



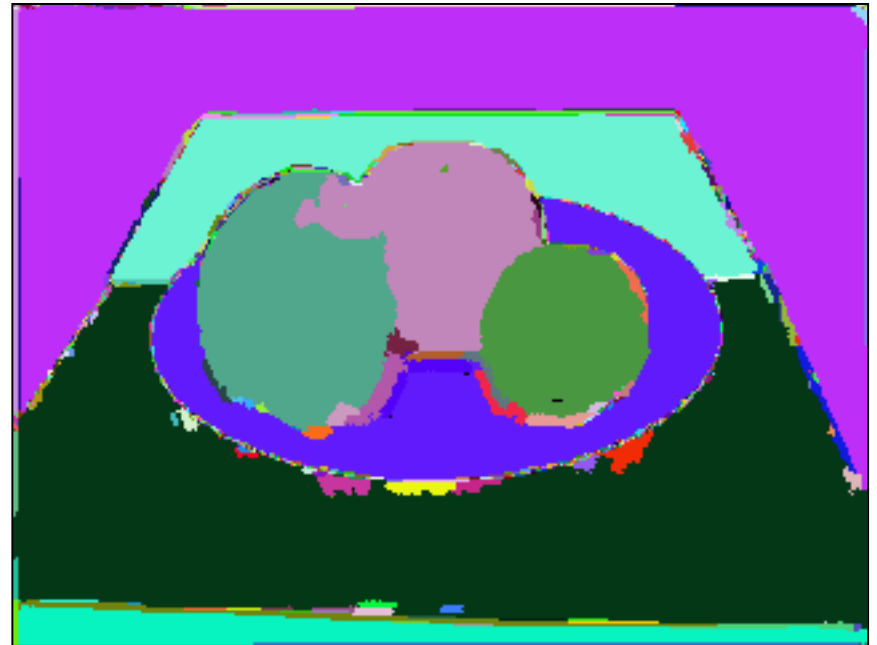
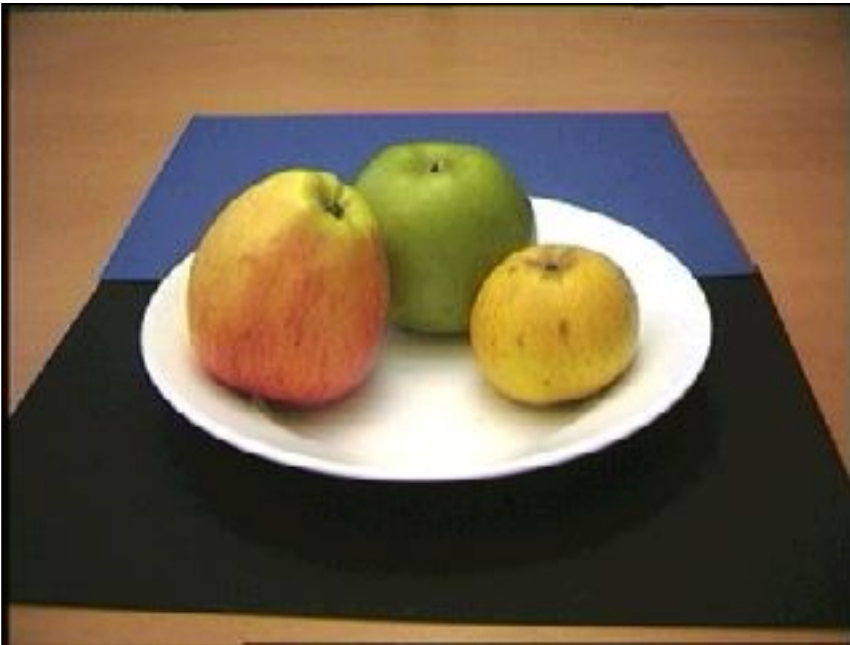
Traitement d'images

Segmentation

NGUYEN Thi Oanh – IPH
oanhnt@soict.hust.edu.vn

Qu'est-ce que la segmentation ?

- La segmentation vise à diviser l'image en morceaux
 - *Ces morceaux correspondent aux objets dans l'image*
- La segmentation est liée à la reconnaissance
 - *Quels objets voit-on dans l'image ?*



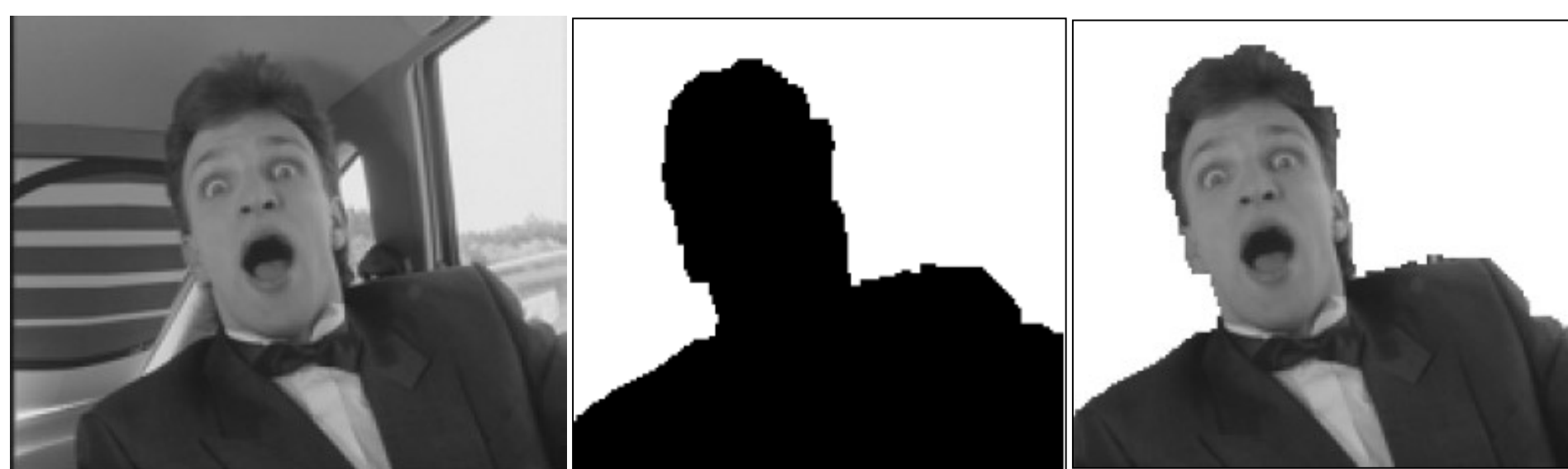
But de la segmentation

- 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



Utilité d'un masque de segmentation

- Les masques (des objets) permettront d'interpréter le contenu de l'image
 - *On traite chaque zone de l'image (masques) séparément*





Qu'est-ce que la segmentation ?

- La segmentation est normalement basée sur:
 - *les discontinuités : contours*
 - les changements abruptes, frontières entre régions...
 - *les zones homogènes : régions*
 - Mêmes couleurs, textures, intensités, ...
- La segmentation est le découpage d'une image en différentes **régions** et/ou **contours**
- **Un contour fermé est équivalent à une région**



Segmentation régions/contours

- **Approches régions**

- Rechercher les zones dans l'image sur un critère d'homogénéité

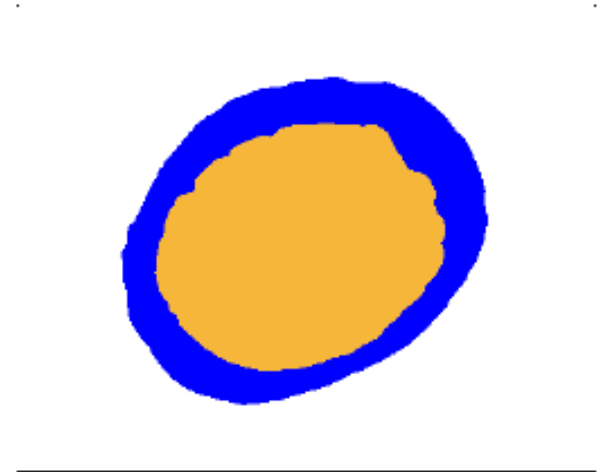
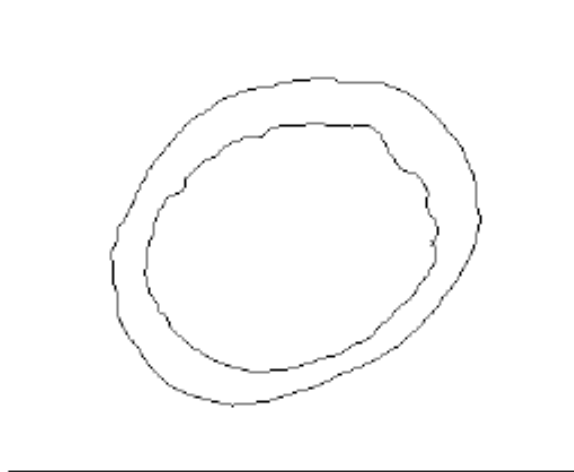
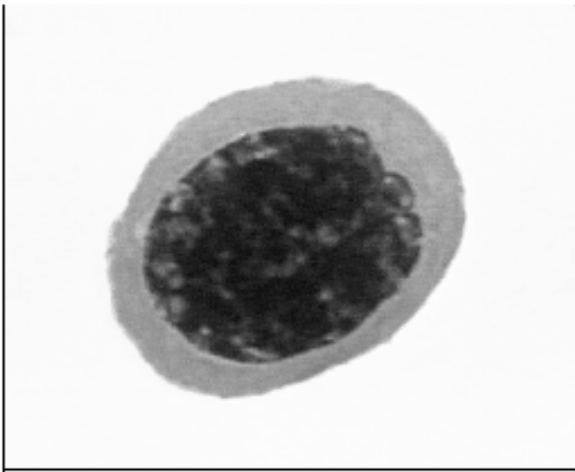
- **Approches contours**

- Rechercher les discontinuités entre régions

- **Approches duales** (régions et contours)

Dualité régions/contours

Un contour fermé est équivalent à une région.





Seuillage



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

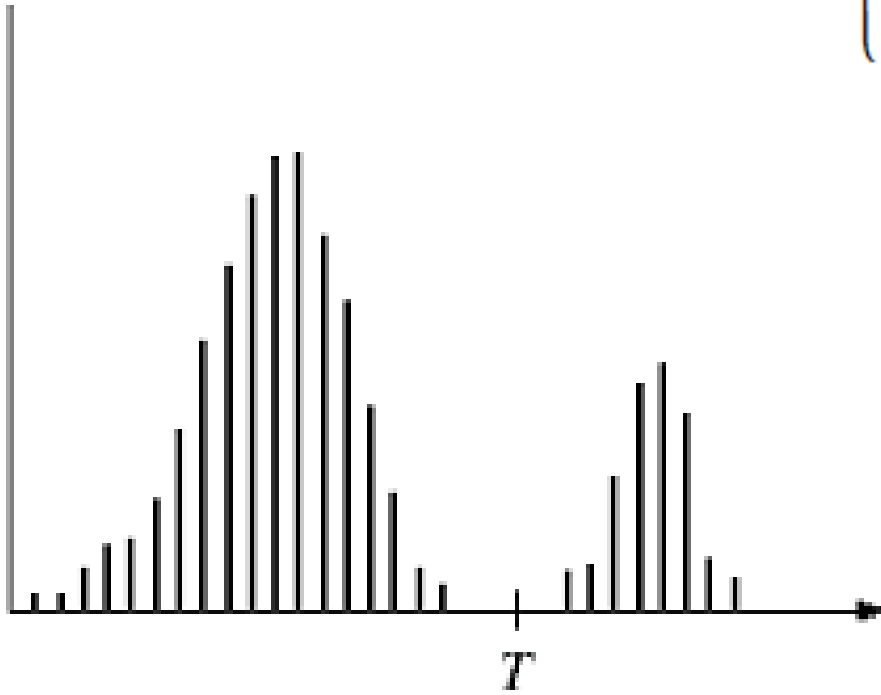


Principe de base

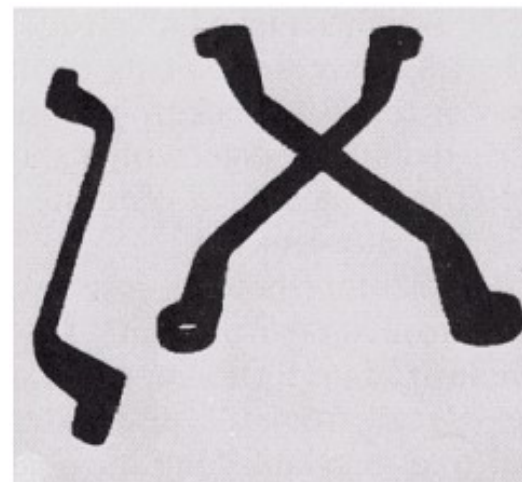
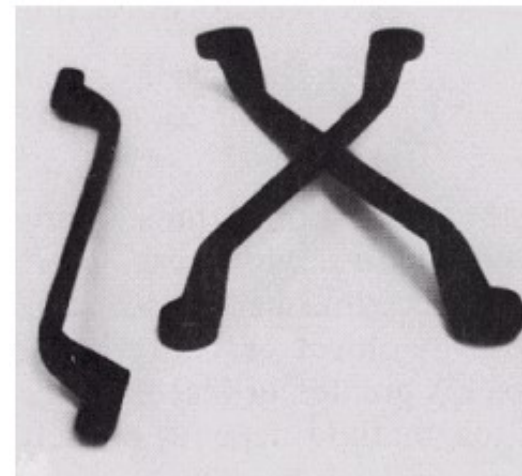
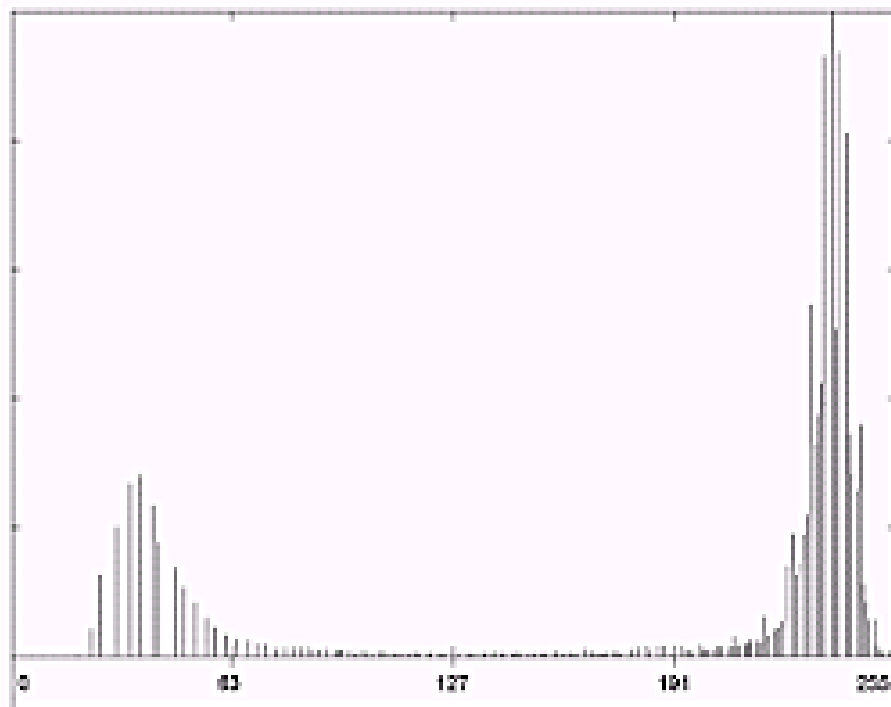
- Seuillage de base (2 classes) :
 - Si *valeur(pixel) \geq seuil* alors *valeur(pixel) = 1*
 - Si *valeur(pixel) < seuil* alors *valeur(pixel) = 0*
- Le résultat du seuillage est une image binaire
 - 0 ou 1 (*qu'on transforme parfois en 0:255 pour l'affichage*)
- **Problème** : choix du seuil !

Seuillage d'histogramme simple

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) \geq T \\ 0 & \text{si } f(x, y) < T \end{cases}$$



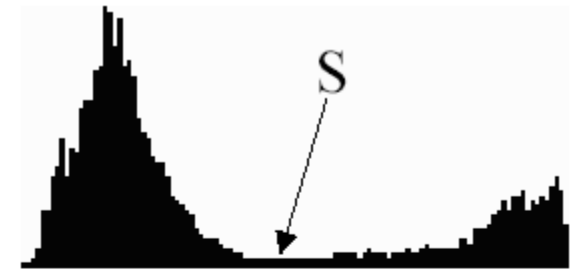
Exemple de seuillage



Seuillage d'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

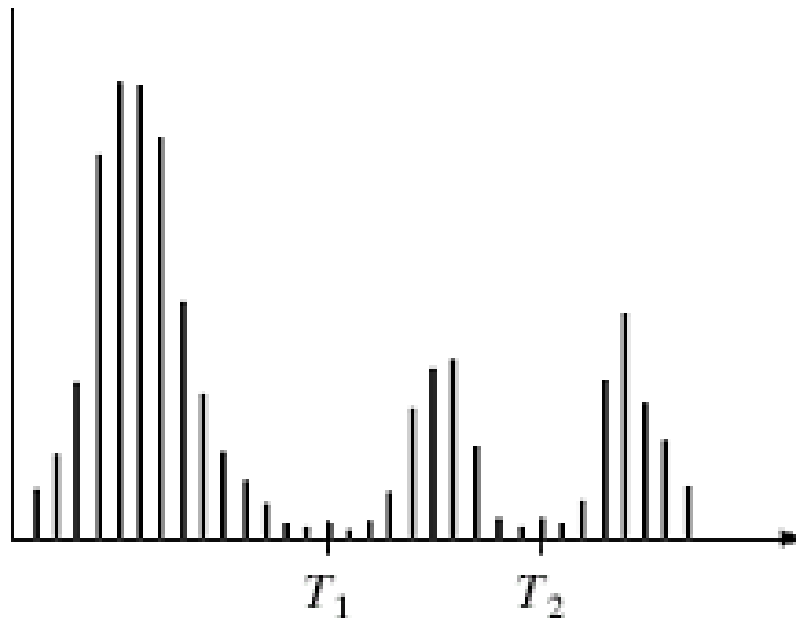


Multi-seuils

- n seuils pour séparer l'image en $n+1$ classes :
 - Si $\text{valeur}(\text{pixel}) < \text{seuil}_1$
 - alors $\text{valeur}(\text{pixel}) = 0$
 - Si $\text{valeur}(\text{pixel}) \geq \text{seuil}_1 \ \&\& \ \text{valeur}(\text{pixel}) < \text{seuil}_2$
 - alors $\text{valeur}(\text{pixel}) = 1$
 - ...
 - Si $\text{valeur}(\text{pixel}) \geq \text{seuil}_n$
 - alors $\text{valeur}(\text{pixel}) = n$
- **Problèmes** : Combien de seuils + Valeurs des seuils !

Seuillage d'histogramme multiple

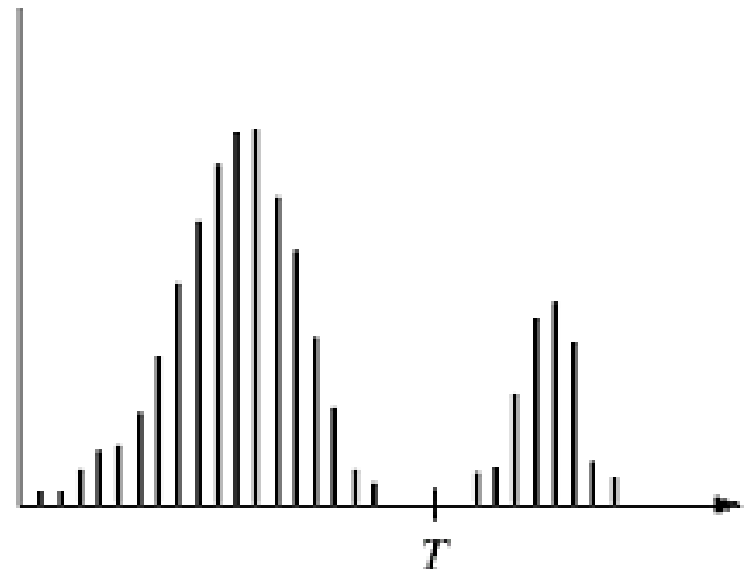
$$g(x, y) = \begin{cases} 2 & \text{si } f(x, y) \geq T_2 \\ 1 & \text{si } f(x, y) \geq T_1 \\ 0 & \text{si } f(x, y) < T_1 \end{cases}$$



Comment définir le seuil ?

- 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 !





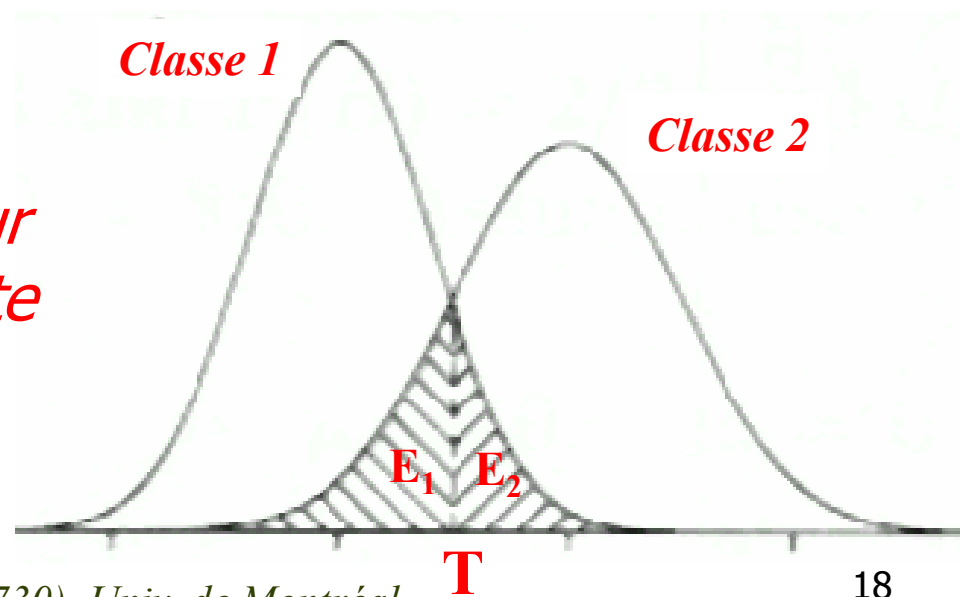
Simple methode à trouver un seuil

- Etape 0 :
 - T_0 = moyenne des niveaux de gris de tous les pixels
- Etape i : Seuillage avec T_i
 - Calcul $m1$ = moyenne du groupe 0 (pixel dont le niveau de gris $< T_i$)
 - Calcul $m2$ = moyenne du groupe 1 (pixel dont le niveau de gris $\geq T_i$)
 - Recalculer la valeur du seuil $T_{i+1} = (m1 + m2)/2$
- Répéter $T_{i+1} \sim T_i$

Choix des seuils (optimaux)

- 2 surfaces (arrière-plan et objet) dans une image
- On suppose des modèles mathématiques pour les distributions (gaussiennes, ...)
- On peut déterminer la probabilité d'erreur de classification dans les classes 1 et 2 (surfaces E_1 et E_2 ci-dessous)
- On cherche alors un seuil T qui causera une erreur minimale

Modéliser les histogrammes pour trouver le meilleur seuil : il existe plusieurs méthodes pour cela !





Seuillage global automatique

- Il existe plusieurs méthodes globales automatiques qui permettent de trouver un seuil à partir d'un histogramme
 - *Otsu, Kittler, ...*
- Mais chaque méthode donne un seuil différent
 - *Quel est le bon seuil ?*
- Il faut tester pour chaque nouvelle application

Exemple : Algorithme de 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

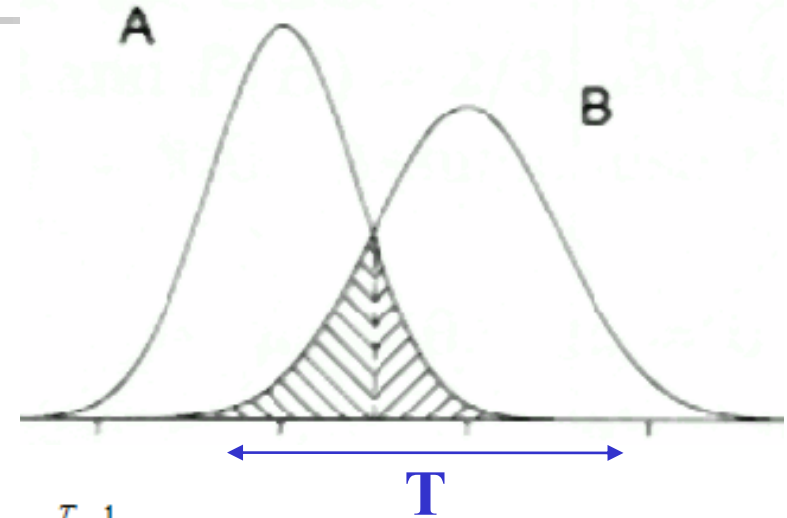
Variances : σ_1^2 et σ_2^2

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



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

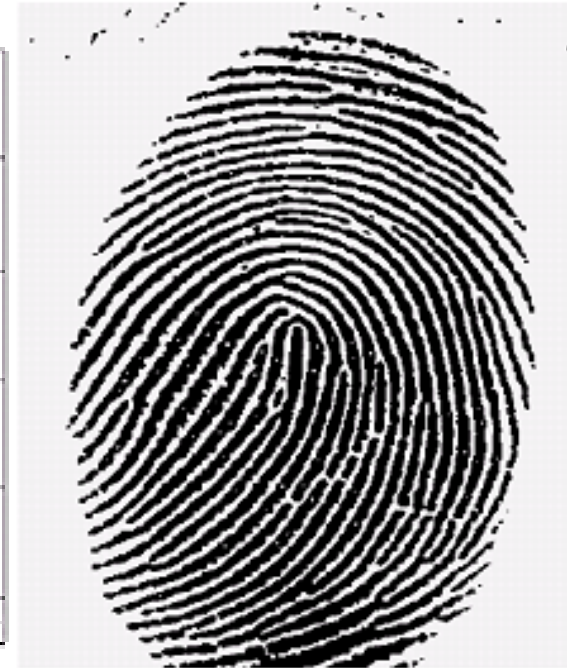
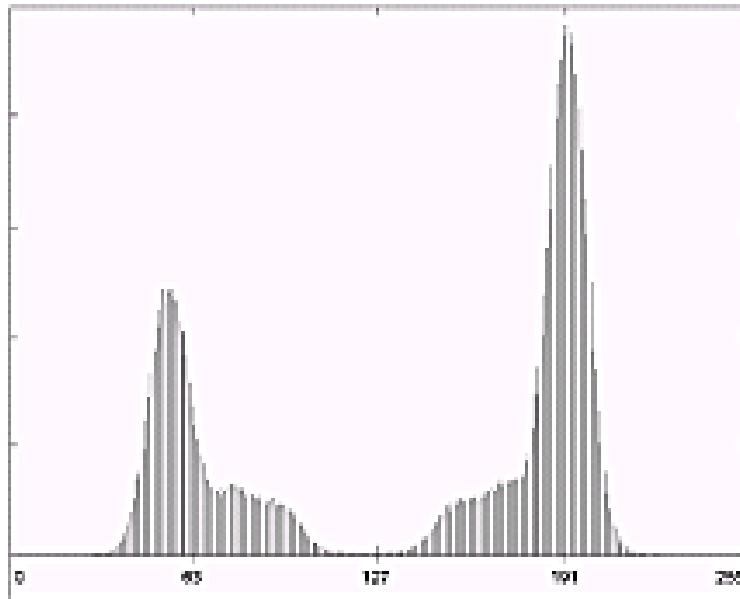
$$\sigma_2^2 = \frac{1}{256 - T} \sum_{i=T}^{255} (h(i) - \mu_2)^2$$

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

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

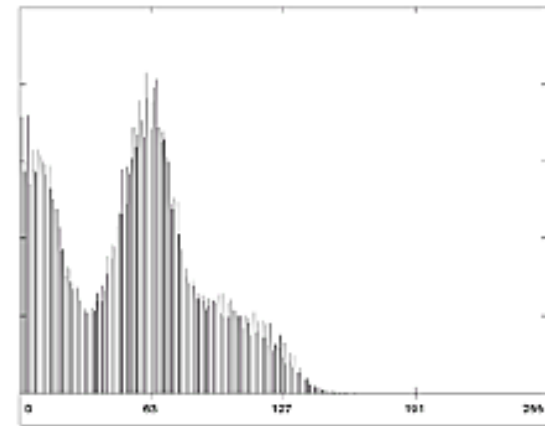
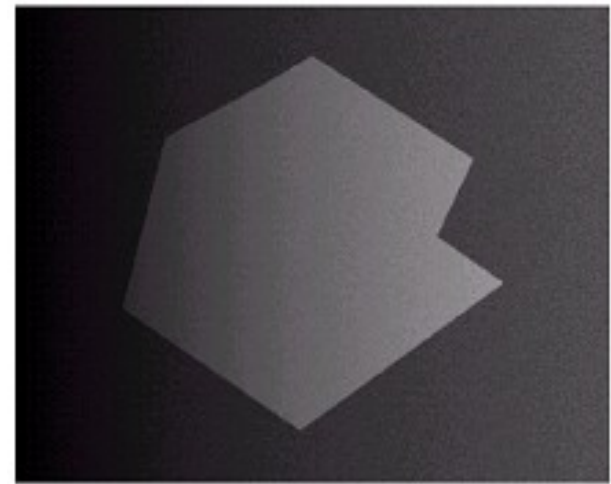
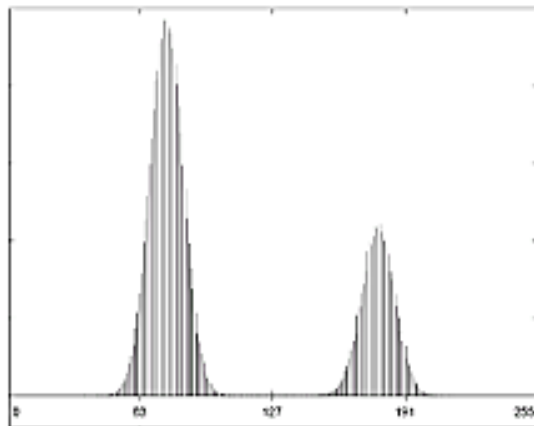
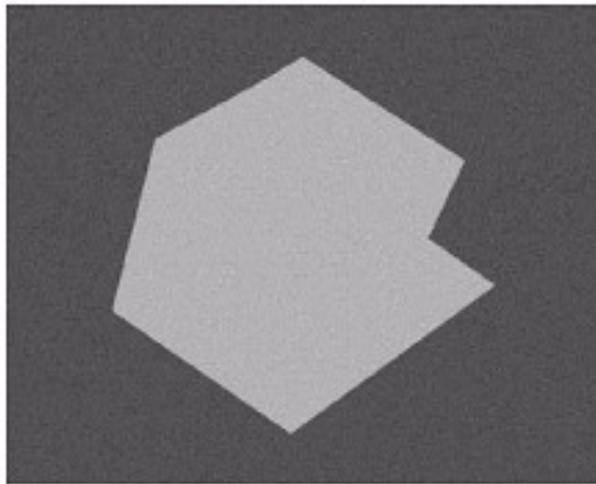
Seuillage global automatique

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



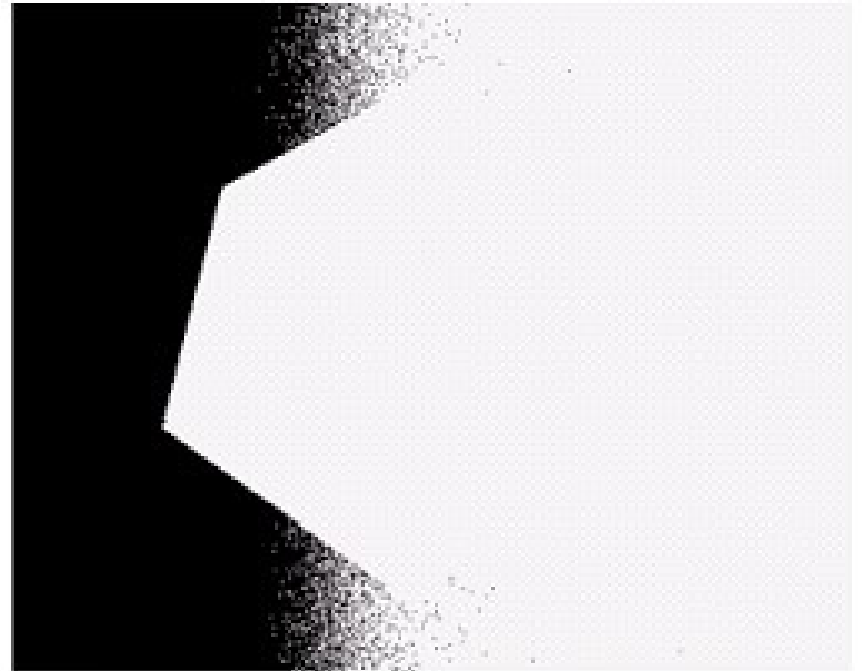
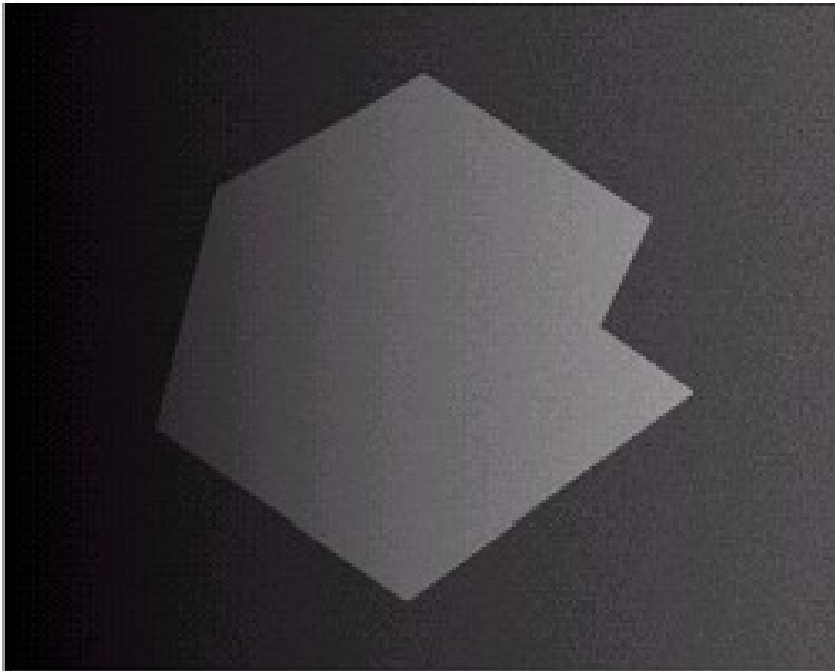
Seuillage global - problème

Problème d'éclairage ?



Seuillage global - problème

- **Problème** : Le seuillage global ne peut traiter ce cas
- **Solution** : seuillage local adaptatif



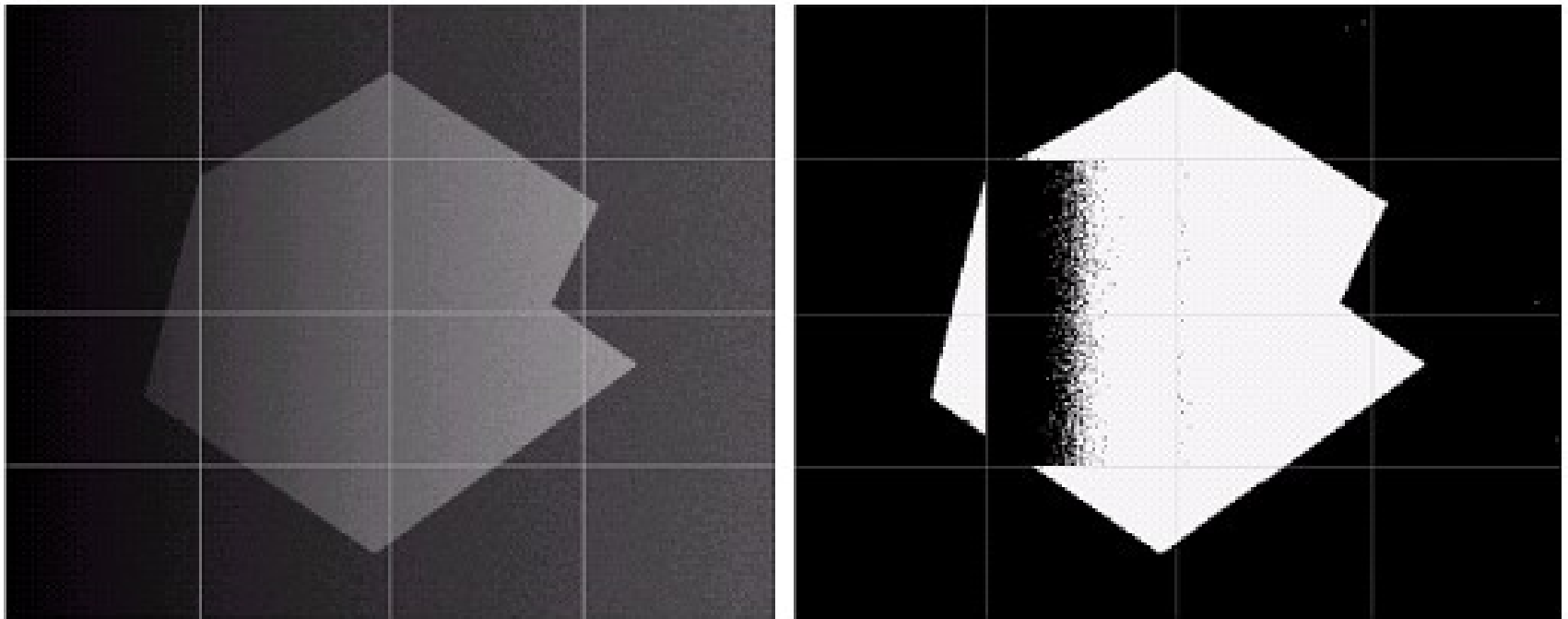


Exemple de seuillage adaptatif

- Nous avons besoin de séparer l'image en sous images, et de traiter chacune avec son propre seuil
- Le choix de la dimension des sous-images est important
- Avant de traiter chaque sous-image, nous vérifions la variance des tons de gris pour décider s'il existe un besoin de segmentation
 - *Exemple : pas besoin si $\text{variance} < 100$*

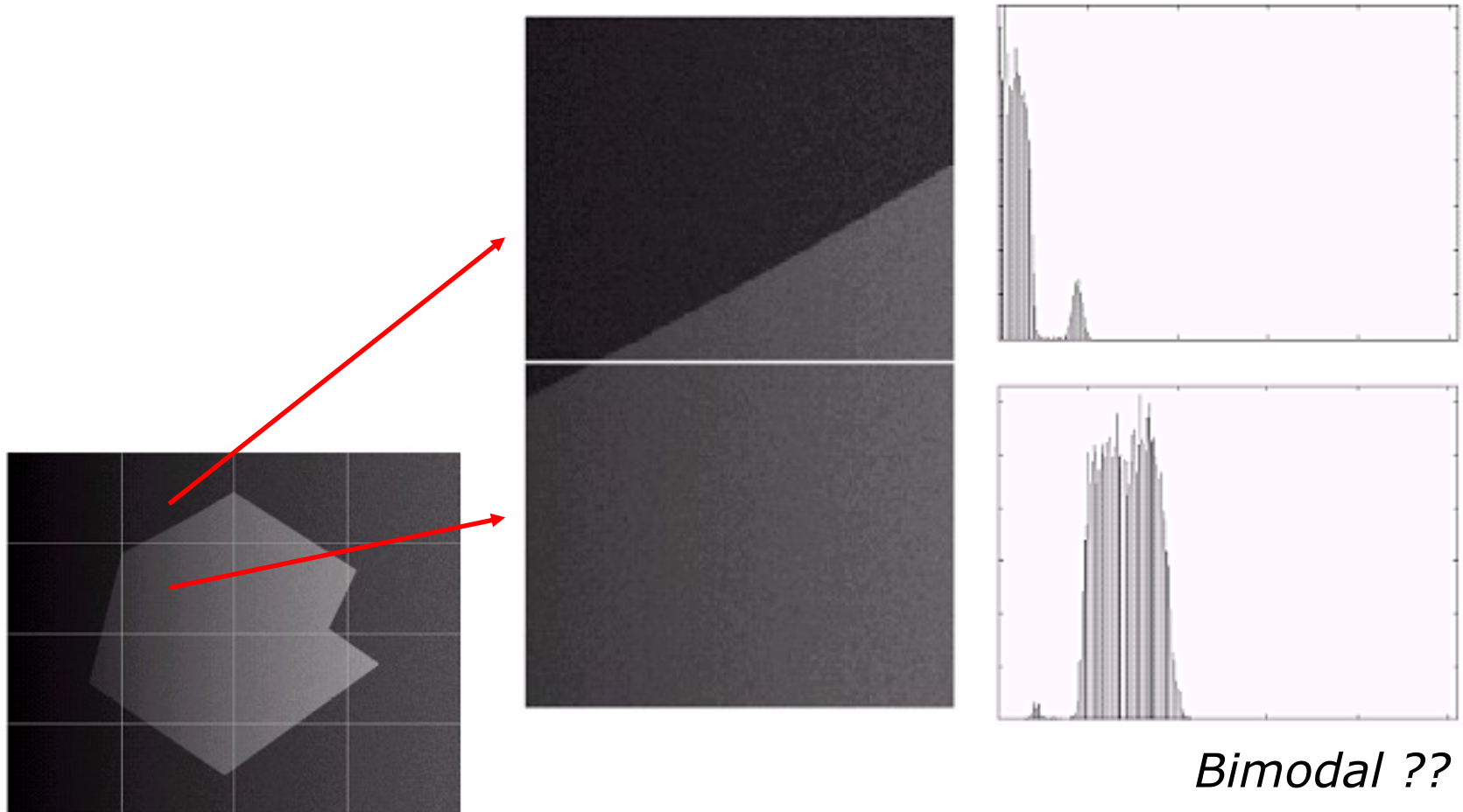
Exemple de seuillage adaptatif

- On divise l'image en sous-images
- On seuille chaque sous-image indépendamment
- Les 4 sous images de coins ne sont pas traitées car $variance < 100$



Exemple de seuillage adaptatif

Bimodal



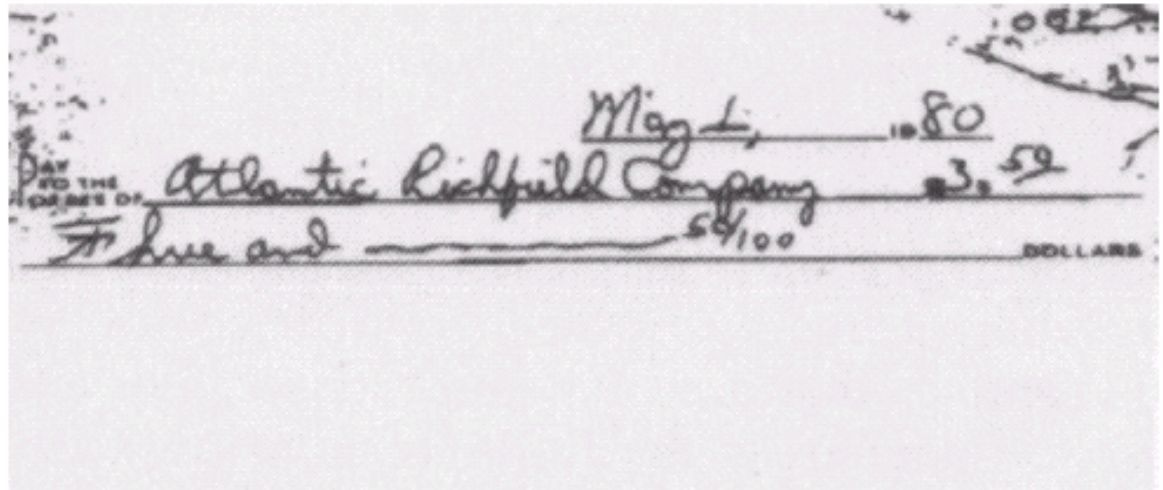
Exemple de seuillage local adaptatif

a

b

FIGURE 10.37

(a) Original image. (b) Image segmented by local thresholding. (Courtesy of IBM Corporation.)





Exemple de seuillage local adaptatif

- Exemples de d'implementation de seuillage adaptatif dans OpenCV (`cv.adaptiveThreshold`)
 - `cv.ADAPTIVE_THRESH_MEAN_C`
 - `cv.ADAPTIVE_THRESH_GAUSSIAN_C`

$$T(i, j) = I(i, j) * \text{Filtre} - C$$

Où :

- C est un constant
- Filtre : `filtre moyennneur` ou `filtre gaussien`.

La taille du filtre est un paramètre

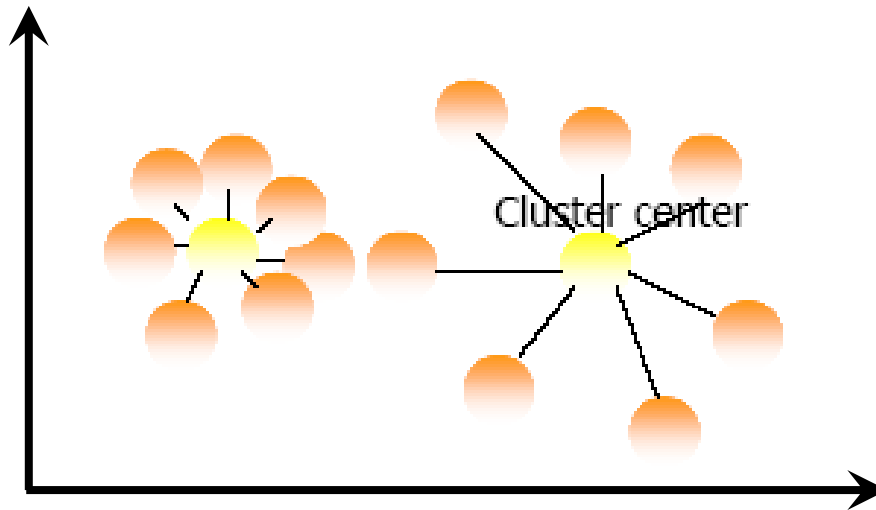


Algorithme des k-moyennes

(K-means)

k-moyennes (*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



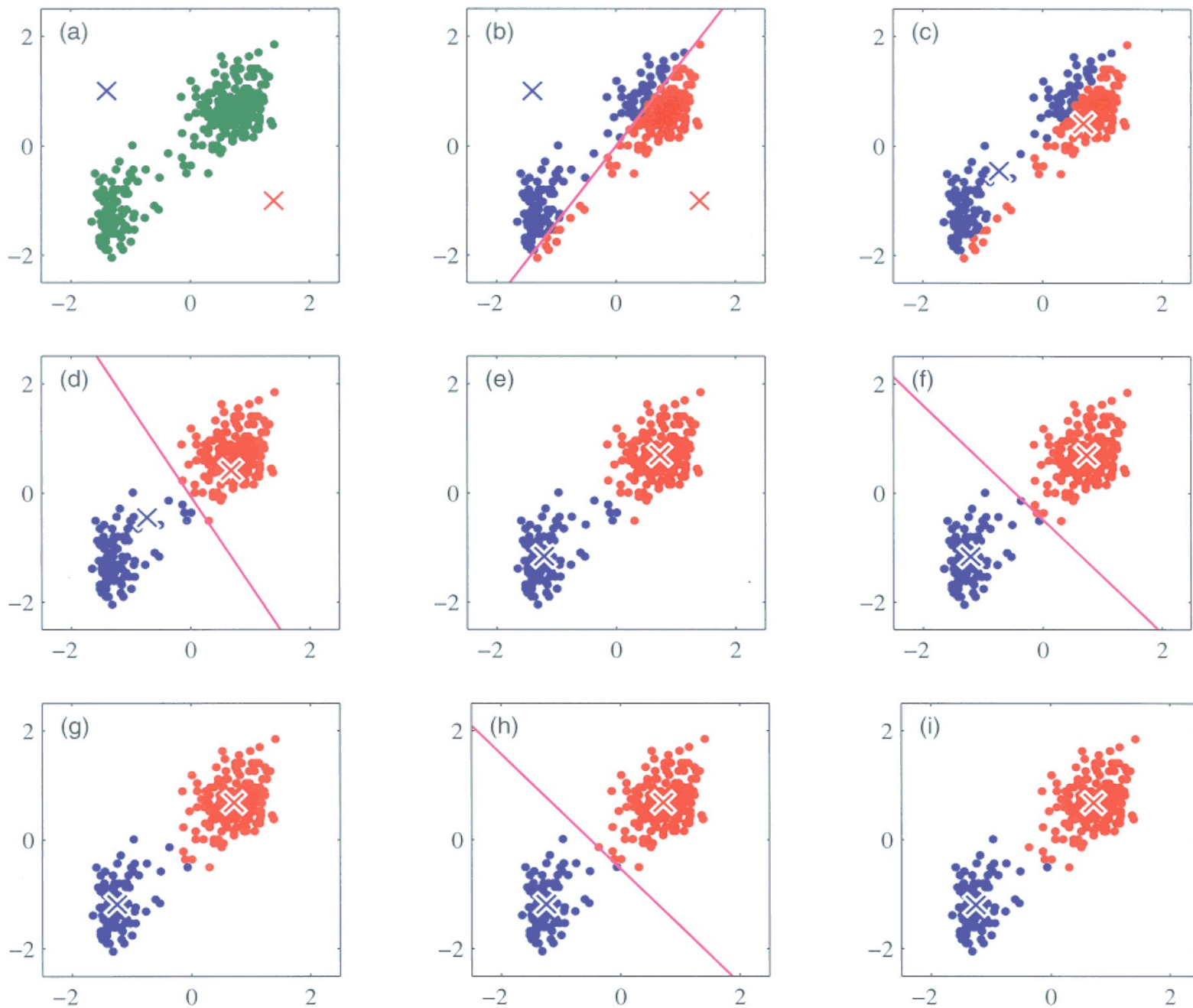
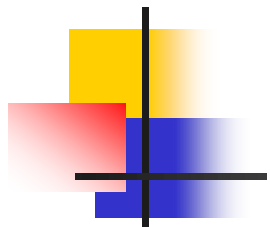


Algorithme des k-moyennes

L'algorithme est composé de 4 étapes :

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



Exemple en segmentation d'images



Image source



*K-moyennes
sur l'intensité*



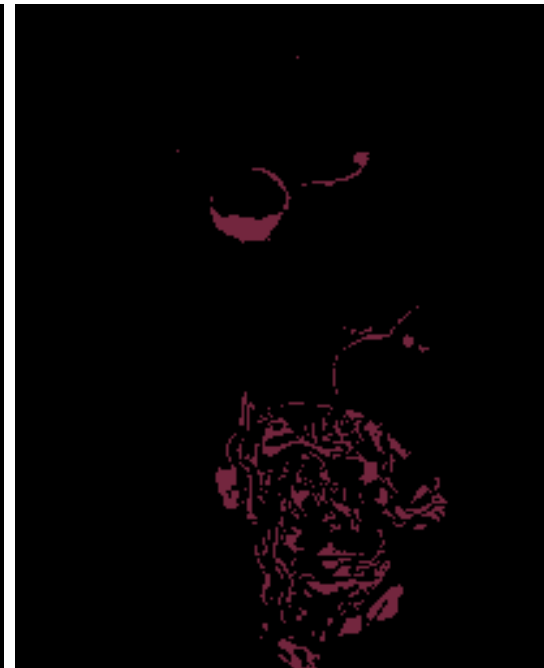
*K-moyennes
sur la couleur*

K-moyennes basé sur l'intensité (gris) ou sur la couleur



Suite...

*K-moyennes sur la
couleur en 11 groupes*





Segmentation en pixels / régions

- Le seuillage est une opération sur les pixels
 - *ne produit pas forcément des régions connexes !*
- On peut utiliser le seuillage pour les régions, mais il faut « nettoyer » le résultats obtenus
 - *éliminer les pixels seuls, conserver les régions*
- Il existe des méthodes de segmentation en régions
 - *conservent la connexité entre les régions*



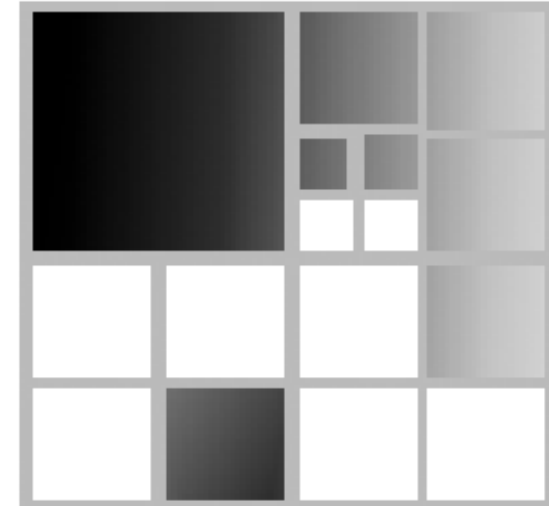
Segmentation en régions

Division-fusion
(*Split-and-Merge*)



Division-Fusion (*split-and-merge*)

- *Etape de division (split)*
 - 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
- *Etape de fusion (merge)*
 - Regrouper les blocs adjacents représentant des régions homogènes selon un prédicat défini

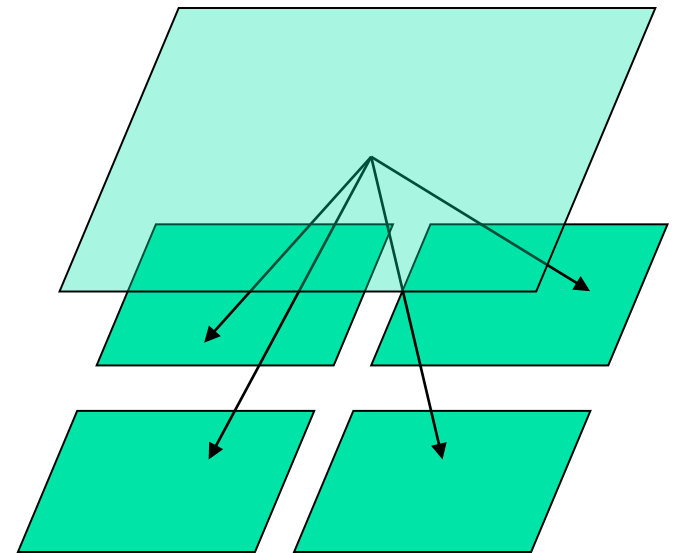


Etape de division

Phase 1 : Créer les zones homogènes = DIVISION (*split*)

Phase 2 : Les regrouper = FUSION (*merge*)

- L'image est stockée dans un arbre
 - Au début, arbre racine = image complète
- Récursivement, chaque feuille F est subdivisée en 4 si elle n'est pas assez homogène
 - les 4 sous-images sont ajoutées en tant que feuilles de F
- L'algorithme poursuit tant qu'il reste des feuilles non homogènes à diviser



Etape de division

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

Image initiale

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

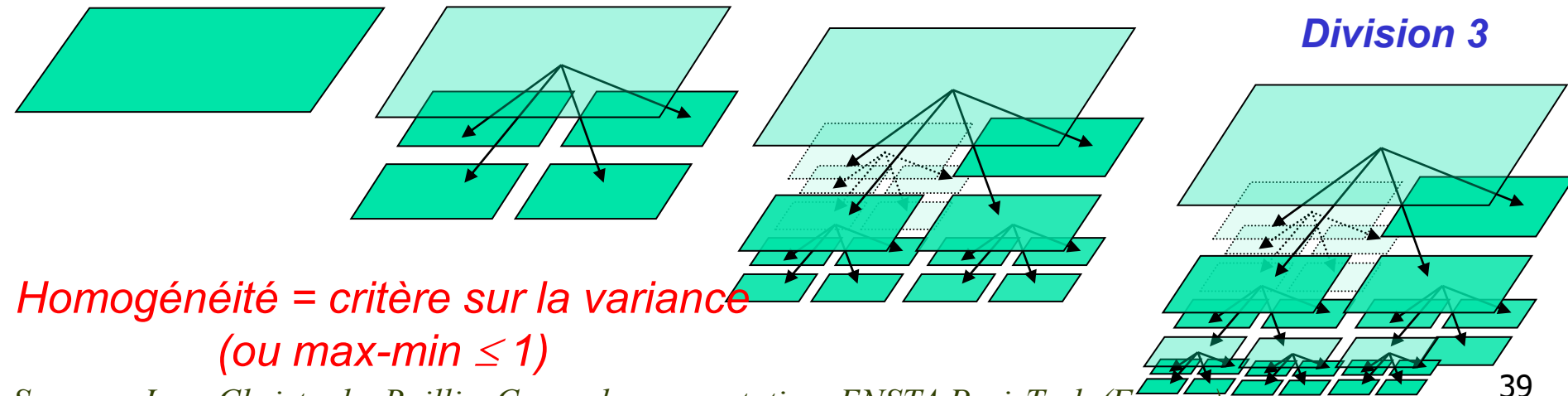
Division 1

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

Division 2

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

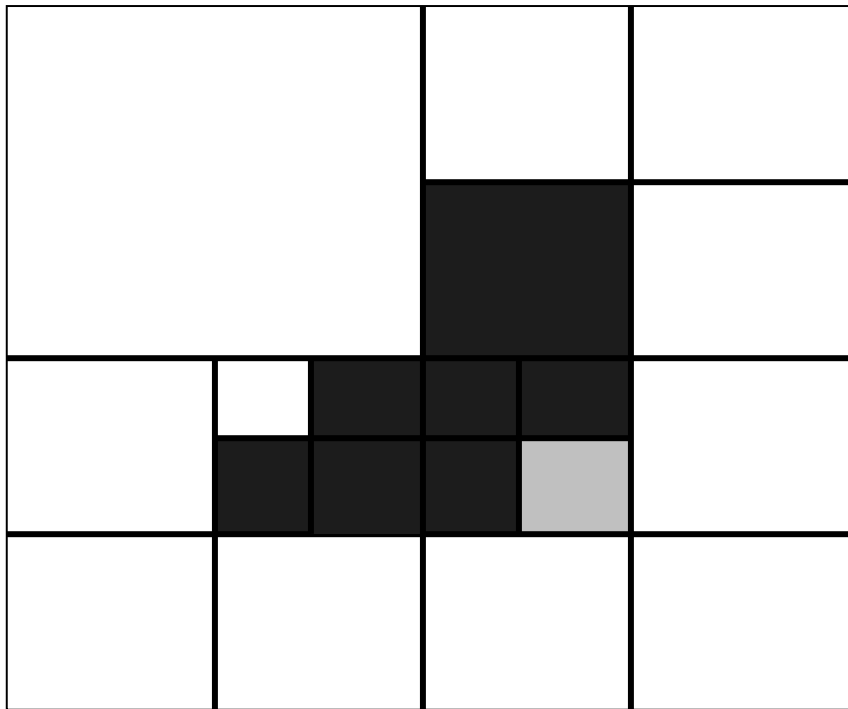
Division 3



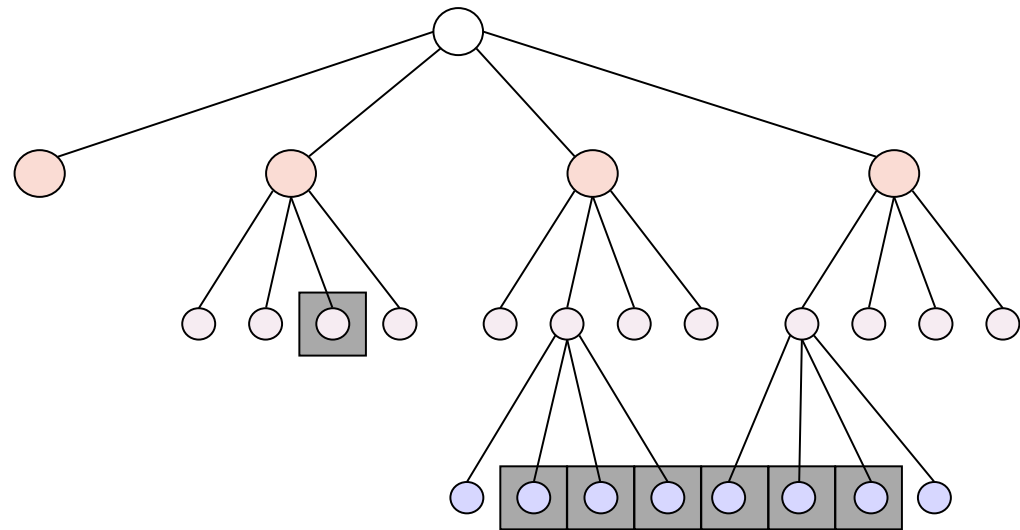
Etape de fusion

Phase 1 : Créer les zones homogènes = DIVISION (*split*)

Phase 2 : Les regrouper = FUSION (*merge*)

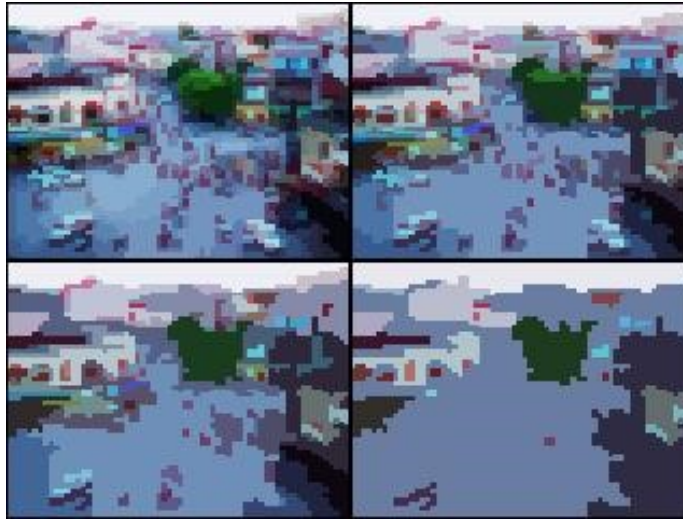


Quadtree



Connecter les régions adjacentes homogènes

Exemples de division-fusion



*Différents seuils
sont utilisés ici*





Segmentation en régions

Croissance de régions
(Region growing)



Croissance de régions

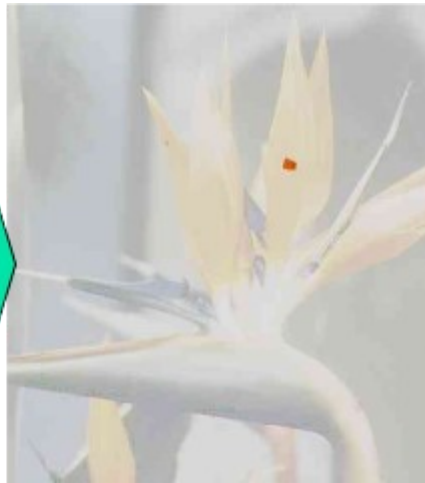
- On débute avec un pixel, et on « ajoute » les pixels voisins qui répondent à un critère d'appartenance :
 - *Variance faible*
 - *Niveau de gris répondant un seuil*
 - ...
- Les pixels initiaux sont appelés « germes », « amorces » ou « semences »
 - *Choix des pixels initiaux automatiques ou semi-automatiques*
- La région « grandit » à partir de son germe
 - *Besoin d'un critère (ou prédicat) pour choisir les pixels à ajouter*

Croissance de régions

- On part d'un germe (*seed*) et on l'étend en ajoutant les pixels voisins qui satisfont le critère d'homogénéité
- Le germe peut être **choisi soit par un humain, soit de manière automatique** en évitant les zones de fort contraste (*gradient important*)



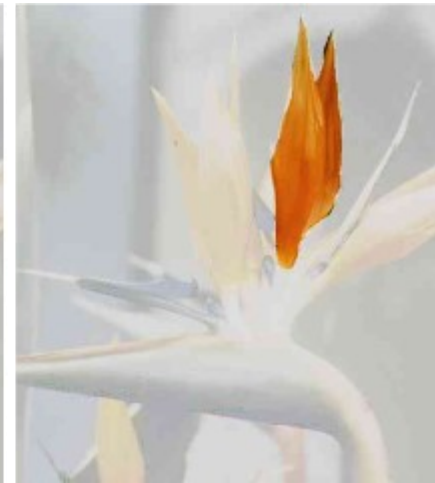
image



germe

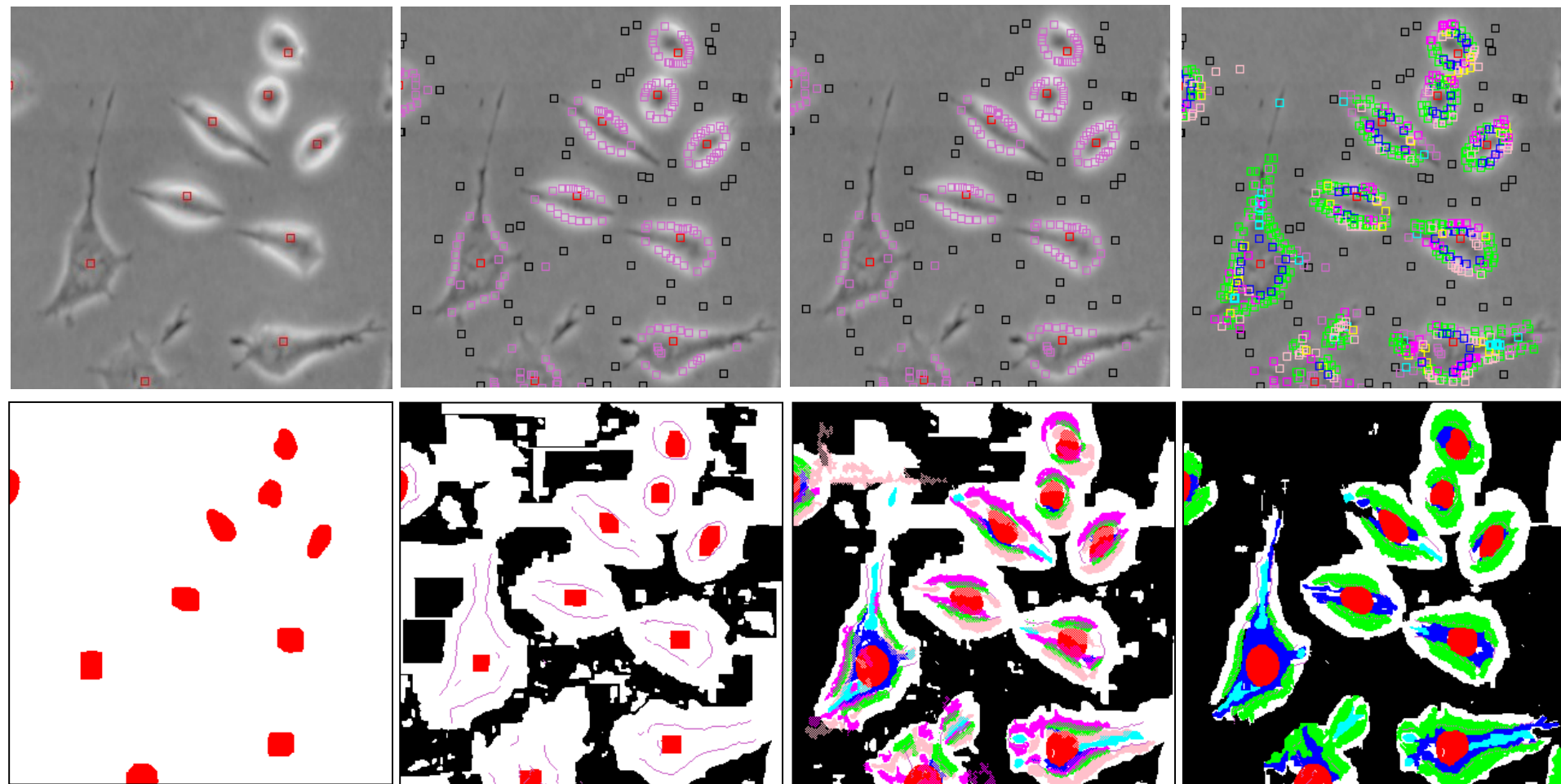


croissance



région finale

Croissance avec plusieurs germes





Segmentation en régions

Ligne de partage des eaux
(*Watershed*)

Segmentation par partage des eaux

On considère la visualisation de l'image en 3D, en utilisant le **ton de gris comme troisième dimension**

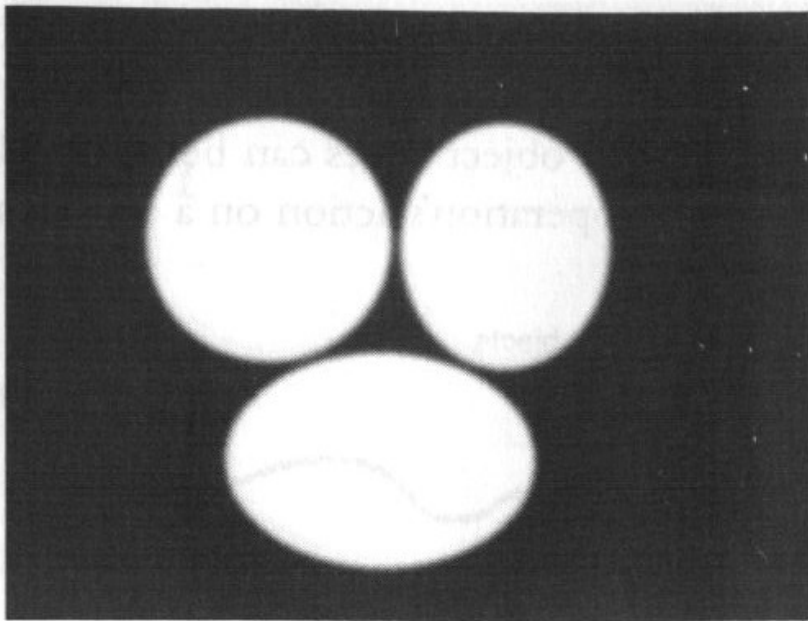
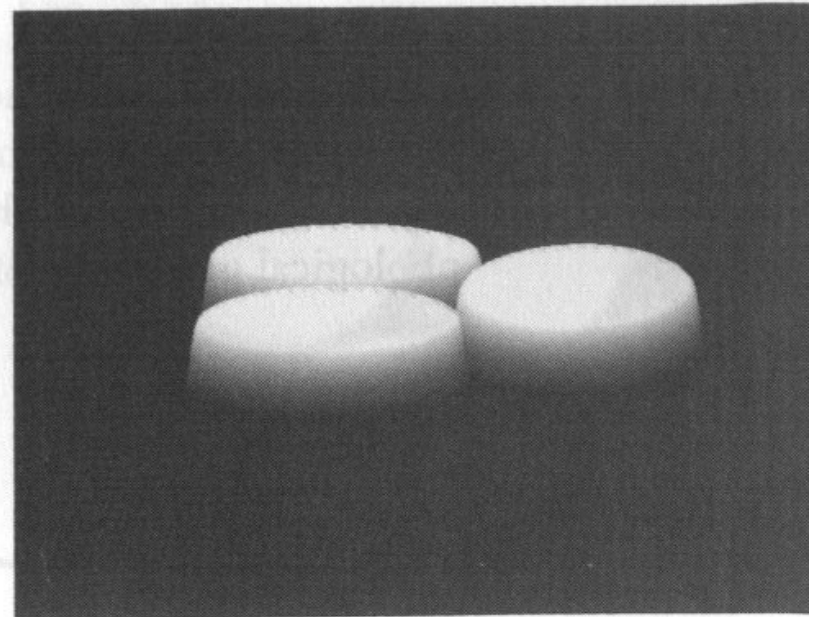


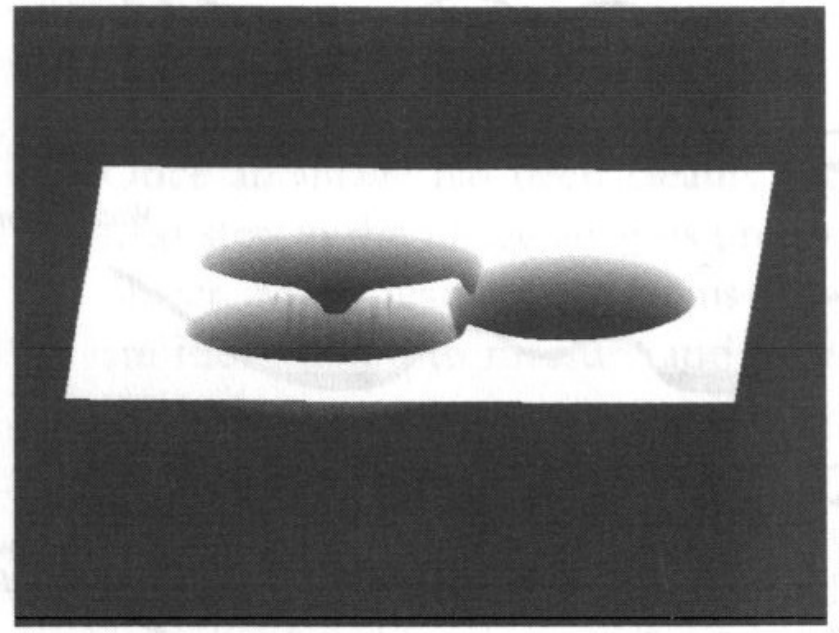
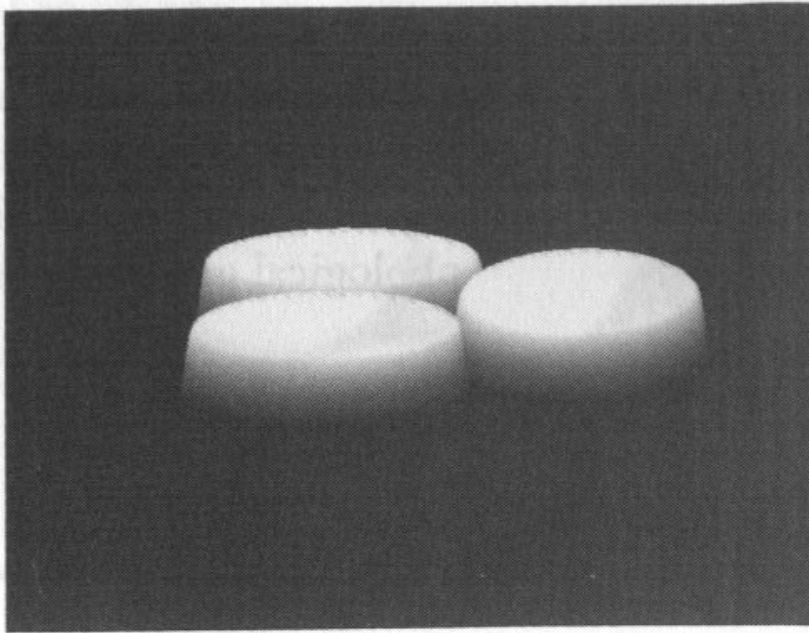
Image 2D



Visualisation en 3D

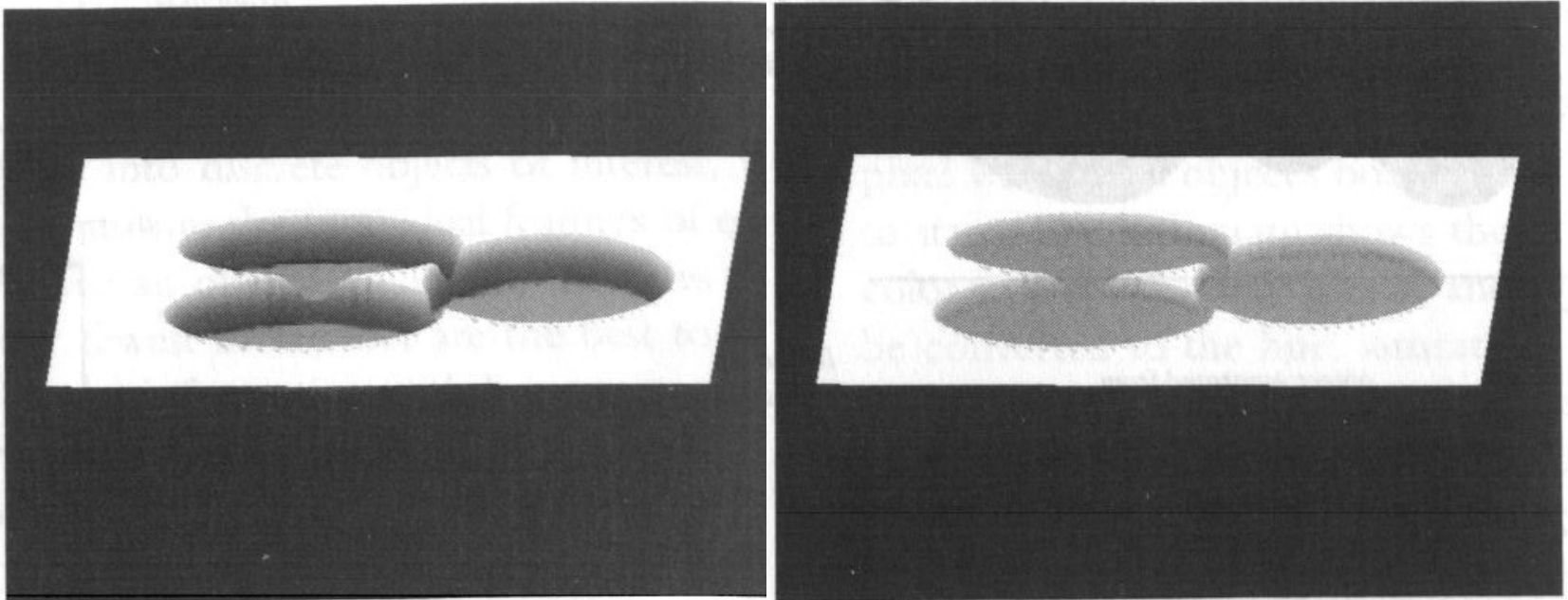
Segmentation par partage des eaux

Ensuite on "complémente" les valeurs pour créer des zones inondables

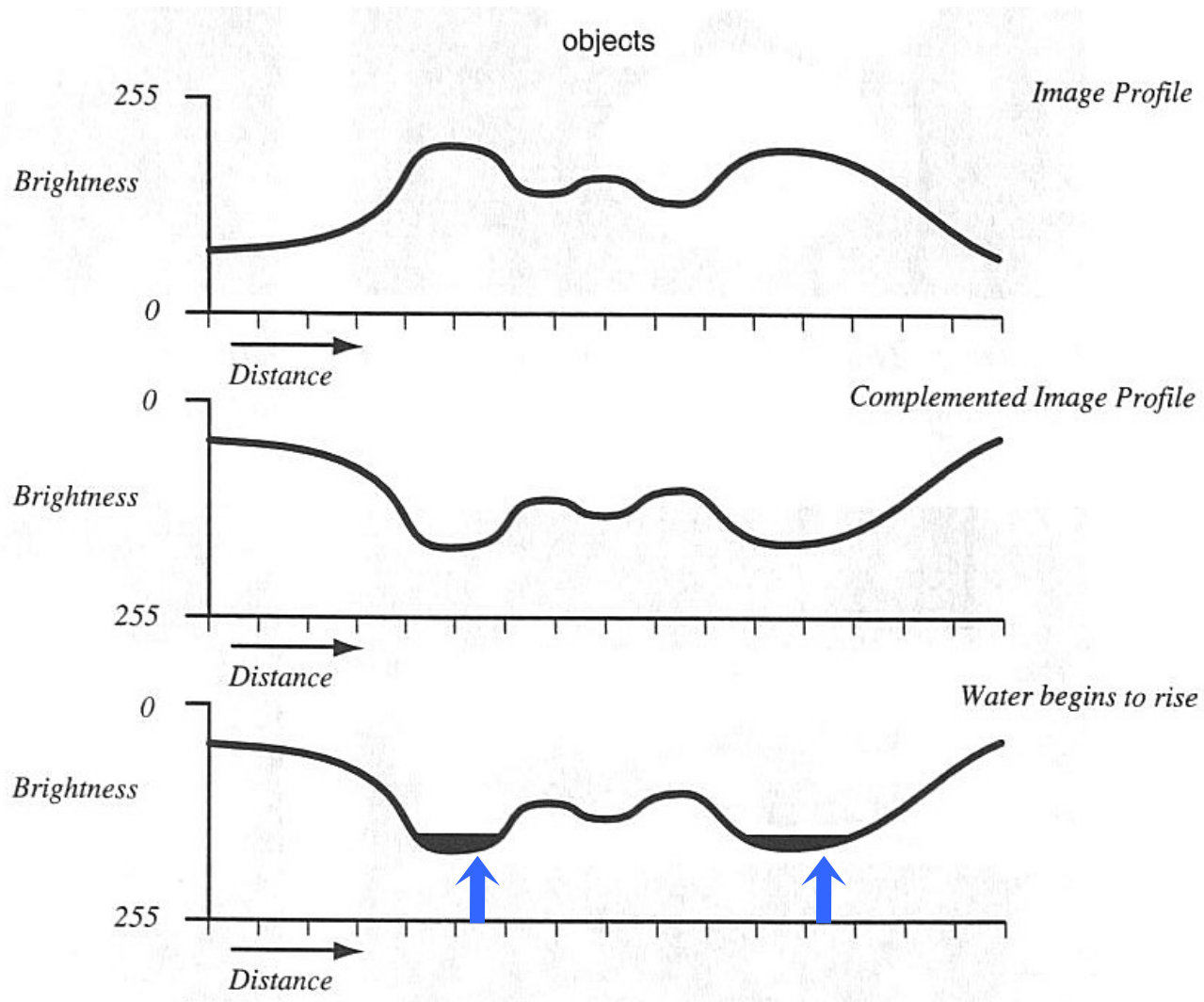


Segmentation par partage des eaux

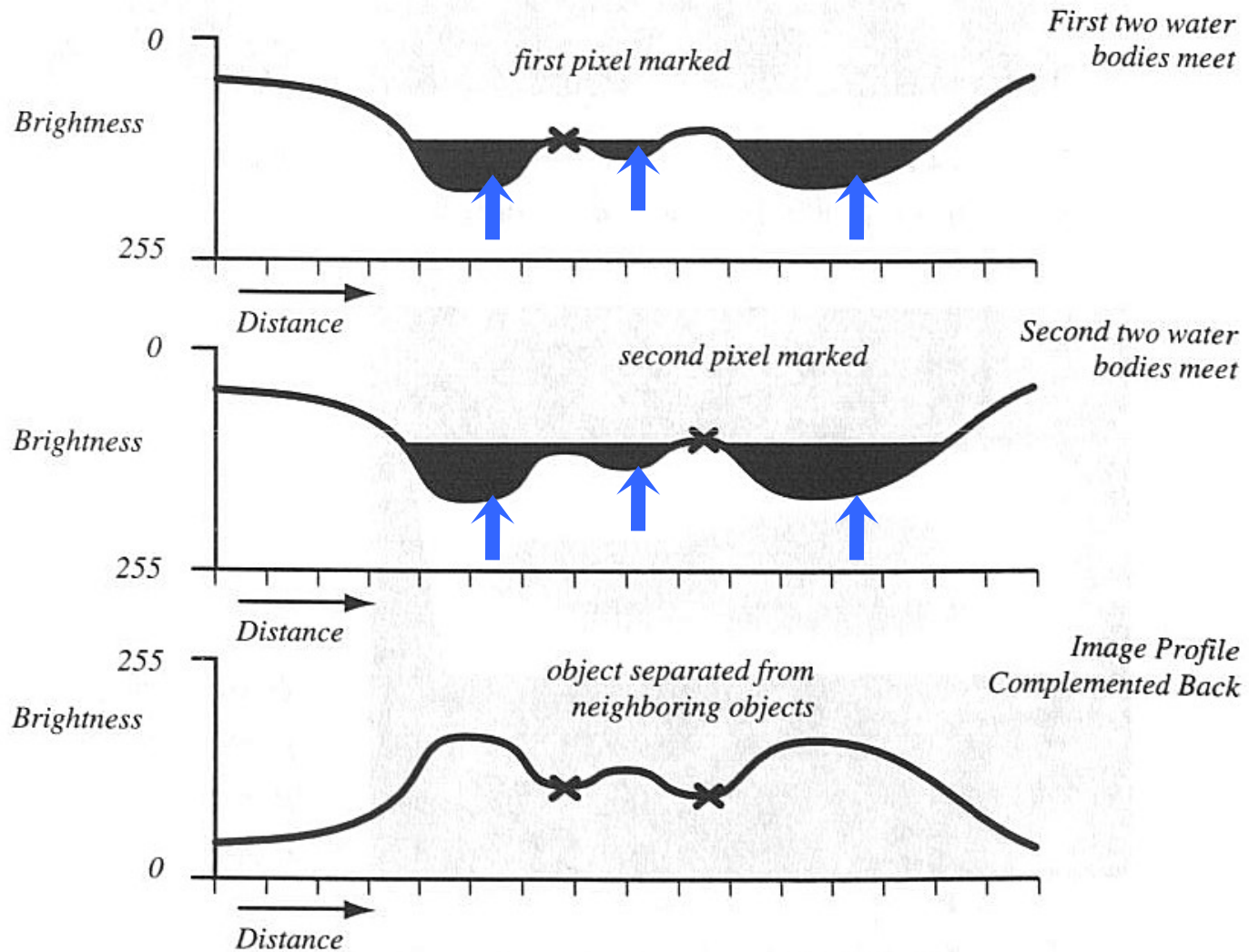
Enfin, on "infiltre" les cavités des zones inondables



Segmentation par partage des eaux



Segmentation par partage des eaux





Projection d'amplitude



Projection d'amplitude

- Pour localiser la region d'une ou plusieurs objets dans des cas particuliers
- Idée générale :
 - calculer la projection d'amplitude (moyenne ou somme des niveaux de gris) selon lignes et colonnes
 - Utiliser ces infos (seuillage par exemple) pour localiser la région d'intérêt

Example

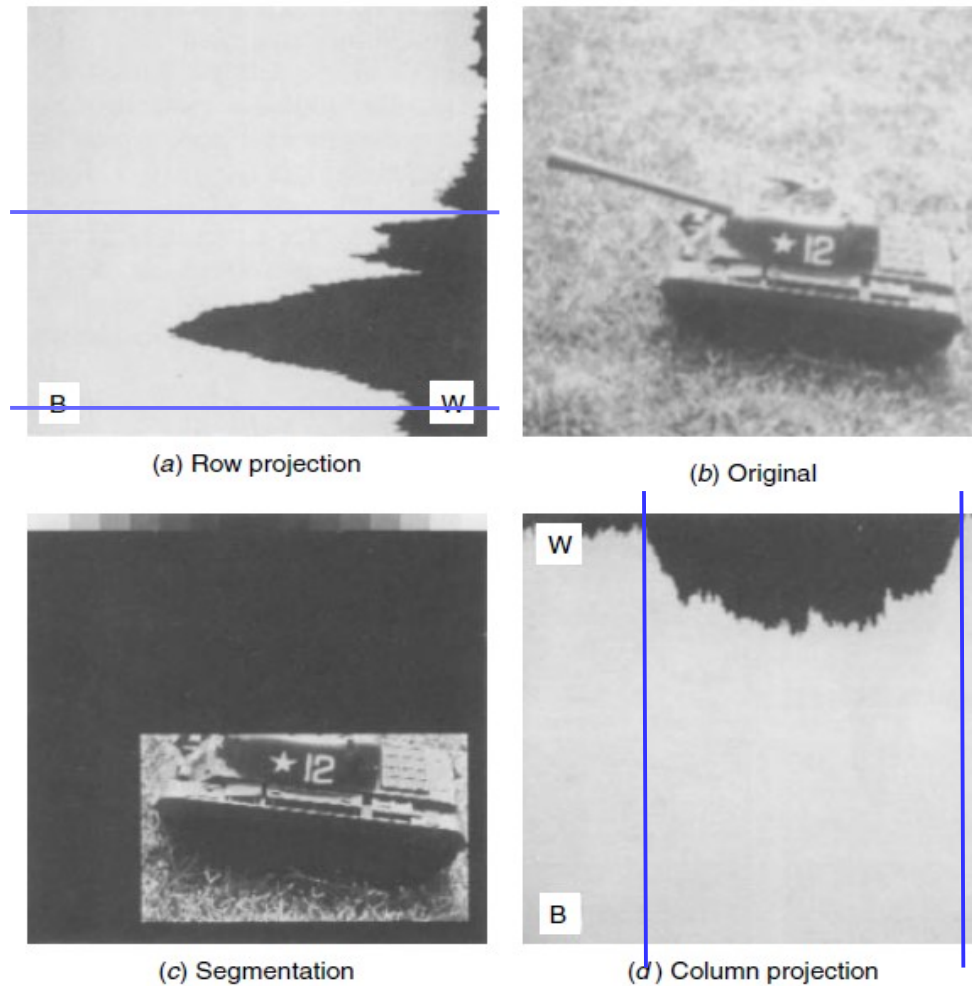
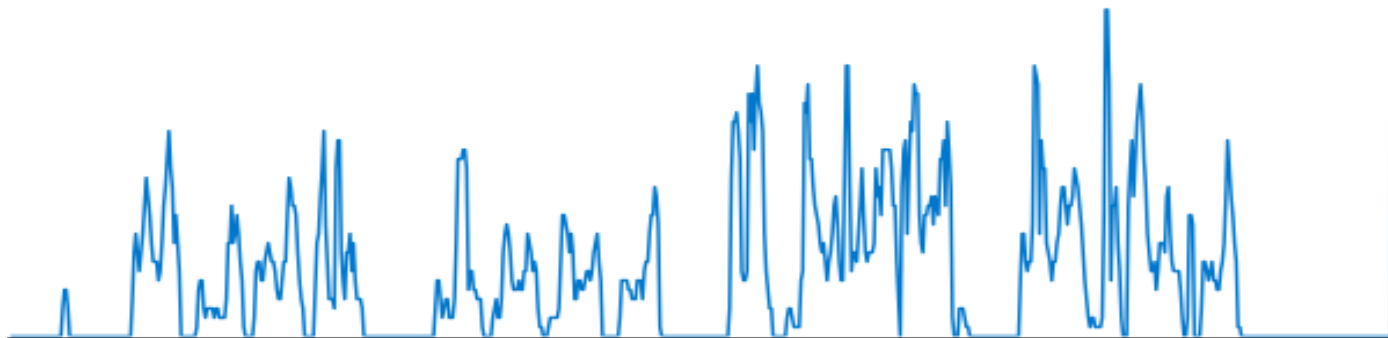


FIGURE 17.1-6. Gray scale projection image segmentation of a toy tank image.

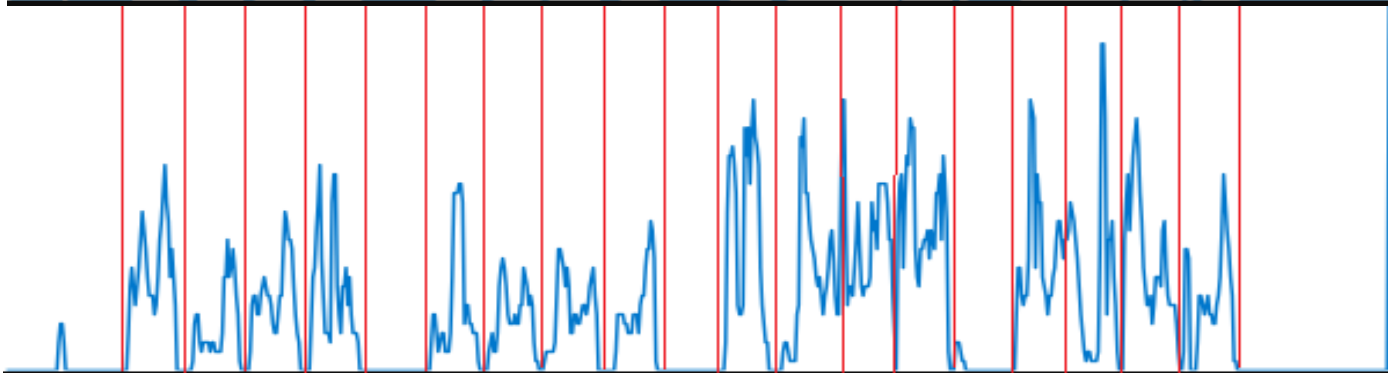
Exemple



Image
originale
Contour



Projection



Frontiere des
lettres



Regions finales



Segmentation basée sur le mouvement

Différence d'images



—



=



- **Détection du mouvement** par différence d'images
- On voit les zones où un mouvement à eu lieu
 - mais on ne possède aucun vecteur de ce mouvement
- **La différence d'images produit deux traces :**
 - Endroit où l'objet était situé (maintenant fond)
 - Endroit où l'objet est maintenant situé (auparavant fond)
 - *Le signe de la différence peut différencier les deux zones*

Différence d'images seuillée



—

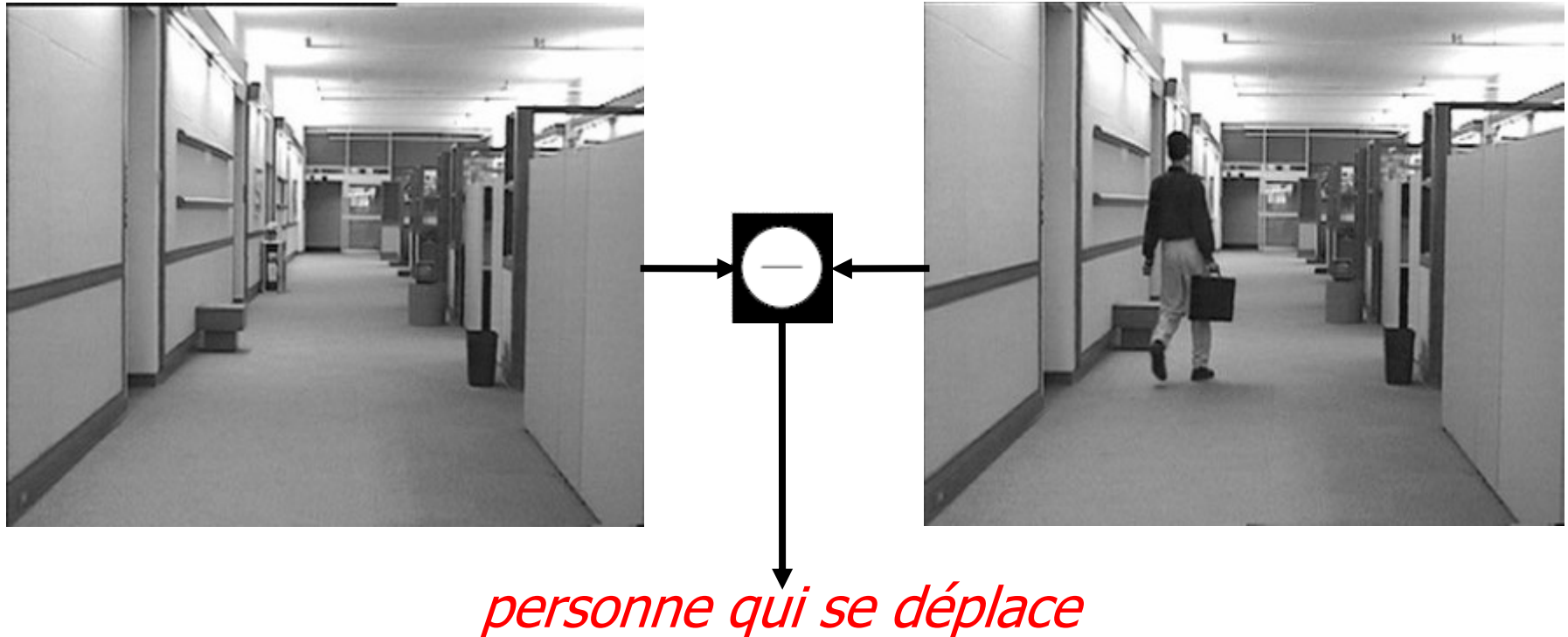


> 50

*Avec un seuillage de la différence,
on obtient plus clairement les
zones de mouvement*

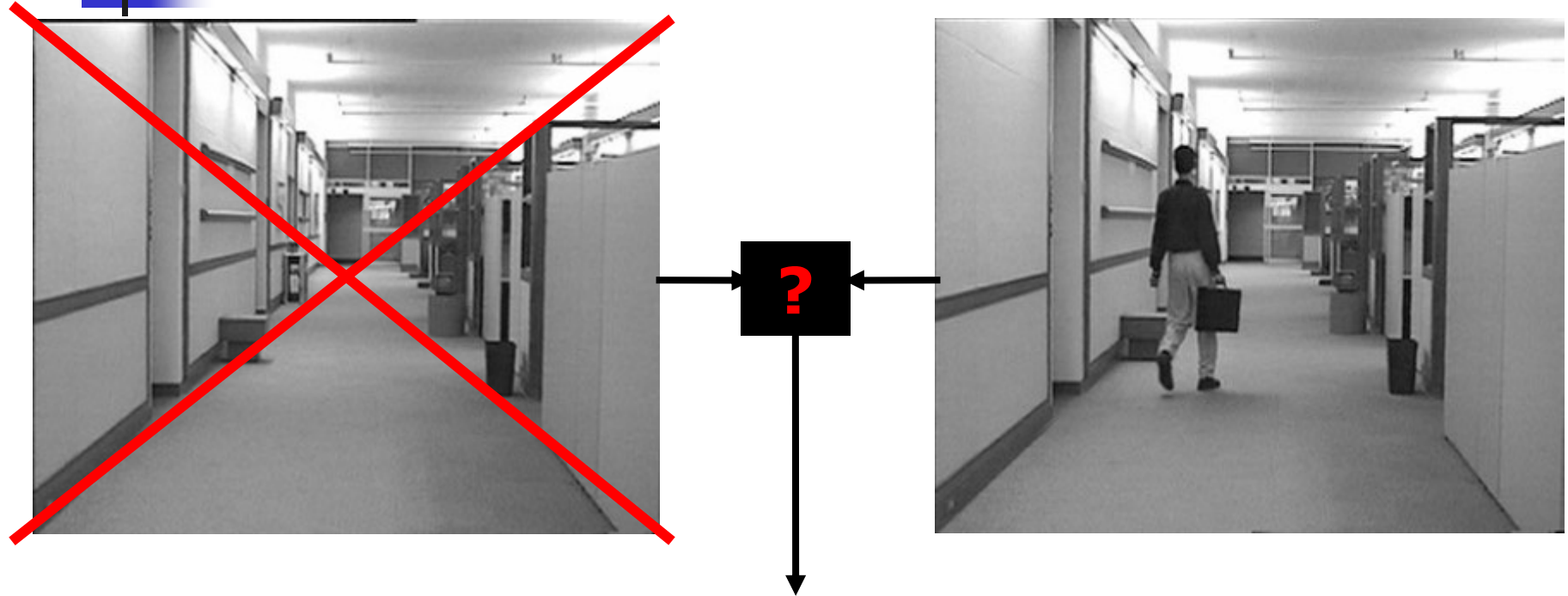


Détection de mouvement



Même si la différence d'images n'est pas la méthode la plus complète pour l'analyse du mouvement, elle est très rapide pour la détection simple du mouvement

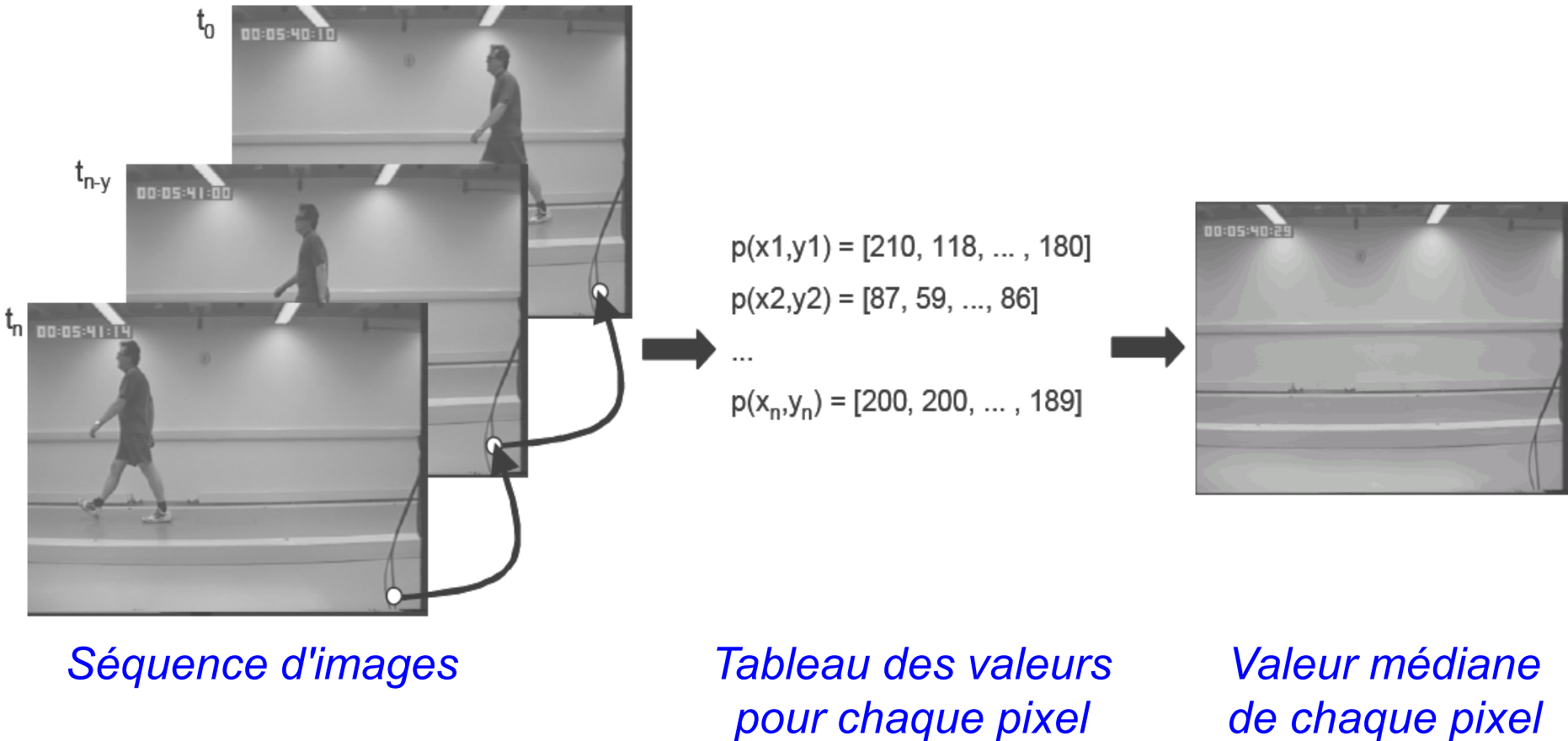
Détection de mouvement



personne qui se déplace ?

Comment faire si on n'a pas l'image d'arrière-plan pour faire la détection ?

Construction de l'arrière-plan



Détection utilisant l'image du fond



Image

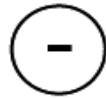


Image du fond



Détection

Après la détection, il reste encore du bruit, qu'il faut nettoyer pour obtenir l'objet qui nous intéresse



Approche Gestalt



Approche Gestalt

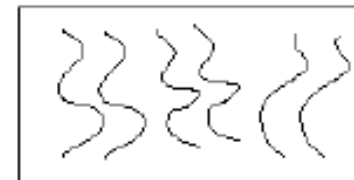
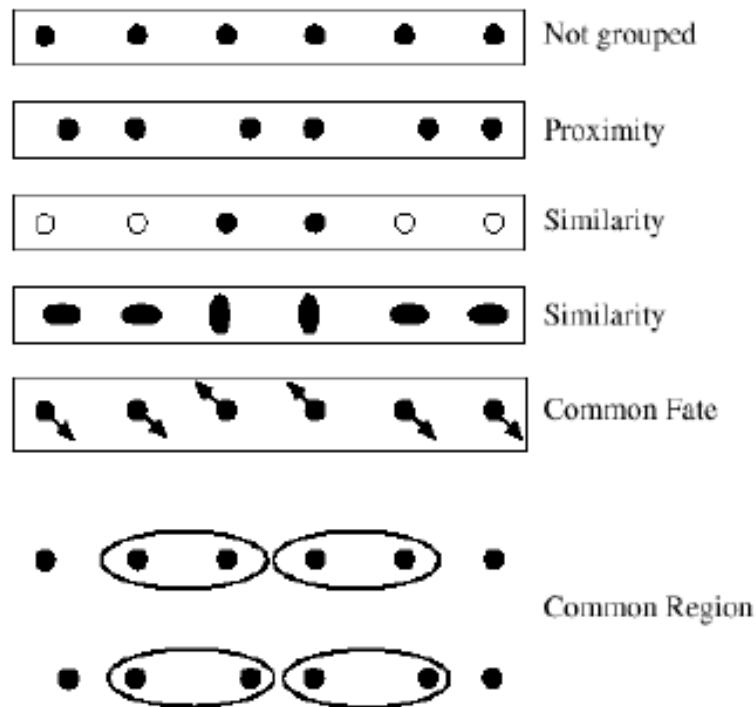
- Une autre approche de la segmentation/groupement s'inspire de la psychologie et de l'observation de l'humain et de son environnement
- Cette approche est basée sur les travaux de Gestalt sur la vision humaine
- Importance du contexte dans cette approche.



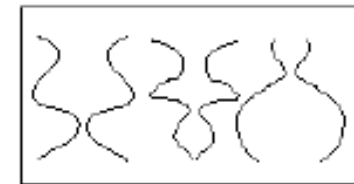
Approche Gestalt

- **Proximité** : objets proches sont regroupés
- **Similarité** : objets semblables sont regroupés
- **Tendance commune** : objets avec un mouvement cohérent semblable sont regroupés
- **Région commune** : objets à l'intérieur d'une même région sont regroupés
- **Parallélisme** : courbes ou objets parallèles sont regroupés
- **Fermeture** : courbes ou objets qui peuvent former des objets fermés sont regroupés
- **Symétrie** : courbes ou objets qui peuvent former des objets symétriques sont regroupés
- **Continuité** : courbes ou objets formant une *continuité* (au sens esthétique) sont regroupés
- **Configuration familière** : courbes ou objets formant des objets connus sont regroupés.

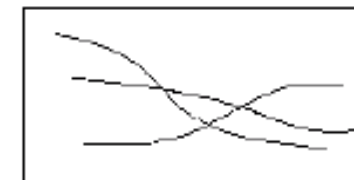
Approche Gestalt



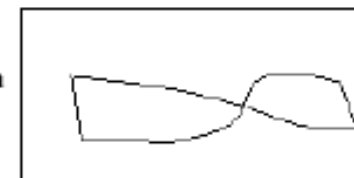
Parallelism



Symmetry



Continuity



Closure

Figure 16.4. Examples of Gestalt factors that lead to grouping (which are described in greater detail in the text). *figure from Gordon, Theories of Visual Perception, page 67 in the fervent hope that permission will be granted*



Approche Gestalt

- Questions sur l'approche Gestalt :
 - Comment intégrer des règles de regroupement dans un algorithme?
 - Comment générer des hypothèses?
- Malgré quelques essais, cette approche reste surtout théorique



Segmentation en régions

Conclusion



Critères, prédicats et paramètres

- Dans toutes les méthodes, il y a des paramètres !
 - *Rien d'universel, il faut tester, au cas par cas*
- **Seuil**
 - Soit on donne une **valeur constante**
 - Soit on donne une **valeur relative** (%, moyenne, ou ...)
 - Soit on donne un **algorithme** pour trouver le seuil
- **Critère/prédicat**
 - **Fonction** : constante, relative ou autre
 - **Dynamique** : évolue au fur et à mesure
 - **Par rapport à la région** :
 - Niveau de gris, couleurs, gradient, ...
 - Moyenne, variance, max/min



Segmentation – conseils

- La segmentation d'une image cause encore aujourd'hui beaucoup de problèmes
 - *Aucune méthode ne fonctionne pour toutes les images*
 - *Pas de garantie, pas de recette miracle !*
- Le **pré-traitement** des images, la **sélection de capteurs** et **sources d'énergie** appropriées, et la **prise contrôlée des images** rendent cette étape plus facile et plus efficace
- La segmentation aide beaucoup pour la reconnaissance, mais elle n'est pas obligatoire dans tous les cas
 - *Voir cours de Vision par Ordinateur*



Segmentation – conseils

- Evaluer le résultat d'une segmentation n'est pas facile
 - *Il dépend de l'application*
 - *Il dépend de ce qu'on veut*
 - *Il est subjectif et varie d'une personne à l'autre*
- Un des principaux problème est de définir le but de la segmentation
 - ***Qu'est-ce qu'on recherche exactement dans l'image ?***
 - *Éléments globaux de l'image ou détails fin de la composition ?*
 - *Présence d'un humain ou détails du visage ?*
- Il faut toujours se poser la question de ce que l'on veut faire avec la segmentation
 - *Cela permet de définir le degré de précision nécessaire*

Comment voit-on image ?

		x =														
		58	59	60	61	62	63	64	65	66	67					
y =	41	210	209	204	202	197	247	143	71	64	80					
	42	206	196	203	197	195	210	207	56	63	58					
	43	201	207	192	201	198	213	156	69	65	57					
	44	216	206	211	193	202	207	208	57	69	60					
	45	221	206	211	194	196	197	220	56	63	60					
	46	209	214	224	199	194	193	204	173	64	60					
	47	204	212	213	208	191	190	191	214	60	62					
	48	214	215	215	207	208	180	172	188	69	72					
	49	209	205	214	205	204	196	187	196	86	62					
	50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
	51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
	52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
	53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
	54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
	55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63



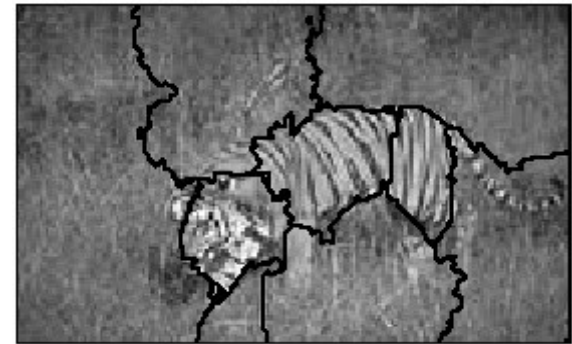
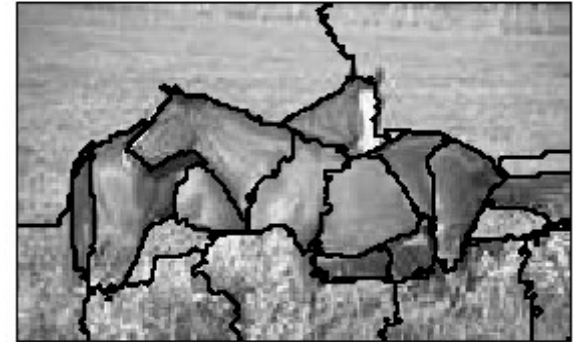
Limites de la segmentation...

La segmentation ne peut pas trouver tous les objets de l'image tel que nous les interprétons !



Limites de la segmentation...

Un objet n'appartient pas à une région !



Source : [Malik 2001].



Références

■ Livres:

- Introduction au Traitement d'Images (Lingrand), section 8.5
- Digital Image Processing 2ed (Gonzalez & Woods), chapitre 10
- Computer Vision: A Modern Approach (Forsyth & Ponce), chapitre 14

■ Jean-Christophe Baillie. Cours de segmentation. ENSTA ParisTech (France)

- <http://uei.ensta.fr/baillie/eng/index.html>
- <http://uei.ensta.fr/baillie/assets/ES322%20-%20Segmentation.ppt>

■ Thomas Boudier. Cours Imagerie Numérique en Biologie II : Algorithmes et développement. Univ. Paris 6 (France).

- <http://www.snv.jussieu.fr/~wboudier/ens/PlanningINB2.html>
- Chap 6 Segmentation :
http://www.snv.jussieu.fr/~wboudier/ens/cours_inb2/06_Segmentation.pdf



Références

- Machine Learning course (COMP4702/COMP770). The University of Queensland (Australie).
 - <http://www.itee.uq.edu.au/~comp4702/material.html>
- Christelle Scharff. Cours de Fouille de données. Pace University (NY, USA).
 - <http://www.csis.pace.edu/~scharff/DMIFI/>
 - Chap 9 : Regroupement : <http://www.csis.pace.edu/~scharff/DMIFI/cluster9.ppt>
- Watershed et ses alternatives:
 - <http://cmm.ensmp.fr/~beucher/wtshed.html>