

Vision par ordinateur

Segmentation

2^{ème} année GI et 3^{ème} année AE3S – ENSMR

Master *Intelligent Processing Systems* –FSR

Cours de Prof. Ibtissam BENMILOUD

Présenté par Dr. Nabila ZRIRA

Année universitaire 2019/2020

Introduction

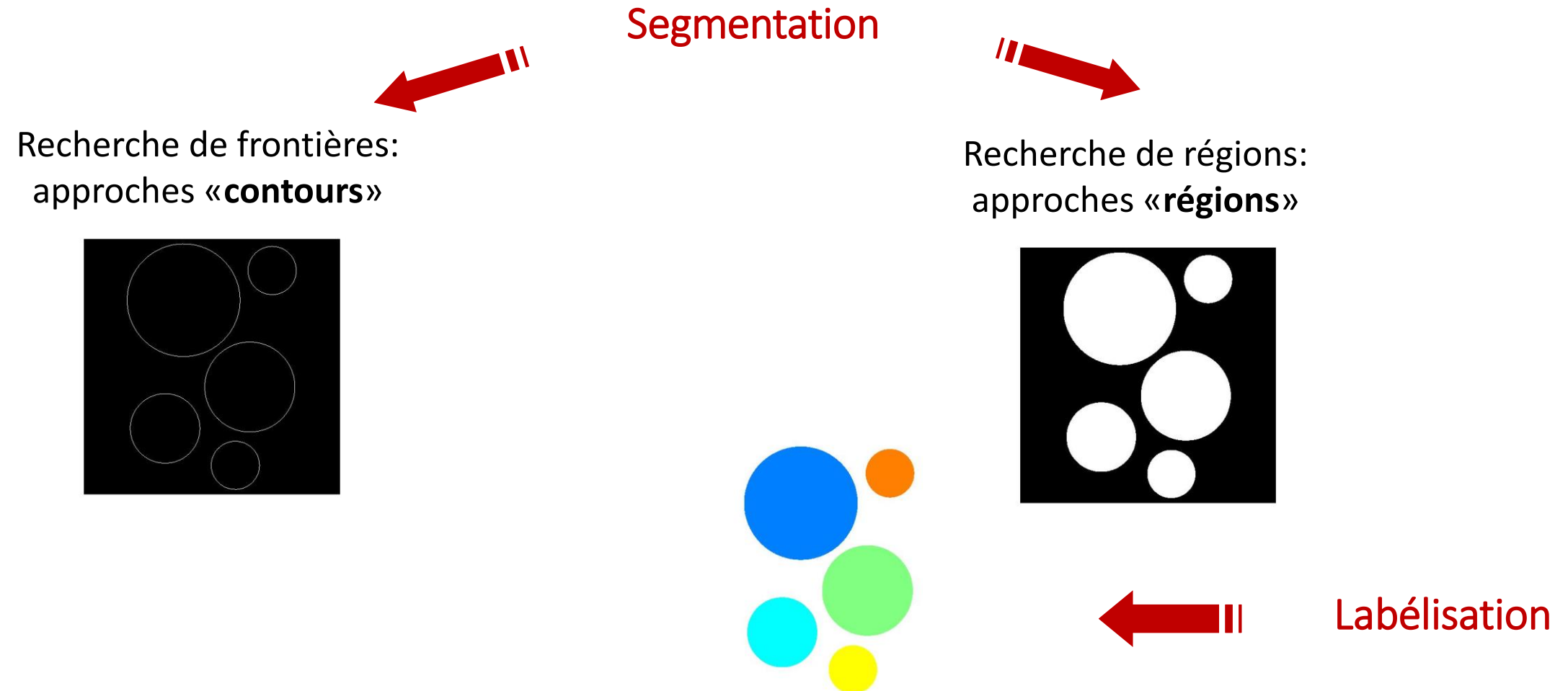
Partitionnement de l'image en régions disjointes

Objectif : identifier les objets d'intérêt dans une image (objet = région à segmenter)

Il existe 2 approches:

- Approche «**contours**»: les régions sont délimitées par les contours des objets qu'elles représentent (séparation)
- Approche «**régions**» : les régions sont déterminées en fonction d'un critère comme: le niveau de gris, la texture ou la couleur

Représentation de la région



Objectifs

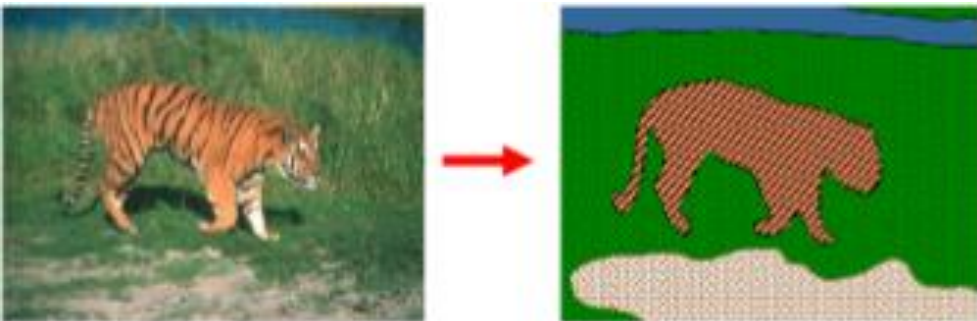
La segmentation est liée à la reconnaissance d'objets présents dans l'image.

Objet = région à segmenter

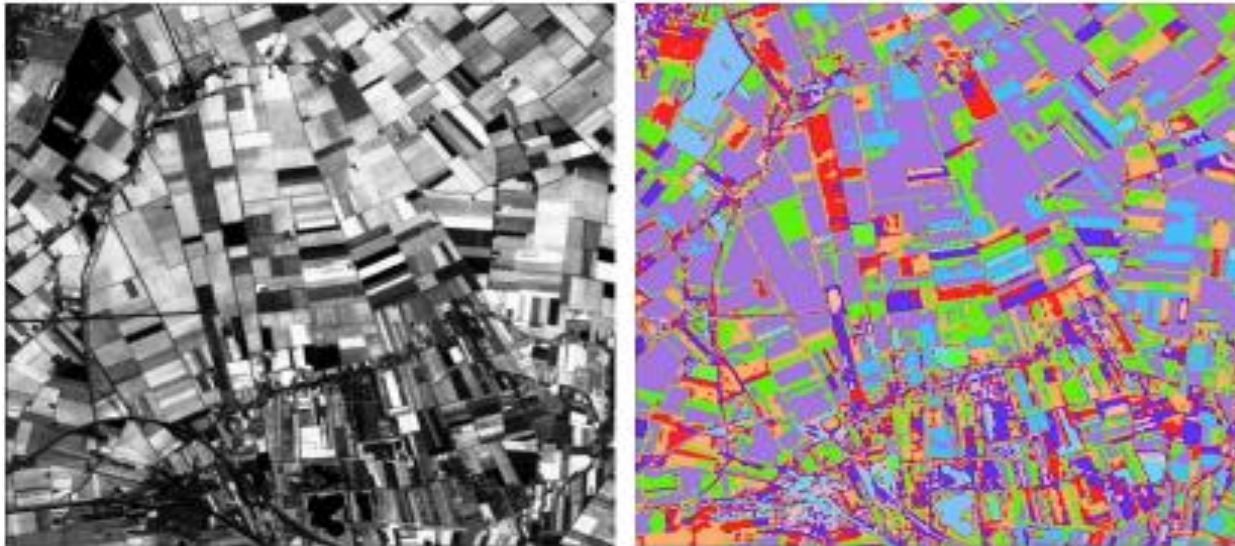
Quels objets voit-on dans l'image ?

Définition d'objets

- Connexe, couleur cohérente, délimité par des contours nets.
- Texture cohérente



Applications



Télédétection: classification d'une région agricole

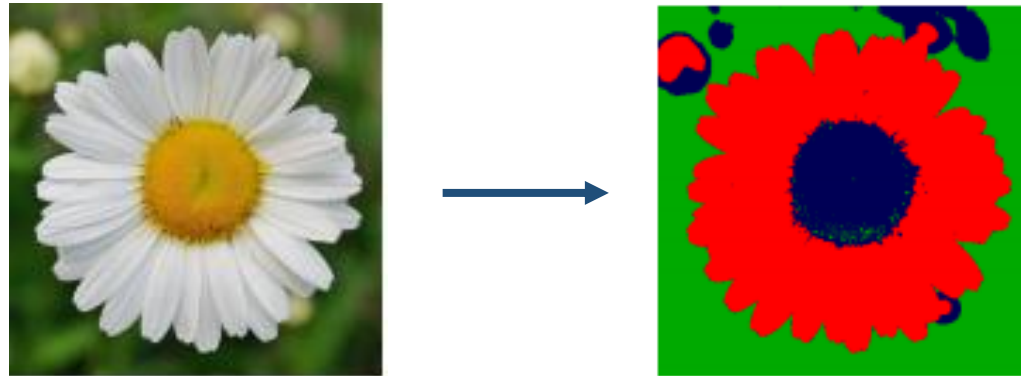
Source INRIA - Projet Ariana



Vidéo: incrustation, effets spéciaux

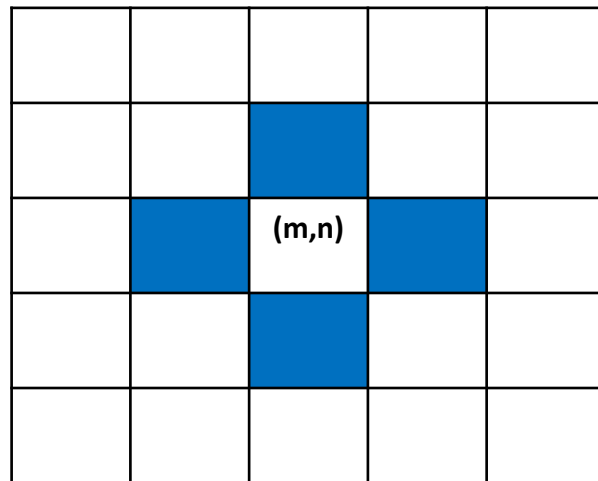
Définition

- La segmentation consiste à partitionner une image suivant un critère d'homogénéité H

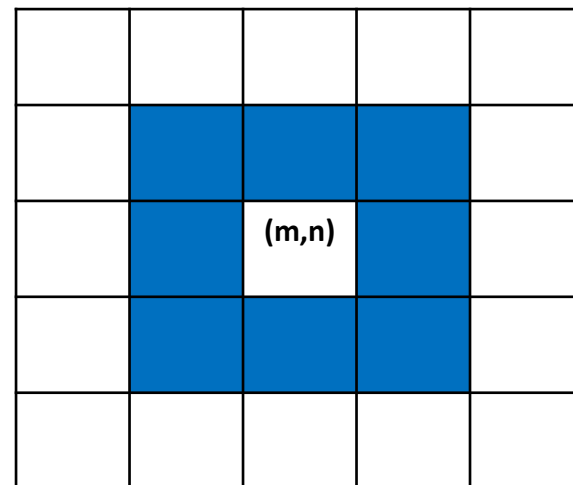


- La partition est un ensemble de régions R_i disjointes qui recouvrent l'intégralité de l'image
- Le critère d'homogénéité H :
 - est vérifié par chaque région R_i
 - n'est pas vérifié pour l'union de deux régions adjacentes

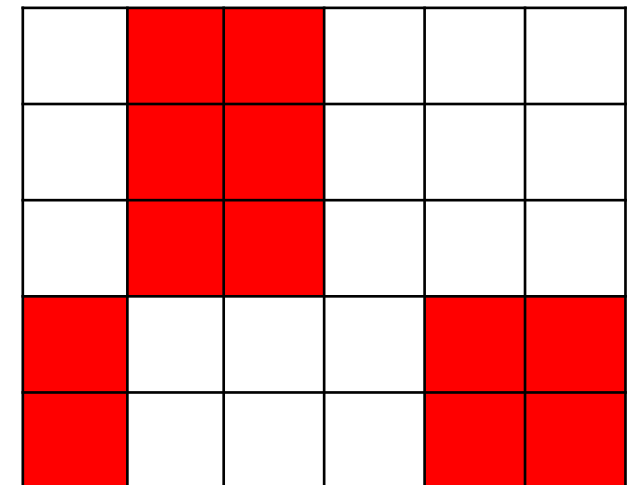
Relations entre les pixels



4-voisinages



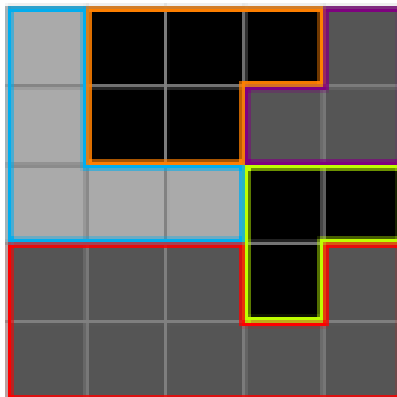
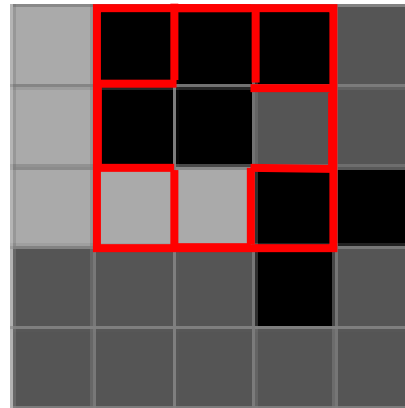
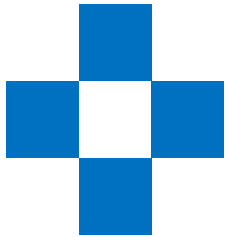
8-voisinages



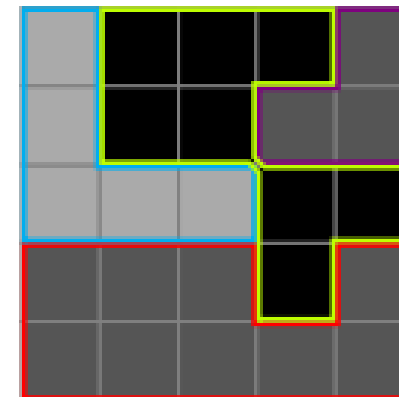
Composante connexe

Relations entre les pixels

Combien de composantes connexes ?



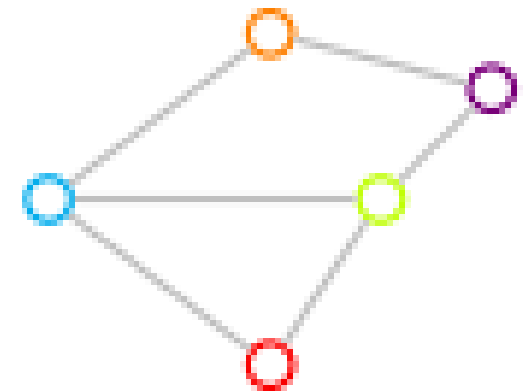
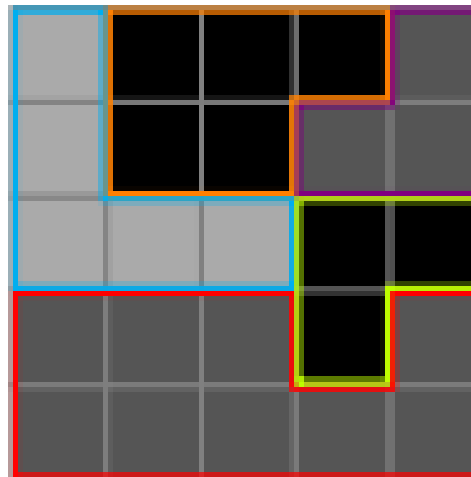
5 composantes connexes en 4-voisinage



4 composantes connexes en 8-voisinage

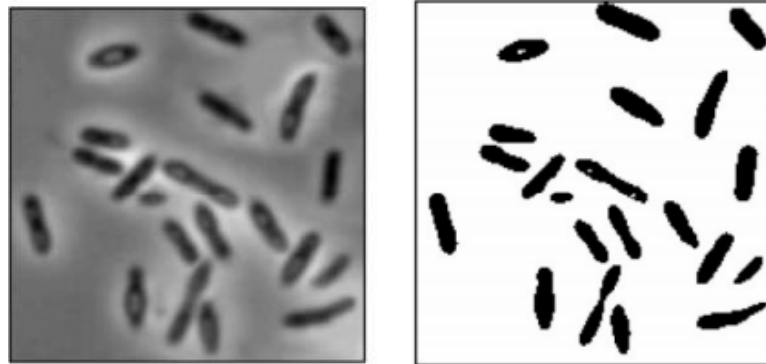
Remarques

- Chaque région est une composante connexe
- Le résultat de la segmentation n'est pas unique (méthode, critère de l'homogénéité, initialisation, etc.)
- Une segmentation peut être interprétée comme un graphe (nœuds = régions, liens entre régions voisines)



Exemple de méthodes de segmentation

La méthode de **seuillage** dont le critère est les **niveaux de gris**

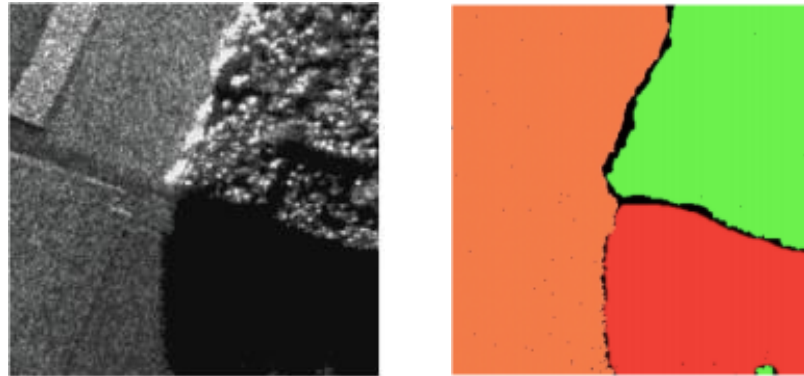


La méthode de **classification** dont le critère est la **couleur**



Exemple de méthodes de segmentation

La méthode de **classification** dont le critère est la **texture**

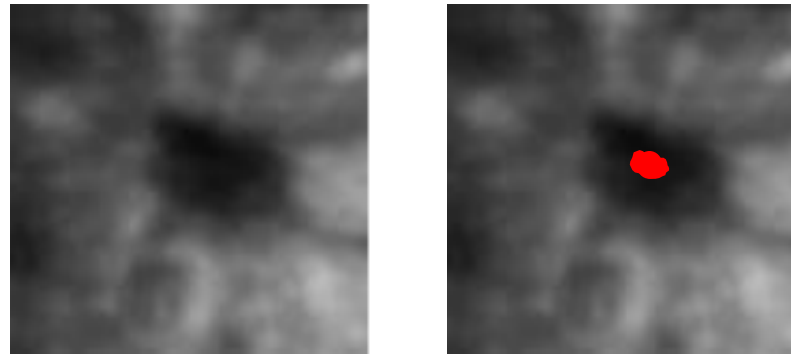


La méthode de **contours actifs** dont le critère est les **contours**

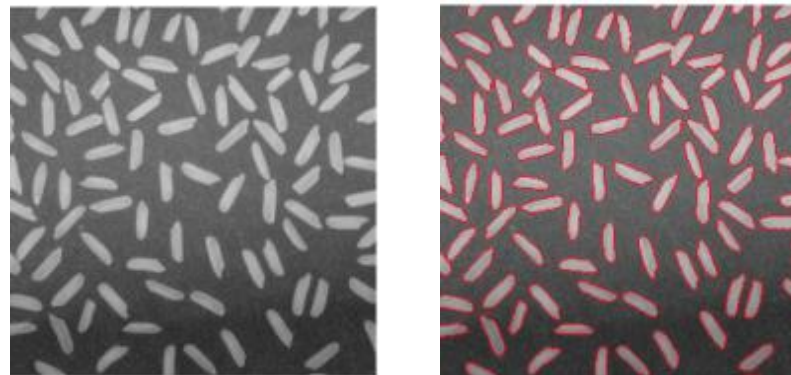


Exemple de méthodes de segmentation

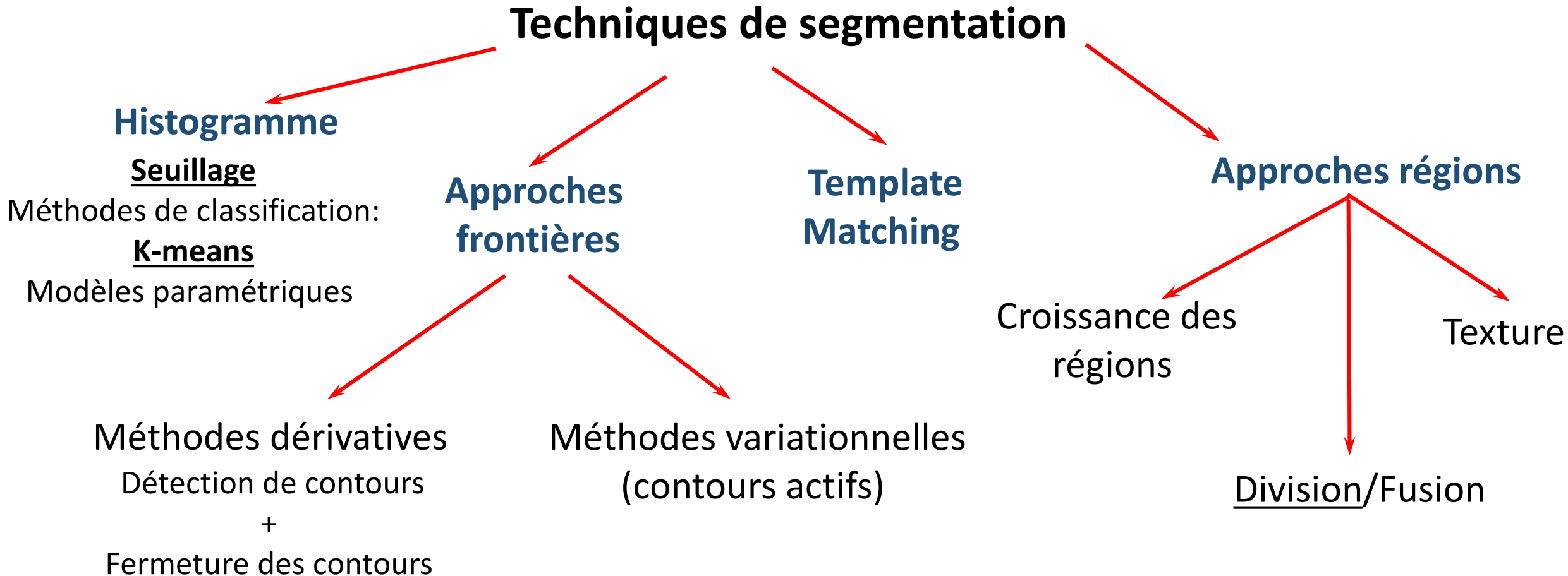
La méthode de **croissance des régions** dont le critère est la **différence d'intensité**



La méthode de **détection de contours** dont le critère est la **recherche des pixels dissemblables**



Vue d'ensemble



Segmentation par seuillage

But: affecter chaque pixel d'une image en niveaux de gris à une classe en fonction d'un seuil S .

C'est une binarisation dans le cas de deux classes

Principe:

- Extraire des seuils à partir de l'histogramme
- Classification d'un pixel p par comparaison aux seuils

Seuillage:

- Trouver le (s) seuil(s) qui sépare à mieux les deux objets (ou plus)
- Exemple de valeur de seuil: la moyenne des niveaux de gris

Deux types de méthodes de seuillage:

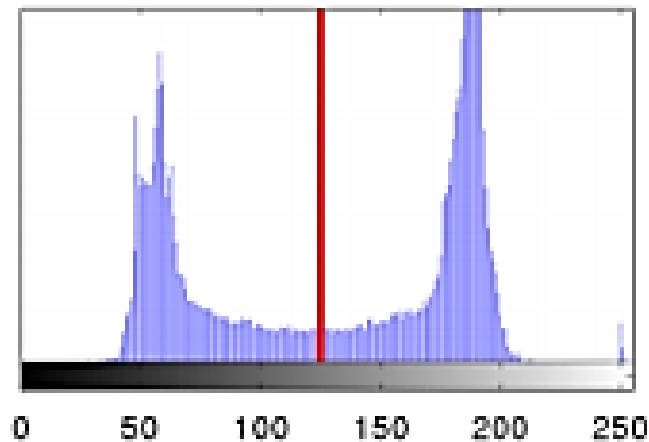
- Seuillage **global**: un seuil pour toute l'image
- Seuillage **adaptatif**: un seuil s'ajustant selon les parties de l'image

Seuillage pour une binarisation

Seuillage à 2 classes:

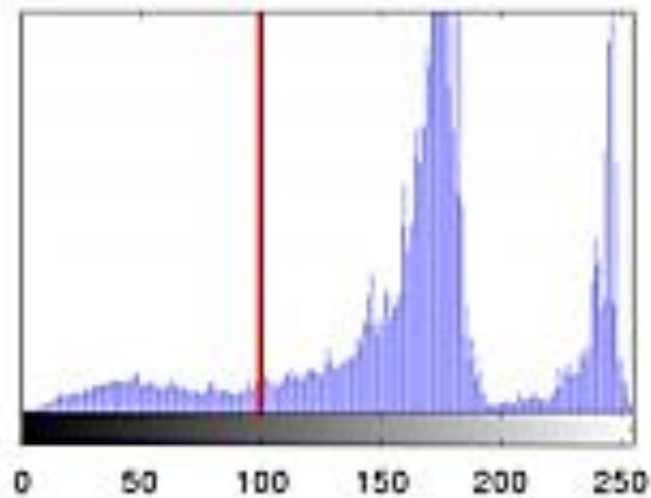
Si valeur(pixel) > seuil	alors	valeur(pixel) = 1
Si valeur(pixel) ≤ seuil	alors	valeur(pixel) = 0

Le résultat est une image binaire



Seuillage pour une binarisation

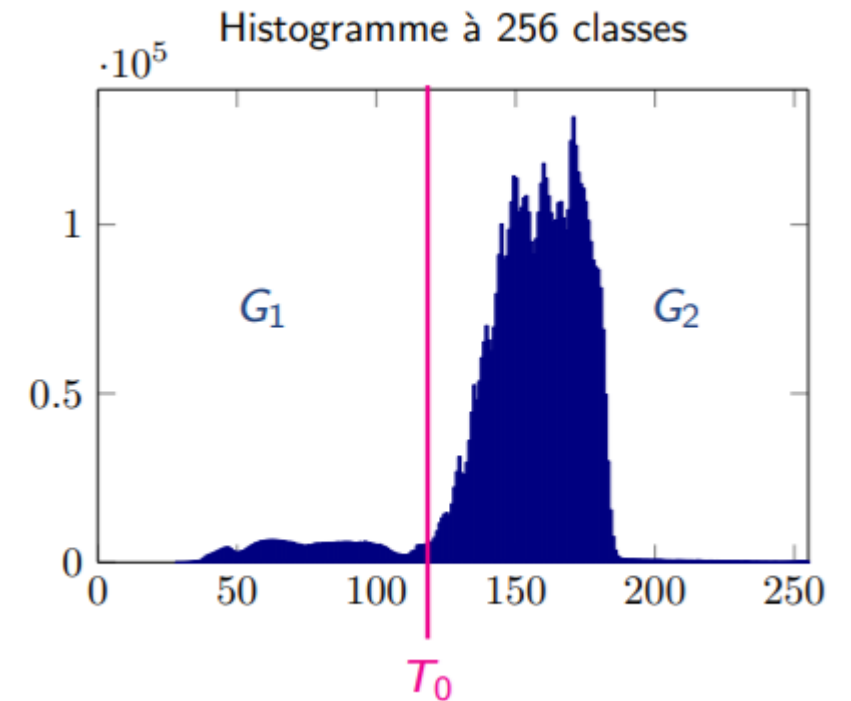
Choix du seuil est moins évident



Seuillage automatique

Algorithme

- Calculer l'histogramme de l'image
- Sélectionner un seuil initial T_0
- Calculer des intensités moyennes μ_1 et μ_2 des groupes G_1 et G_2 .
- Calculer le nouveau seuil $T = (\mu_1 + \mu_2)/2$
- Continuer jusqu'à que les variations de T soient inférieures à ε (défini par l'utilisateur)



Méthode de «Otsu»

La méthode de Otsu permet de déterminer un seuil optimal qui **minimise** la variance **intra-classe** σ_w^2 (raffinement de la méthode du seuillage automatique)

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Cela revient à **maximiser** la variance **inter-classe** σ_b^2

$$\sigma_b^2(t) = q_1(t)q_2(t)(\mu_1(t) - \mu_2(t))^2$$

La variance σ^2 des intensités de l'image est $\sigma^2 = \sigma_w^2 + \sigma_b^2$

Méthode de «Otsu»

La probabilité de chaque classe:

$$q_1(t) = \sum_{i=1}^t P(i) \quad \text{et} \quad q_2(t) = \sum_{i=t+1}^{256} P(i)$$

- Avec
 - $P(t) = \frac{h_t}{N}$ la probabilité de t
 - h_t : le nombre de pixels du niveau de gris t
 - N : Nombre total de pixels

La moyenne de chaque classe:

$$\mu_1(t) = \frac{1}{q_1(t)} \sum_{i=1}^t i P(i) \quad \text{et} \quad \mu_2(t) = \frac{1}{q_2(t)} \sum_{i=t+1}^{256} i P(i)$$

La variance de chaque classe :

$$\sigma_1^2(t) = \frac{1}{q_1(t)} \sum_{i=1}^t (i - \mu_1(t))^2 P(i) \quad \text{et} \quad \sigma_2^2(t) = \frac{1}{q_2(t)} \sum_{i=t+1}^{256} (i - \mu_2(t))^2 P(i)$$

Méthode de «Otsu»

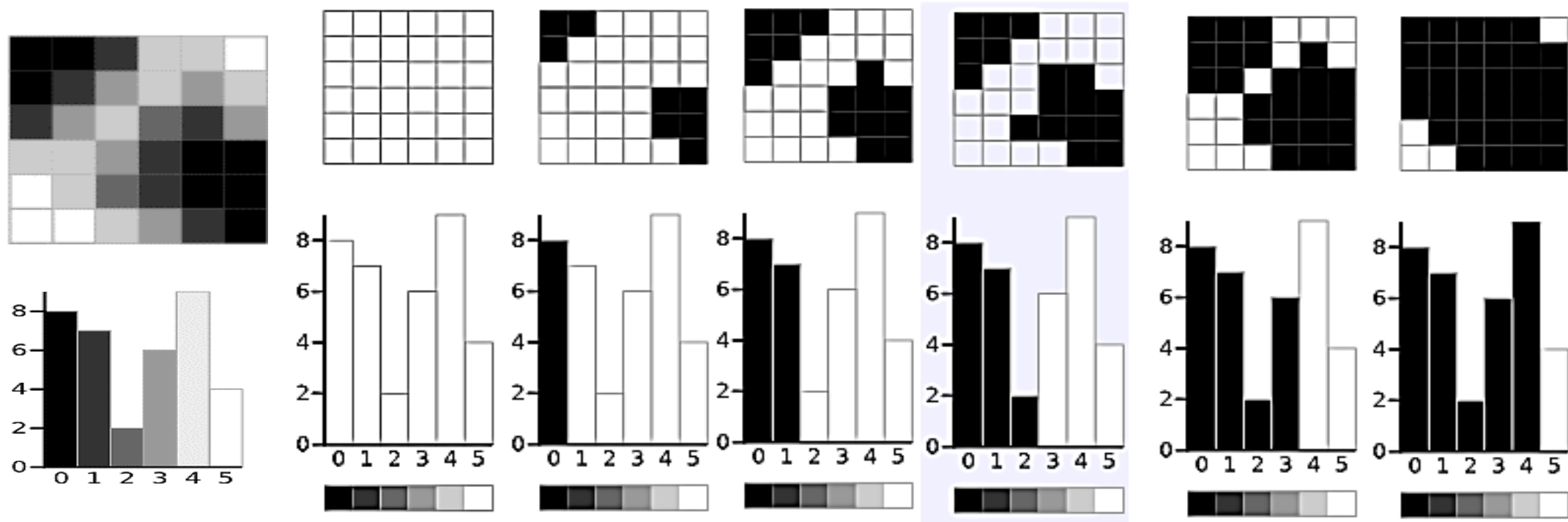
Implémentation de la méthode

Calculer pour tous les seuils $t \in [1, 256]$ la variance intra-classe $\sigma_w^2(t)$ et retenir le seuil T qui minimise $\sigma_w^2(t)$

En Matlab:

`thresh = graythresh(Im)`

Etapes d'«Otsu»



Seuil	S=0	S=1	S=2	S=3	S=4	S=5
Variance intra-classe	$\sigma_w^2=3.1196$	$\sigma_w^2=1.5268$	$\sigma_w^2=0.5561$	$\sigma_w^2=0.4909$	$\sigma_w^2=0.9779$	$\sigma_w^2=2.2491$
Variance inter-classe	$\sigma_B^2=0$	$\sigma_B^2=1.5928$	$\sigma_B^2=2.5635$	$\sigma_B^2=2.6287$	$\sigma_B^2=2.1417$	$\sigma_B^2=0.8705$

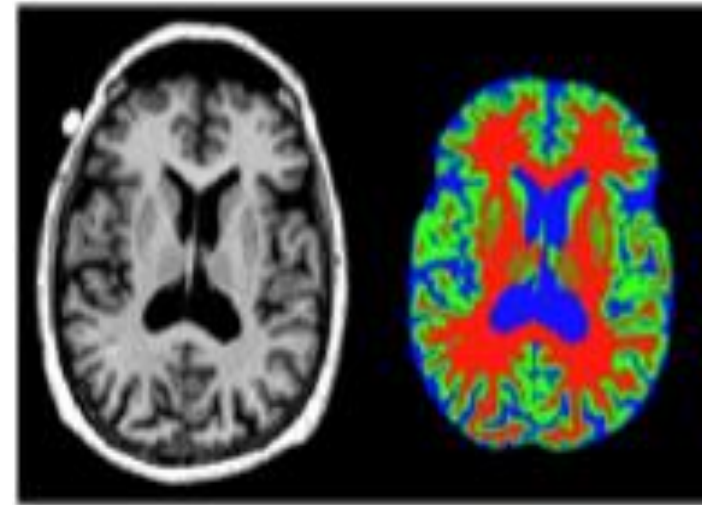
Seuillage multi-seuils

Lorsque plusieurs modes sont visibles sur l'histogramme, il est possible d'utiliser plusieurs seuils pour aboutir à plusieurs classes

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

En Matlab:

thresh = **multithresh**(Im,N)



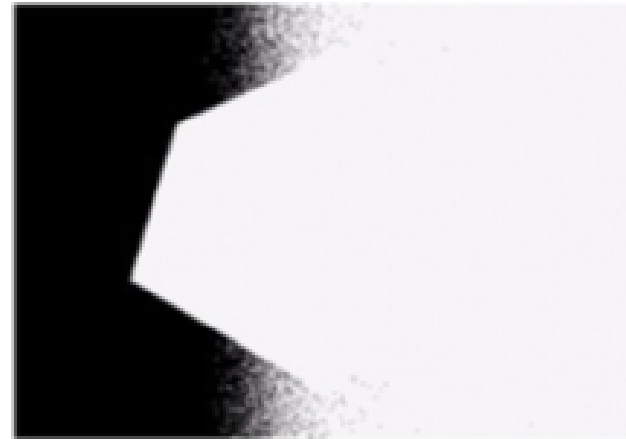
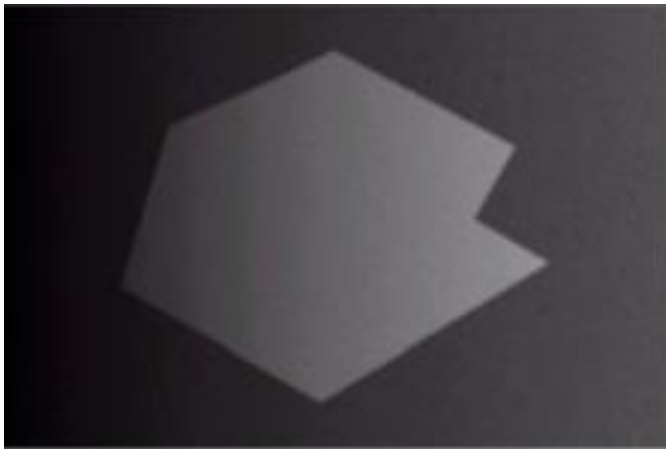
■ $r \in [0, T_1]$

■ $r \in]T_1, T_2]$

■ $r \in]T_2, 2^K - 1]$

Cas de l'éclairage non uniforme

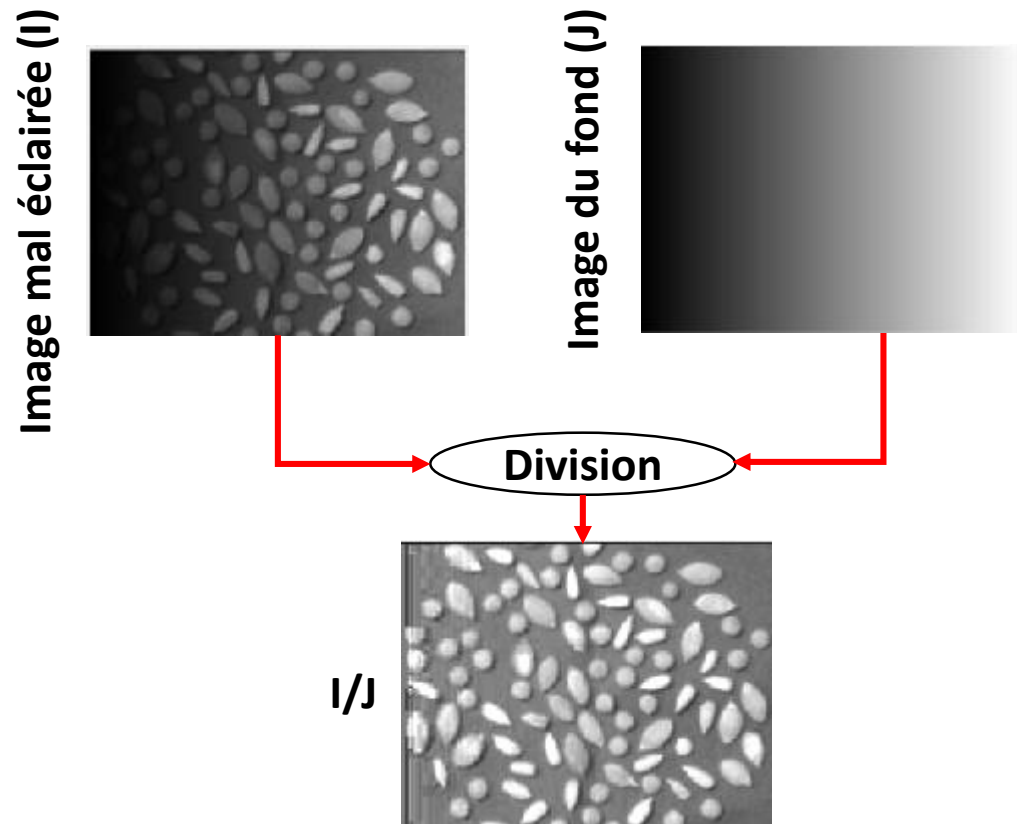
La variation d'illumination ne permet pas de seuiller l'image correctement.



Plusieurs solutions sont possibles

Cas de l'éclairage non uniforme

Le défaut d'éclairage est connu, on utilise l'opération de la division pour corriger l'image avant le seuillage



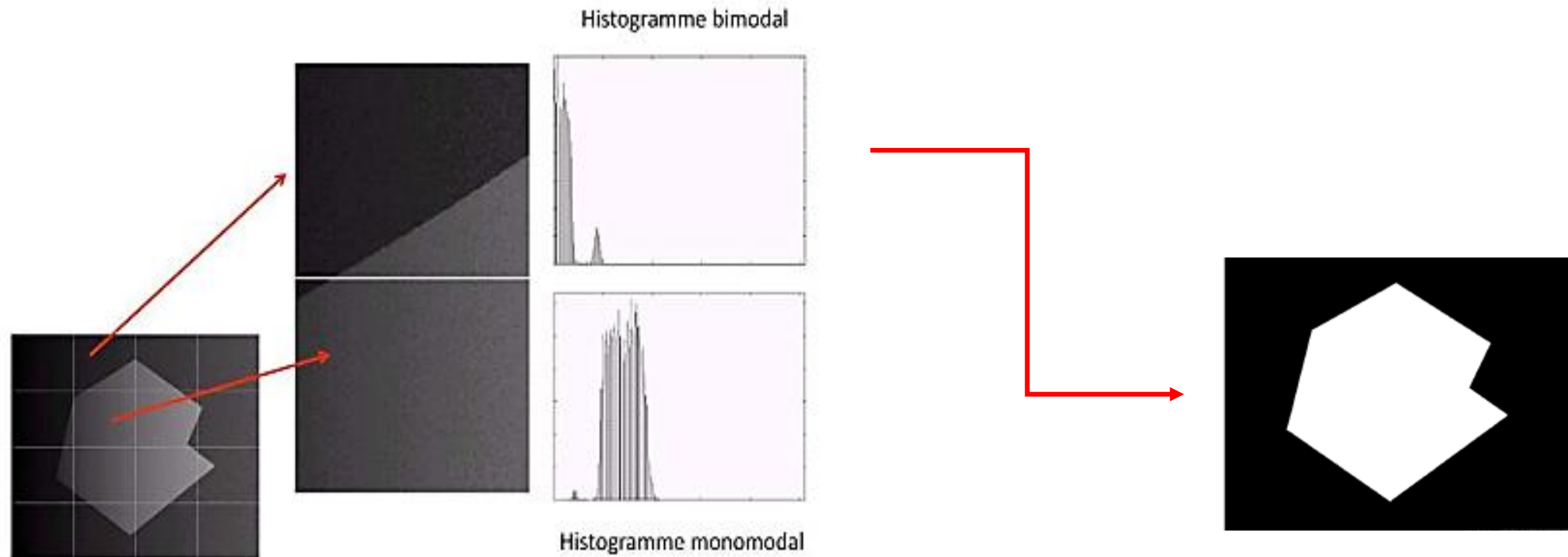
Seuillage par Otsu



Cas de l'éclairage non uniforme

Le défaut de l'éclairage est inconnu ➡ Seuillage adaptatif

- On divise l'image en un certain nombre de sous-régions
- On calcule le seuil pour chaque région indépendamment



Avantages/Inconvénients

Avantages:

- Universel, temps réel, simplicité
- Fonctionne bien sur des histogrammes multimodaux

Inconvénients:

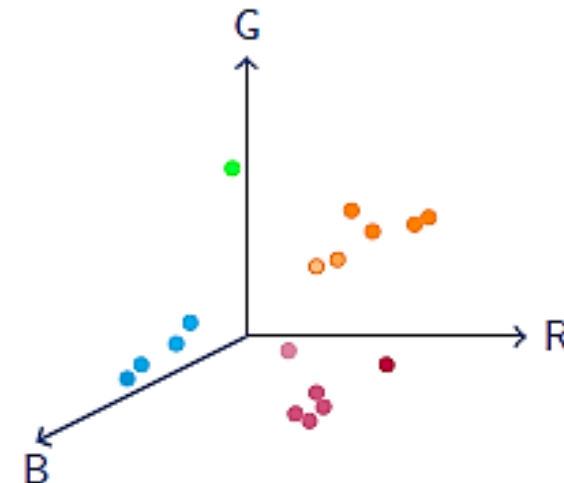
- Apparition de faux éléments (aucune prise en compte de la composante spatiale)
- Nombre de modes souvent différents du nombre de classes attendues
- Il faut connaître le nombre de classes



Deux modes mais trois classes

Méthodes de clustering: K-means

- Extension du seuillage d'histogramme aux images couleurs
- Chaque pixel contient 3 composantes R, V et B
 - 3 histogrammes pour représenter chaque bande.
- Difficulté à exploiter la méthode basée sur l'histogramme
- Principe des méthodes de clustering: regrouper les vecteurs en groupes homogènes.



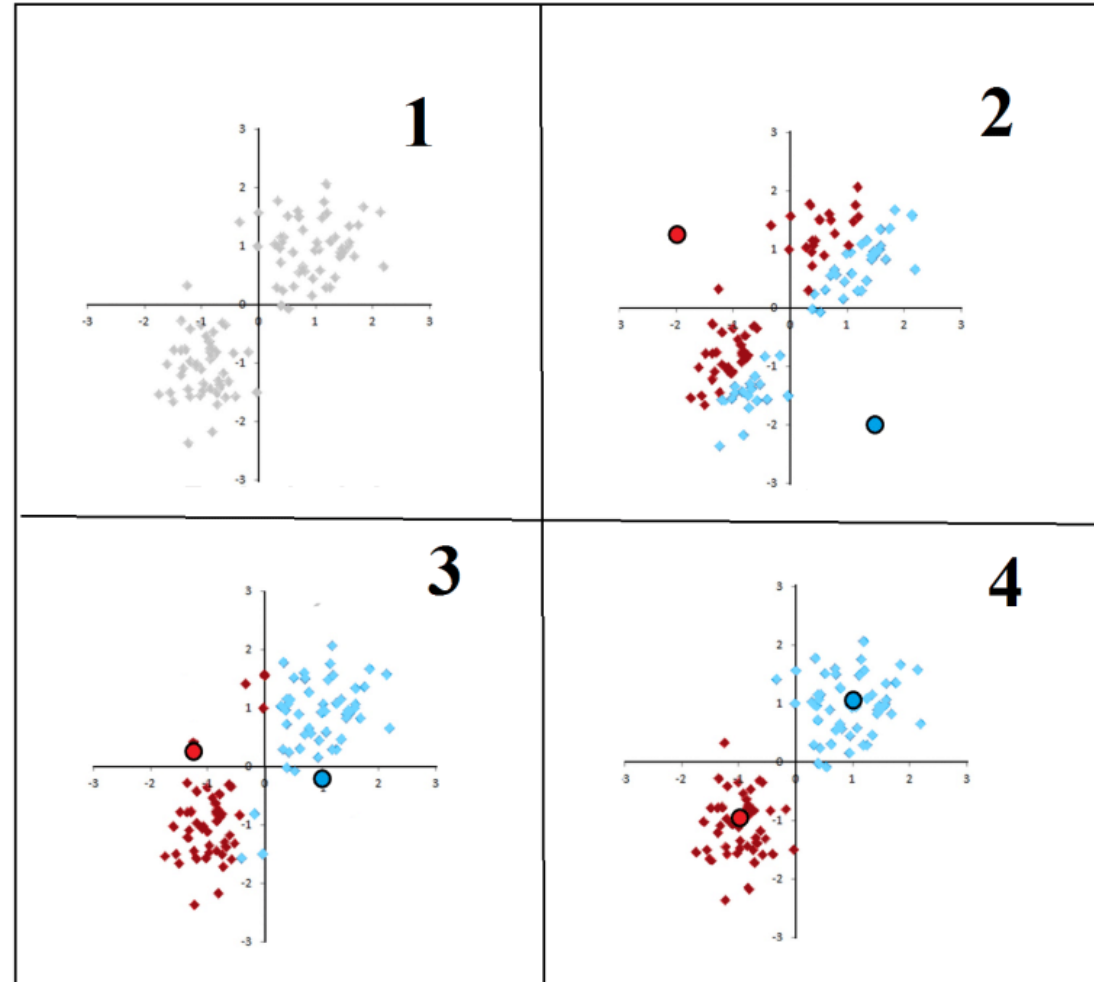
Algorithme de K-means

On veut diviser les points en k groupes (clusters) (k est donné à l'avance)

Algorithme

- Partitionnement aléatoire des points en k groupes
- Calcul du centroïde de chacun des groupes
- Répéter tant que les centroïdes varient:
 - Pour chaque point:
 - Calcul des distances du point à tous les centroïdes
 - Affectation au point au groupe le plus proche
 - Calcul du centroïde de chacun des groupes

Exemple en 2D



Avantages/Inconvénients

Avantages

- Méthode simple
- Implémentation facile
- Fonctionne correctement lorsque les clusters sont sphériques



Inconvénients

- Nécessite de connaître le nombre de classes
- Sensible à l'initialisation
- Peut être lent en grande dimension
- Échoue pour des structures non sphériques



- Sensible aux valeurs aberrantes



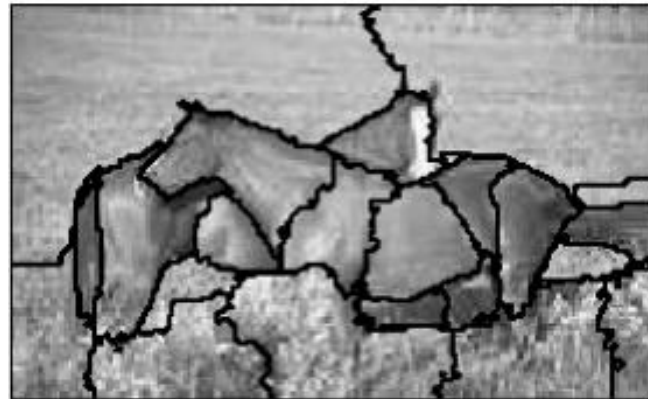
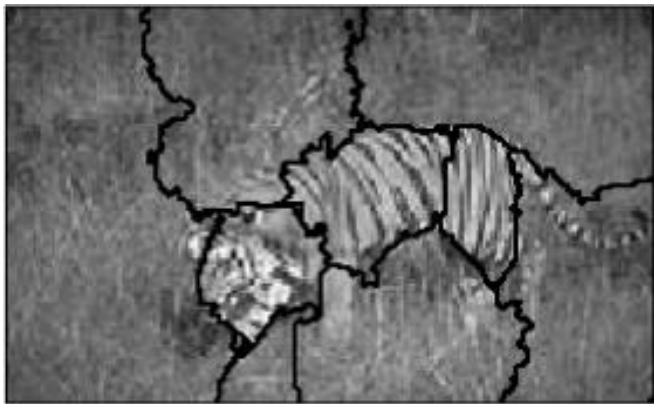
Limites de la segmentation

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



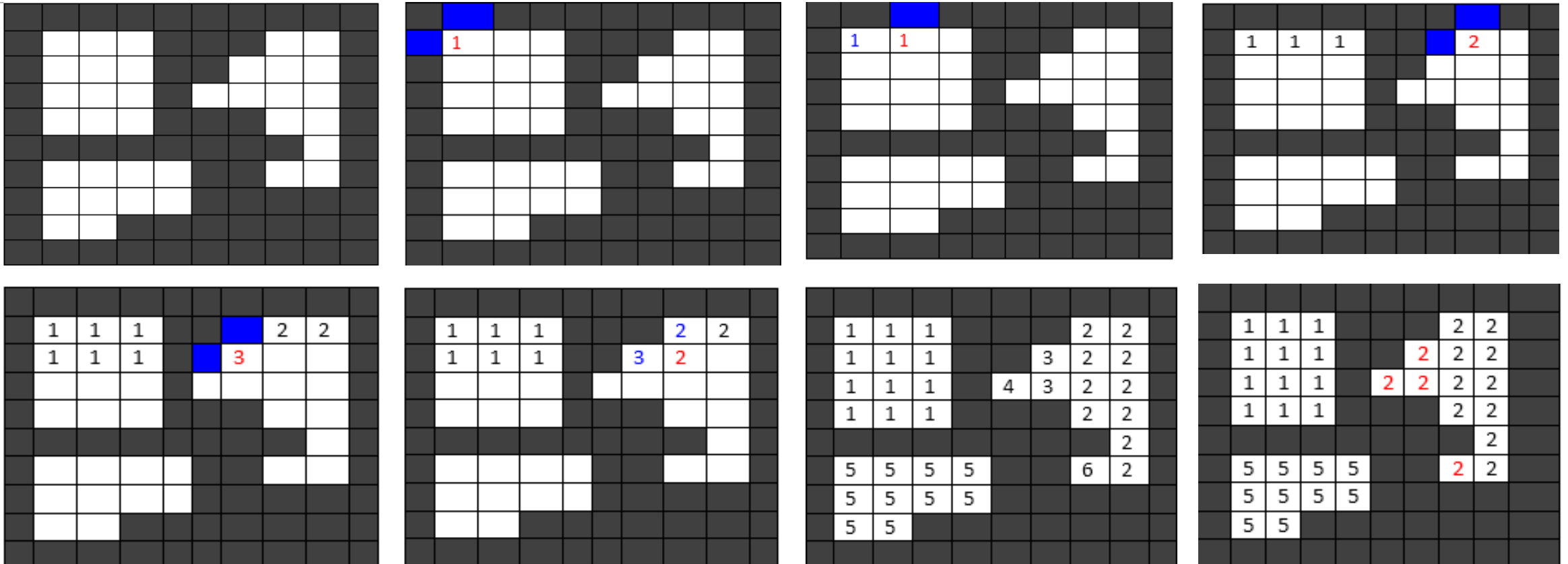
Limites de la segmentation

Un objet n'appartient pas à une région !



Etiquetage des régions (labels)

Un **plan d'étiquettes** est une image de même dimension et parallèle au plan image. Il indique les étiquettes correspondant aux pixels (toutes les étiquettes sont différentes).



Utilité d'une image «étiquette»

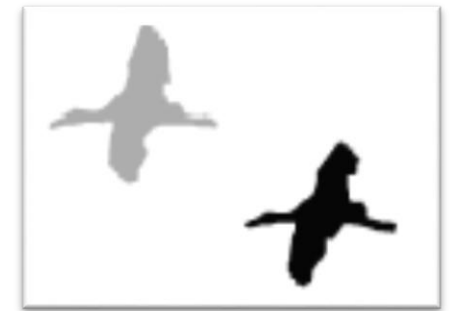
Cas d'une image binaire

- Numériquement, on ne distingue pas les 2 oiseaux

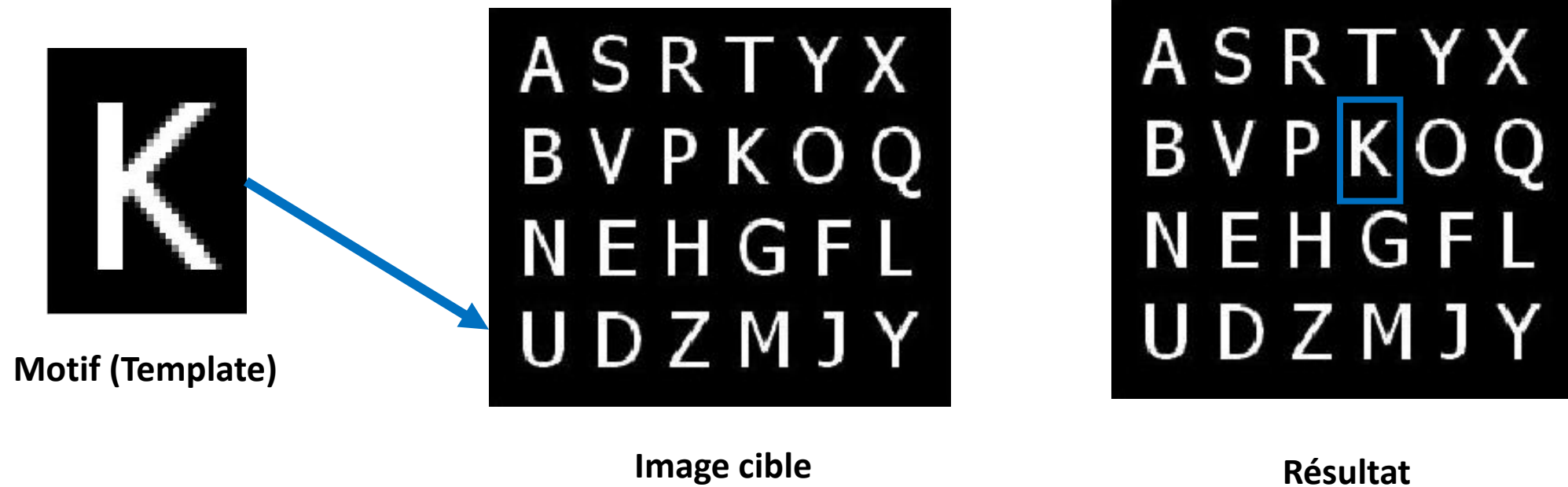


Cas d'une image «étiquette»

- Chaque oiseau est une entité propre et peut être étudié individuellement.



Template Matching



La fonction de Matlab est `normxcorr2(Motif,ImageCible)`

Template Matching

Basée sur la recherche de la position spatiale d'un motif (objet) M connu dans une image I

La recherche se fait par le calcul de l'intercorrélation bidimensionnelle $C(x, y)$ entre l'image I et l'image motif M :

$$C(x, y) = \frac{1}{wh} \sum_{i=-h/2}^{i=h/2} \sum_{j=-w/2}^{j=w/2} \frac{(I(x + i, y + j) - \mu_{xy})(M(i + \frac{h}{2}, j + \frac{w}{2}) - \mu_M)}{\sigma_{xy}\sigma_M}$$

- w et h représentent respectivement la largeur et la hauteur de l'image motif M .
- μ_{xy} et σ_{xy} sont respectivement la moyenne et l'écart type des pixels dans la fenêtre autour du pixel (x, y) de l'image source I
- et μ_M et σ_M sont respectivement la moyenne et l'écart type des pixels de l'image motif M