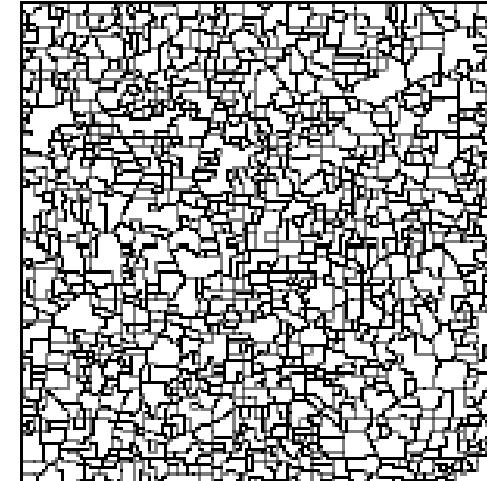
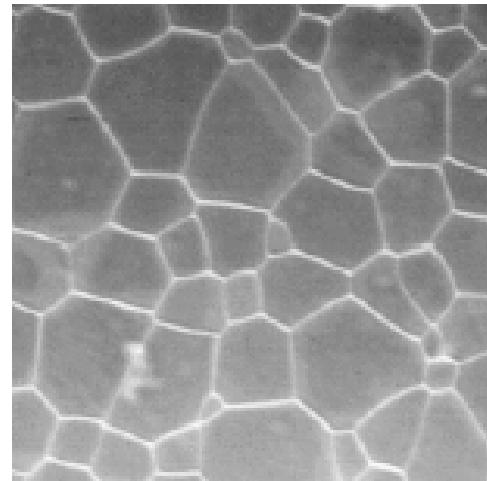
- Ligne de partage des Eaux : Morphologie Math.
- Principe :
 - ↳ En partant du principe d'une analogie de l'image avec un relief géodésique, recherche les plateaux du relief ce qui lui permet de trouver des régions d'intensité uniforme sans utiliser le gradient.
- Les germes correspondent aux minima locaux.
- Inconvénients
 - ↳ Sur-segmentation
 - ↳ Il est souvent nécessaire d'opérer certaines transformations pour se ramener à cette problématique.
 - ↳ Un résultat pas toujours comme on l'attendrait



Segmentation - Décision : Approche Région : LPE

➤ Ligne de partage des Eaux :

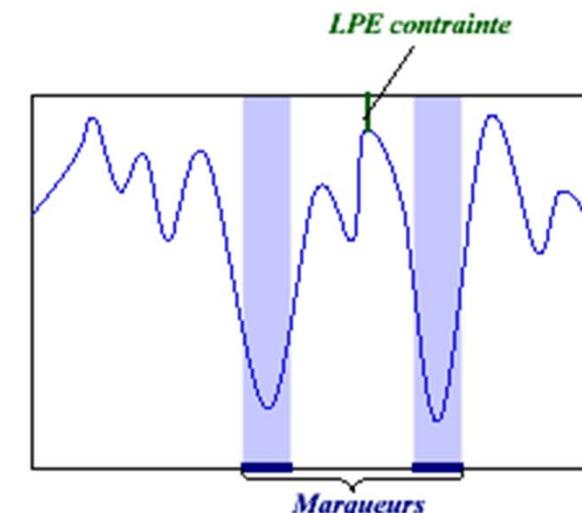
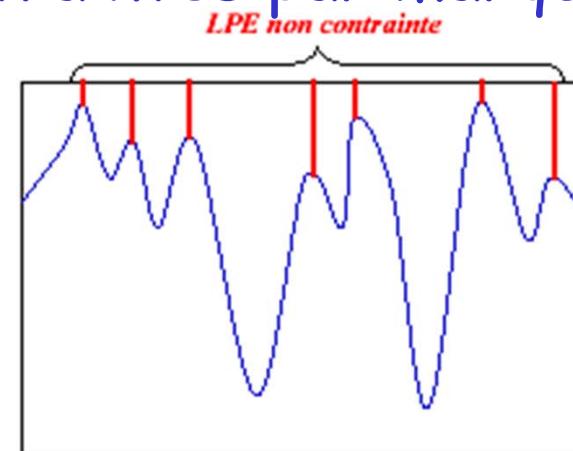
↳ Problème de la sur-segmentation



Segmentation - Décision : Approche Région : LPE

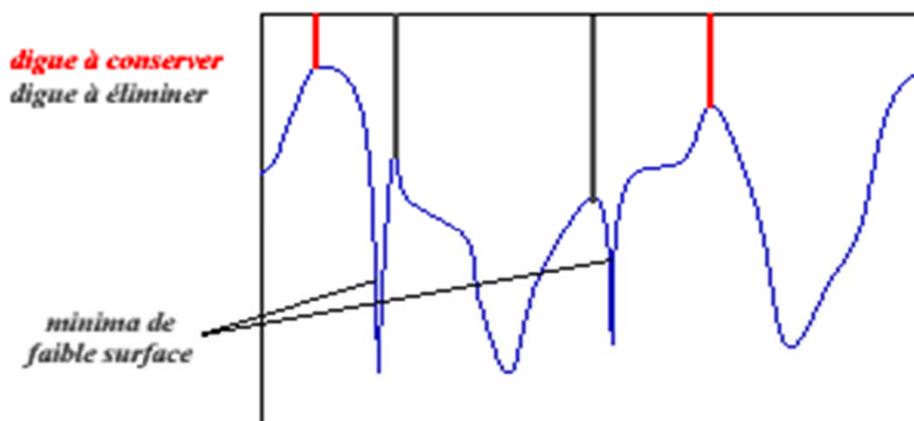
➤ Ligne de partage des Eaux :

↳ Contraintes par marquage

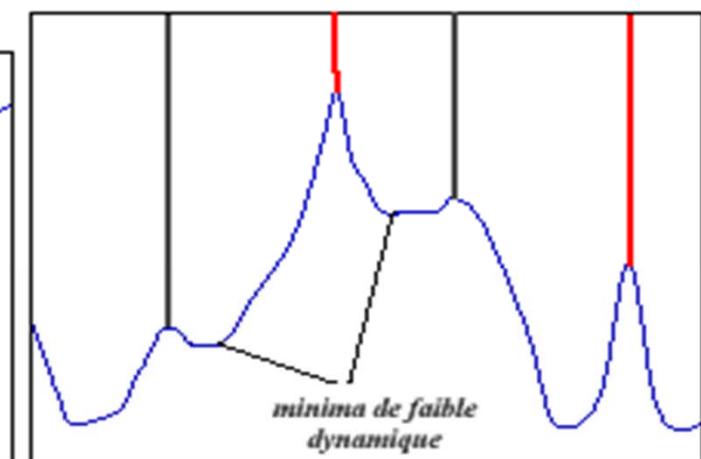


↳ Contraintes par filtreage

— = *digue à conserver*
— = *digue à éliminer*



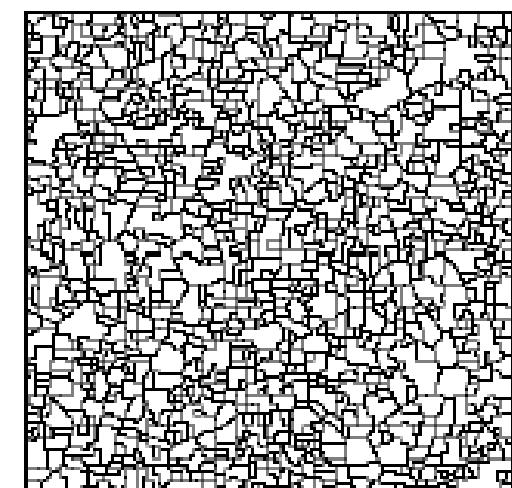
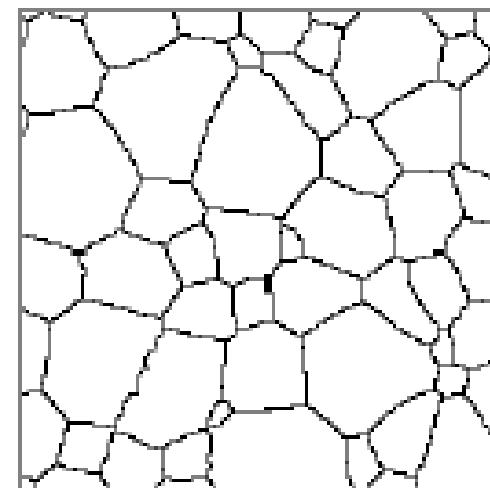
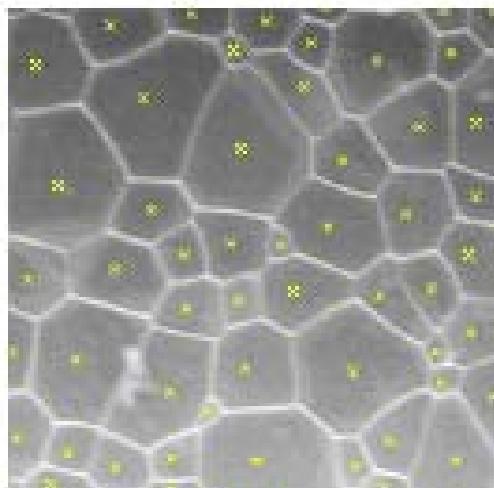
La surface



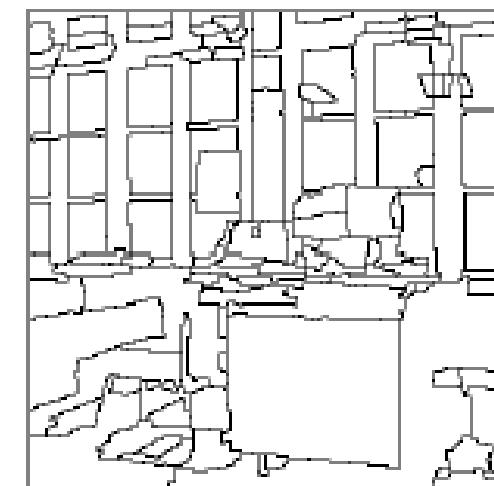
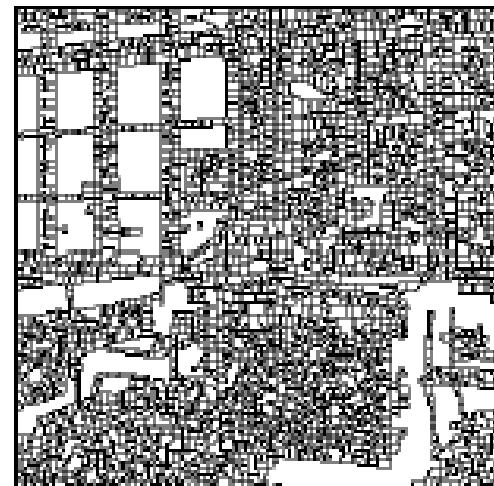
La profondeur

Segmentation - Décision : Approche Région : LPE

➤ Ligne de partage des Eaux : avec contraintes



LPE contrainte



Segmentation - Décision : Approche Région : LPE

➤ Ligne de Partage des Eaux : Application

exemple 2



Séquence du Taxi de Hambourg



marqueurs : objets mobiles + fond



LPE brute



LPE contrainte

Segmentation - Décision : Approche Classification de pixels

➤ La segmentation par classification (clustering) consiste à regrouper les pixels par classes de même critère, mais contrairement à l'approche région, on ne fait pas intervenir de notion de voisinage

↳ Classification sur l'histogramme

- Binarisation si 2 classes
- Classification couleur sur histogramme R, V et B

↳ Classification sur n'importe quelle caractéristique

- K-moyennes (K-means)
- Nuées Dynamiques
- Bayes
- ... voir RDF(si méthodes supervisées)

Segmentation - Décision : Approche Classification de pixels

➤ Classification sur Histogramme :

↳ Seuillage - Binarisation

- Méthode simple et très populaire pour la segmentation d'objets dans les images numériques.

↳ Le seuillage peut être de nature

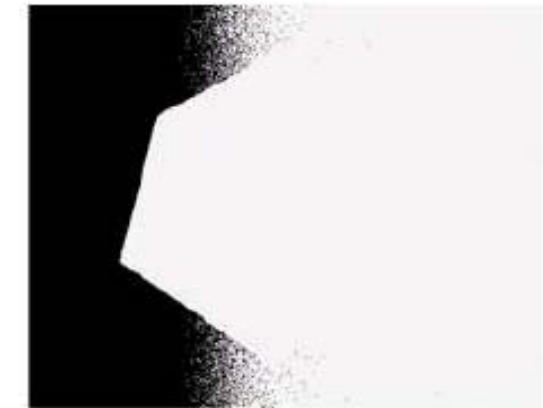
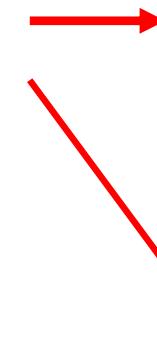
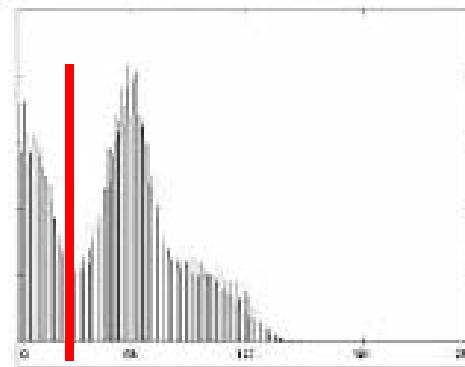
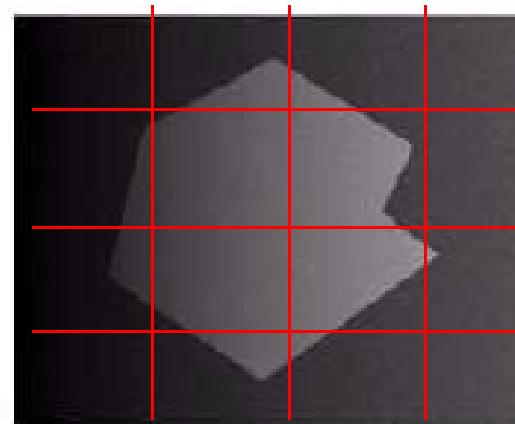
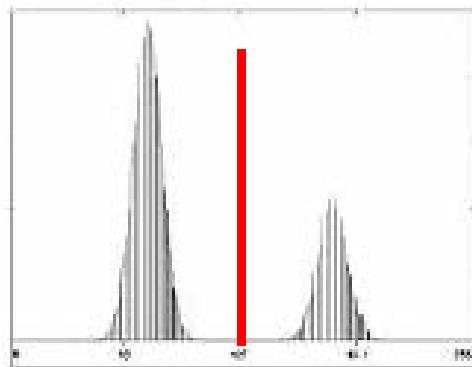
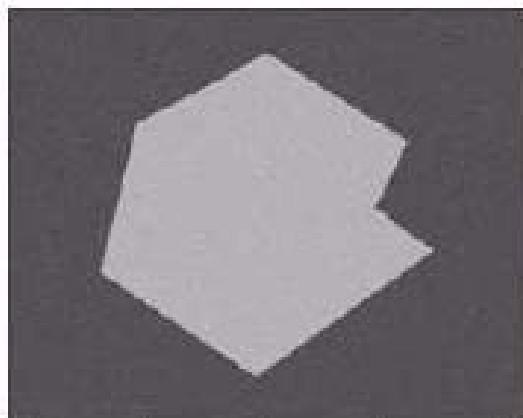
- Globale : un seuil pour toute l'image
- Locale: un seuil pour une portion de l'image
- Adaptative: un seuil qui s'ajuste selon les images/parties de l'image.

↳ Inconvénients:

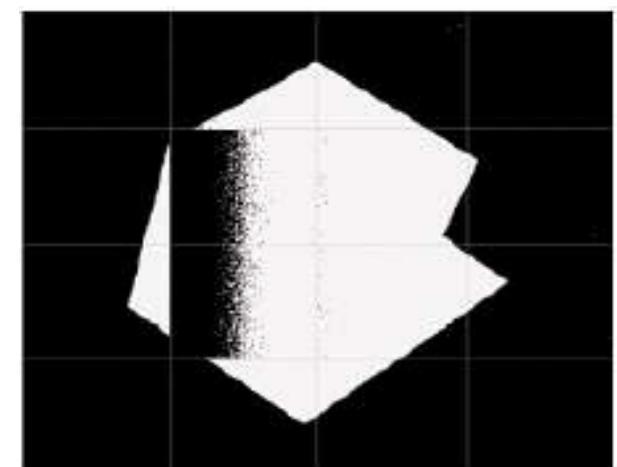
- Connaître le nombre de classes
- Apparition de faux éléments (aucune prise en compte de la composante spatiale)
- Choix du ou des seuils : seuillage automatique (Otsu)

Segmentation - Décision : Approche Classification de pixels

➤ Classification sur Histogramme :

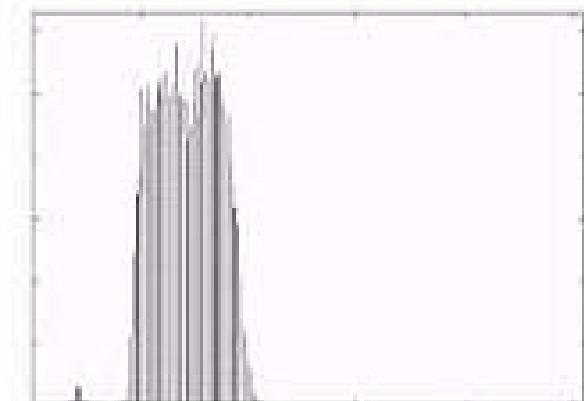
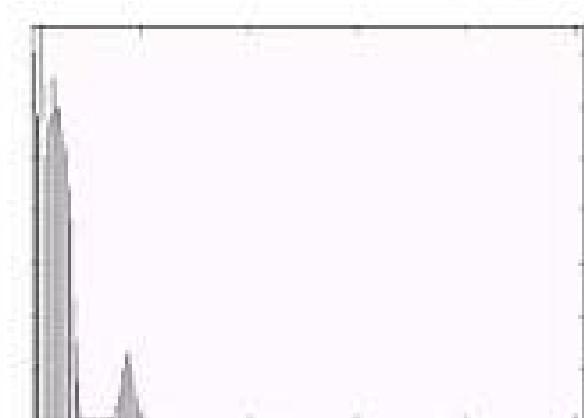
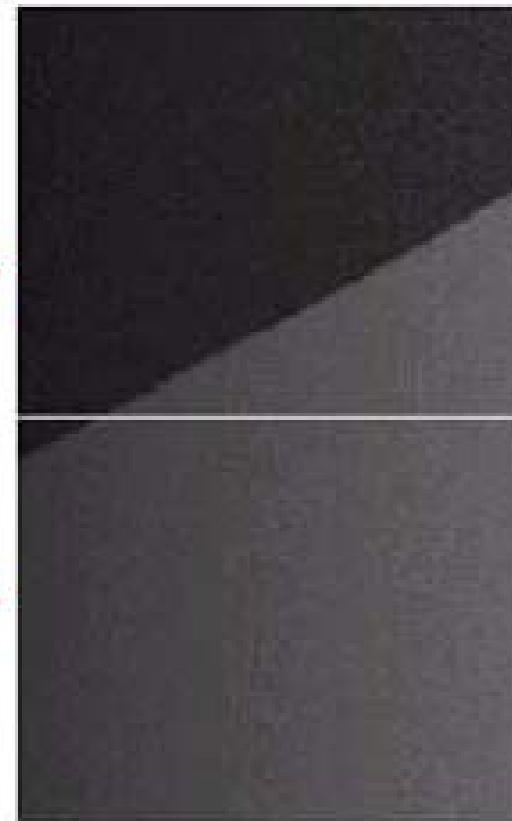
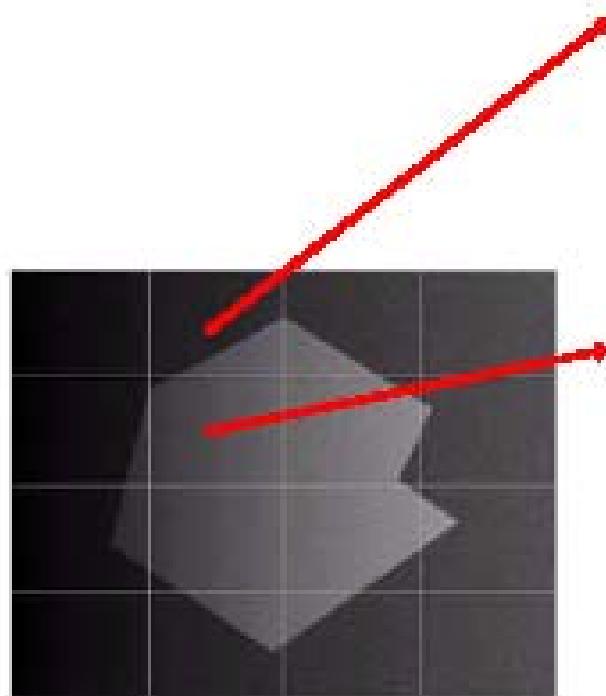


Seuillage adaptatif



Segmentation - Décision : Approche Classification de pixels

➤ Classification sur Histogramme :

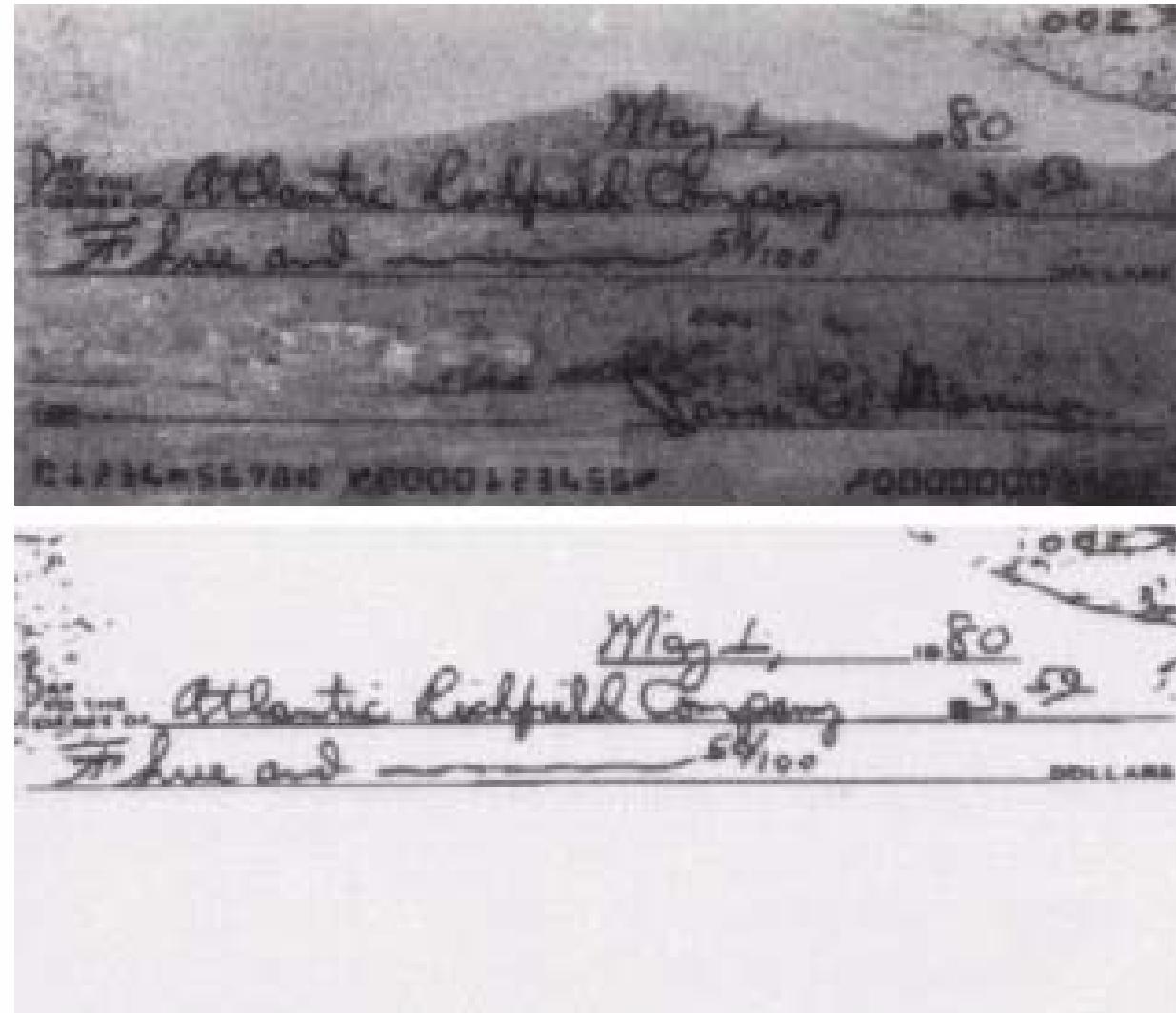


Bimodal ??

Seuil de calcul sur variance $\sigma > 100$

Segmentation - Décision : Approche Classification de pixels

➤ Classification sur Histogramme : Application



Segmentation - Décision : Approche Classification de pixels

➤ Segmentation par Classificateur

↳ K-moyennes (K-Means)

- Division des pixels en K groupes (clusters)
- K doit être fixé à l'avance (paramètre)

↳ Algorithme :

- On fixe les centres des groupes
- Algorithmes de découpage en k groupes de variance minimale selon un critère de distance
- On alloue chaque pixel au centre le plus proche

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of } i^{\text{'}} \text{th cluster}} \|x_j - \mu_i\|^2 \right\}$$

- x peut être n'importe quel vecteur de paramètres

Segmentation - Décision : Approche Classification de pixels

➤ Segmentation par Classificateur K-moyennes



Image source



K-moyennes
sur l'intensité

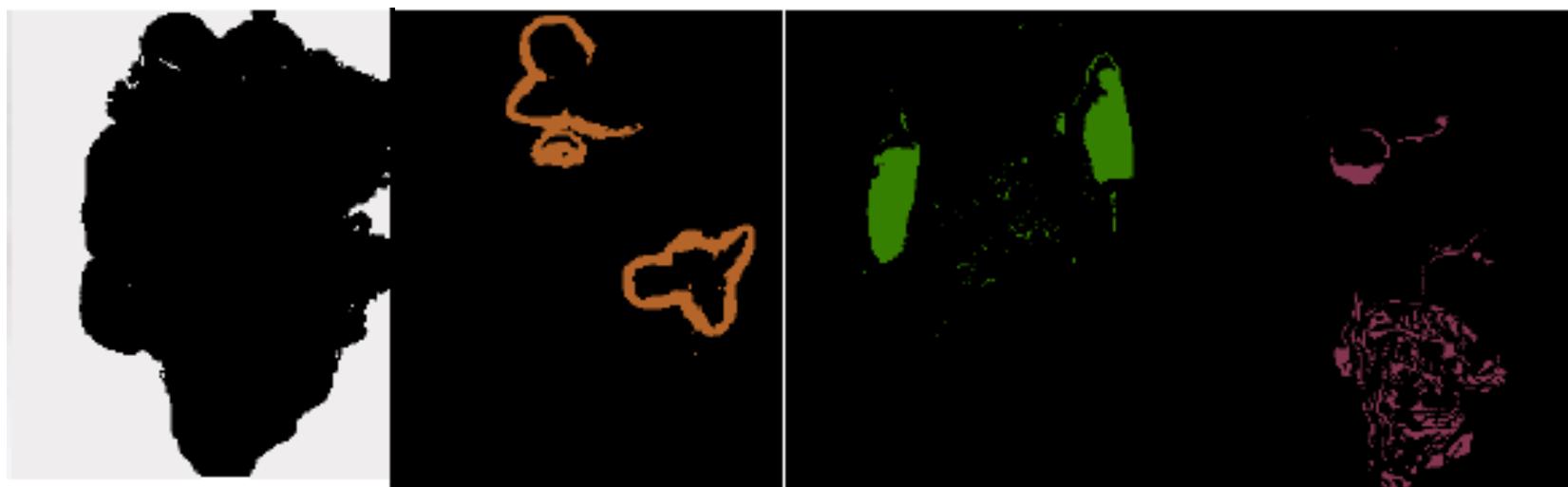
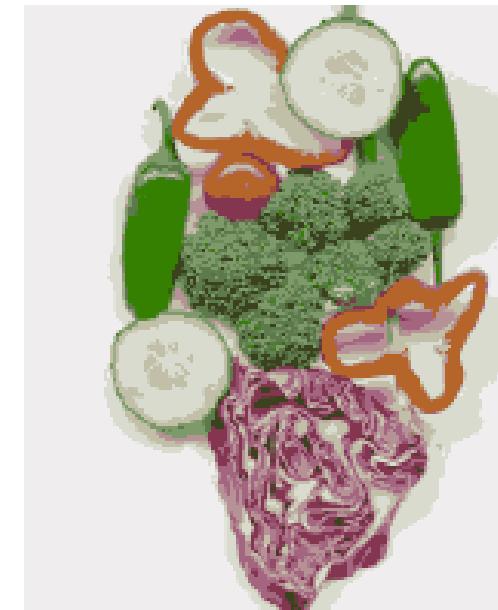


K-moyennes
sur la couleur

Segmentation - Décision : Approche Classification de pixels

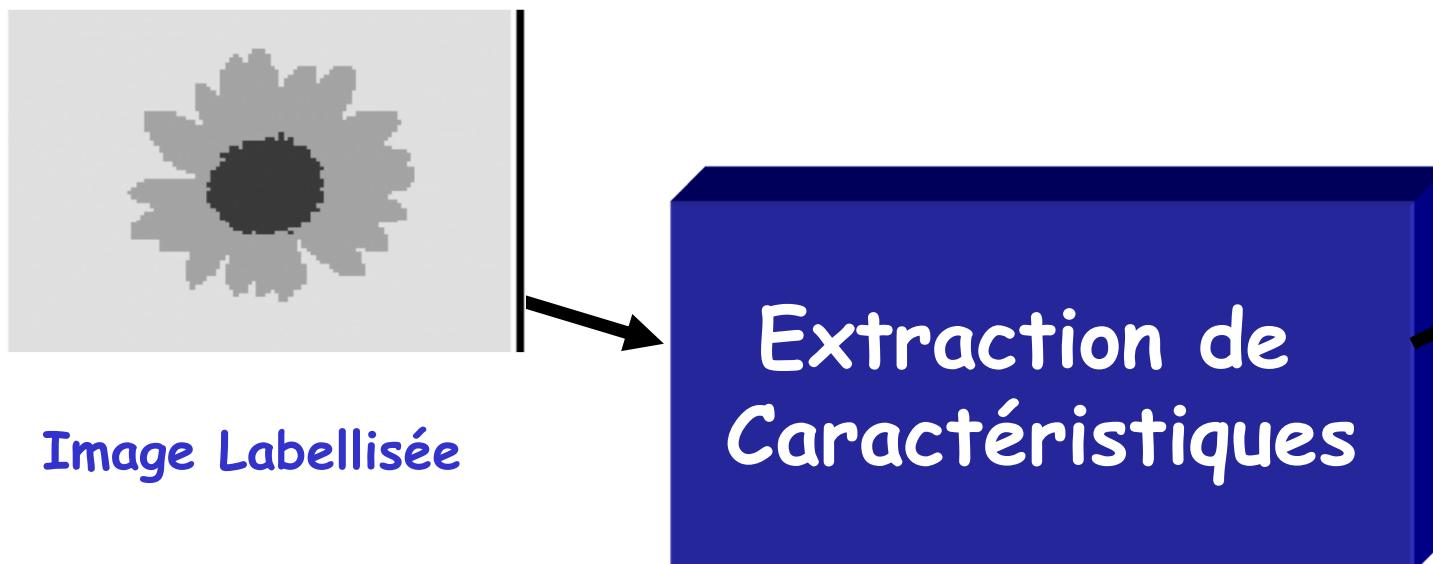
➤ Segmentation par Classificateur K-moyennes

↳ 11 classes



Segmentation - Décision : Extraction de caractéristiques

- La caractérisation correspond à l'extraction d'un vecteur caractéristique qui sera exploité par les étapes suivantes (Interprétation - Décision)
- Elle s'appuie sur la géométrie discrète et la topologie
- Elle dépend fortement de l'objectif du traitement:
 - ↳ RdF, Compression, Analyse, ...



Vecteur caractéristique :

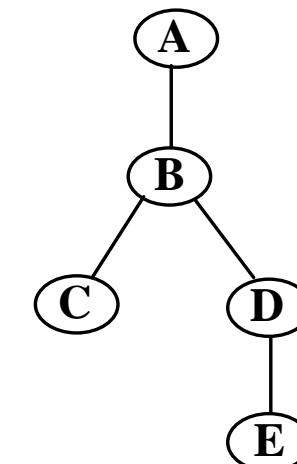
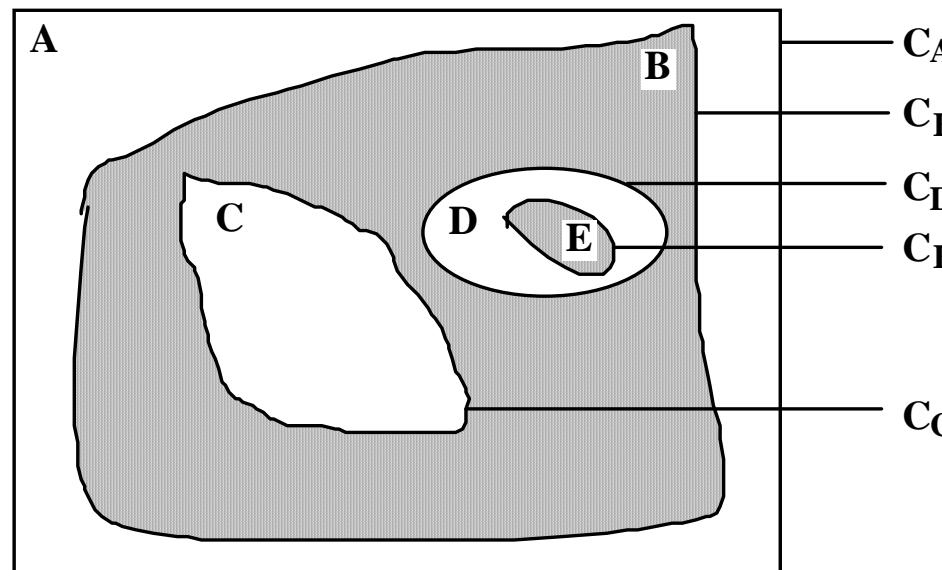
- Surface,
- Périmètre,
- Nombre d'Euler,
- Chaîne de codes,
- ...

➤ Notion de topologie - Relation d'imbrication:

- *Courbe frontière*: courbe fermée de l'image qui sépare deux régions.
- *Extérieur ou fond*: région de l'image qui contient l'ensemble des courbes frontières.
- *Région* : groupe de pixels connectés par une ou plusieurs propriétés.
- *Frontière d'objet*: courbe frontière de niveau de gris "1" en son intérieur.
- *Objet*: intérieur d'une frontière d'objet.
- *Frontière d'évidement*: courbe frontière inclue dans un objet ou dans un objet interne ayant le niveau de gris "0" en son intérieur.
- *Trou ou évidement*: intérieur d'une frontière d'évidement.
- *Frontière d'objet interne* : courbe frontière inclue dans un évidement de niveau de gris "1" en son intérieur.
- *Objet interne* : intérieur d'une frontière d'objet interne.
- *Forme* : objet, objet interne ou évidement.
- *Forme parent* : Forme qui englobe une ou plusieurs autres formes.
- *Forme fille* : Forme englobée par une autre forme.
- *Relation d'imbrication* : relation qui lie les formes parents et les formes filles

➤ Notion de topologie - Relation d'imbrication:

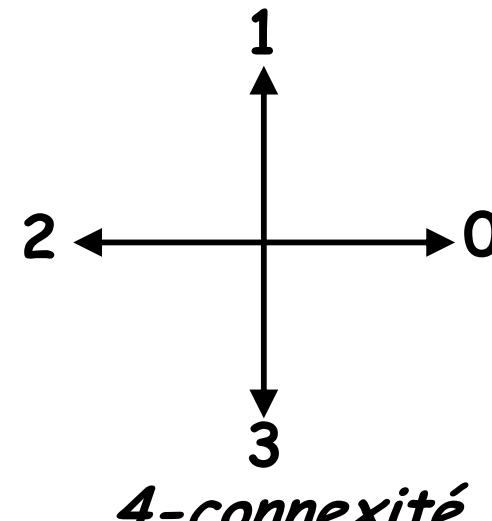
- ↳ CA, CB, CC, CD, CE représentent respectivement les contours des formes A, B, C, D. La forme A étant le fond de l'image, CA est assimilé à son cadre
- ↳ L'arbre donne les relations d'imbrications entre les différentes formes : B, C, D, E sont imbriquées dans A. Les évidements C et D ont le même niveau d'imbrication.



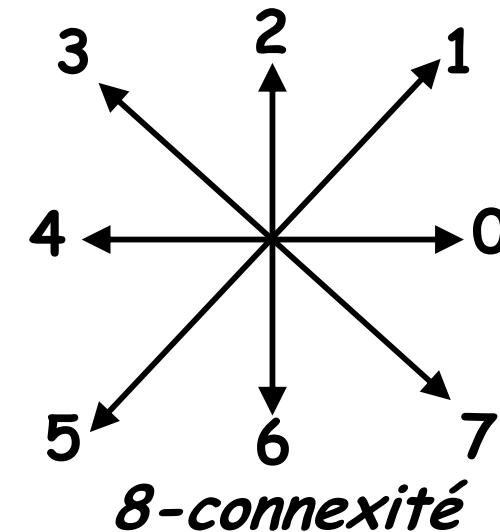
- Nombre d'Euler : $1 - \text{Nb d'évidements de la forme}$
- Exemple : Nb d'Euler (B) = - 1

Segmentation - Décision : Codage d'un contour

➤ Codage de Freeman:



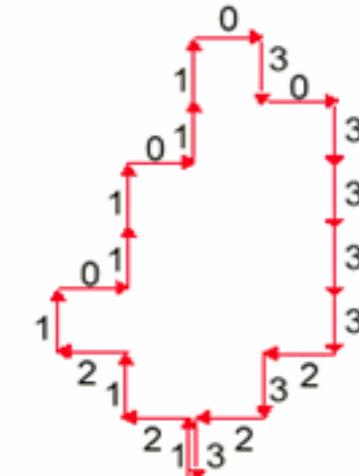
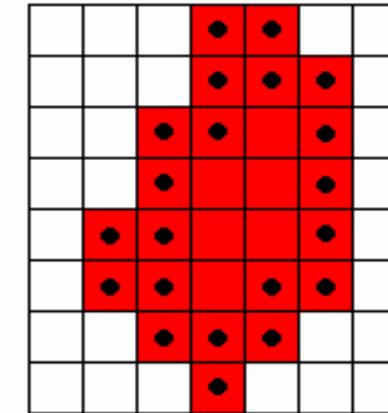
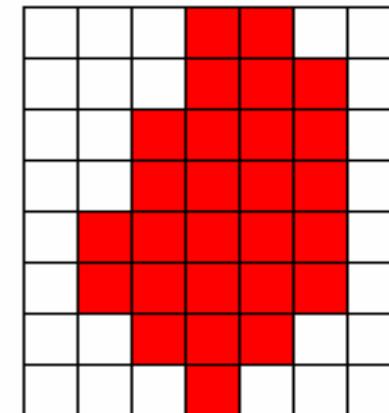
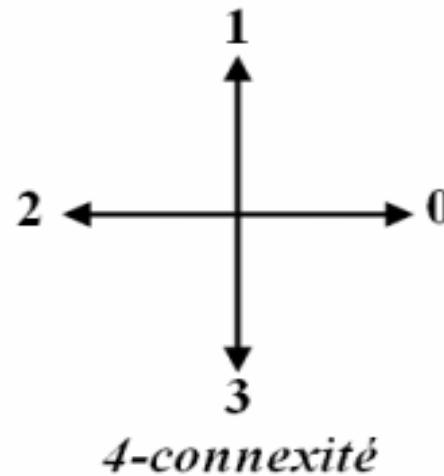
→ Codage sur 2 bits



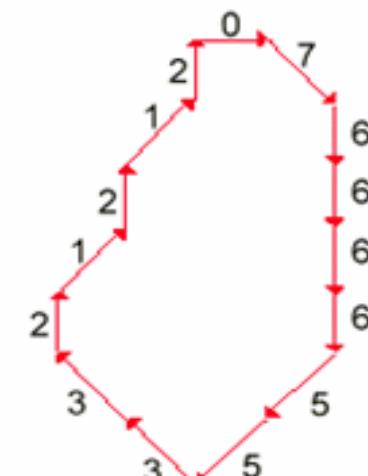
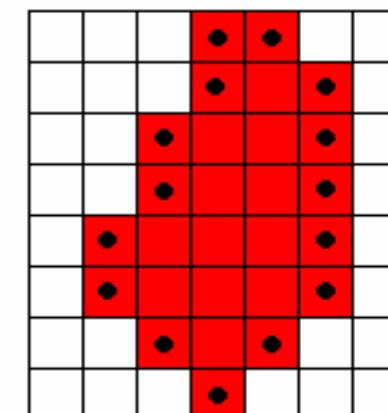
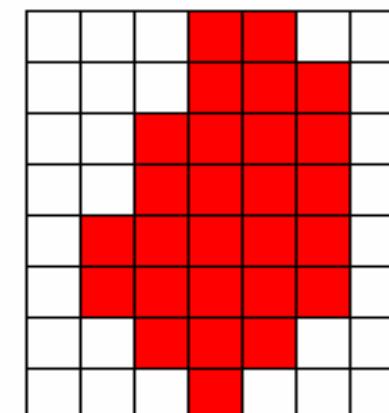
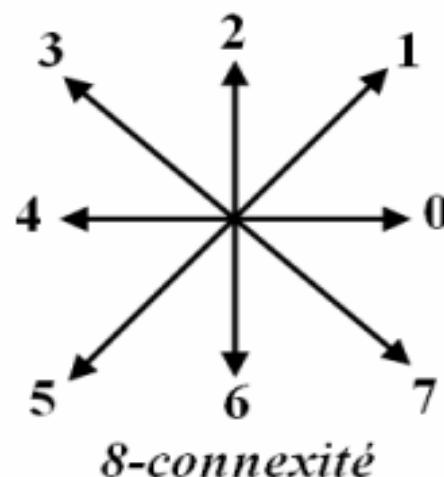
→ Codage sur 3 bits

- Principe : on part d'un pixel du contour et on code le contour en le parcourant dans le sens des aiguilles d'une montre.

Segmentation - Décision Codage d'un contour : code de Freeman



0303333232312121011011



076666553321212

➤ Principe de codage :

- ↳ L'image binaire de départ sera remplacée par un tableau de nombres qui utilisera les nombres :
 - ↳ 0 pour les pixels de fond,
 - ↳ 1 pour les pixels de l'objet,
 - ↳ 2 pour repérer la position de voisinage du pixel courant,
 - ↳ 3 pour la case du premier pixel (1) rencontré lors d'un balayage de gauche à droite et de haut en bas de l'image,
 - ↳ 4 pour le remplacement d'un pixel par cette valeur dès que l'on sait que la case où se trouve le pixel appartient au contour.
- ↳ A l'issue de l'algorithme, les pixels du contour seront remplacés par la valeur 4 ; alors, l'intérieur de l'objet conservera la valeur 1 des pixels, l'extérieur conservant des 0.

➤ Principe de codage :

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

Image initiale

Initialisation
1er pixel rencontré
par balayage de la
gauche vers la droite
et du haut vers le bas,
remplacé par un "3". "2"
"au dessus de ce pixel"

0 2 0 0 0	0 <u>0</u> 2 0 0	Remplacement du pixel "0" de l'avant demière exploration par un "2" en gras
0 3—1 0 0	0 3—4 0 0	Le "2" précédent est remplacé par "0". souligné
0 1 1 0 0	0 1 1 0 0	
0 0 0 1 0	0 0 0 1 0	
0 0 0 0 0	0 0 0 0 0	

1ère étape
Exploration depuis "2", dans
le sens des
aiguilles d'une montre du
voisinage 8 de "3"
jusqu'à rencontrer un
point de contour "1" souligné
que l'on remplace par un "4"

Segmentation - Décision : Codage d'un contour

➤ Principe de codage :

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

2 ième étape

On reprend autour de "4" l'exploration des positions de voisinage 8, comme autour du "3" précédent

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

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

3 ième étape

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

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

4 ième étape

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

➤ Principe de codage :

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

5 ième étape

0	0	0	0	0
0	3	—	4	0
0	4	—	4	//
0	2	0	4	0
0	0	0	0	0

0	0	0	0	0
2	3	—	4	0
0	4	—	4	//
0	0	0	4	0
0	0	0	0	0

6 ième et dernière étape

En retrouvant le "3", un dernier point de contour est généré

0	0	0	0	0	0
0	4	—	4	0	0
0	4	—	4	//	
0	0	0	4	0	
0	0	0	0	0	0

Segmentation - Décision : Border Following: Suzuki-Abe Method

(1) Select one of the following:

(a) If $f_{ij} = 1$ and $f_{i,j-1} = 0$, then decide that the pixel (i, j) is the border following starting point of an outer border, increment NBD, and $(i_2, j_2) \leftarrow (i, j - 1)$.

(b) Else if $f_{ij} \geq 1$ and $f_{i,j+1} = 0$, then decide that the pixel (i, j) is the border following starting point of a hole border, increment NBD, $(i_2, j_2) \leftarrow (i, j + 1)$, and $\text{LNBD} \leftarrow f_{ij}$ in case $f_{ij} > 1$.

(c) Otherwise, go to (4).

(2) Depending on the types of the newly found border and the border with the sequential number LNBD (i.e., the last border met on the current row), decide the parent of the current border as shown in Table 1.

(3) From the starting point (i, j) , follow the detected border: this is done by the following substeps (3.1) through (3.5).

(3.1) Starting from (i_2, j_2) , look around clockwise the pixels in the neighborhood of (i, j) and find a nonzero pixel. Let (i_1, j_1) be the first found nonzero pixel. If no nonzero pixel is found, assign $-NBD$ to f_{ij} and go to (4).

(3.2) $(i_2, j_2) \leftarrow (i_1, j_1)$ and $(i_3, j_3) \leftarrow (i, j)$.

(3.3) Starting from the next element of the pixel (i_2, j_2) in the counterclockwise order, examine counterclockwise the pixels in the neighborhood of the current pixel (i_3, j_3) to find a nonzero pixel and let the first one be (i_4, j_4) .

(3.4) Change the value f_{i_3, j_3} of the pixel (i_3, j_3) as follows:

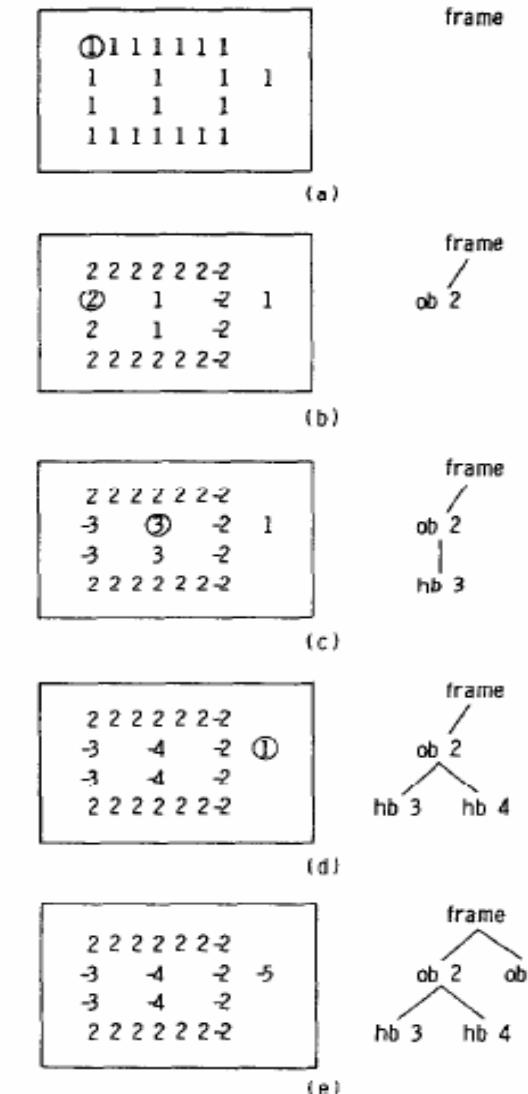
(a) If the pixel $(i_3, j_3 + 1)$ is a 0-pixel examined in the substep (3.3), then $f_{i_3, j_3} \leftarrow -NBD$.

(b) If the pixel $(i_3, j_3 + 1)$ is not a 0-pixel examined in the substep (3.3) and $f_{i_3, j_3} = 1$, then $f_{i_3, j_3} \leftarrow NBD$.

(c) Otherwise, do not change f_{i_3, j_3} .

(3.5) If $(i_4, j_4) = (i, j)$ and $(i_3, j_3) = (i_1, j_1)$ (coming back to the starting point), then go to (4); otherwise, $(i_2, j_2) \leftarrow (i_3, j_3)$, $(i_3, j_3) \leftarrow (i_4, j_4)$, and go back to (3.3).

(4) If $f_{ij} \neq 1$, then $\text{LNBD} \leftarrow |f_{ij}|$ and resume the raster scan from the pixel $(i, j + 1)$. The algorithm terminates when the scan reaches the lower right corner of the picture.

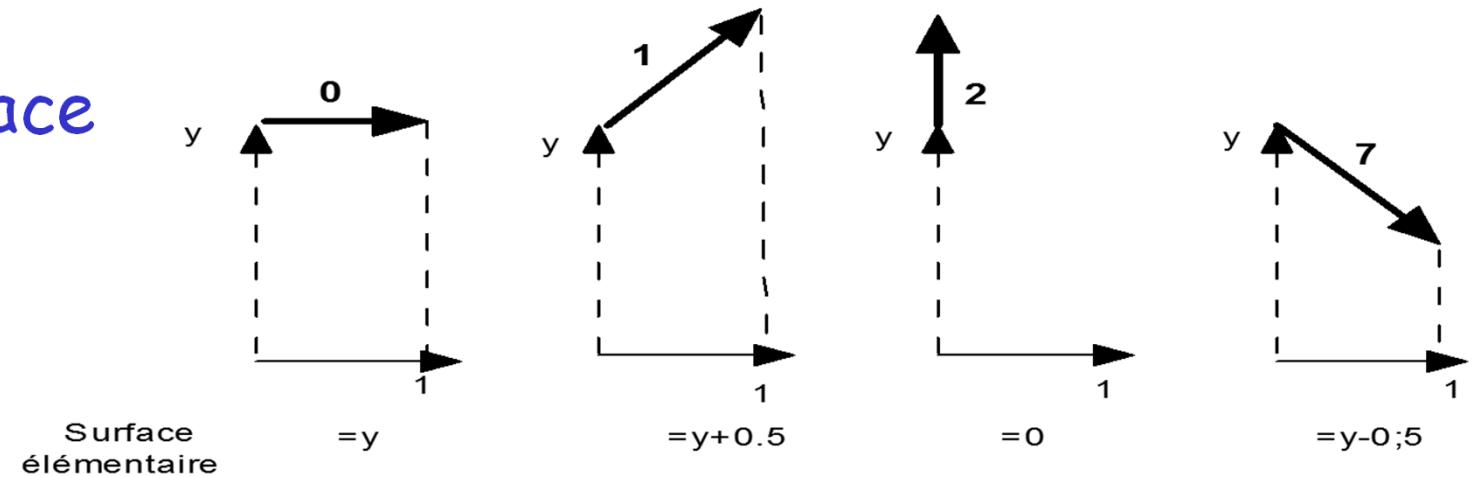


Segmentation - Décision :

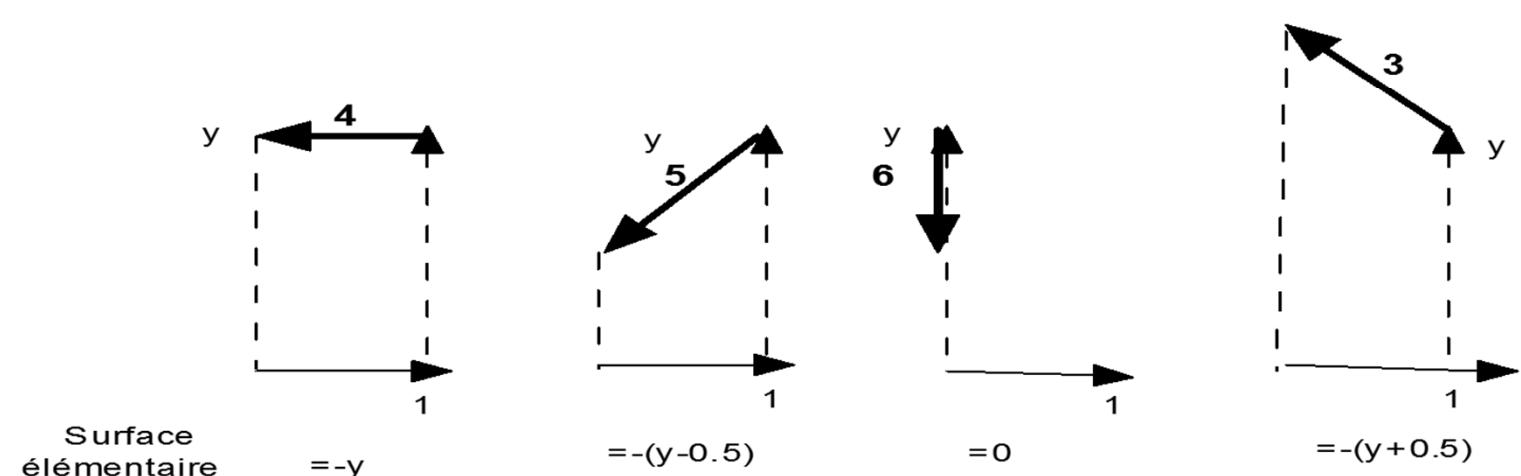
Calcul de surface

Calcul de la surface
d'un objet à
partir du code
de freeman:

Partie supérieure



Partie inférieure

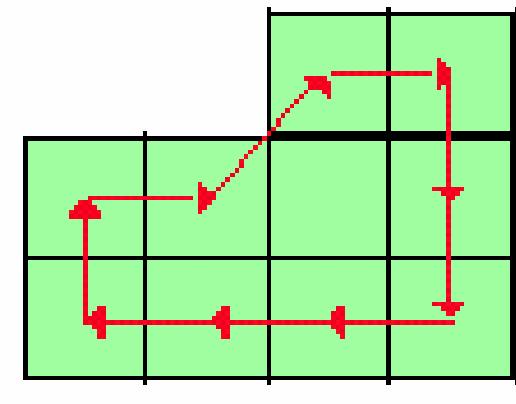


➤ Calcul du périmètre d'un objet à partir du code de freeman:

↳ Trois critères sont possibles pour calculer Pi :

- Compter le nombre de paires adjacentes (p,q) : $p \in S$ et $q \in \bar{S}$
- Compter le nombre d'étapes pour suivre le bord par l'algorithme de suivi de contour.
- Même critère que précédemment, mais les étapes obliques sont comptées pour $\sqrt{2}$.

Ex pour critère 3 : péri (X) = 9.414



Segmentation - Décision Exemple de Caractéristiques

➤ Il existe de nombreuses autres mesures, primitives ou caractéristique à calculer sur une image

↳ Regroupées par catégories :

- Statistiques (moyenne, écart type, moments, ...),
- Géométriques,
- Topologiques,
- Etc...

↳ Diamètres de Ferets.

↳ Indice de compacité ($C = \frac{4\pi S}{l^2}$), circularité, rectangularité, ...

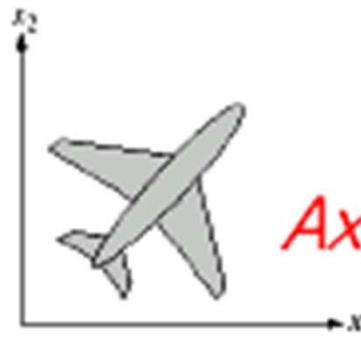
↳ Rectangle encadrant,

↳ Axes principaux,

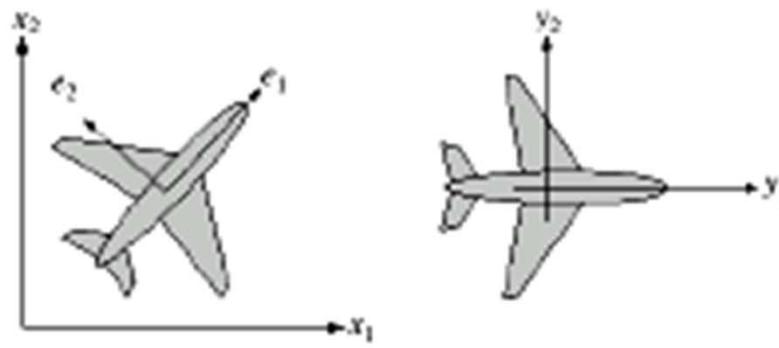
↳ Orientation,

↳ Ellipse (petit axe, grand axe)

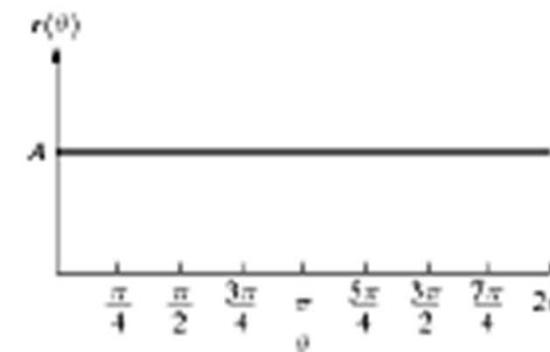
Segmentation - Décision : Exemple de Caractéristiques



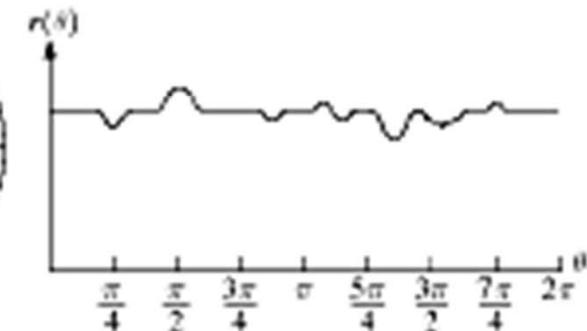
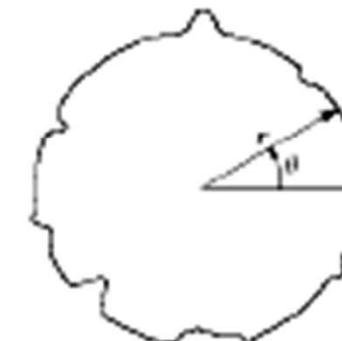
Axes d'inertie



Concavités



Profil angulaire



Segmentation - Décision : Exemple de calcul : IRIS

Exemple : 3 variétés d'Iris sont classifiées par la longueur et la largeur de leurs pétales

- Nous avons donc trois classes
 - ↳ Iris virginica, Iris versicolor, Iris setosa w1, w2 et w3
- Chaque fleur est évaluée par deux descripteurs
 - ↳ Longueur des pétales, largeur des pétales



virginica



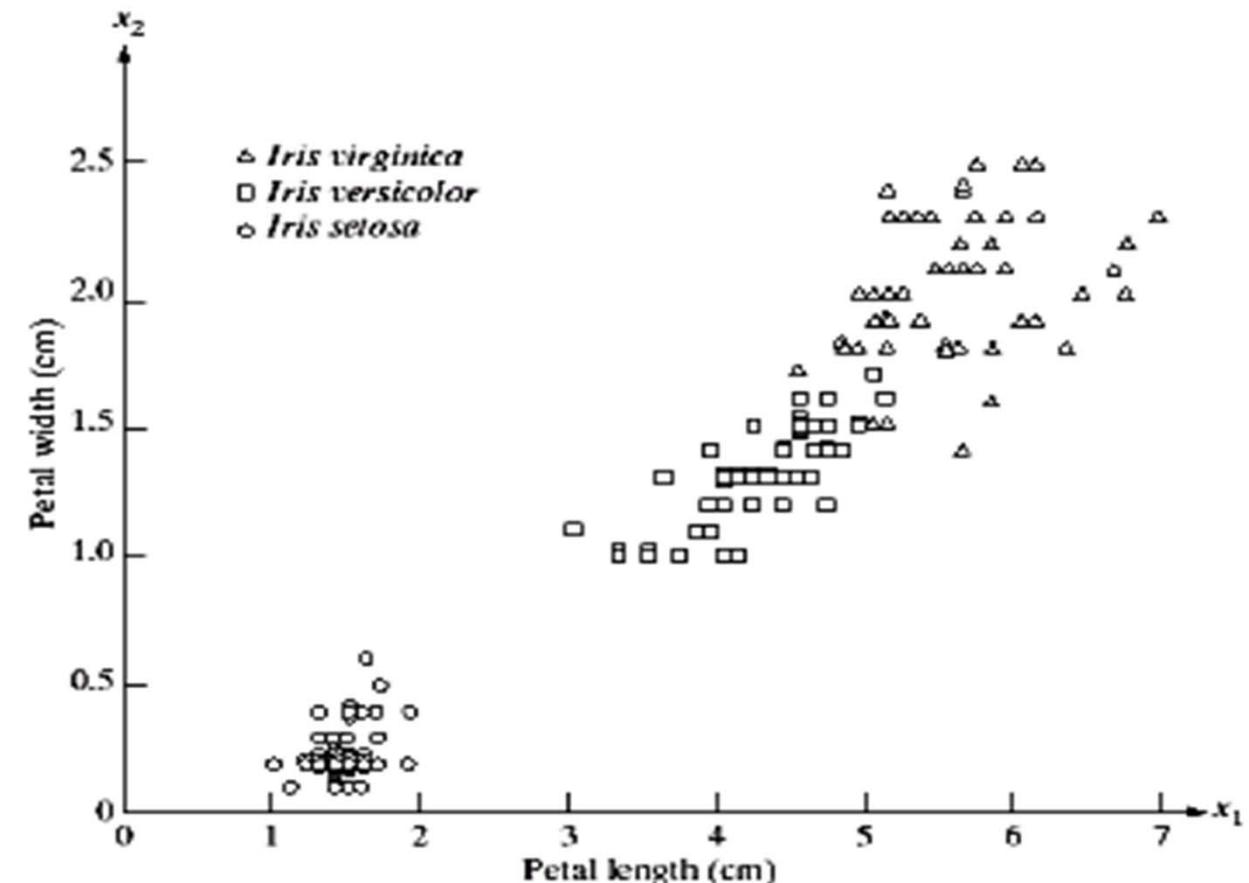
versicolor



setosa

Segmentation - Décision : Exemple de calcul : IRIS

- La variété Setosa est bien différenciée des deux autres variétés.
- Il est difficile de différencier les deux autres variétés sans erreur.
- Il s'agit d'un problème avec la sélection des caractéristiques pour décrire la forme.



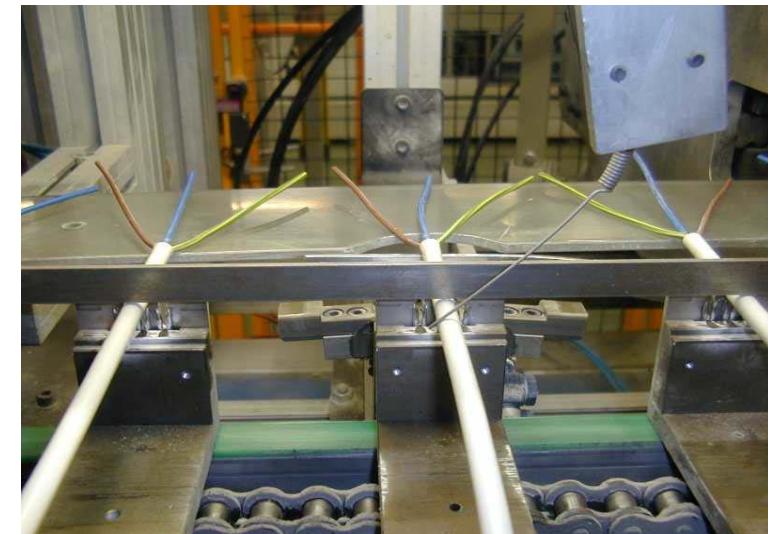
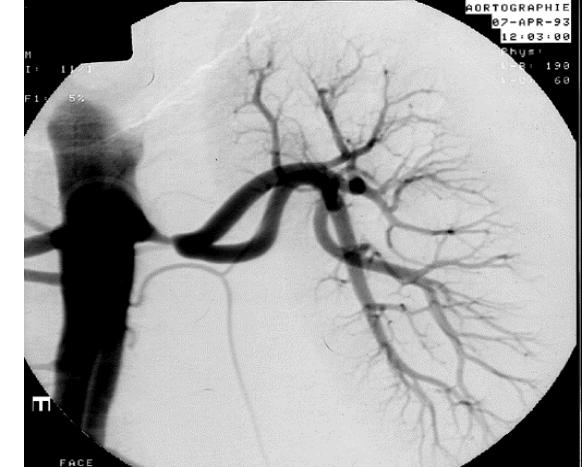
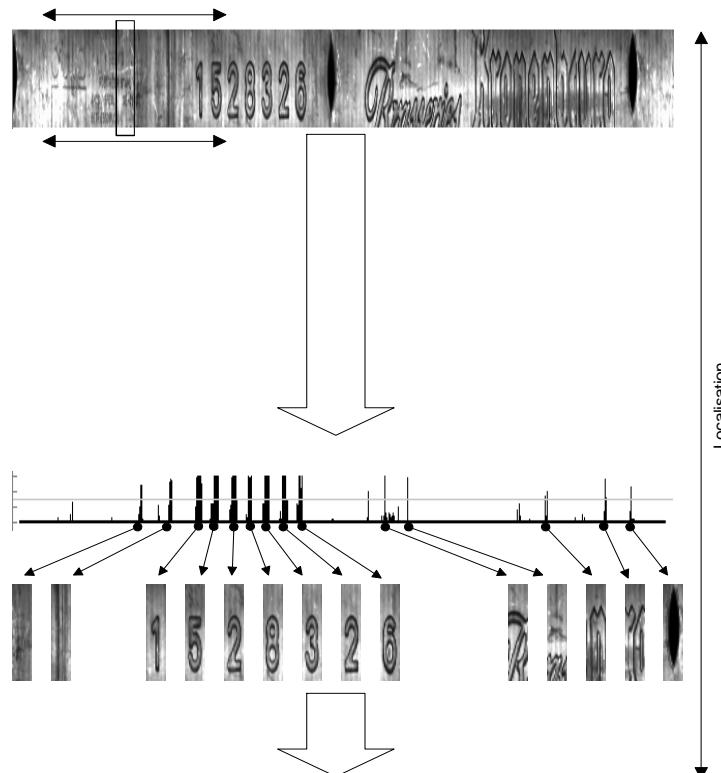
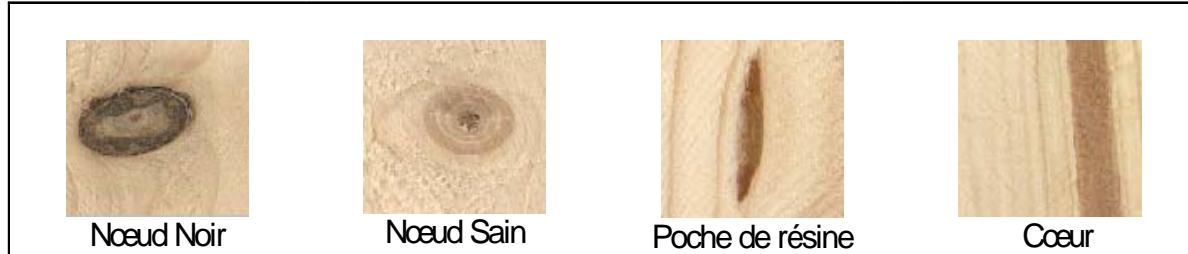
↳ Importance
de la sélection de
caractéristiques
Significatives !

- Etapes de haut niveau: Intelligence Artificielle
 - ↳ En fait regroupent plusieurs opérations souvent indissociables:
 - la décision est intimement liée à l'objectif fixé
 - l'interprétation également



Segmentation - Décision :

Quelques exemples de Rdf

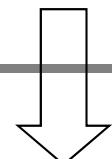


- C'est un rond, c'est un carré,
- le feu est vert,
(je passe, ou je m'arrête)
- votre électrocardiogramme est normal
- c'est une facture téléphone

Distance avec des formes de références
Classe = action possible

Diagnostic = détection : signal ou bruit
Modèle = les « règles » (même source)

- odeur : c'est une madeleine
- caractère - écriture
(c'est une lettre, un mot,...)
- parole (forme temporelle)
- voix (c'est Chirac aux guignol),
- visage (vision)
- identification (visage + voix + odeur + empreintes)
- une voiture
- il va peut-être pleuvoir



Capteur complexe Aspects humains
Complexité de la tâche
Modélisation par apprentissage
Temps (système évolutif :environnement)
Complexité de l'espace des caractéristiques
Invariances
Fusion - (informations hétérogènes)
Concept imprécis
Notion d'incertitude

➤ Classification :

↳ Modèle de classes / Frontières de classes

➤ Détection de défauts :

↳ Modélisation de la « non-normalité »

➤ Identification :

↳ Modélisation d'individus parmi des inconnus

➤ Vérification

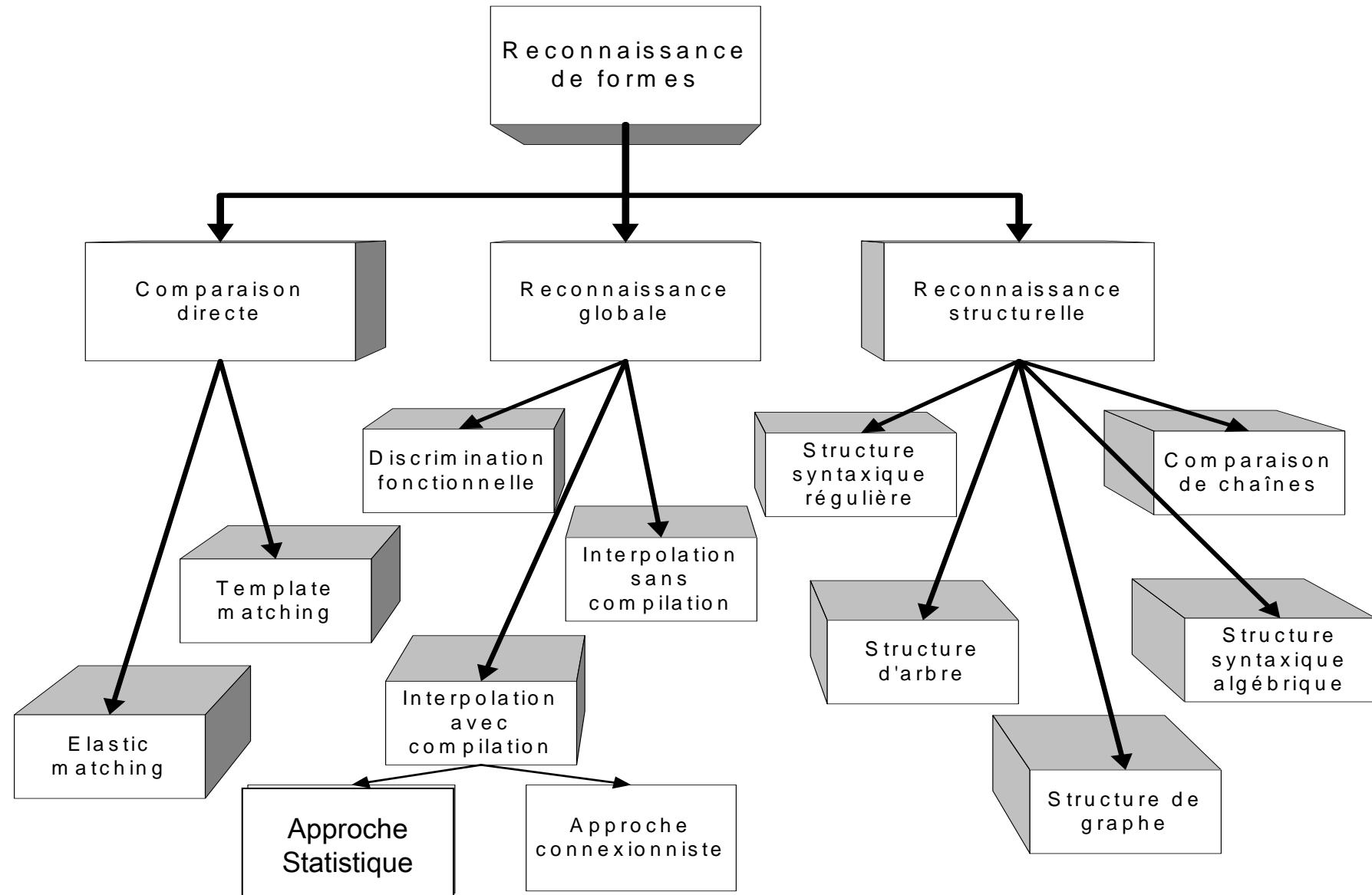
↳ Identification d'un individu d'un ensemble connu sans étiquetage

➤ Reconnaissance de formes

↳ Etiquetage d'un individu parmi un ensemble connu

Segmentation - Décision :

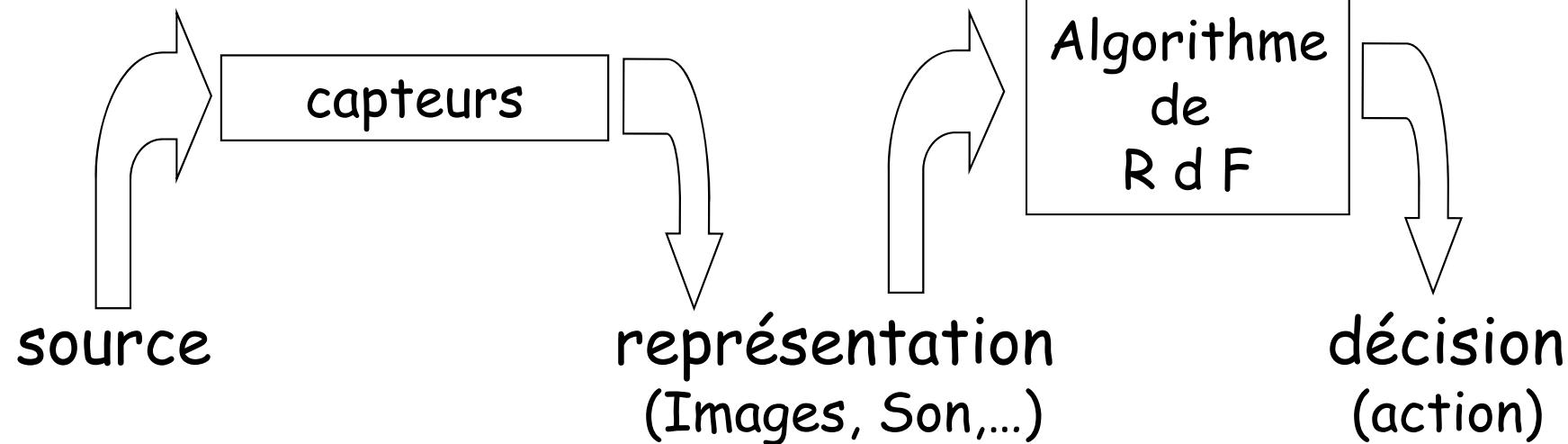
Les différentes catégories de la RdF



Segmentation - Décision :

Les différentes phases :
Méthodes de comparaison Directe

Template matching:
Corrélation, RdN,...



espace
des
sources

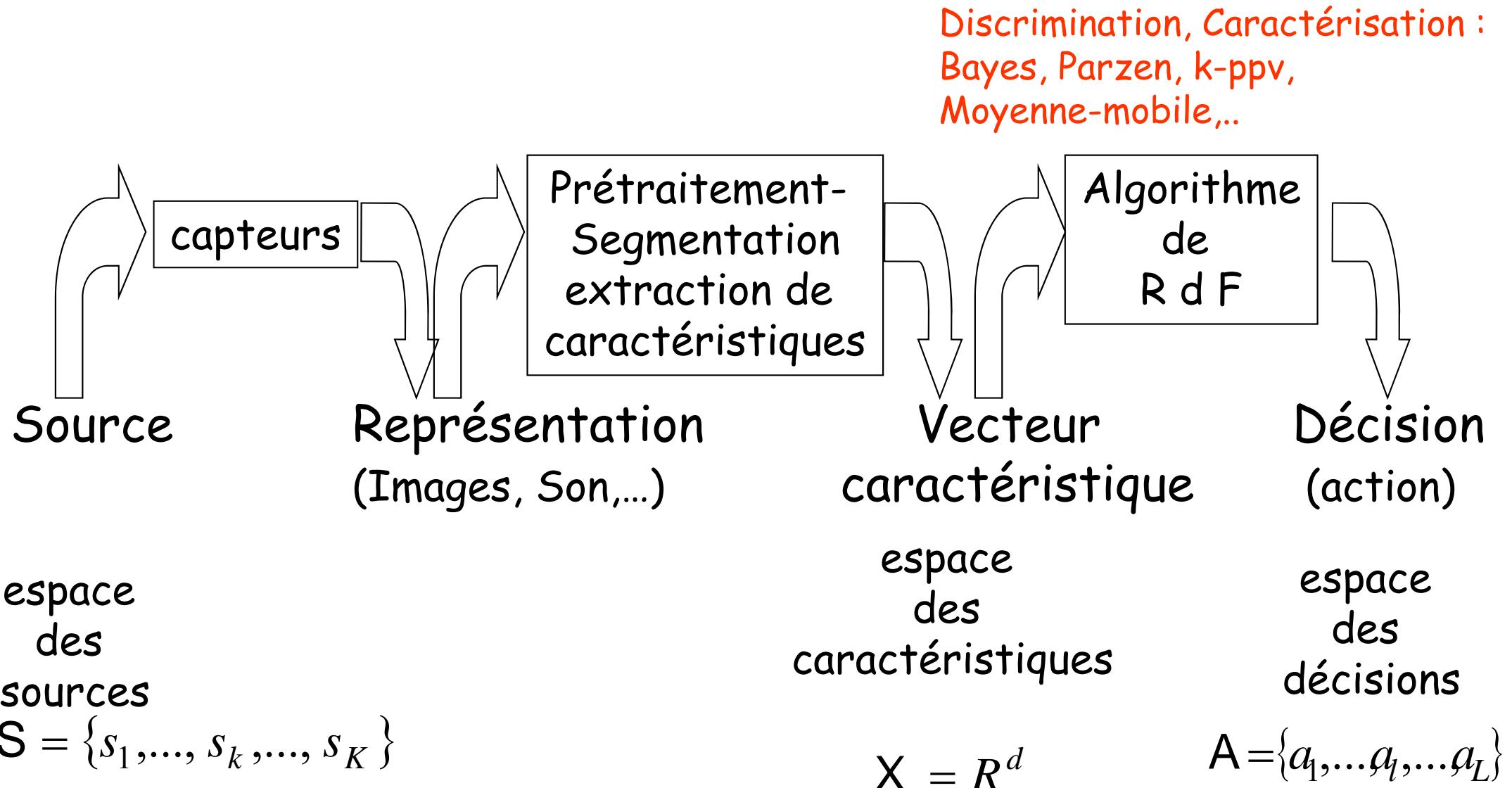
$$S = \{s_1, \dots, s_k, \dots, s_K\}$$

espace
des
décisions

$$A = \{a_1, \dots, a_j, \dots, a_L\}$$

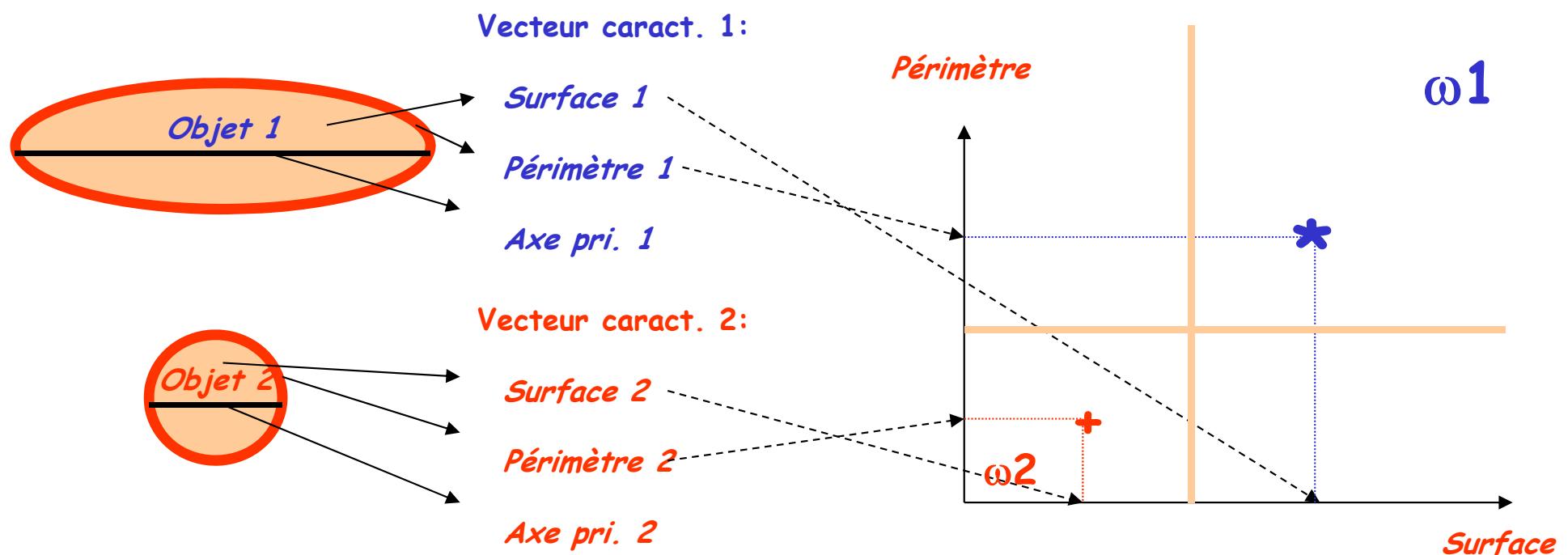
Segmentation - Décision :

Les différentes phases :
Méthodes de reconnaissance globale 1/4



On appelle **caractéristique**(ou descripteur) une information qui peut être mesurée sur la forme à reconnaître.

Le vecteur caractéristique est composé des plusieurs caractéristiques choisies pour représenter la forme (Espace de représentation)



Segmentation - Décision :

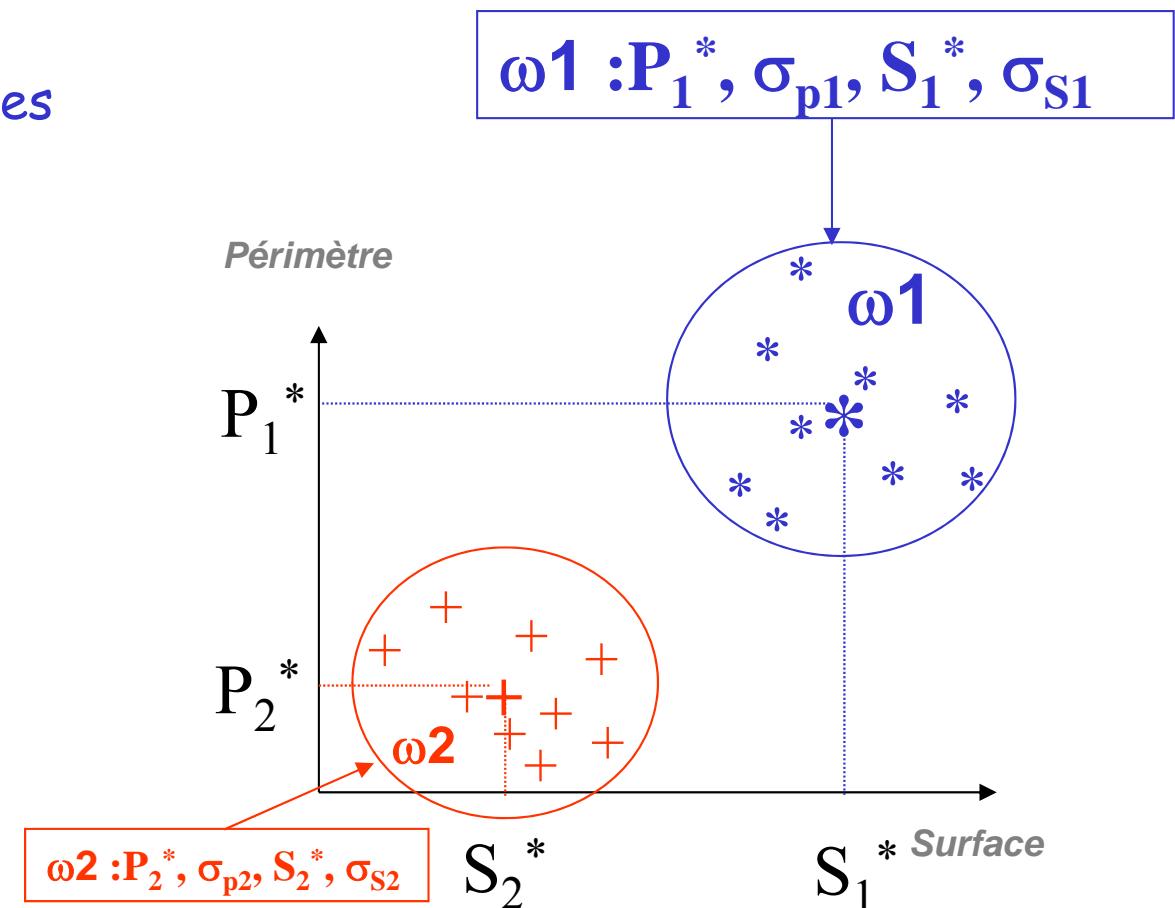
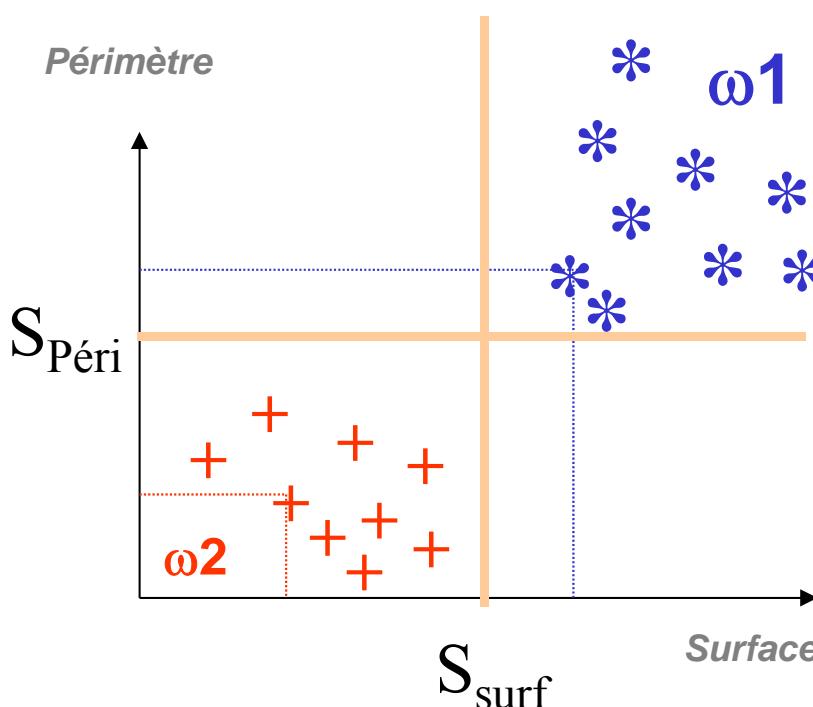
Les différentes phases :
Méthodes de reconnaissance globale 3/4

➤ En extension (caractérisation) :

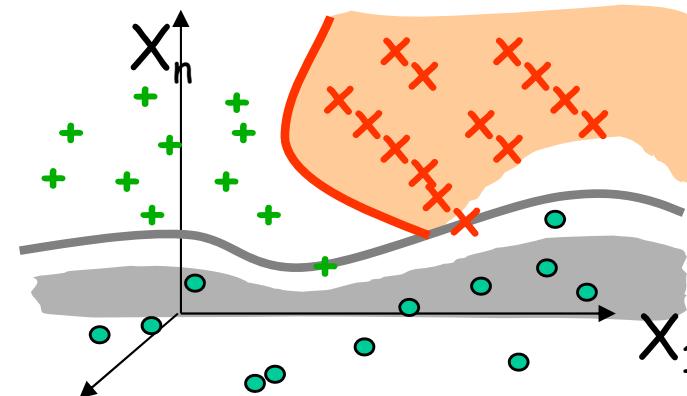
- ↳ Un ou plusieurs exemplaires : Prototypes de la classe (K-ppv).
- ↳ Ensemble d'exemplaires "résumés" : Modèle de la classe (gaussienne).

➤ En intension (discrimination) :

- ↳ Un sous espace de R^n .
- ↳ Une frontière entre les formes

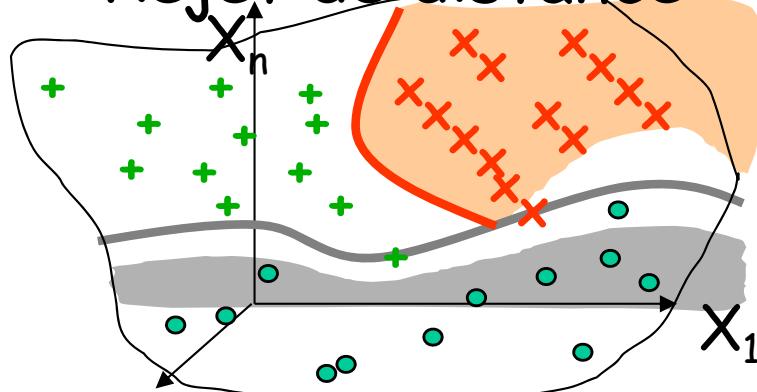


prototypes

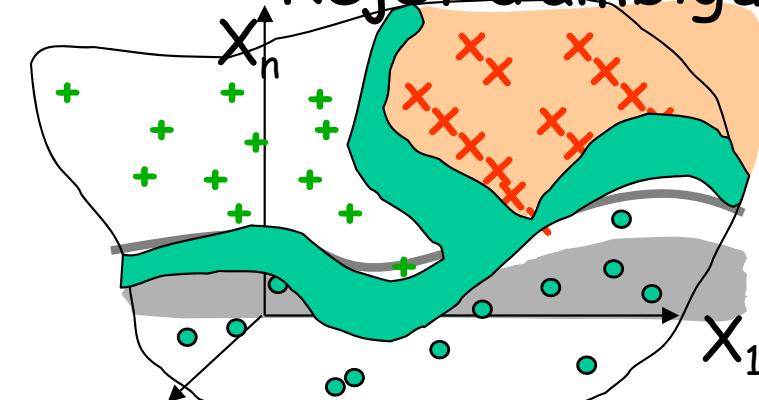


Frontière de décision
fonction $b(x)=0$
telle que $D(x-e) \neq D(x+e)$

Rejet de distance



Rejet d'ambiguité



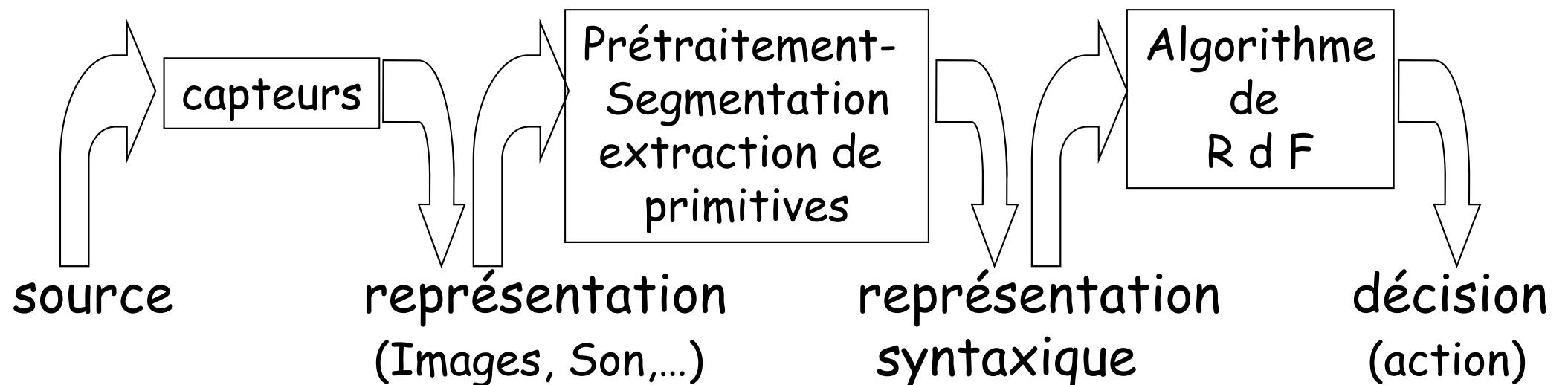
Variabilité intra-classe et séparabilité des classes :
Minimiser la distance intra-classe
Maximiser la distance inter-classes

Segmentation - Décision :

Les différentes phases :
Méthodes de reconnaissance structurelle

Vocabulaire : {A,B,C}

Analyse syntaxique,
Inférence grammaticale,
Chaîne de Markov



espace
des
sources

$$S = \{s_1, \dots, s_k, \dots, s_K\}$$

Phrase

$$X = AAABAACCCA A$$

espace
des
décisions

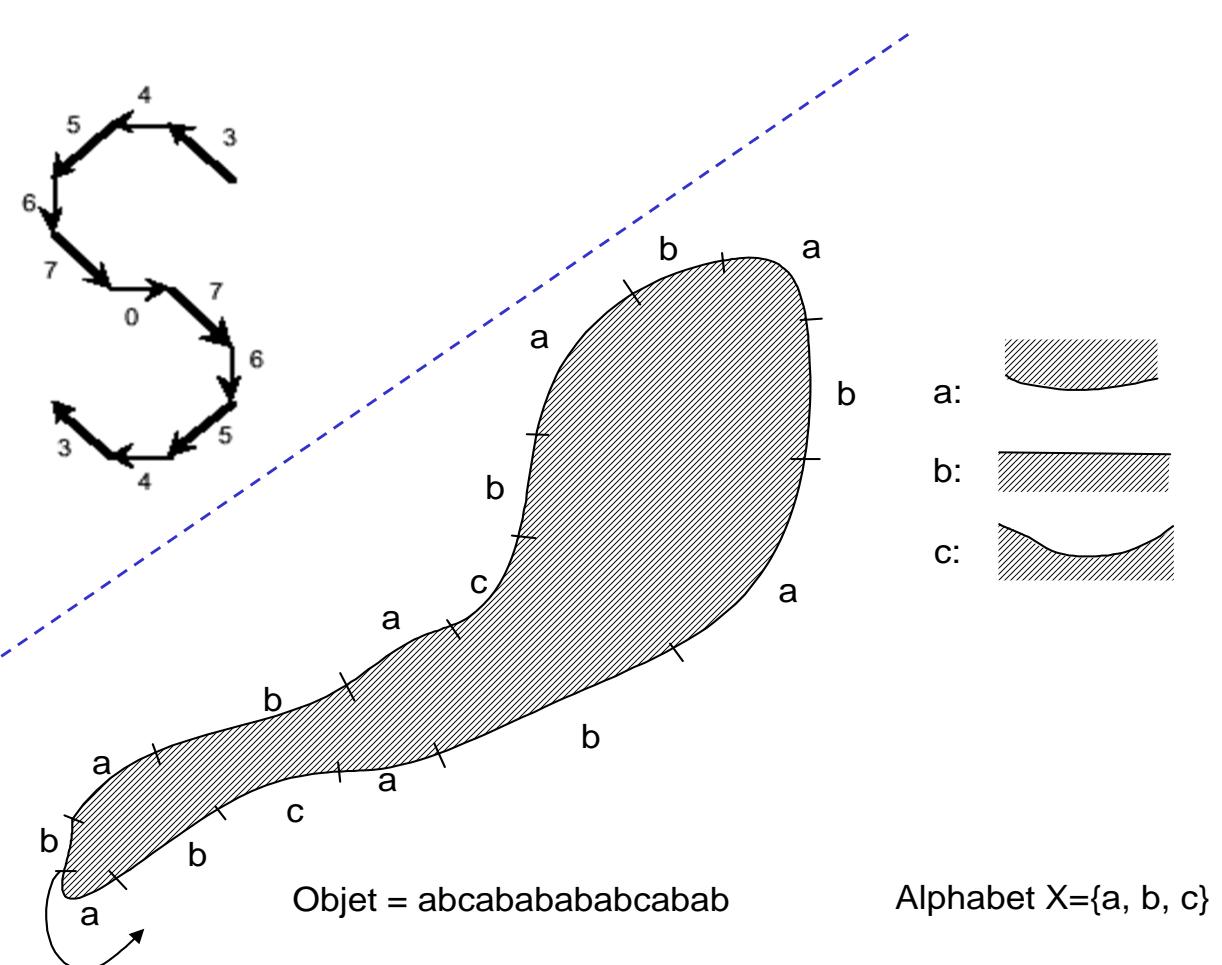
$$A = \{a_1, \dots, a_l, \dots, a_L\}$$

Segmentation - Décision :

Les différentes phases :
Méthodes de reconnaissance structurelle

➤ On appelle **primitive** une composante élémentaire d'une forme, les primitives ne sont pas décomposables. Par exemple : un segment de droite, une boucle ...

Lettre S formée de segments
(munis des codes de Freeman) :



Segmentation - Décision^{RdF:} 2 points de vue : utilisateur - concepteur

Utilisation :

Une forme
(vecteur forme
des caractéristiques)

Algorithme
de
Reconnaissance
des Formes

C'est
la forme
« y »

Apprentissage :

Lot d'apprentissage

Données

Connaissances

Méthode de construction
de l'Algorithme de
Reconnaissance
des Formes

Méthodes
Supervisées



- *Lots d'apprentissage* : « par cœur », taille, choix, ordre d'occurrence des échantillons,...
- **Auto apprentissage** : horizon d'observation, facteur d'oubli,...
- **Evaluation des performances** :
Capacité de prédiction du modèle
La vitesse de classification, d'induction du modèle et sa maintenance sont des éléments secondaires
- **Validation croisée**: vérité terrain (ground truth), « leave one out ».

- *L'important est la capacité de prédiction du modèle*
 - ↳ La vitesse de classification, d'induction du modèle et sa maintenance sont des éléments secondaires
- *La matrice de confusion*

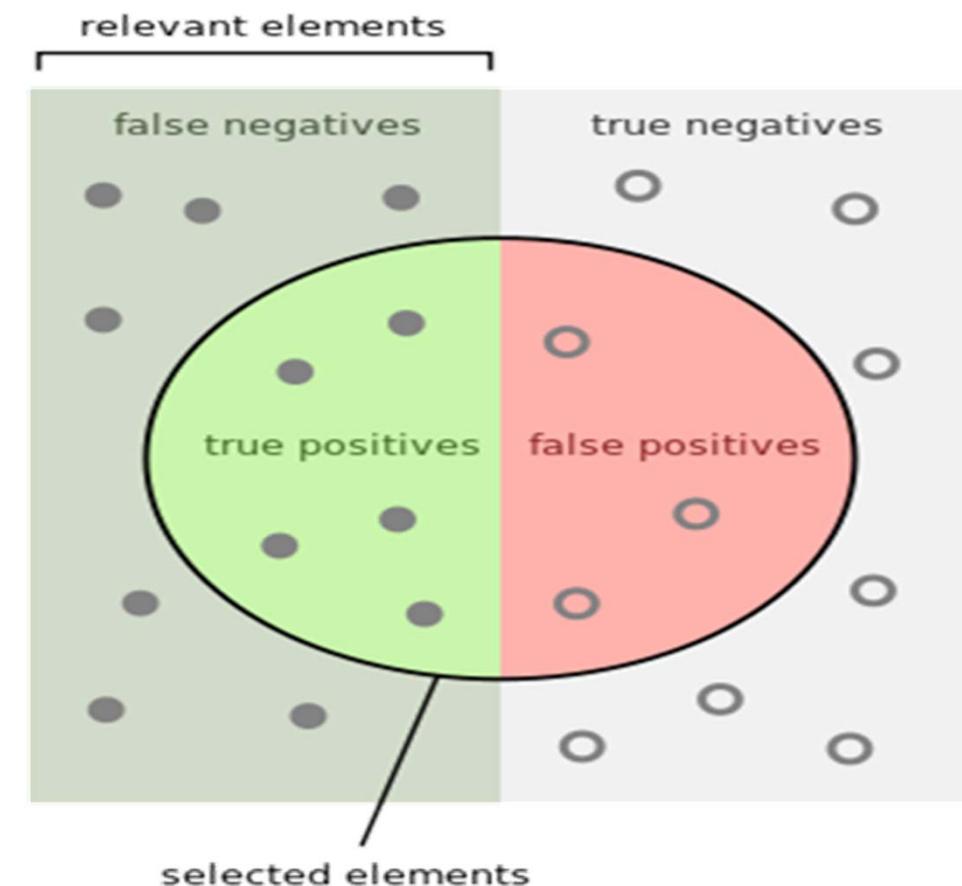
		PREDICTED CLASS	
		Class=Yes	Class=No
TRUE CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)



$$\text{Accuracy (Justesse)} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

		Classe prédite	
		Class=Yes	Class>No
Classe réelle	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

Taux d'Erreur ou Taux de Reconnaissance (Accuracy) : évaluation duale

Mémorisation : taux calculé sur le lot d'apprentissage

Généralisation : taux calculé sur un lot test « inconnu »

Justesse totale : mémorisation généralisation

↳ *La justesse ne tient pas compte de la répartition des données dans les classes*

		Classe prédictive	
Classe réelle		Class=Yes	Class>No
	Class=Yes	9990	0
	Class>No	10	0

- ↳ *Si le modèle classe toutes les données dans la classe yes,*
- *sa justesse est*
 $(9990+0) / (9990 + 0 + 10 + 0) = 99.9 \%$
 - *mais il est incapable de reconnaître une donnée de la classe No*
- ↳ *Est-ce un classificateur utile ?*

$$\text{Precision} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{a}{a+b} = \frac{TP}{TP+TN}$$

$$F\text{-measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2TP}{2TP + FP + FN}$$

C (prédict / réel)

#	Classe prédite	
Classe réelle	Class=Y	Class=N
	Class=Y	a
Class=N	c	d

- Précision est sensible à $C(\text{Yes}|\text{Yes})$ & $C(\text{Yes}|\text{No})$
- Rappel est sensible à $C(\text{Yes}|\text{Yes})$ & $C(\text{No}|\text{Yes})$
- F-mesure (moyenne harmonique) n'est pas sensible à $C(\text{No}|\text{No})$
 - plus proche de la valeur la plus faible (entre p et r)
 - donc une valeur forte signifie que la justesse et le rappel est bon

- L'apprentissage :
ce qu'un bébé réalise en deux ans, aucune machine n'est aujourd'hui capable de le réaliser. (besoin d'exemples)
- Choix du nombre de classes : classes surnuméraires
- Modéliser l'information, dépasser la complexité
- Fusion de données hétérogènes
- Sélection d'entrées
- Prise en compte du temps
- Représentation des incertitudes
- Gestion des imprécisions

- Le "No Free Lunch Theorem" dit qu'en l'absence d'information *a priori* sur le problème à traiter, il n'y a pas d'algorithme d'apprentissage supérieur à un autre!
- Il y a autant de problèmes pour lesquels un algorithme est supérieur à un autre qu'il y a de problèmes où c'est l'inverse!
- En moyenne, sur tous les problèmes de classification possibles, les algorithmes de classifications ont les mêmes performances!
- Le "Ugly Ducking Theorem" dit qu'il n'y a pas d'ensemble de caractéristiques meilleur qu'un autre pour l'ensemble des problèmes (ou en l'absence d'*a priori* sur la nature du problème).
- La qualité d'un ensemble de caractéristiques dépend donc du problème.

- Ensemble ces deux théorèmes nous disent qu'il n'y a tout simplement pas d'algorithme ou ensemble de caractéristiques "universellement" meilleur
- La théorie et les algo. ne suffisent donc pas! Il faut connaître le problème.

La RdF est un sujet empirique!

Comme le traitement d'Images 😞