Chapitre 6 : Morphologie mathématique

Joël Lefebvre (UQÀM)

INF600F – Traitement d’images

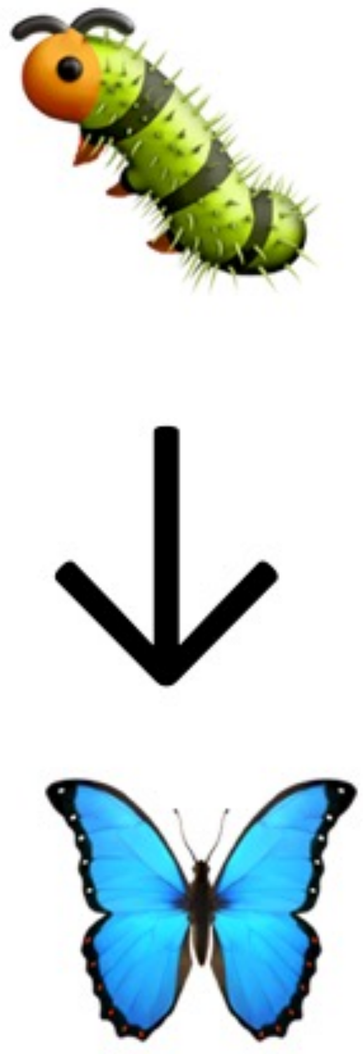
Automne 2024

Annonces

- **Atelier** : Segmentation / morphologie, TP3
- **Correction de l’examen** :
  - Remise des notes cette semaine.
  - Consultation la semaine prochaine.
- **TP3** : remise le dimanche 24 novembre avant 23h59
- **Mercredis-Recherche** (https://youtu.be/E46x3yPLZAs)
- **À venir** :
  - **TP4** (S11 à S13)
  - **Examen final** (S15, 11 décembre 2024)
  - **Évaluation des enseignements**

Survol du cours

- Introduction
- Utilisation des images binaires
- Morphologie mathématique (binaire)
- Algorithmes basés sur la morphologie
- Morphologie mathématique (niveaux de gris)



Le terme **« opération morphologiques »** vient de **métamorphose** (ou *morphing) en* biologie, qui signifie « changer radicalement de forme » (Antidote)

Références

- (Chityala, 2020) Ch9 : *Morphological Operations*
- (Burger, 2009) Vol1, Ch7 : *Morphological Filters*
- (Gonzalez, 2018) Ch9 : *Morphological Image Processing*
- (Russ, 2016) Ch8 : *Processing Binary Images*

Opérations sur les images binaires

Chapitre 6 : Morphologie mathématique

Joël Lefebvre (UQÀM)

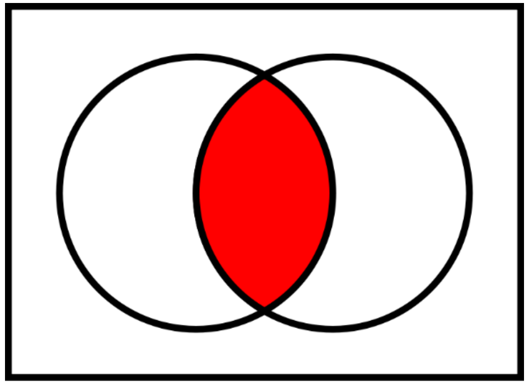
INF600F – Traitement d'images

Opérations booléennes

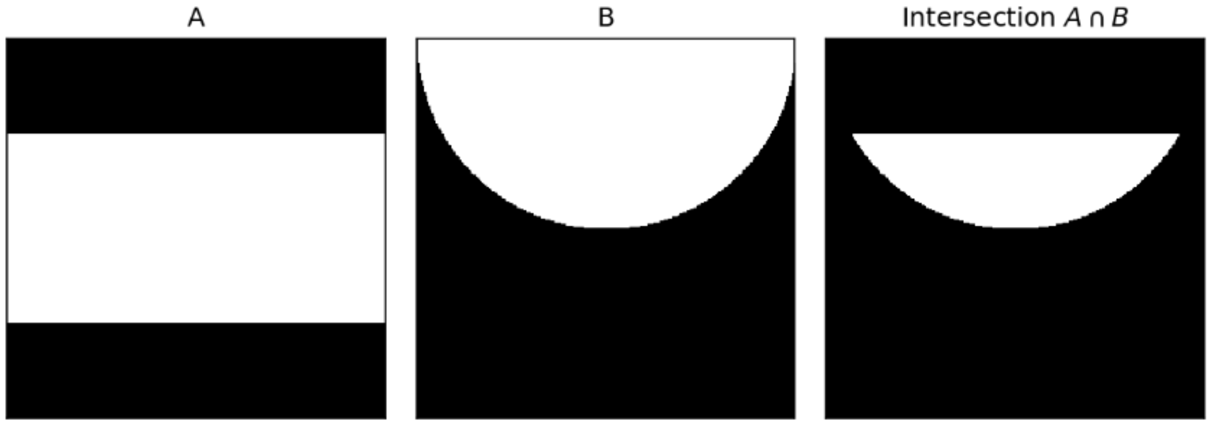
- Image binaire : 0 / 1, False / True, 0 / 255, …
- Opérations logiques ponctuelles entre 2 images
- Plusieurs opérateurs disponibles
  - ET (*AND*) : Intersection
  - OU (*OR*) : Union
  - NON (NOT) : Inversion
  - OUX (XOR) : Exclusion

Opérateur booléen : Intersection ∩

- Noté ET (AND en anglais)
- Notation mathématique : $A \cap B$
- Notation Python : **numpy.logical_and(A, B)**



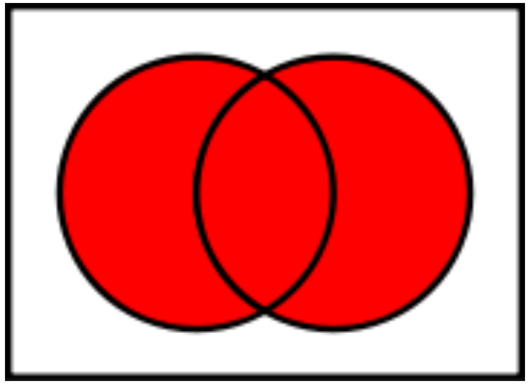
Intersection entre A et B,
Diagramme de Venn (Wikipedia)

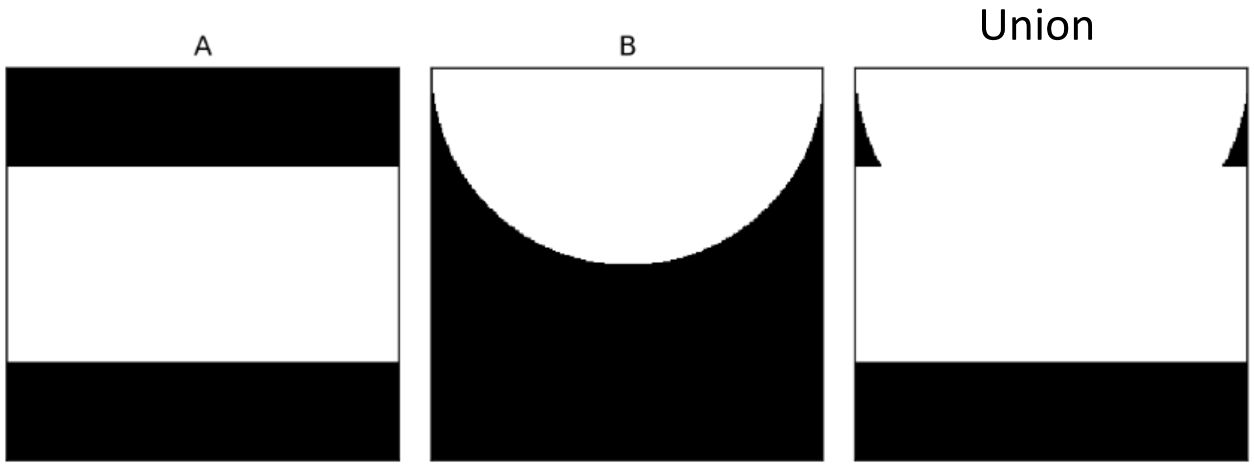


Notation alternative en python pour A et B
numpy.ndarray : C = A & B

Opérateur booléen : Union ∪

- Noté OU (OR en anglais)
- Notation mathématique : $A \cup B$
- Notation Python : **numpy.logical_or(A, B)**



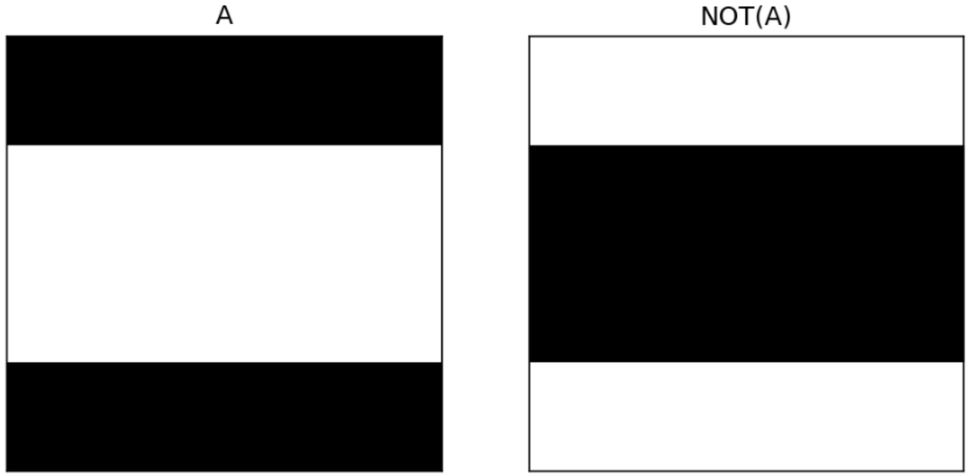


Union entre A et B,
Diagramme de Venn ([Wikipedia](#))

Notation alternative en python pour A ou B
numpy.ndarray : C = A|B

Opérateur booléen : Inversion

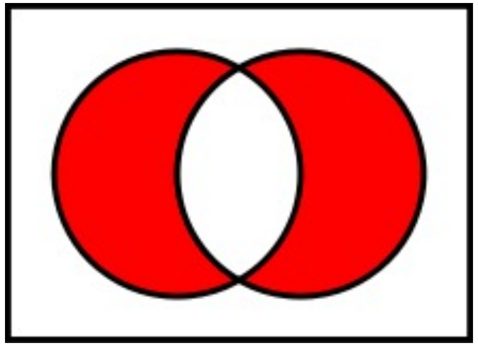
- Noté NON (NOT en anglais)
- Notation mathématique : $\bar{A}$ ou A^c
- Autre nom : **complément**
- Pour chaque pixel, VRAI et Faux sont inversés
- Notation Python : **numpy.logical_not(A)**



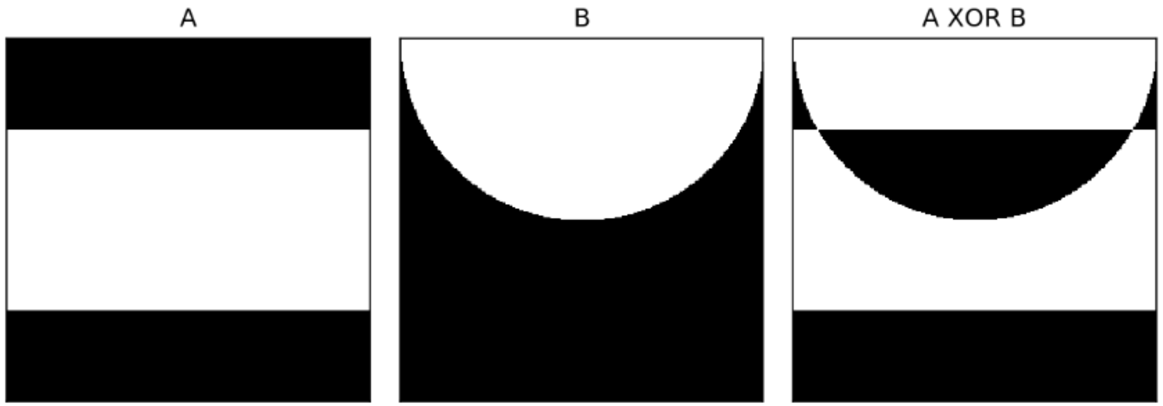
Notation alternative en python pour A numpy.ndarray de type bool : C = ~A

Opérateur booléen : OU exclusif

- Noté OUX (XOR en anglais)
- Aussi nommé : disjonction exclusive
- Notation Python : **numpy.logical_xor(A, B)**



Disjonction exclusive entre A et B, Wikipedia



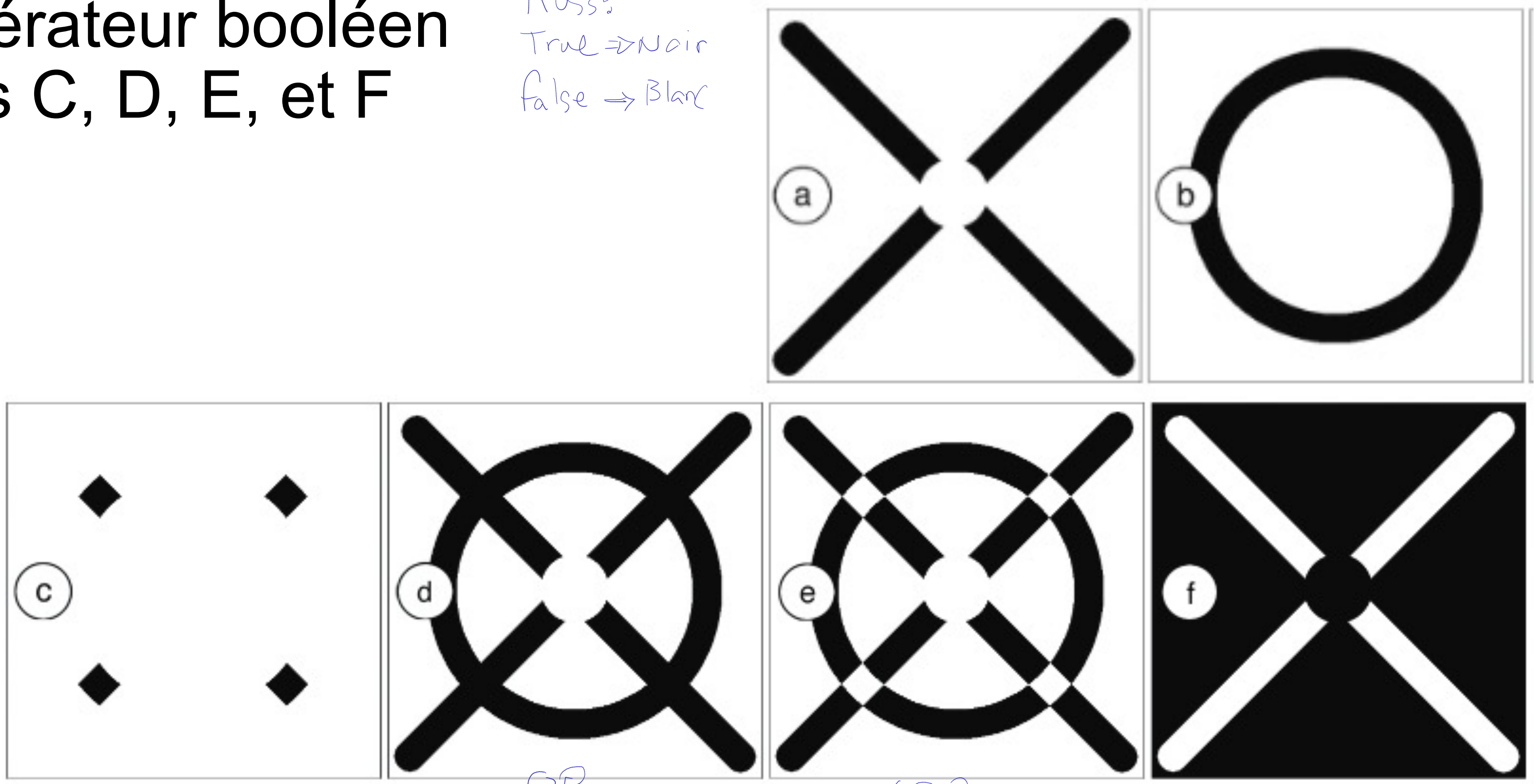
Quiz : Opérateurs booléens

- Associez un opérateur booléen avec les images C, D, E, et F
- Choix :
  1. A OR B
  2. A XOR B
  3. NOT A
  4. NOT B
  5. A AND B
  6. B XOR A

Russ:
True => Noir
False -> Blanc

Image binaire A

Image binaire B



OR

XOR

NOT A

Source : Fig8.1-8.2, Russ2016

Combinaison d’opérations booléennes

- Les opérateurs AND, OR, XOR et NOT peuvent être **combinés** selon les applications
- **Cas spécial :** XOR peut être remplacé par une combinaison de AND, OR et NOT
- A XOR B ≡ [NOT(A) AND B] OR [A AND NOT(B)]

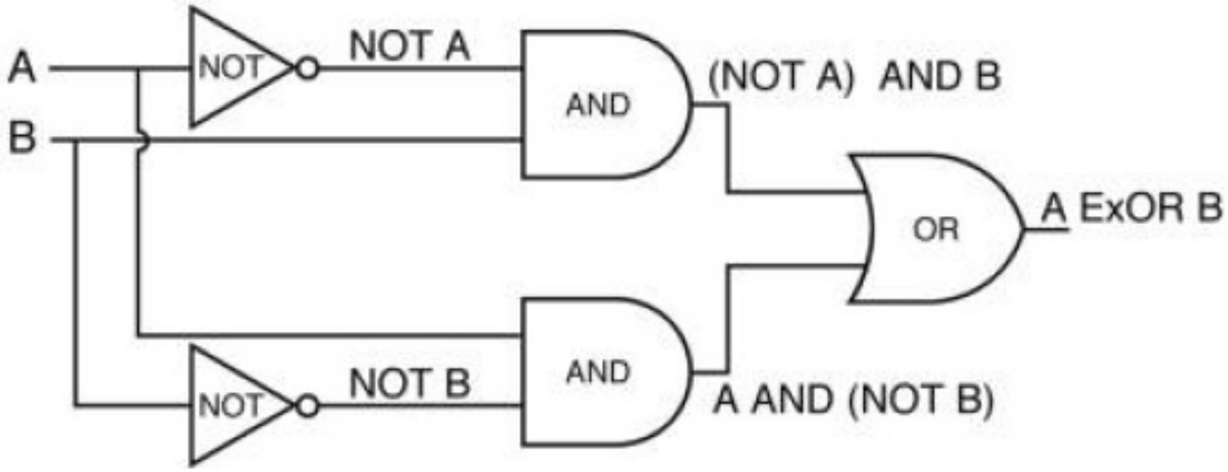
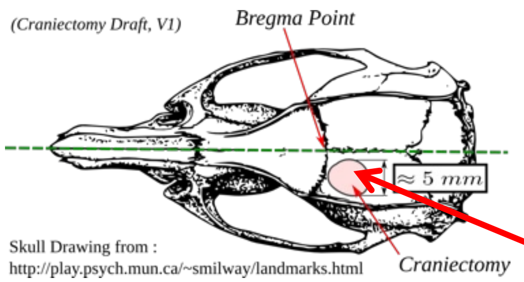
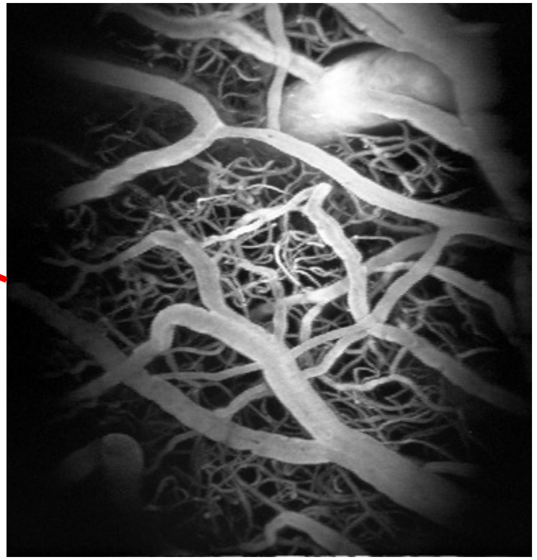


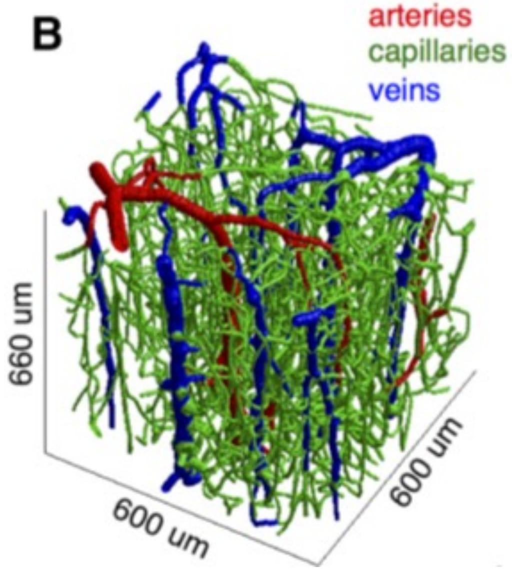
Diagramme d’un circuit utilisé pour créer l’opérateur A XOR B à partir d’opérateurs AND, OR et NOT

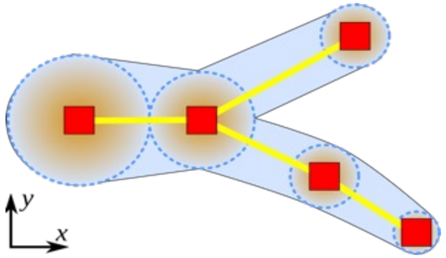
A	B	A XOR B
VRAI	VRAI	FAUX
FAUX	FAUX	FAUX
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI

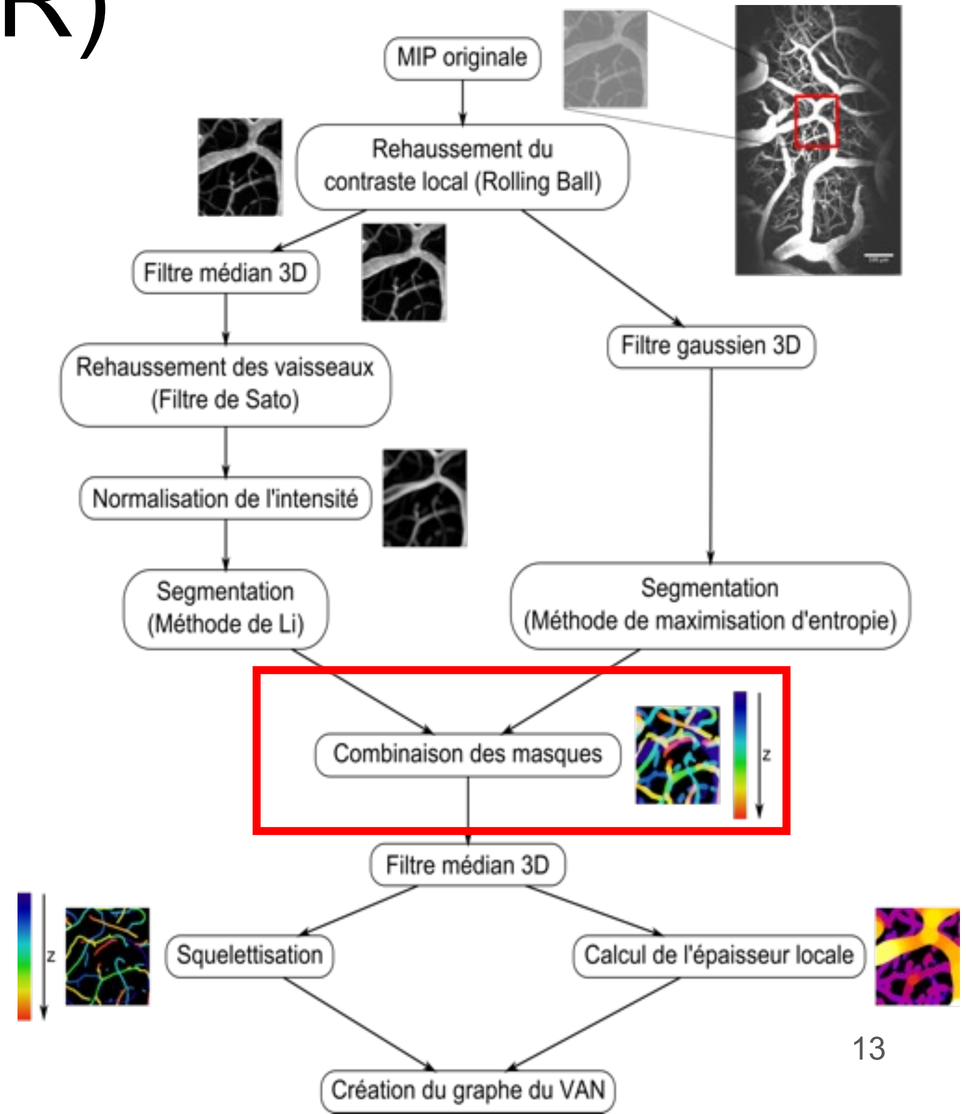
Exemple : Segmentation des vaisseaux sanguins à plusieurs échelles et combinaison booléenne des résultats (OR)





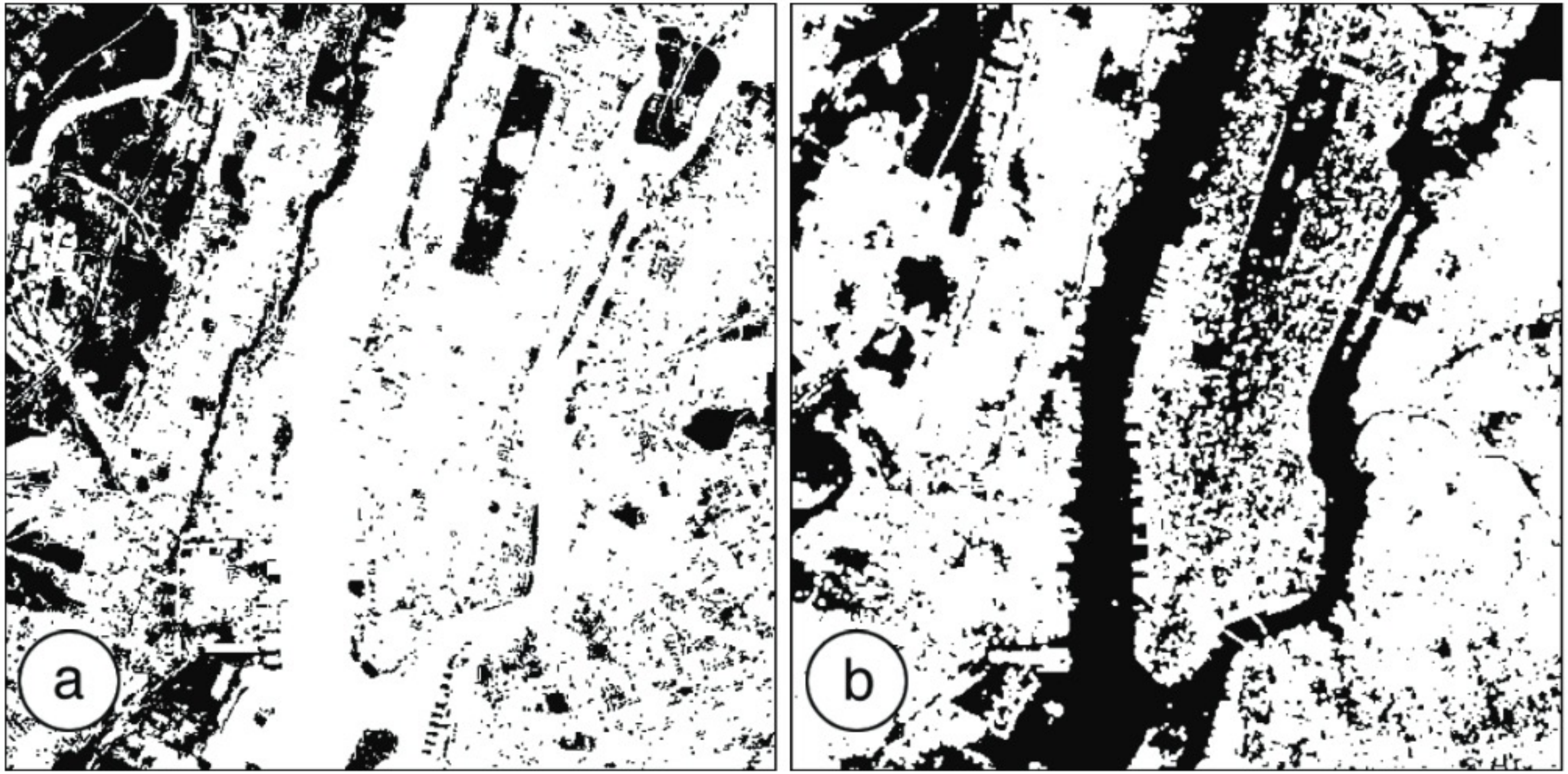






Exemple : Opérateur AND pour trouver la végétation en imagerie satellite (1)

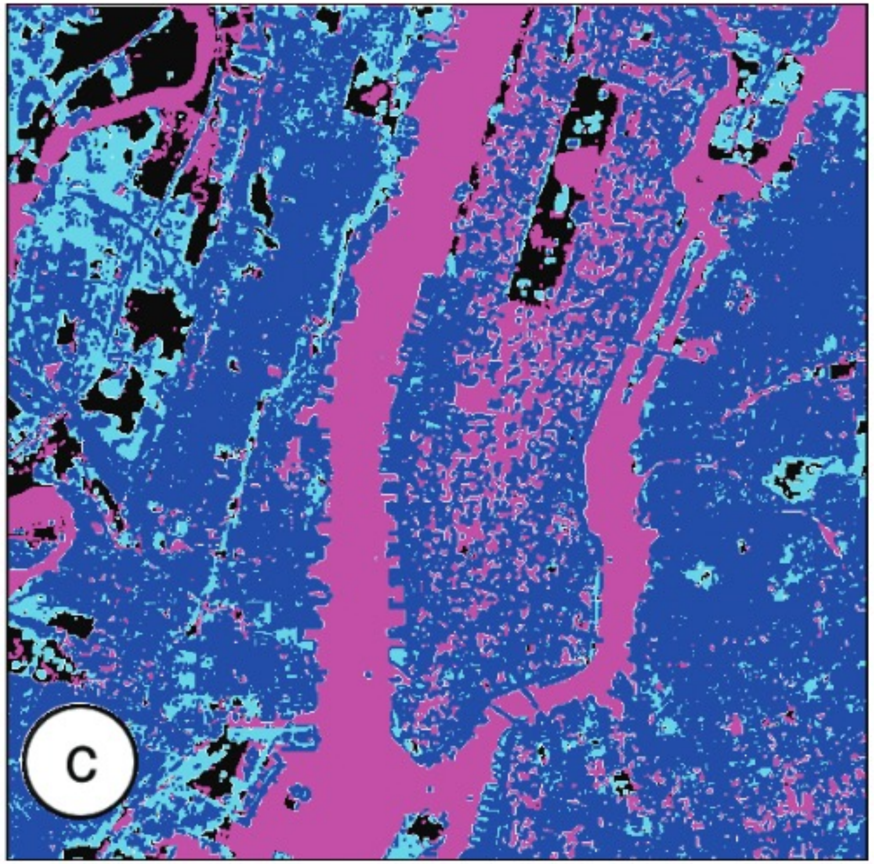
Proche infrarouge | Lumière visible (bleu)



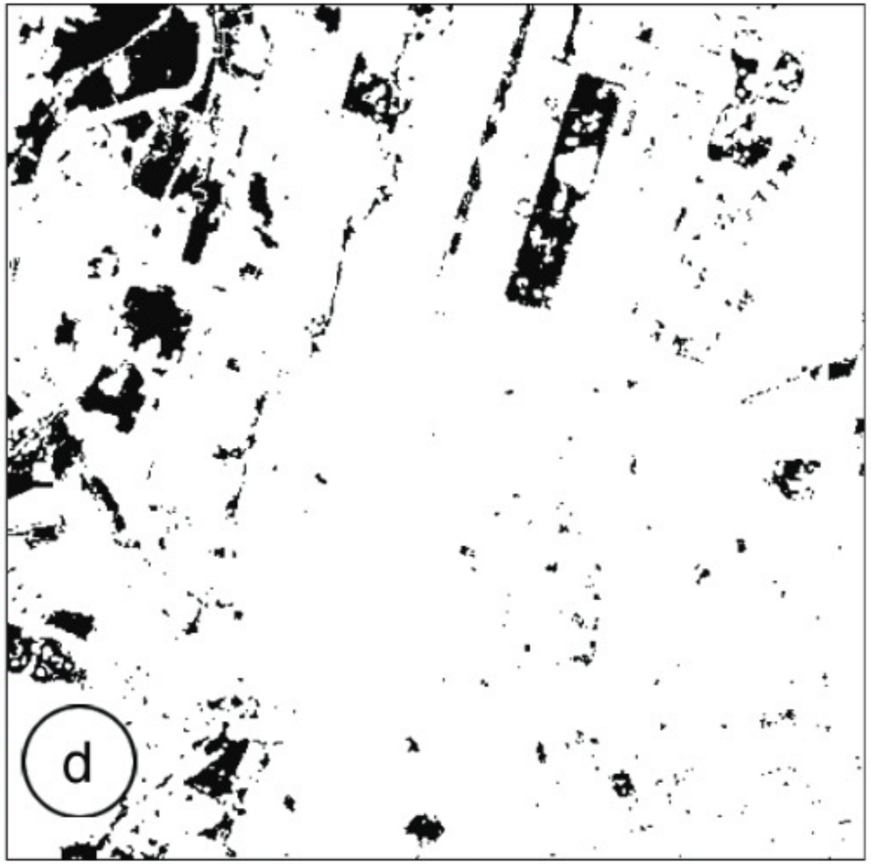
Source : Fig8.4, Russ2016

Exemple : Opérateur AND pour trouver la végétation en imagerie satellite (2)

Combinaison en couleur



A AND B

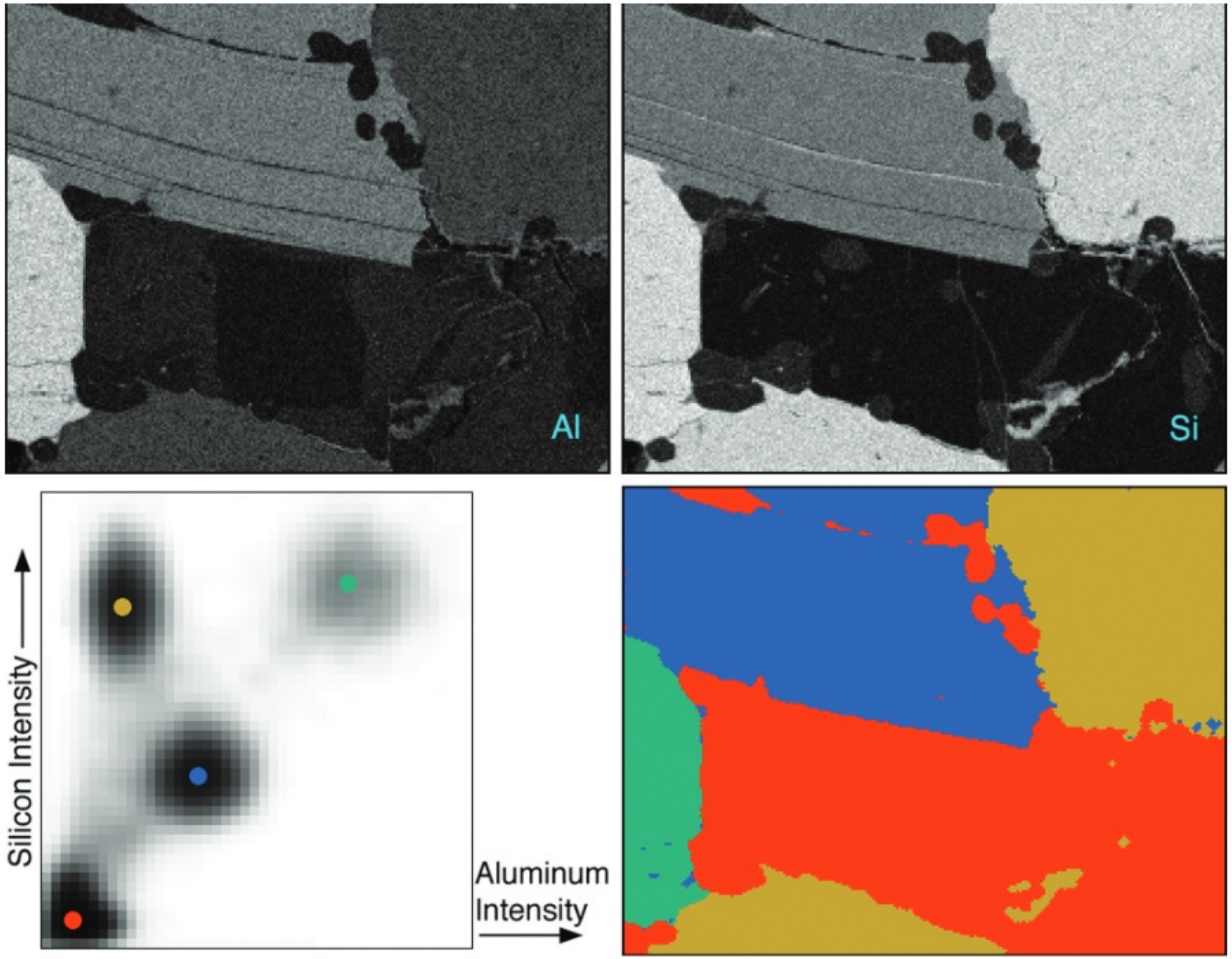


Source : Fig8.4, Russ2016

Exemple : Combinaison de plusieurs types d’images pour l’analyse des matériaux

- 2 images par Rayons X sensibles aux concentrations en silicium et en aluminium

Un graphe de **colocalisation** (co-occurrence) utilise l’intensité des pixels pour chaque position dans les deux images en tant que coordonnées pour créer un **histogramme conjoint**.



Russ2016, Fig8.

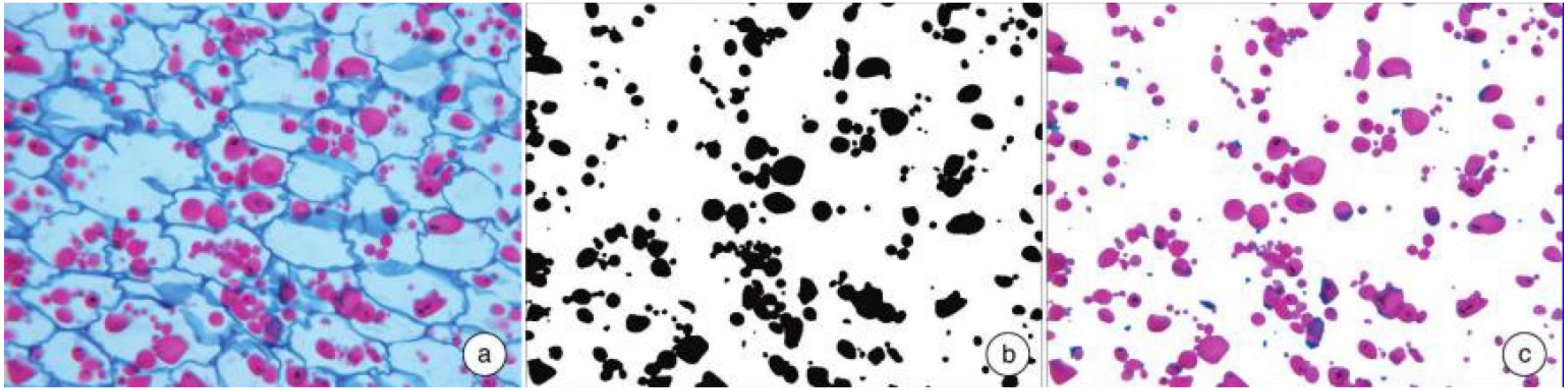
Combinaison d'une image binaire et d'une image en niveaux de gris

- Image binaire : masque
- Permet d'isoler dans l'image grise les pixels segmentés dans le masque

Tranche fine de pomme de terre marquée et observée au microscope.

Segmentation basée sur la couleur verte

Masque appliqué à l'image

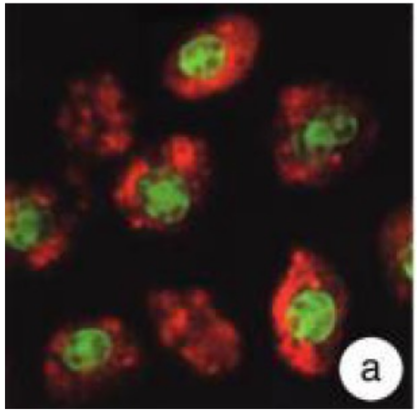


Russ2016, Fig8.14

Exemple : Noyaux cellulaires

- **Application :** Trouver les cellules ayant un noyau visible (Microscopie optique avec marqueurs fluorescents)

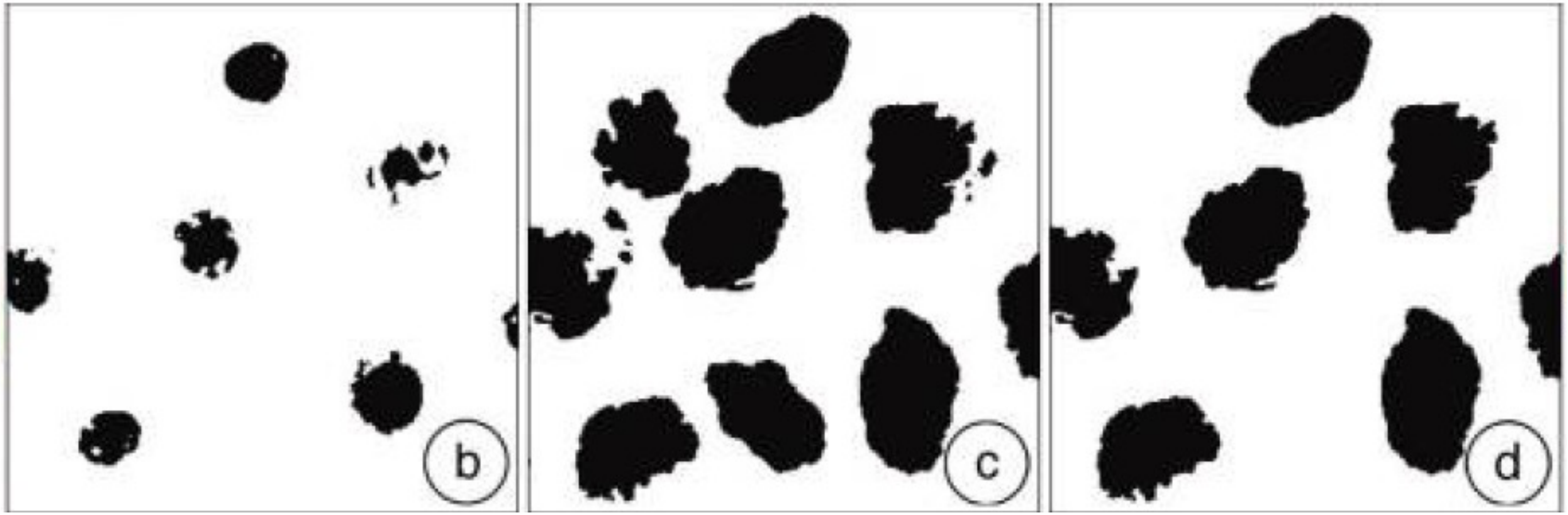
Source : Russ Fig 8.26



Seuillage (canal vert) : Noyaux

Seuillage (canal rouge) : Cellules

Cellules ayant un noyau



Morphologie binaire

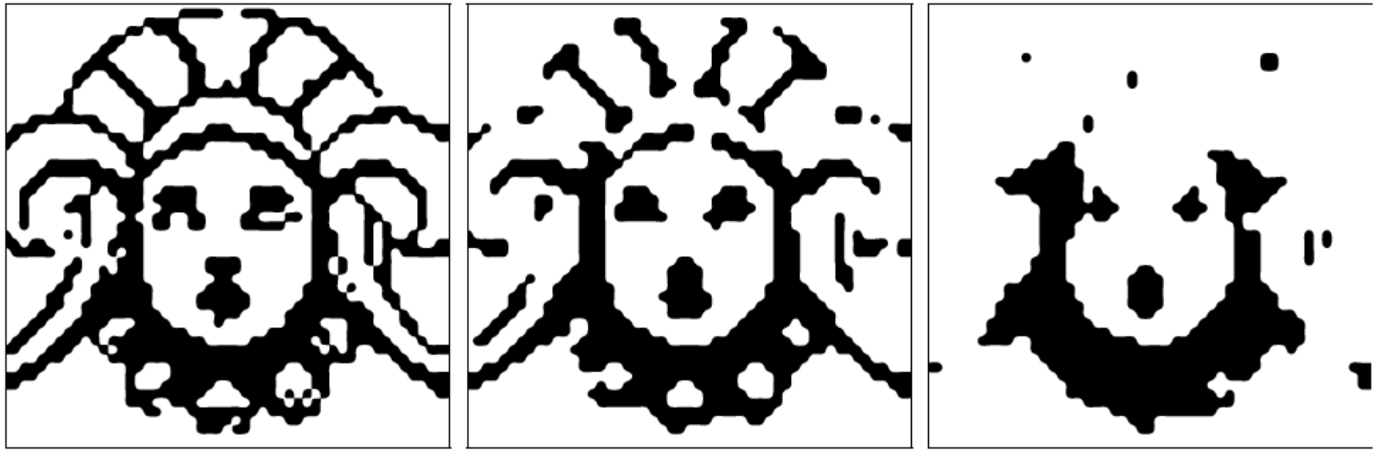
Chapitre 6 : Morphologie mathématique

Joël Lefebvre (UQÀM)

INF600F – Traitement d’images

Motivation

- Les **filtres médians** peuvent modifier les structures 2D.
- **Exemples** : arrondir des coins, remplissage de trous
- L'effet varie selon la **forme locale** des structures.
- La modification des structures de l'image d'une manière prévisible : **<u>Filtre morphologique</u>**



Filtrage d'une image binaire (gauche) avec un **filtre médian** de taille 3x3 (milieu) et 5x5 (droite)

(Burger 2009, Fig 7.1)

Filtre morphologique

- Originellement développé pour les images binaires
  - Valeurs de 0 / 1 (ou noir & blanc)
- Provenance des images binaires
  - Impression,
  - Transmission (ex: FAX),
  - Segmentation (ex: seuillage global)
- Pour ce chapitre
  - **Avant-plan :** valeur de 1 affichée en noir
  - **Arrière-plan :** valeur de 0 affichée en blanc
  - Par analogie avec un document imprimé

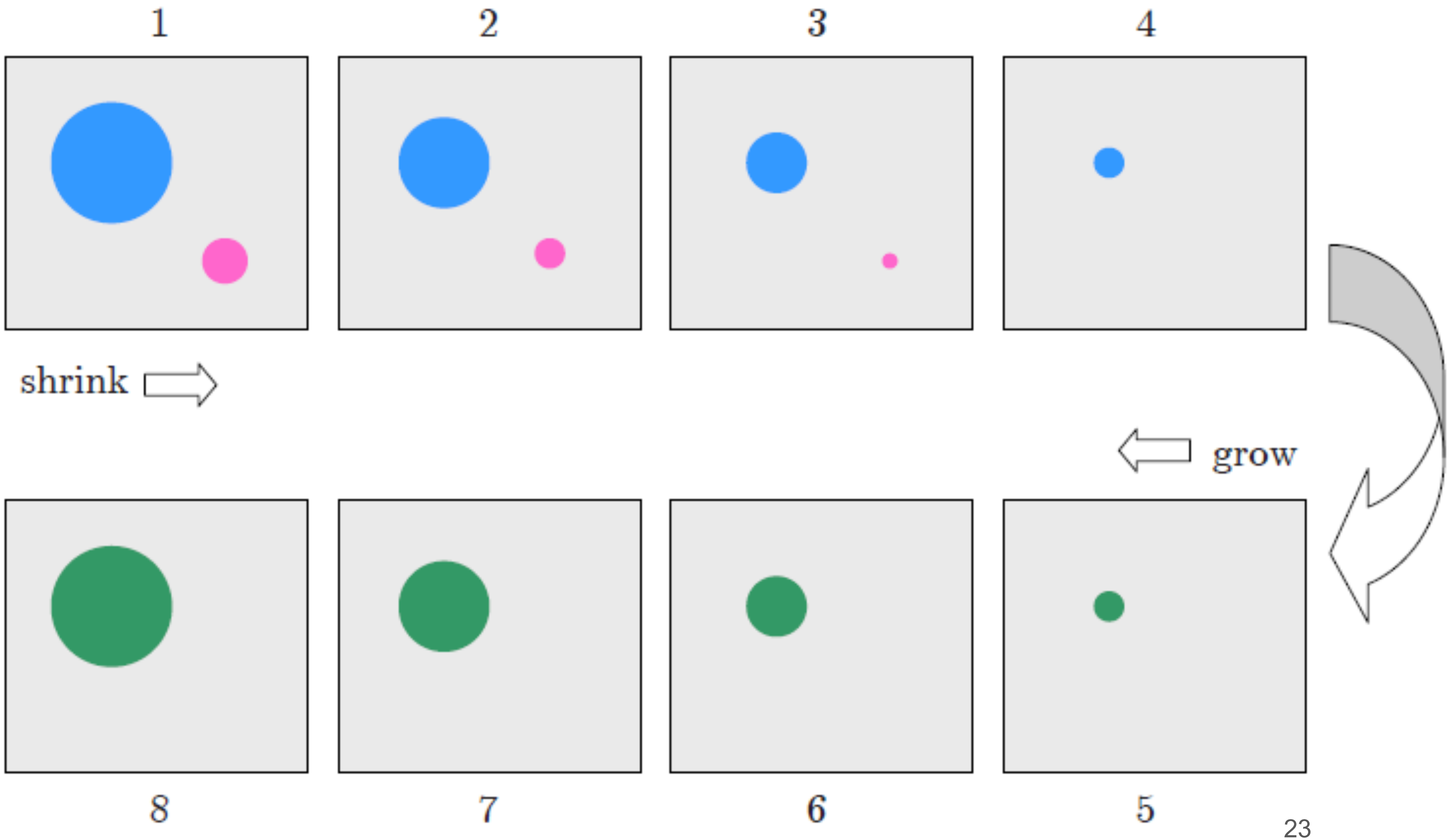
Contraction et Expansion (1)

- **<u>Observation :</u>** Un filtre médian 3x3 peut arrondir les grandes structures et retirer les petites structures (points, lignes fines) d’une image binaire
- Peut être utile pour éliminer les objets d’une image en fonction de leur taille (ex. : bruit ou saleté)
- **Comment contrôler la taille et la forme des structures affectées par l’opération ?**
- **Idée générale :** contraction et expansion

Contraction et expansion (2)

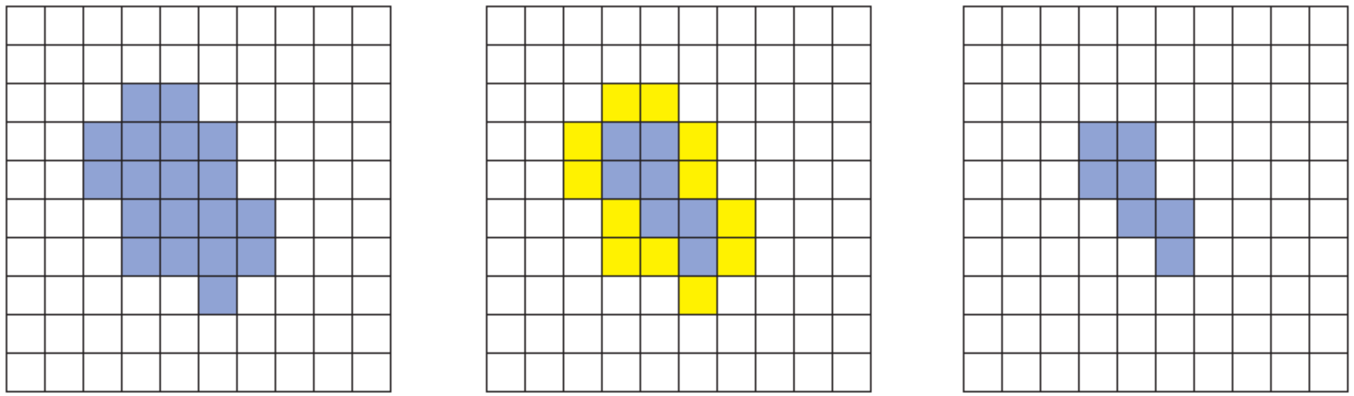
Une contraction isotopique suivie d'une expansion **isotropique** permet de retirer les petits objets (ex. : cercle rose)

(Burger 2009, Vol1, Fig 7.2)



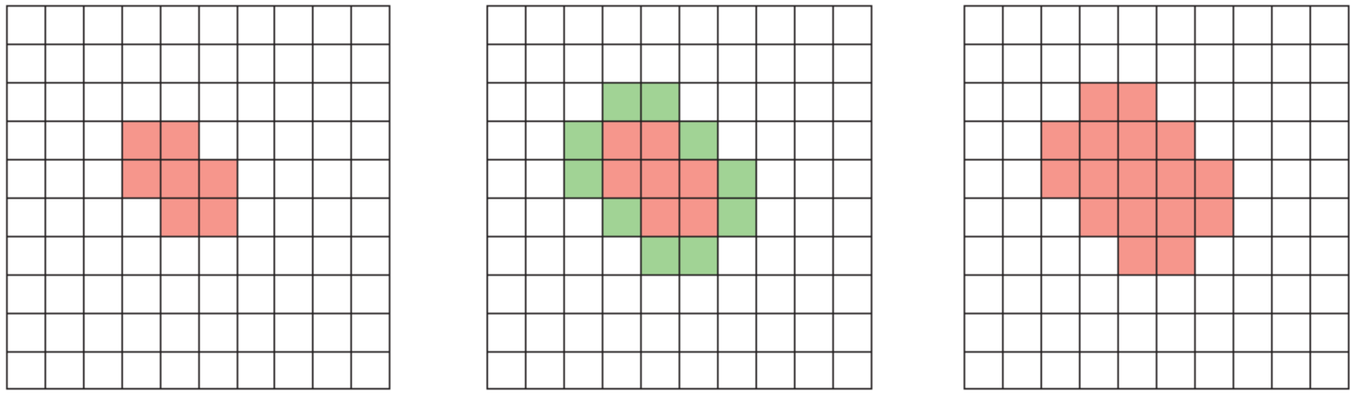
Représentation discrète de la contraction et de l’expansion

- Contraction



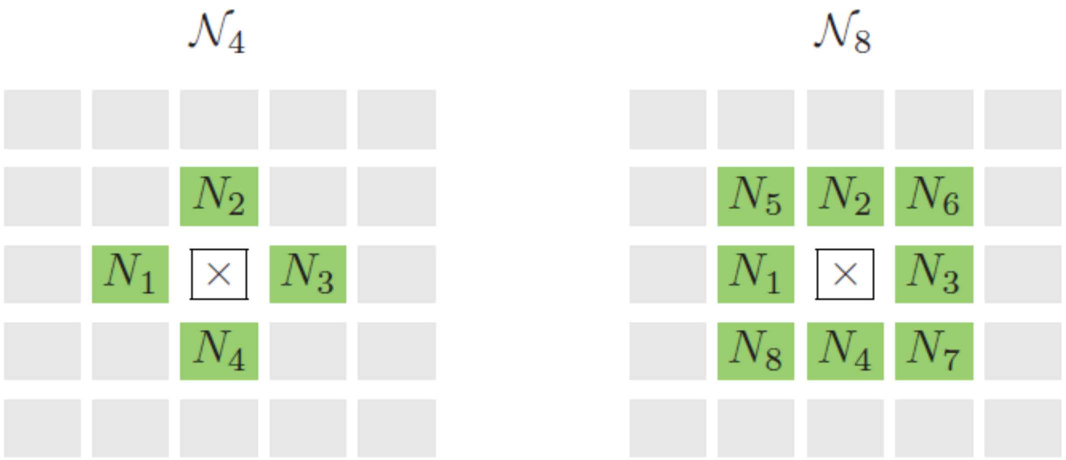
- Expansion

(Burger 2009, Vol1, Fig 7.3, 7.4)



Rappel : Voisinage d'un pixel

- Pour l'expansion / contraction, on a besoin de définir ce que sont 2 pixels adjacents
- 2 définitions de voisinage :
  - N_4 : **4 pixels adjacents** (horizontal et vertical)
  - N_8: **8 pixels adjacents** (horizontal, vertical, diagonal)



(Burger 2009, Vol1 Fig 7.5)

Opérations morphologiques de base

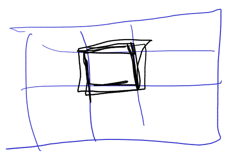
- **Opérations de base** : Contraction et Expansion
- Filtres morphologiques correspondants :
  - **Érosion** morphologique (contraction)
  - **Dilatation** morphologique (expansion)
- L'érosion et la dilation sont **plus générales**, et permettent de réaliser des opérations complexes en les combinant.

Élément structurant

- Similaire au noyau d'un filtre linéaire
- Décrit la **forme** de l'opération morphologique
- Formé de valeurs 0 et 1 uniquement

$$H(i,j) \in \{0,1\}$$

- Le **point chaud (*hot spot*)** est l'origine du filtre
- L'origine n'est pas nécessairement le centre du filtre et n'a pas nécessairement une valeur = 1



Élément – 0 : Vide
Élément – 1 : ●

(Burger 2009, Vol1 Fig 7.6)

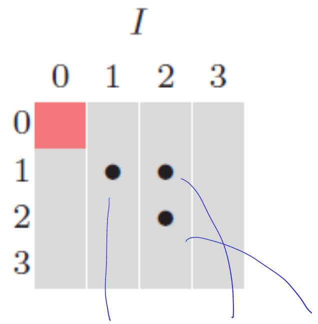
Ensemble de points (1)

- Image binaire $I(u,v) \in \{0,1\}$
- Ensemble des points de l'avant-plan

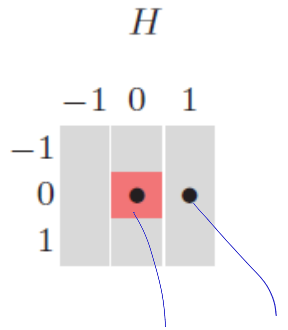
$$\mathcal{Q}_I = \{\boldsymbol{p} \mid I(\boldsymbol{p}) = 1\}$$

Où $\boldsymbol{p} = (u,v)$ est la coordonnée d'un pixel

- Autant l'image et que l'élément structurant peuvent être représentés par un ensemble $\mathcal{Q}$



$I \equiv \mathcal{Q}_I = \{(1,1),(2,1),(2,2)\}$



$H \equiv \mathcal{Q}_H = \{(0,0),(1,0)\}$

Ensemble de points (2)

- Notation utile pour représenter des opérations morphologiques fondamentales sur les images binaires
- **L'inversion** d'une image binaire $I \rightarrow \bar{I} = I^c$ (*p. ex. échanger l'arrière-plan et l'avant-plan)* est équivalent à construire **l'ensemble complémentaire**:

tous les points →
(0,0), (0,1), ...
(1,0), (2,0) ...

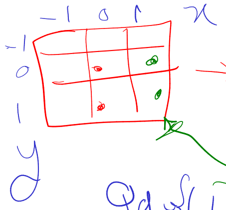
$$\mathcal{Q}_{\bar{I}} = \bar{\mathcal{Q}}_I = \{\boldsymbol{p} \in \mathbb{Z}^2 \mid \boldsymbol{p} \notin \mathcal{Q}_I\}$$

- La combinaison d'images avec **l'opération *OU*** *(OR)* est **l'union** des ensembles :

$$\mathcal{Q}_{I_1 \vee I_2} = \mathcal{Q}_{I_1} \cup \mathcal{Q}_{I_2}$$

Translation (*décalage*) d'une image binaire *Ensemble de points*

- Une translation d'une image binaire par un vecteur $\boldsymbol{d}$ résulte en une nouvelle image



$Q_I = \{(0,0), (0,1)\}$

$d = (1,0)$

$$I_{\boldsymbol{d}}(\boldsymbol{p}+\boldsymbol{d}) = I(\boldsymbol{p})$$

$Q_d = \{(1,0), (1,1)\}$

si c'est plus que 1 on va ajouter un ligne

- On peut représenter la **translation** comme une opération sur l'ensemble de points Q

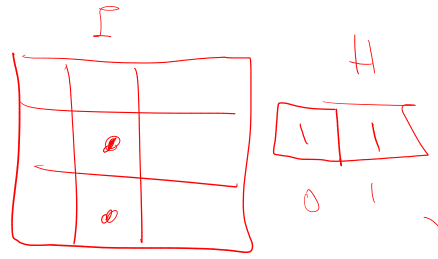
$Q_R = \{(0,0), (0,-1)\}$

$$I_{\boldsymbol{d}} \equiv \{(\boldsymbol{p}+\boldsymbol{d}) | \boldsymbol{p} \in I\}$$

- Une **réflexion** (effet miroir) d'une image binaire par rapport à son origine est :

$$H^* \equiv \{-\boldsymbol{p} \mid \boldsymbol{p} \in H\}$$

Dilatation morphologique



$Q_I = \{(0,0),(0,1)\}$ $Q_H = \{(0,0)(0,1)\}$

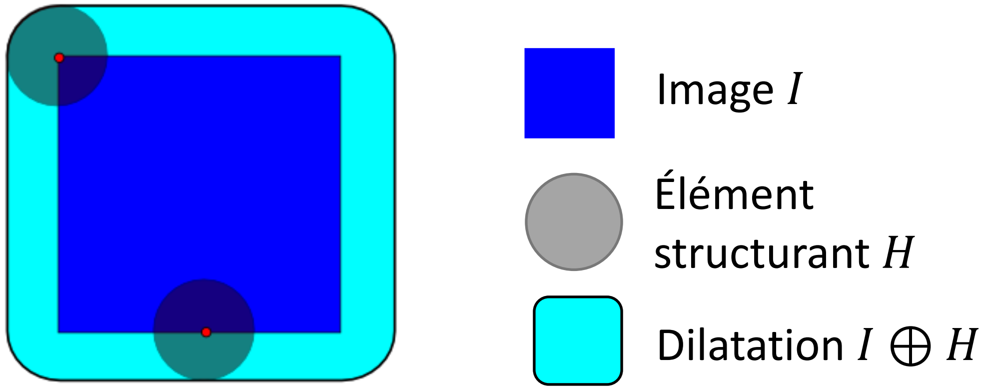
$Q_D = \{(0,0),(0,1),(0,1),(0,2)\}$

- Représente la **croissance** d'une région

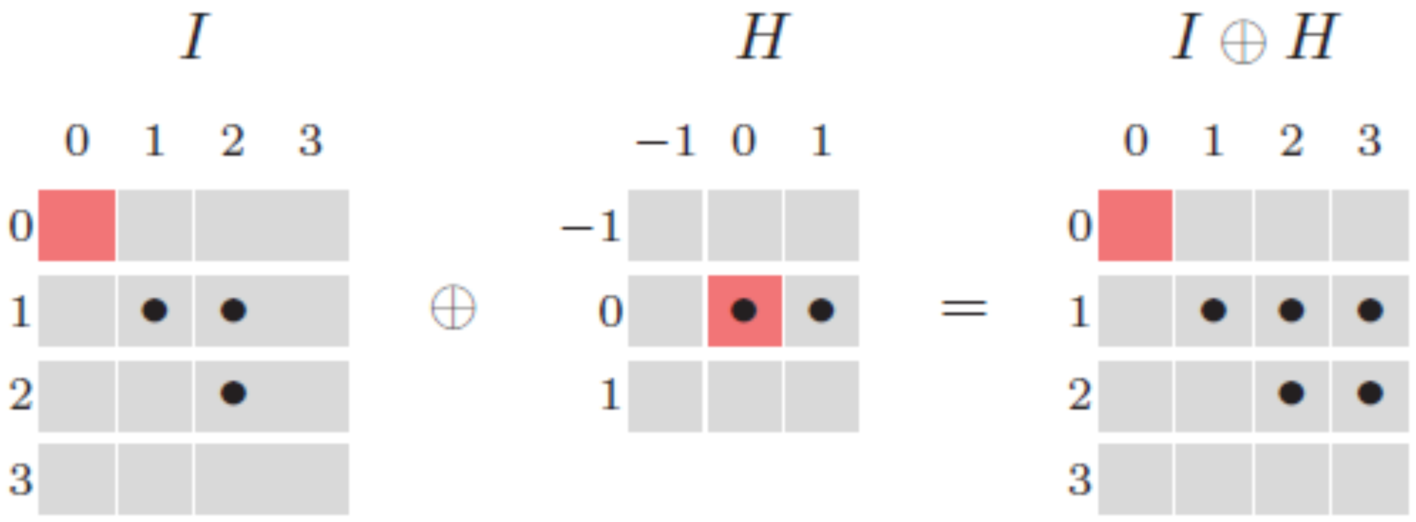
$$I \oplus H \equiv \{(\boldsymbol{p} + \boldsymbol{q}) \mid \text{pour tout } \boldsymbol{p} \in I, \boldsymbol{q} \in H\}$$

- Le résultat d'une dilatation est la somme (vectorielle) de toutes les paires de coordonnées possibles entre l'image I et l'élément H

Wikipedia



Exemple : Dilatation morphologique



(Burger 2009, Vol1 Fig 7.8)

$$I \equiv \{(1,1),(2,1),(2,2)\},\ H \equiv \{(0,0),(1,0)\}$$

$$\begin{aligned} I \oplus H \equiv \{\ & (1,1)+(0,0),\ (1,1)+(1,0), \\ & (2,1)+(0,0),\ (2,1)+(1,0), \\ & (2,2)+(0,0),\ (2,2)+(1,0)\ \} \end{aligned}$$

5 points

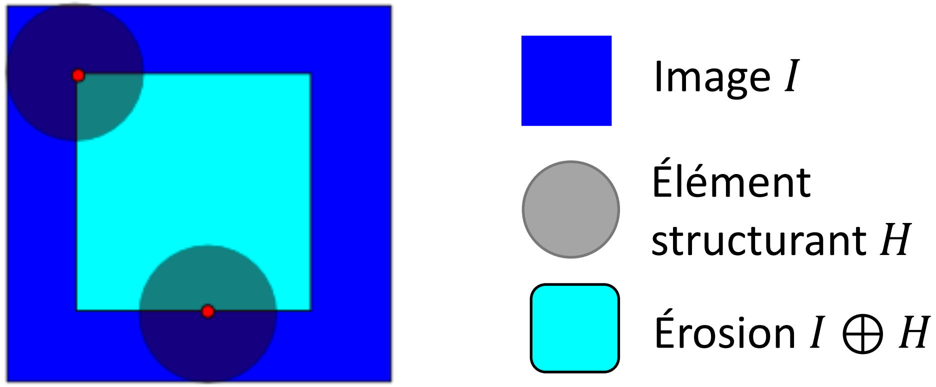
Érosion morphologique

- **Inverse de la dilatation**

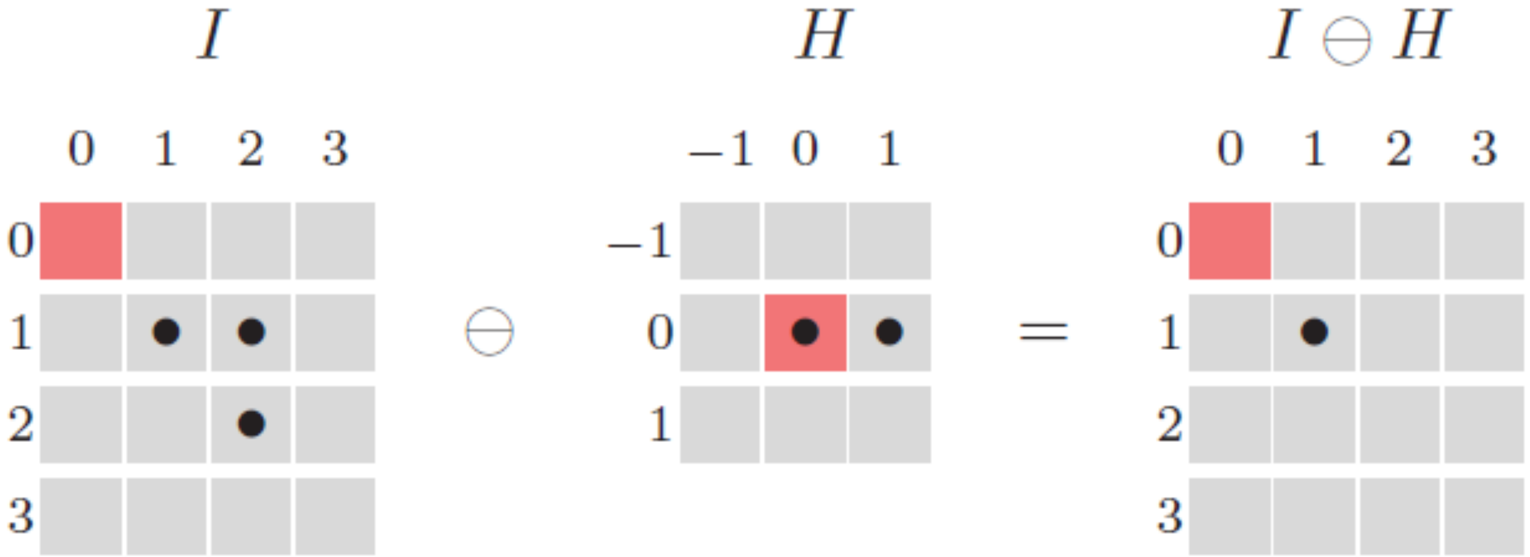
$$I \ominus H \equiv \{\boldsymbol{p} \in \mathbb{Z}^2 \mid (\boldsymbol{p} + \boldsymbol{q}) \in I, \text{pour tout } \boldsymbol{q} \in H\}$$

- Un point $\boldsymbol{p}$ est inclus dans le résultat **si (et seulement si)** l'élément structurant H, lorsqu'il est placé sur $\boldsymbol{p}$, est entièrement contenu dans l'avant-plan de l'image originale

[Wikipedia](#)



Exemple : Érosion morphologique



(Burger 2009, Vol1 Fig 7.9)

$$I \equiv \{(1,1),(2,1),(2,2)\},\ H \equiv \{(\mathbf{0},\mathbf{0}),(\mathbf{1},\mathbf{0})\}$$

$$I \ominus H \equiv \{\,(1,1)\,\}\ \text{because}$$

$$(1,1)+(\mathbf{0},\mathbf{0})=(1,1)\in I \quad \textbf{and} \quad (1,1)+(\mathbf{1},\mathbf{0})=(2,1)\in I$$

Dilatation et érosion : Propriétés (1)

- L’opération de dilatation est **commutative**

$$I \oplus H = H \oplus I$$

- La dilatation est **associative**

$$(I_1 \oplus I_2) \oplus I_3 = I_1 \oplus (I_2 \oplus I_3)$$

- On peut donc remplacer une dilatation à l’aide d’un **grand élément** structurant par **plusieurs dilatations** à l’aide d’éléments structurants plus petits

$$I \oplus H_{grand} = \left(\cdots\left((I \oplus H_1) \oplus H_2\right) \oplus \cdots \oplus H_K\right)$$

Dilatation et érosion : Propriétés (2)

- Un **élément neutre δ** pour une dilatation est

$$I \oplus \delta = \delta \oplus I = I \text{ avec } \delta \equiv \{(0,0)\}$$

- Une érosion **n'est pas** commutative

$$I \ominus H \neq H \ominus I$$

- Une **combinaison** de dilatations et d'érosions est ***associative***

$$(I_1 \ominus I_2) \oplus I_3 = I_1 \ominus (I_2 \oplus I_3)$$

Dilatation et érosion : Propriétés (3)

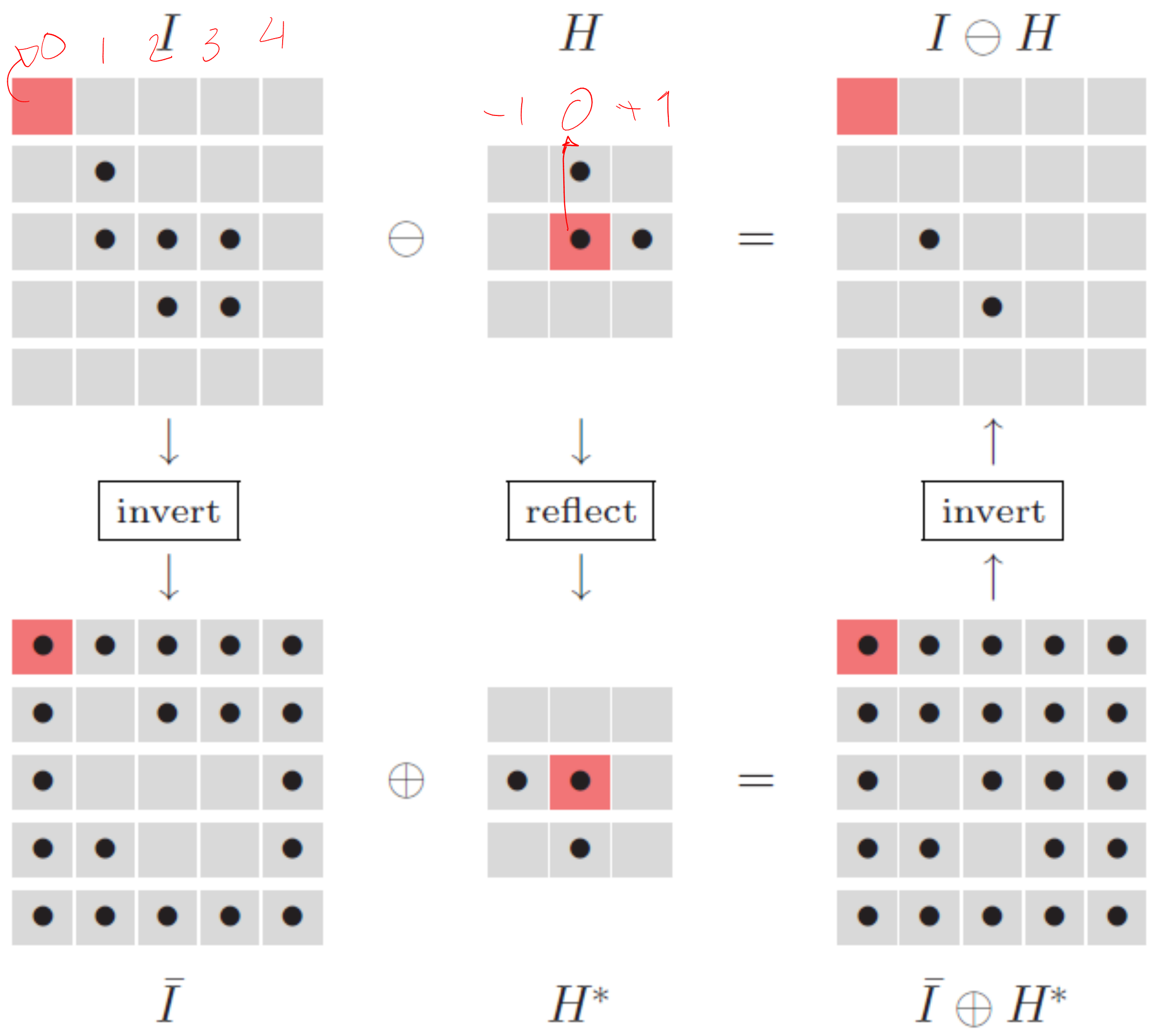
- La dilation et l’érosion **ne sont pas** mutuellement inverses (c.-à-d. une dilatation ne peut pas être inversée par une érosion subséquente)
- La dilation et l’érosion sont liées par :

$$I \oplus H \equiv \overline{(\bar{I} \ominus H^{*})}$$

$$I \ominus H \equiv \overline{(\bar{I} \oplus H^{*})}$$

- **Une dilation de l’avant-plan est équivalente à une érosion de l’arrière-plan qui est ensuite inversée, en utilisant H^* (la réflexion de H)**

Exemple : Lien entre dilatation et érosion

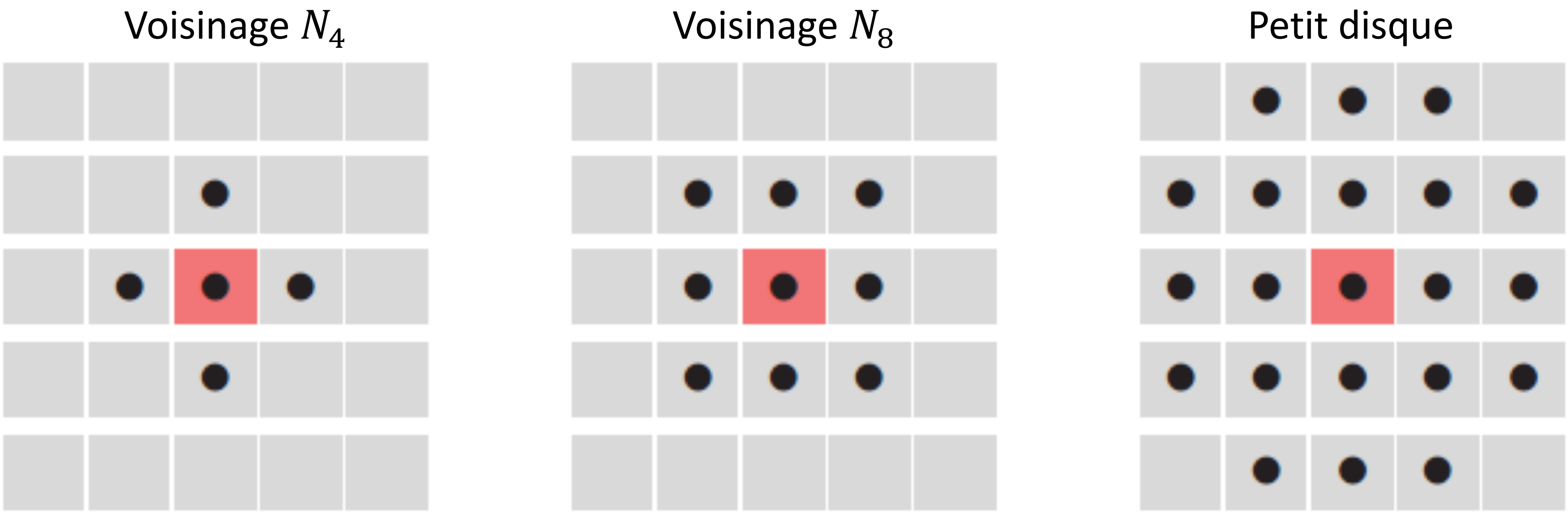


(Burger 2009, Vol1 Fig 7.10)

Conception d’un filtre morphologique

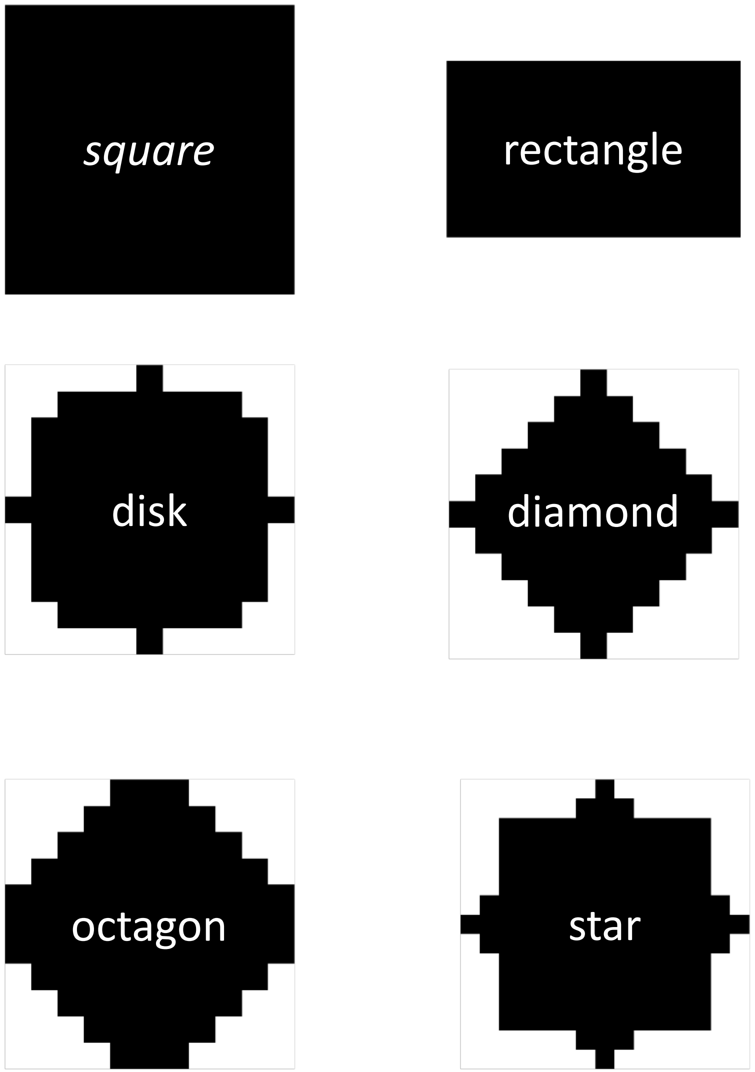
- Pour utiliser un filtre morphologique, il faut spécifier :
  - Le **type d’opérations** (dilatation, érosion, …)
  - La forme et le contenu de **l’élément structurant** H
- La taille et la forme de H dépendent de l’application

(Burger 2009, Vol1 Fig 7.11)



Exemples d’éléments structurants

- skimage.morphology.square
- skimage.morphology.rectangle
- skimage.morphology.diamond
- skimage.morphology.disk
- skimage.morphology.cube
- skimage.morphology.octahedron
- skimage.morphology.ball
- skimage.morphology.octagon
- skimage.morphology.star



Application : Extraction d'un contour (1)

- **But :** Extraire le **contour** d'un **objet segmenté**
- On doit choisir un type de voisinage (N_4 ou N_8)
- Algorithme en 2 étapes
  1. Érosion de l'image originale pour retirer les pixels de contours de l'avant-plan

$$I' = I \ominus H_n$$

  2. Intersection de l'image originale avec l'inverse de l'érosion effectuée à l'étape 1

$$B = I \cap \overline{I'} = I \cap \overline{(I \ominus H_n)}$$

- **Note :** un élément structurant de type N_4 produit un contour de type N_8, et vice versa

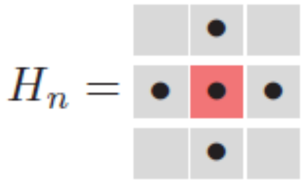
$N8 \longrightarrow N4$

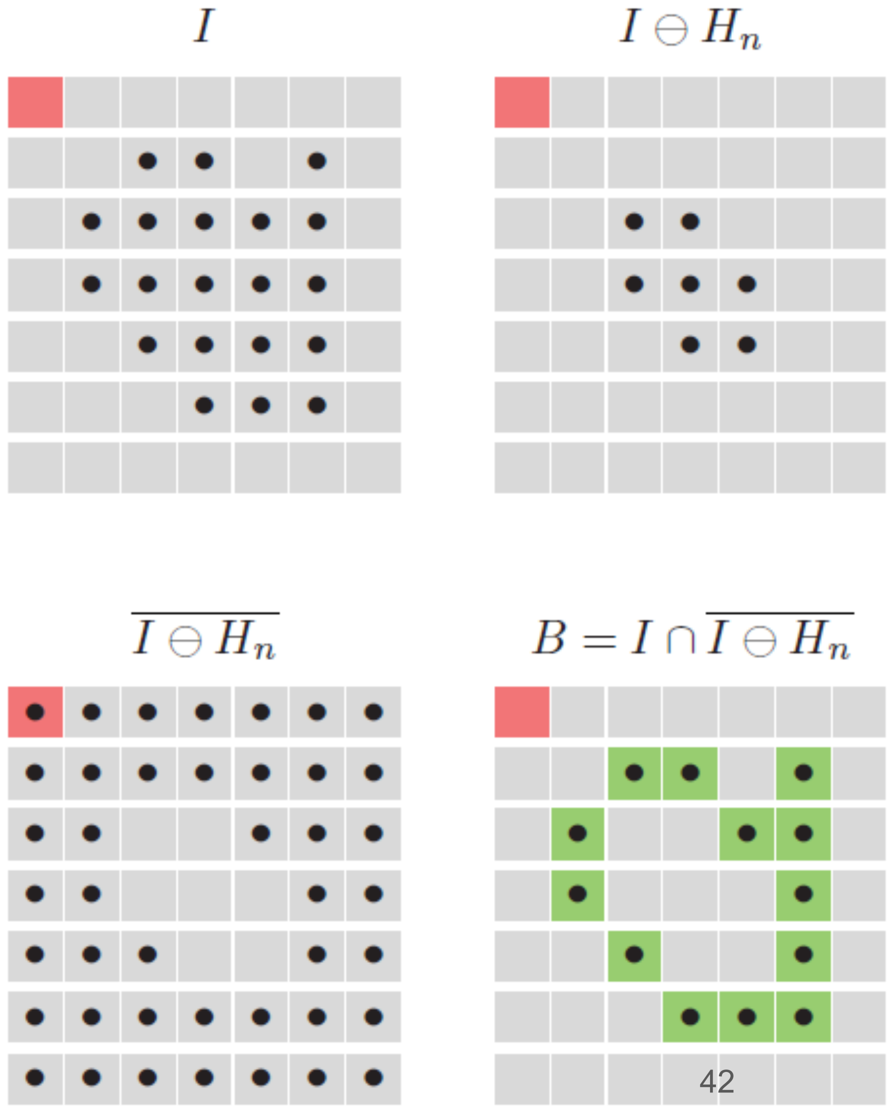
$N4 \longrightarrow N8$

Application : Extraction d’un contour (2)

- On peut utiliser un ou exclusif (XOR) pour obtenir le même résultat

$$B(u,v) = \mathrm{XOR}\big(I(u,v), I'(u,v)\big)$$

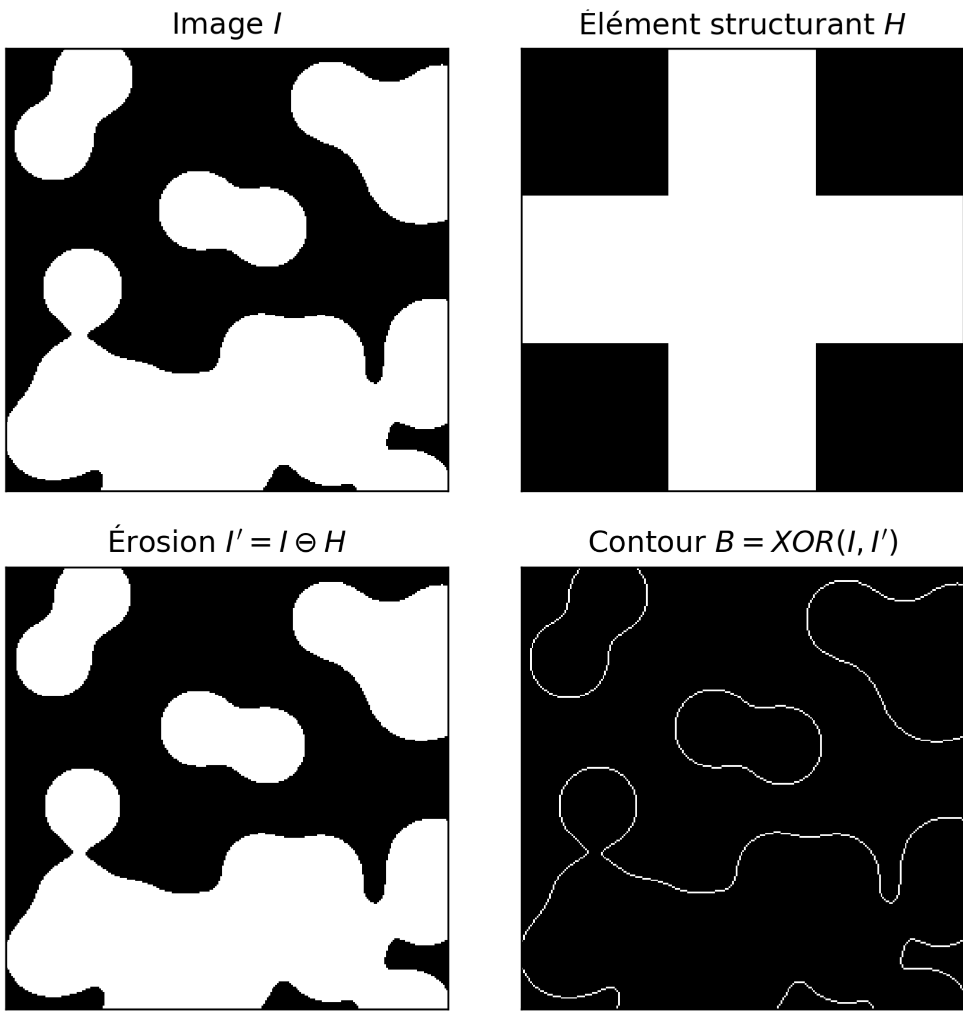




(Burger 2009, Vol1 Fig 7.16)

Application : Extraction d’un contour (3)

```python
import skimage
import numpy as np

# Image binaire à traiter
img = skimage.data.binary_blobs()
img = img[0:256, 0:256]

# Création de l'élément structure H (N4)
H = np.zeros((3,3))
H[0,1] = 1; H[1,0] = 1;
H[2,1] = 1; H[1,2] = 1;
H[1,1] = 1

# Érosion morphologique
img_p = skimage.morphology.binary_erosion(img, H)

# Calcul de l'intersection entre I et I'
B = np.logical_xor(img, img_p)
```



Algorithmes basés sur la morphologie mathématique

Chapitre 6 : Morphologie mathématique

Joël Lefebvre (UQÀM)

INF600F – Traitement d'images

Opérations composites

- **Combinaison** des érosions / dilatations
- Exploitation de la dualité de ces transformations
- Opérations composites les plus communes
  - **Ouverture morphologique** (*opening*, noté $\circ$)
  - **Fermeture morphologique** (*closing*, noté $\cdot$)
- Probablement les opérations morphologiques les plus utilisées en pratique

Ouverture morphologique

- Désigne une **érosion suivie d'une dilatation** avec le même élément structurant H

$$I \circ H = (I \ominus H) \oplus H$$

- Tous les éléments d'avant-plan plus petits qu'une certaine taille sont éliminés de l'image. La forme des objets restants est arrondie par la dilatation.

Wikipedia

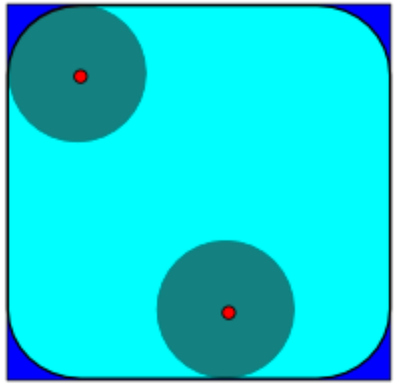


Image I

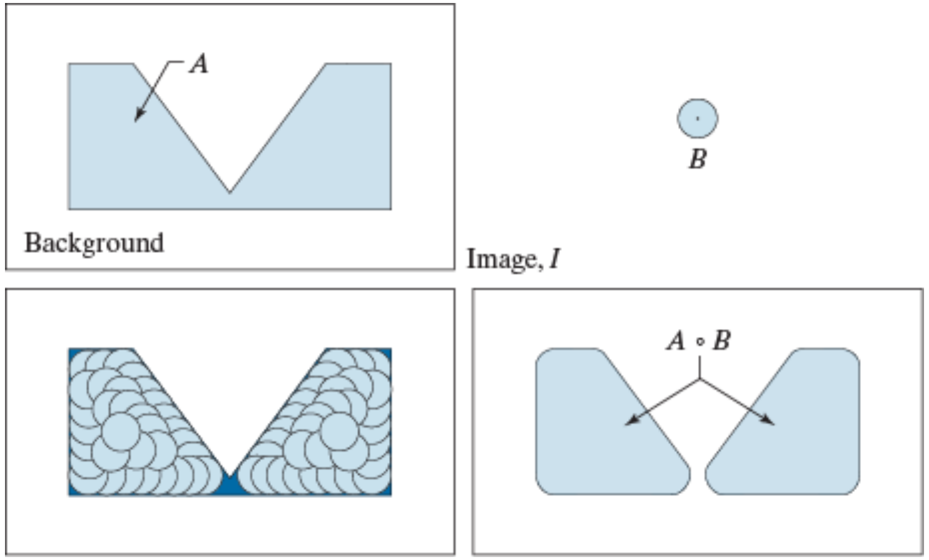
Élément structurant H

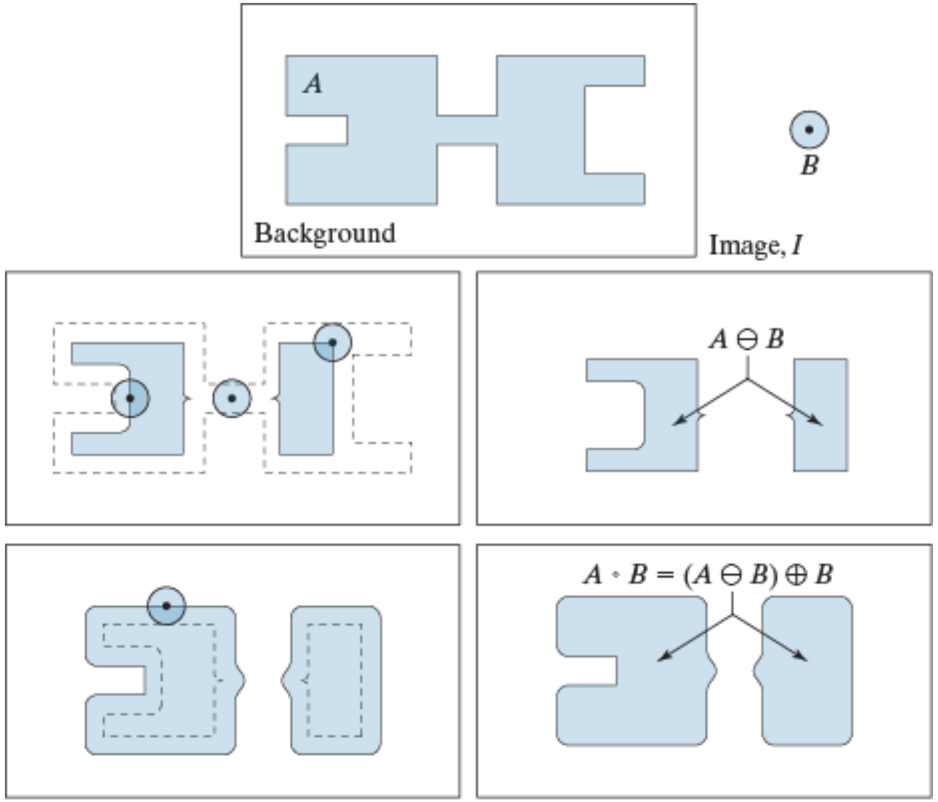
Ouverture $I \circ H$

Exemple : Ouverture morphologique

$$I \circ H = (I \ominus H) \oplus H$$

(Gonzalez & Woods, Fig 9.8)
Image I et élément structurant B.
Ouverture morphologique





(Gonzalez & Woods, Fig 9.10)
Image I et élément structurant B.
Étapes d’une ouverture morphologique

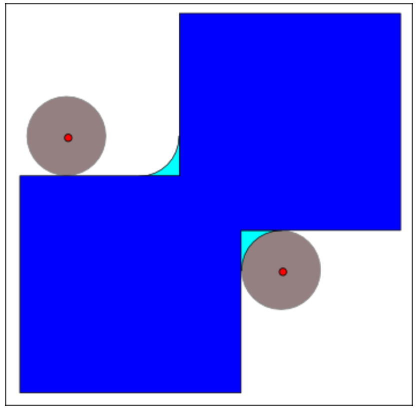
Fermeture morphologique

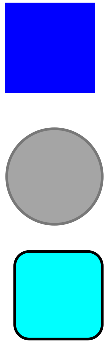
- Désigne une **dilatation suivie d'une érosion** avec le même élément structurant H

$$I \cdot H = (I \oplus H) \ominus H$$

- **Retire (*ferme*) les trous et fissures** dans l'avant-plan qui sont plus petits que l'élément structurant.

[Wikipedia](https://en.wikipedia.org)

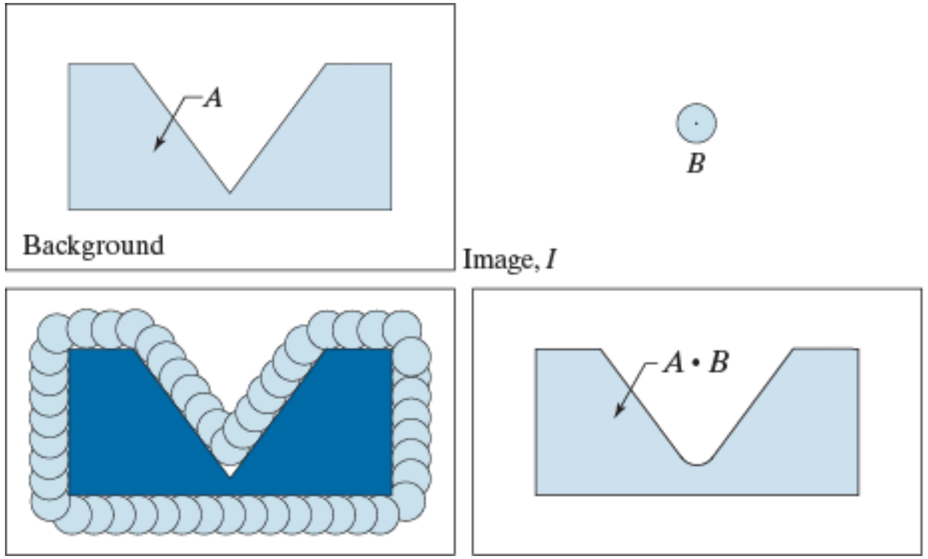


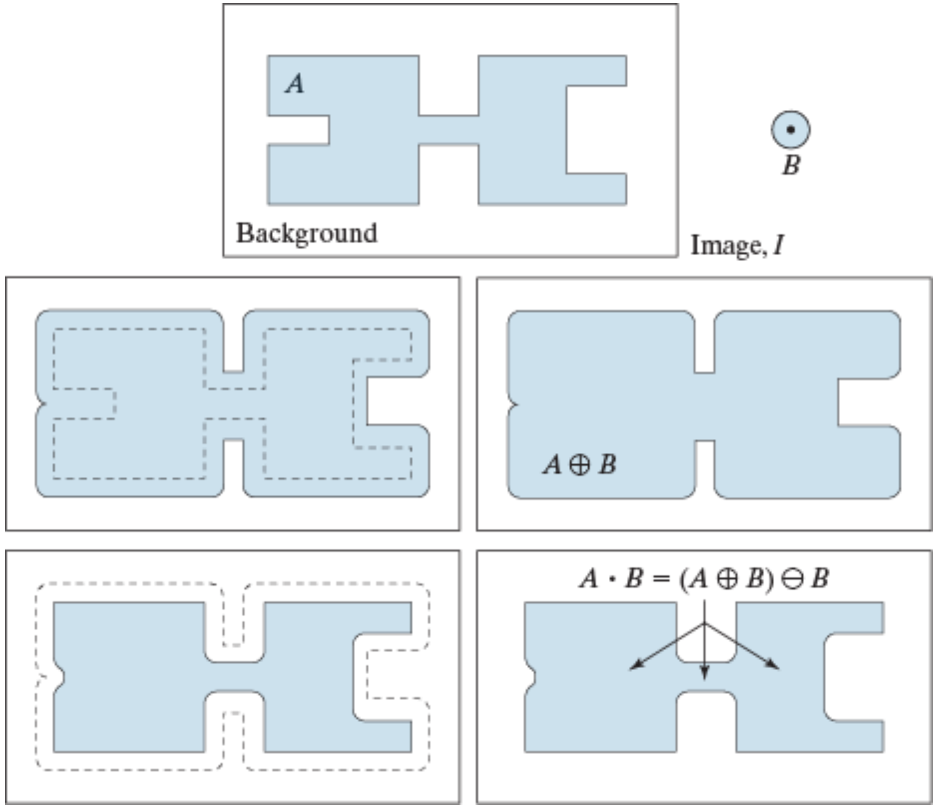


Exemple : Fermeture morphologique

$$I \cdot H = (I \oplus H) \ominus H$$

(Gonzalez & Woods, Fig 9.9)
Image I et élément structurant B.
Fermeture morphologique





(Gonzalez & Woods, Fig 9.10)
Image I et élément structurant B.
Étapes d'une fermeture morphologique

Application : Filtrage par ouverture et fermeture morphologiques (1)

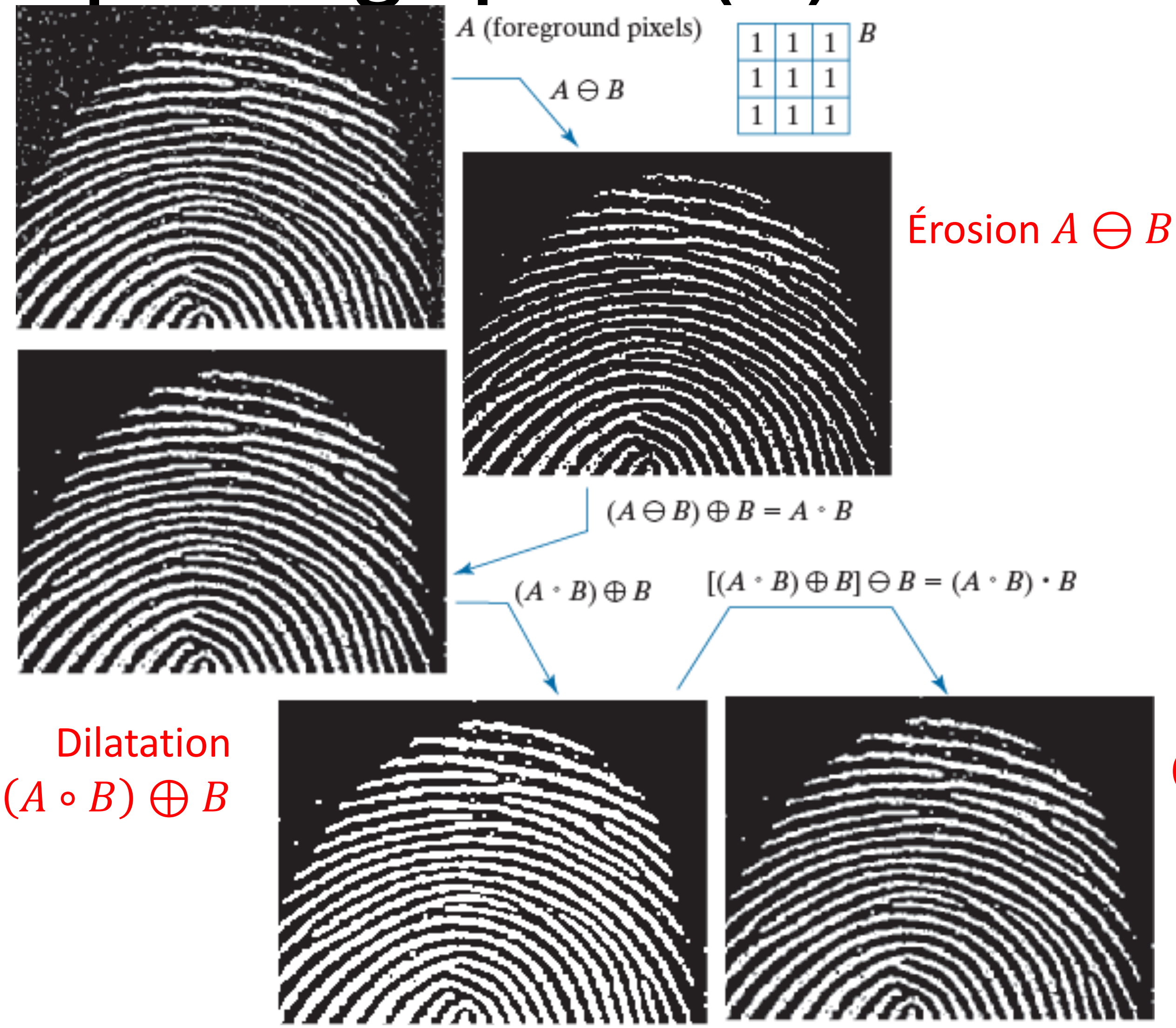
Image à corriger. Empreinte digitale bruitée

Érosion $A \ominus B$

Ouverture $A \circ B$

Dilatation $(A \circ B) \oplus B$

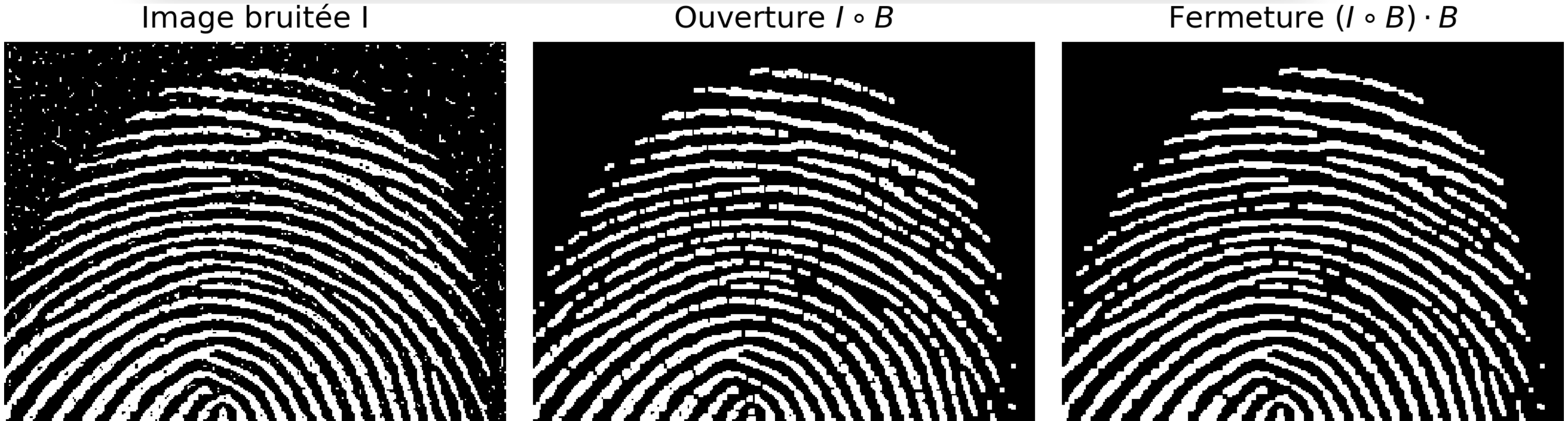
Fermeture $(A \circ B) \cdot B$



(Gonzalez & Woods, Fig 9.11)

Application : Filtrage par ouverture et fermeture morphologiques (2)

```python
# Élément structurant
B = np.ones((3,3))

# Ouverture
img_p = skimage.morphology.binary_opening(img, B)

# Fermeture
img_p2 = skimage.morphology.binary_closing(img_p, B)
```



Ouverture et fermeture : Propriétés

- L'ouverture et la fermeture sont **idempotentes**, c.-à-d. que leur résultat est **final**.

Si tu change pas ⇒ la meme resultat

- Des applications successives de l'ouverture ne changent pas le résultat

$$I \circ H = (I \circ H) \circ H = \big((I \circ H) \circ H\big) \circ H = \cdots$$

- *Idem* pour la fermeture
- *L'ouverture et la fermeture sont des **opérations duales***

$$I \circ H = \overline{(\bar{I} \cdot H)} \text{ et } I \cdot H = \overline{(\bar{I} \circ H)}$$

Transformation *Hit-or-Miss* (HMT) (1)

- Outil de base en **détection de formes**
- Soit I une image binaire formée de pixels d'avant-plan A et d'arrière-plan $\bar{A}$
- Utilisation de **2 éléments structurants**
  - B_1 : Forme dans l'avant-plan ***(Hit)***
  - B_2 : Forme dans l'arrière-plan ***(Miss)***
- Transformation HMT :

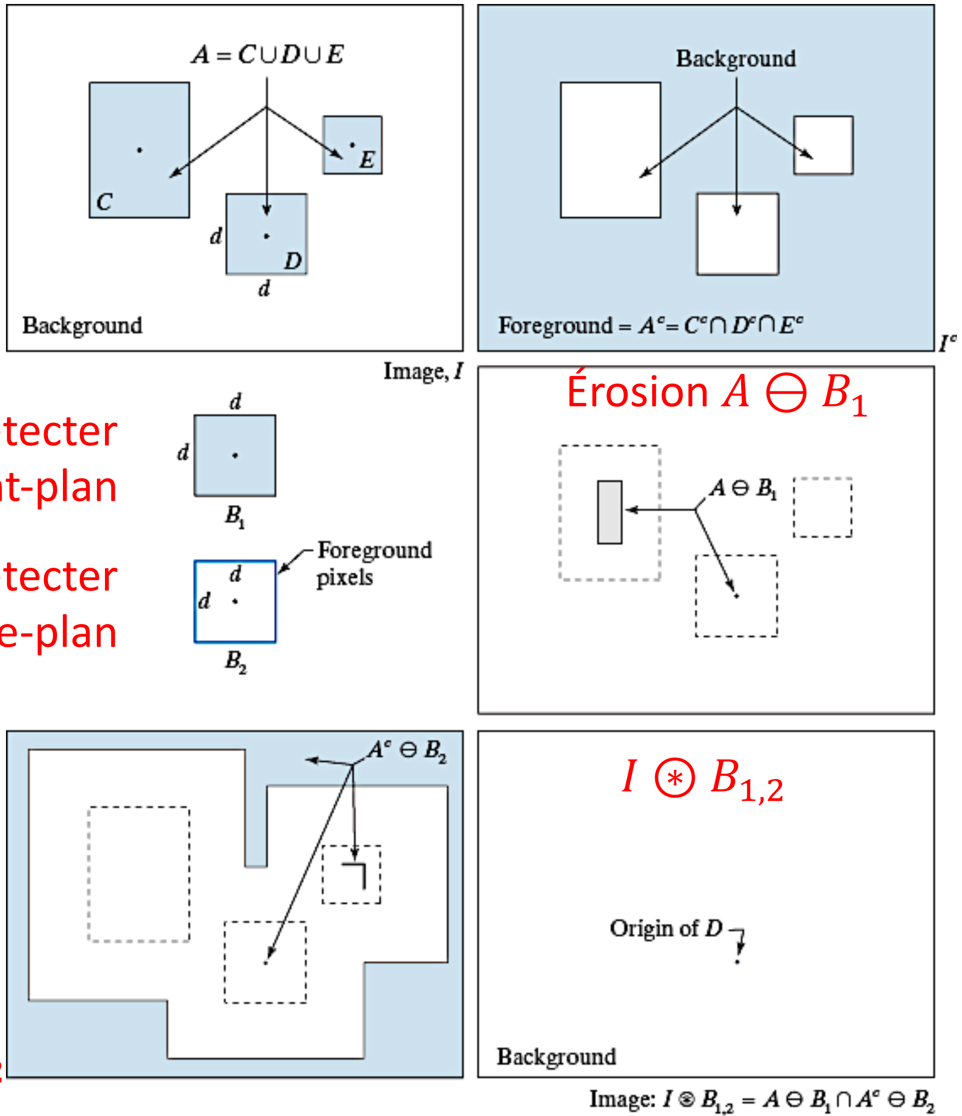
$$\begin{aligned} I \circledast B_{1,2} &= \{z \mid (B_1)_z \subseteq A \text{ et } (B_2)_z \subseteq \bar{A}\} \\ &= (A \ominus B_1) \cap (\bar{A} \ominus B_2) \end{aligned}$$

Transformation *Hit-or-Miss* (HMT) (2)

$$I \circledast B_{1,2} = (A \ominus B_1) \cap (\bar{A} \ominus B_2)$$

B_1 : Forme de l’objet à détecter dans l’avant-plan

B_2 : Forme de l’objet à détecter dans l’arrière-plan



Érosion $\bar{A} \ominus B_2$

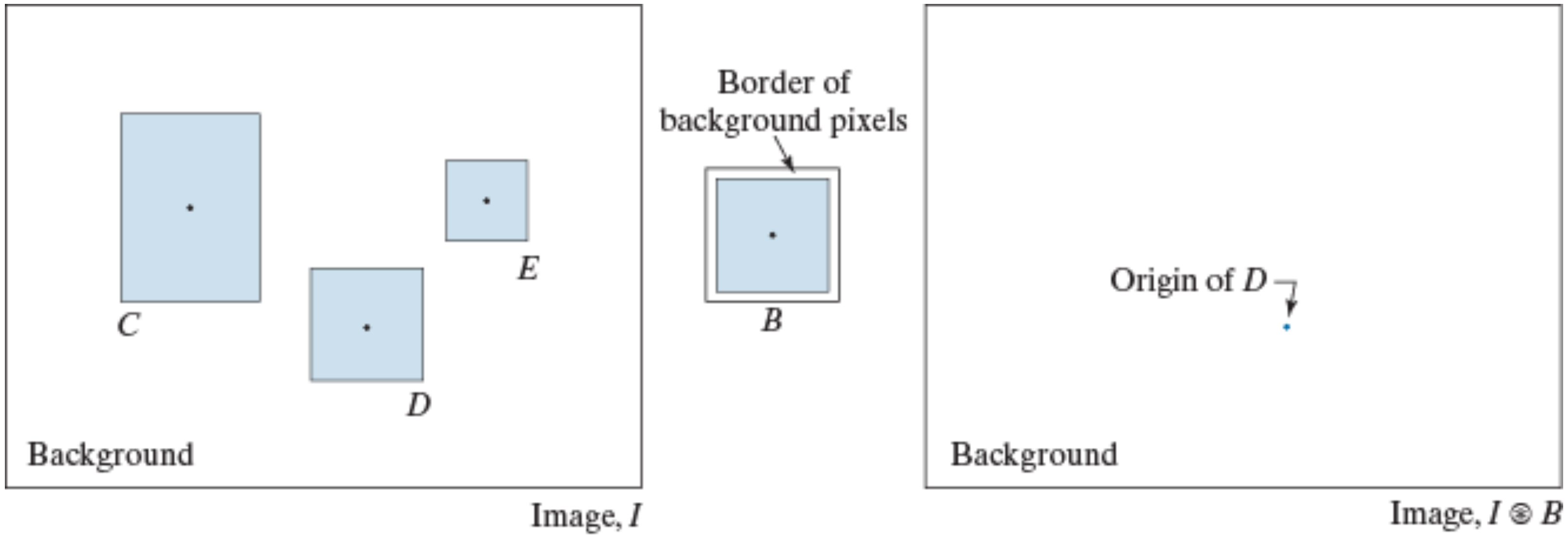
(Gonzalez & Woods, Fig 9.12)

Transformation *Hit-or-Miss* (HMT) (3)

- Combinaison de B_1 et B_2 en un seul élément structurant B
- La transformation devient

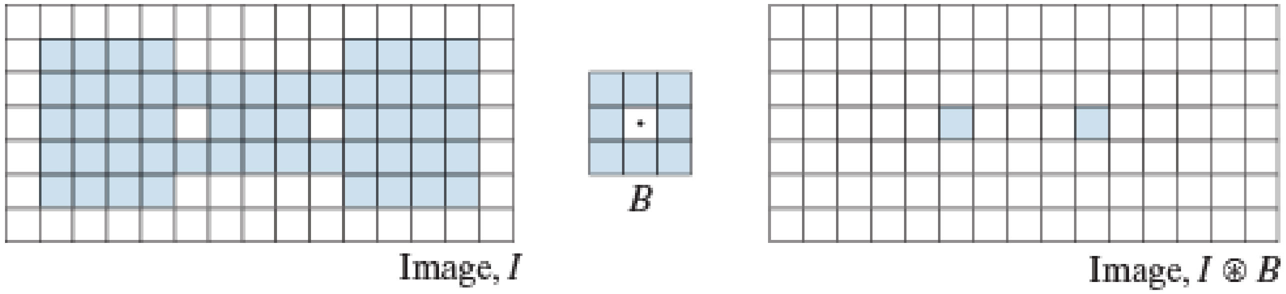
$$I \circledast B = \{z \,|(B)_z \subseteq I\}$$

- Pour chaque position z on vérifie si l'élément structurant est entièrement inclus dans l'image
- N'utilise pas l'érosion morphologique

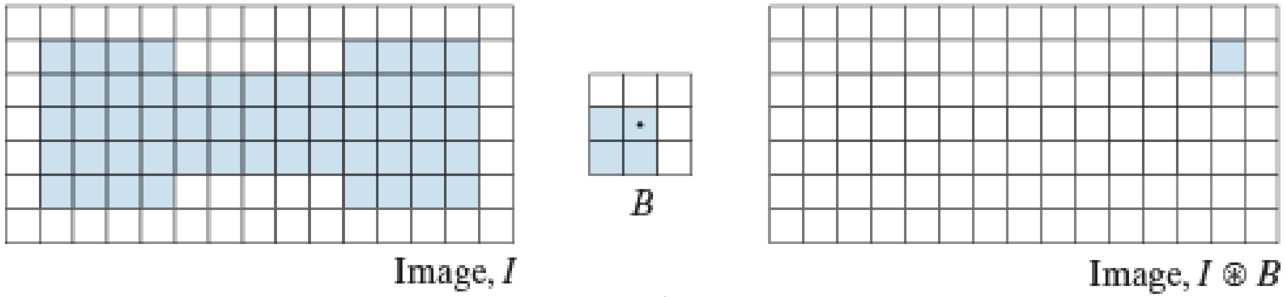


Exemple de détecteurs HMT

Trous de 1 pixel

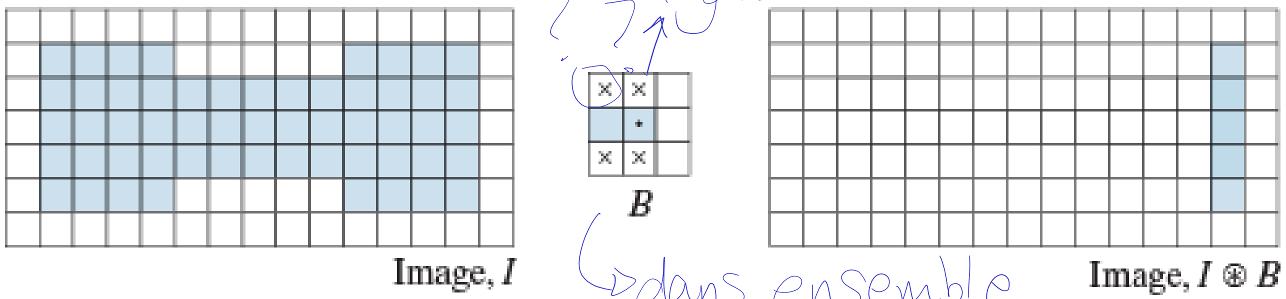


Coin supérieur droit



Bordure droite

? → ignore



↳ dans ensemble on met pas

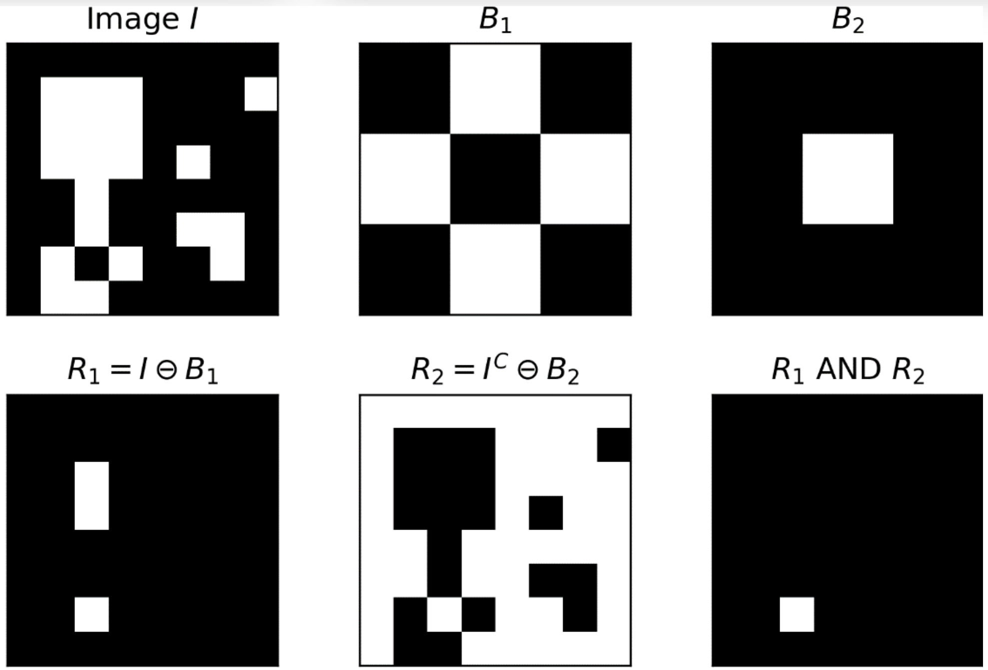
x : Désigne un pixel *non important*, il n'est ni dans B_1, ni dans B_2

Exemple Python pour HMT

```python
# Élément B1 d'avant-plan (Hit)
B1 = np.zeros((3,3))
B1[0,1] = 1; B1[1,0] = 1
B1[1,2] = 1; B1[2,1] = 1

# Élément B2 d'arrière-plan (Miss)
B2 = np.zeros((3,3))
B2[1,1] = 1
```

```python
# 1. Érosion de l'image par B1
R1 = skimage.morphology.binary_erosion(img, B1)

# 2. Érosion de l'inverse de l'image par B2
R2 = skimage.morphology.binary_erosion(~img, B2)

# Intersection entre R1 et R2
R = R1 & R2
```

Détection de trous de 1 pixel



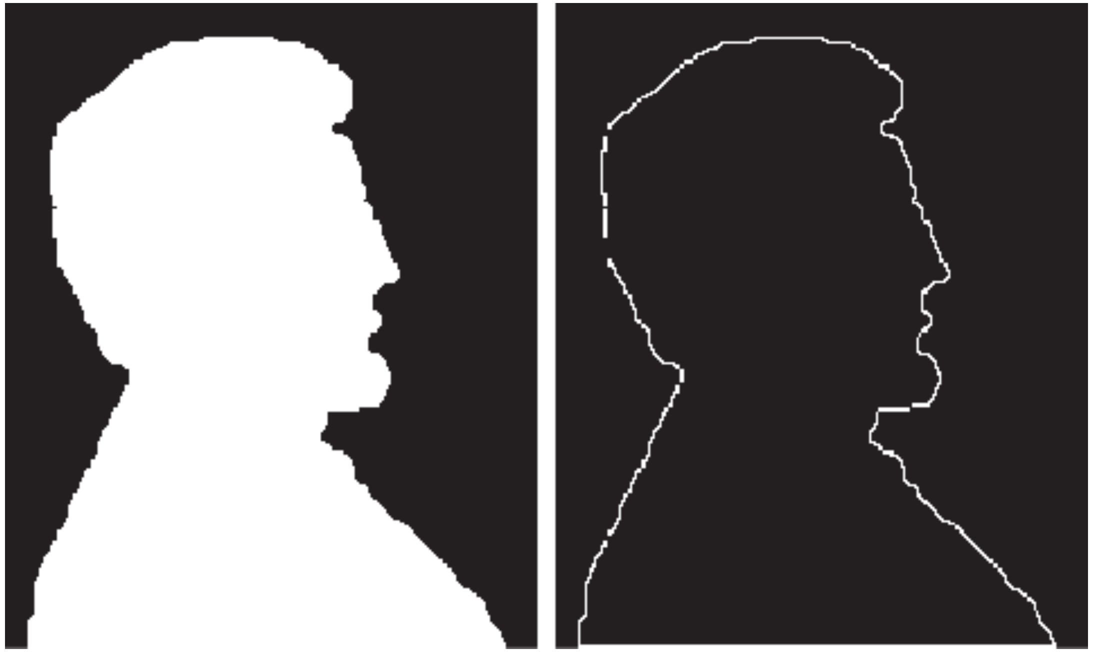
Quelques algorithmes morphologiques

- Extraction de contours
- Remplissage de trous (*Hole filling)*
- Extraction des composants connectés
- Enveloppe convexe (*Convex Hull*)
- Amincissement (*Thinning)*
- Élargissement (*Thickening*)
- Squelettisation

Application : Extraction de contours

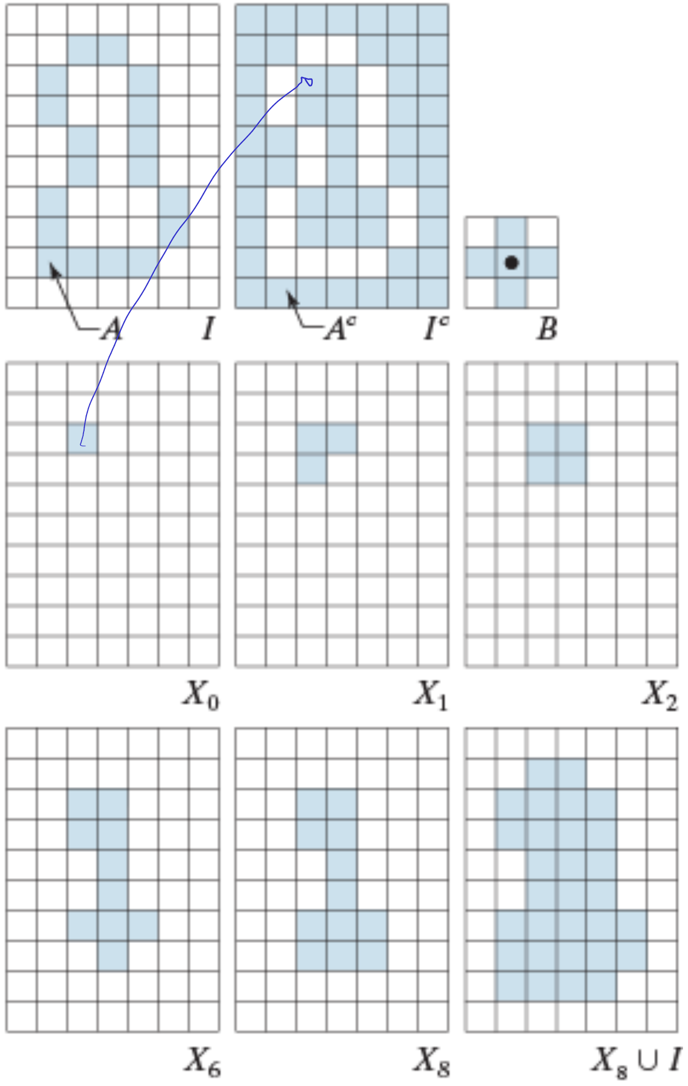
- Intersection entre l'image et son érosion

$$B = I - (I \ominus H)$$



Application : Remplissage de trous (1)

- Dilatation conditionnelle
$$X_k = (X_{k-1} \oplus B) \cap I^c$$
- **Condition initiale :** X_0 est une matrice nulle à l'exception des pixels qu'on sait être un trou
- **Critère d'arrêt :** $X_k = X_{k-1}$
- **Image sans trous :**
$$X_k \cup I$$

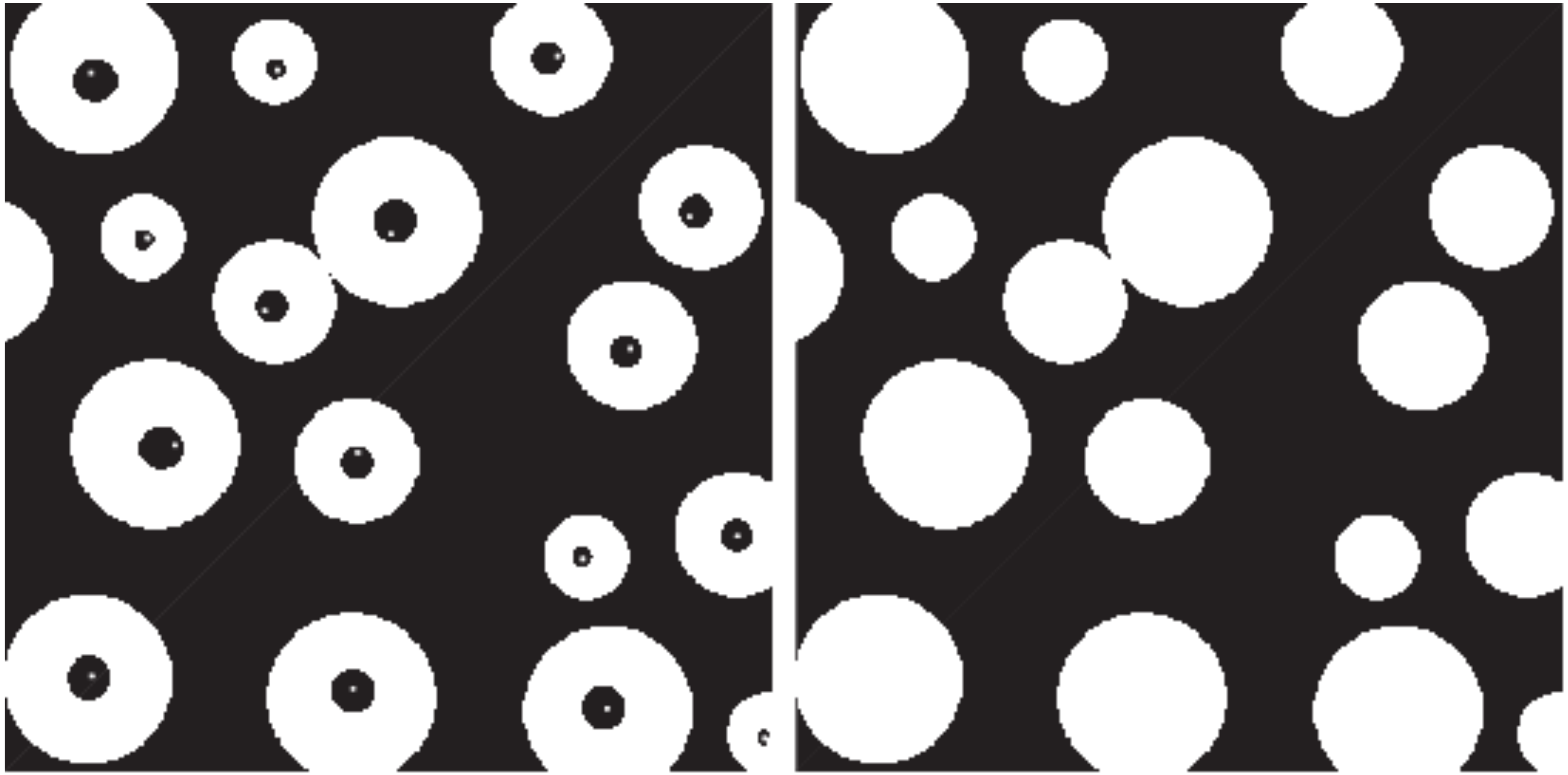


(Gonzalez & Woods, Fig 9.17)

Application : Remplissage de trous (2)

- scipy.ndimage.morphology.binary_fill_holes

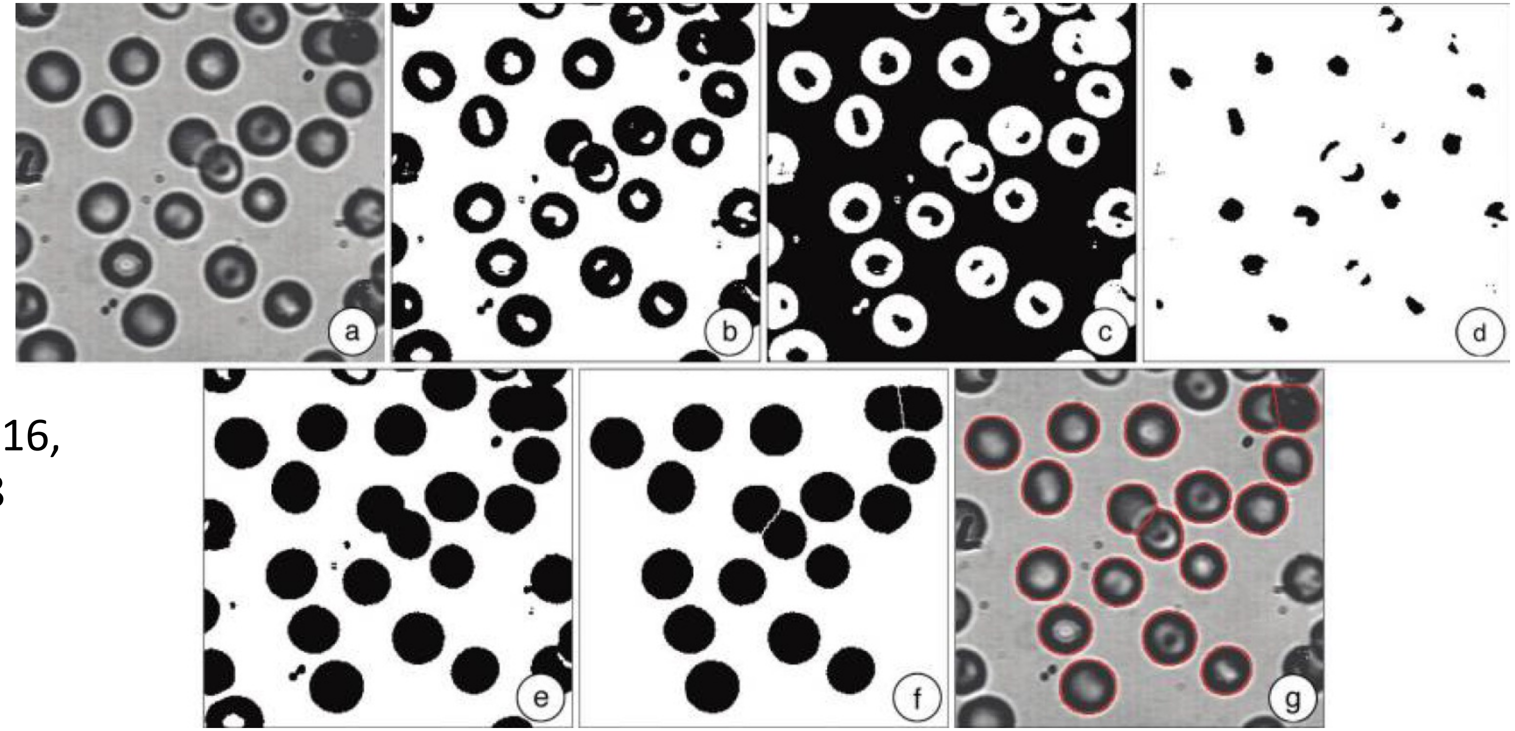
(Gonzalez & Woods, Fig 9.18)



Application : Remplissage de trous (3)

- Détection des globules rouges à partir d'images acquises par microscopie optique.

Image originale | Seuillage global | Inversion | Retire l'arrière-plan touchant la bordure



Ross2016, Fig8.18

Remplissage des trous | Séparation watershed | Résultat

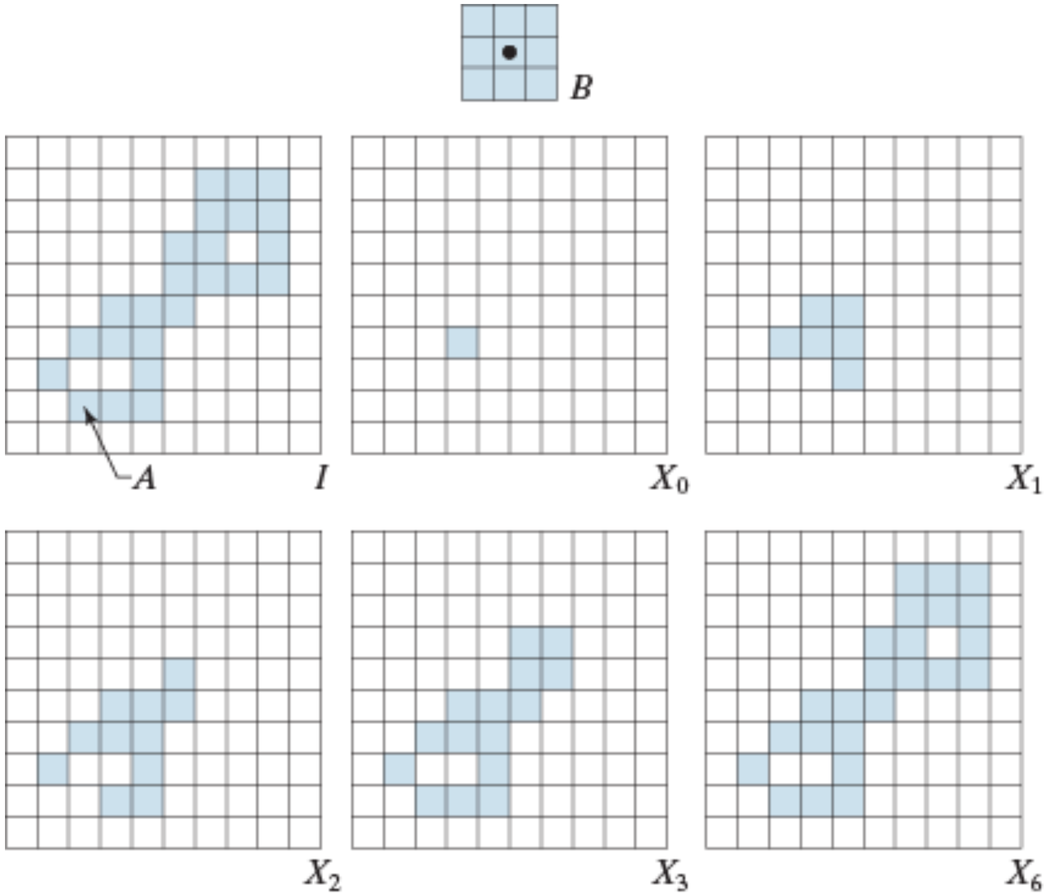
Application : Extraction des composants connectés (1)

- Dilatation conditionnelle

$$X_k = (X_{k-1} \oplus B) \cap I \text{ pour } k = 1, 2, 3, \cdots$$

- **Condition initiale :** X_0 est une matrice nulle à l'exception des pixels qu'on sait être un objet
- **Critère d'arrêt :**

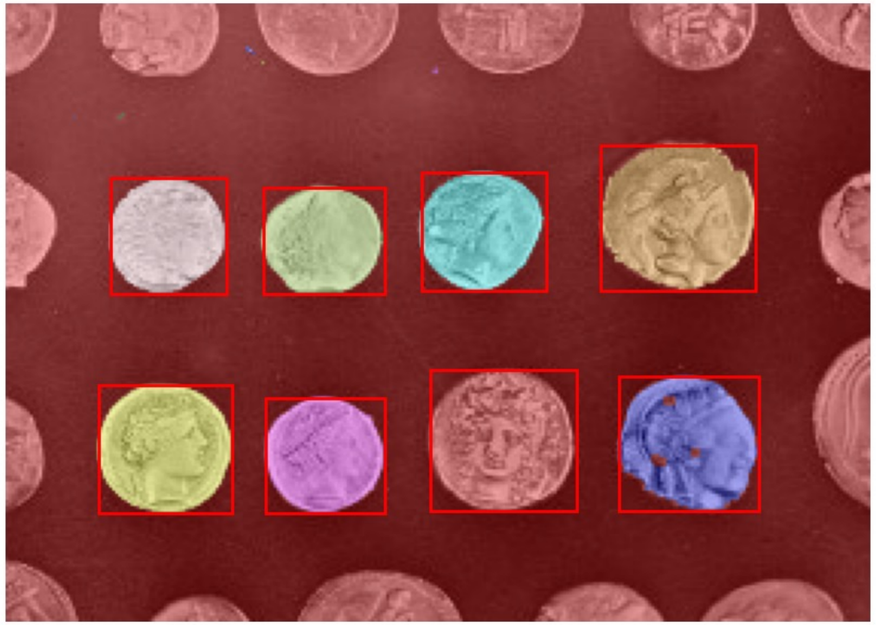
$$X_k = X_{k-1}$$



(Gonzalez & Woods, Fig 9.19)

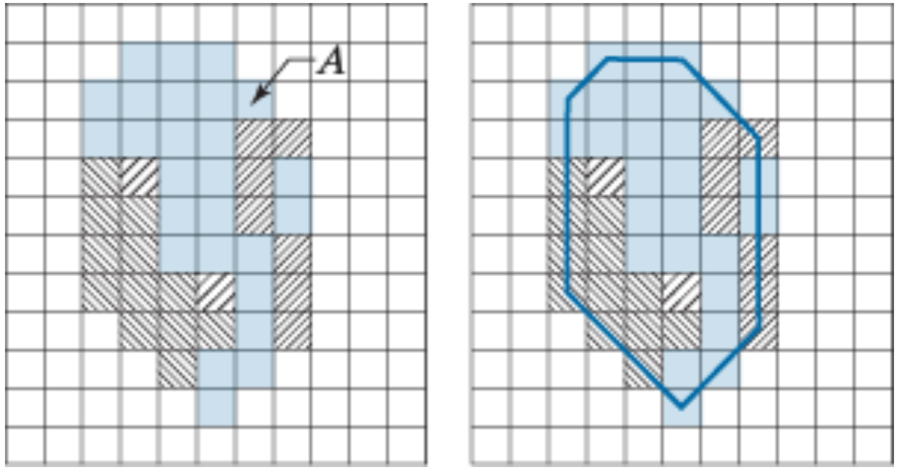
Application : Extraction des composants connectés (2)

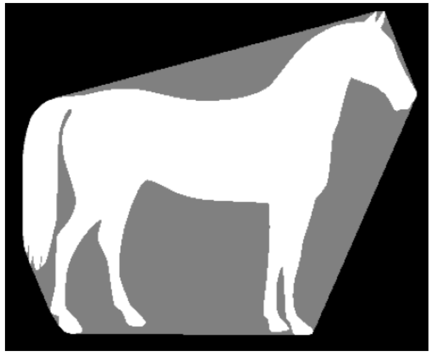
- **Exemple :** Étiqueter des régions d'une image (URL)
- Utilise **skimage.measure.label** qui considère les composants connectés
- Chaque couleur représente un composant connecté différent



Application : Enveloppe convexe

- Trouve **la plus petite forme convexe** permettant **d'envelopper** les pixels de l'avant-plan
- Transformation itérative de type *Hit-or-Miss* conditionnelle avec 4 éléments structurants
- C'est une approximation de l'enveloppe convexe optimale
- Exemple Python : [URL]





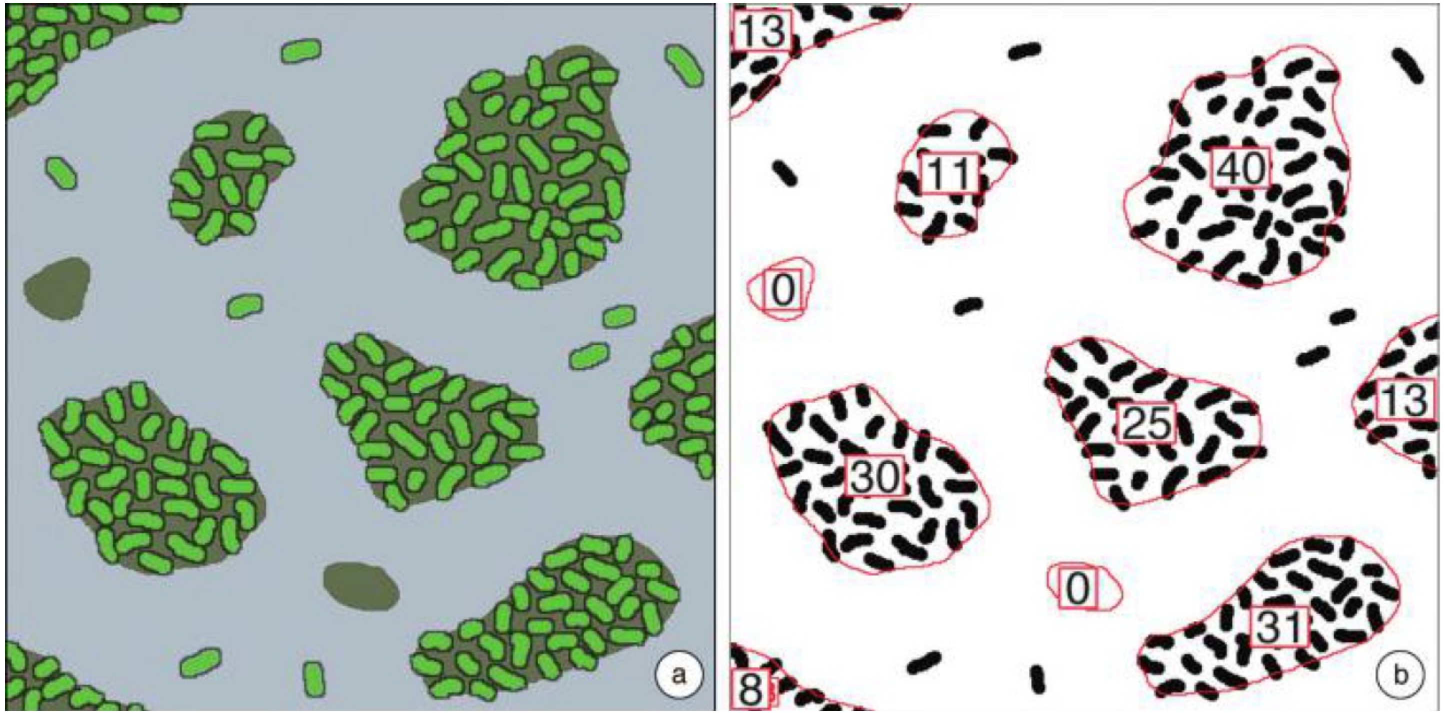
[skimage.morphology.convex_hull_image](image)

[skimage.morphology.convex_hull_object](image)

(Gonzalez & Woods, Fig 9.22)

Exemple : Enveloppe convexe

- Calcul automatique du nombre de bactéries par colonie et du nombre de colonies dans un boîte de Petri
- Extraction des composants connectés et des enveloppes convexes.



Source : Russ
Figure 8.29

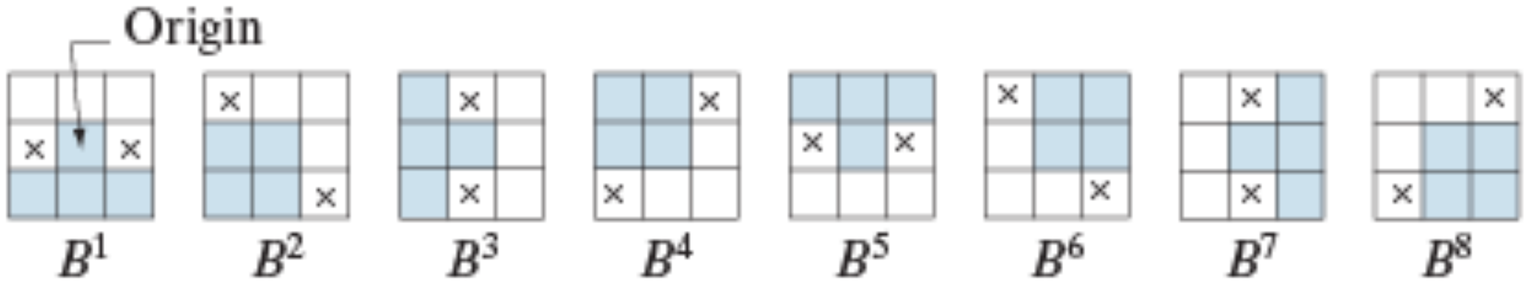
Application : Amincissement (*Thinning*)

- Utilisé pour retirer certains pixels d'avant-plan de l'image binaire
- Intersection entre une image et le complément d'une transformation *Hit-or-Miss*

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

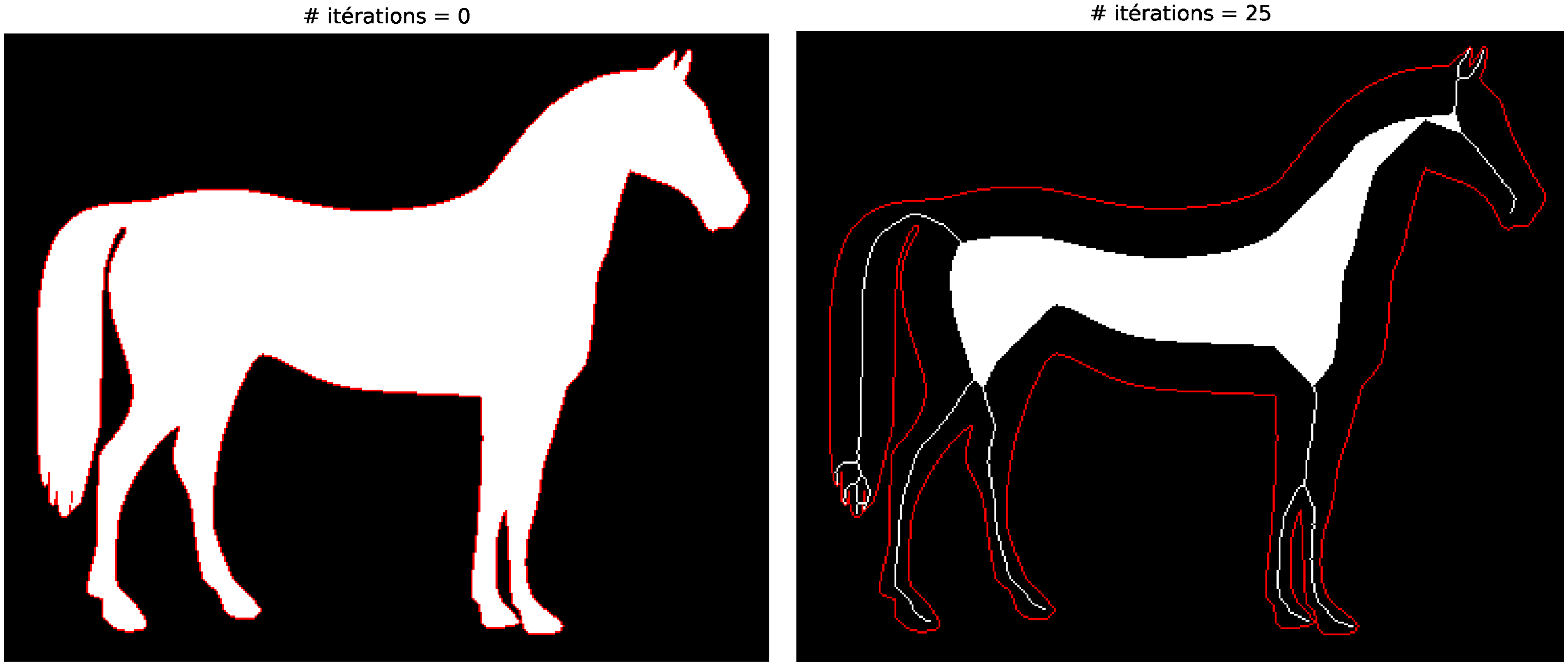
- Pour décrire un amincissement symétrique, on peut utiliser une séquence d'éléments structurants

$$\{B\} = \{B^1, B^2, B^3, \cdots, B^n\}$$



Exemple : Amincissement (*Thinning*)

- **img_t = skimage.morphology.thin(img, max_iter=n_iter)**

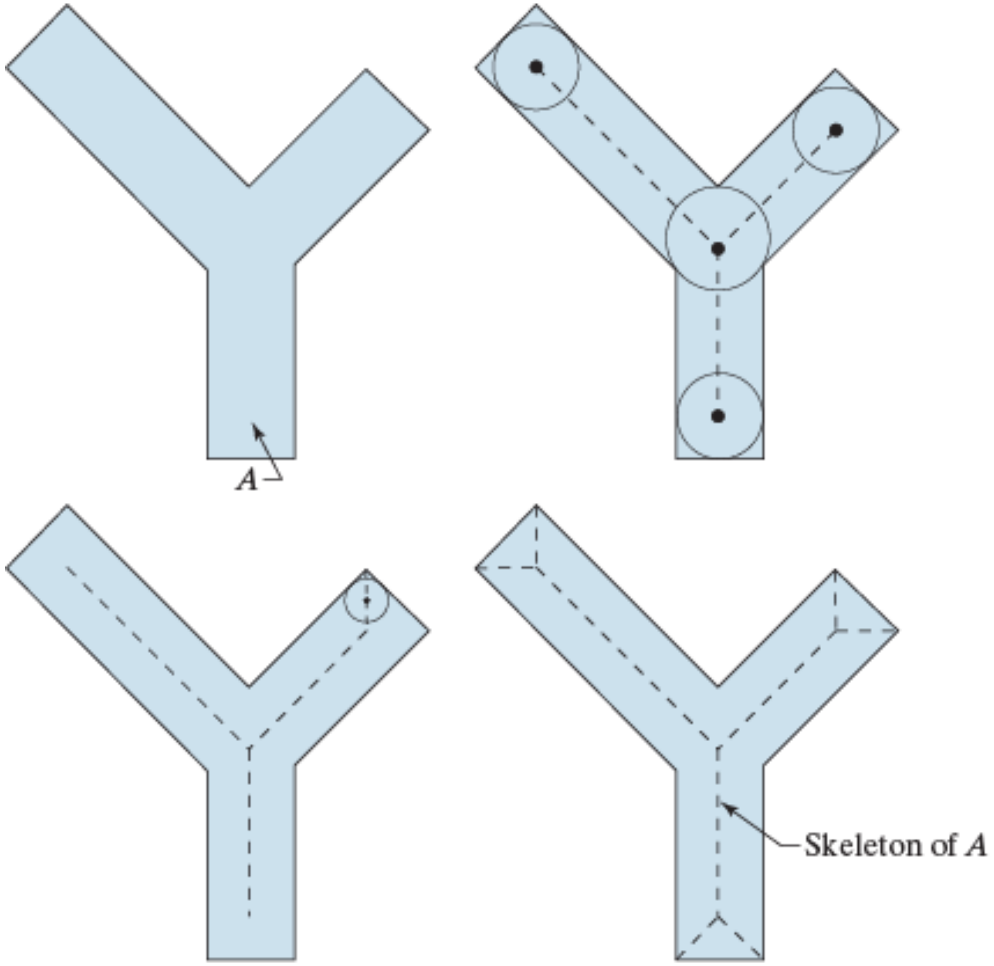


Application : Élargissement

- Ajouter des pixels à l'avant-plan de façon sélective
$$A \odot B = A \cup (A \circledast B)$$
- Peut être défini par une séquence d'opérations en utilisant plusieurs filtres.
- Inverse dual de l'amincissement
  - Élargissement de l'avant-plan = amincissement de l'arrière-plan
  - Souvent seul l'amincissement est implémenté

Application : Squelettisation

- **Squelette** : Positions des plus grands cercles pouvant être inclus dans l'objet d'avant-plan
- Peut être construit à partir d'érosions successives
- On peut reconstruire l'objet A à partir de son squelette et de dilatations successives
- Plusieurs versions de squelettes dans **skimage.morphology**
  - `skeletonize(image)`
  - `thin(image)`
  - `medial_axis(image)`



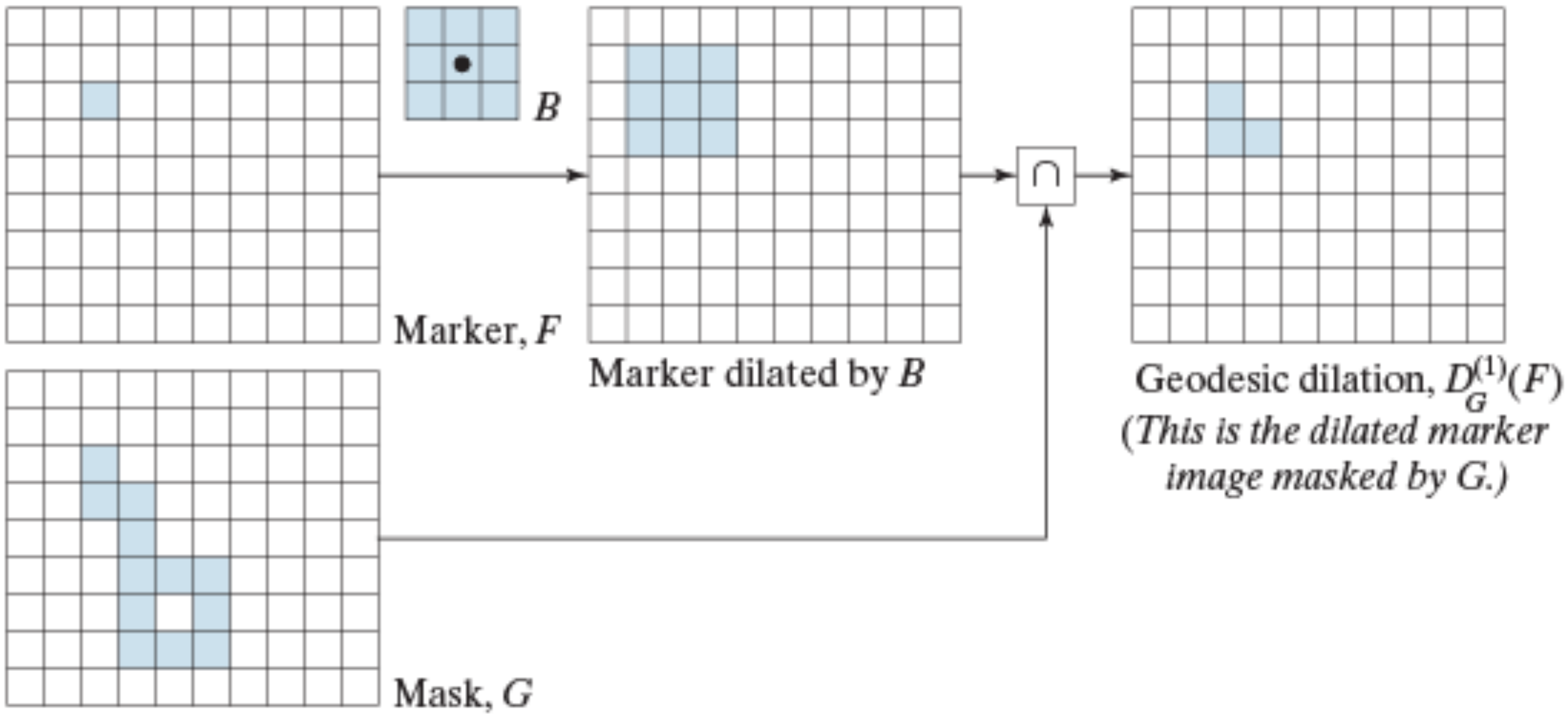
Reconstruction morphologique

- Utilise 2 images (F, G) et un élément structurant (B)
  - F : **Marqueur**. Représente les points initiaux de la reconstruction
  - G : **Masque**. Représente une contrainte (conditions) de la reconstruction
  - B : **élément structurant** pour définir le type de connectivité (ex: 4 ou 8 voisins)
- Basé sur la dilatation et l'érosion géodésique

Dilatation géodésique

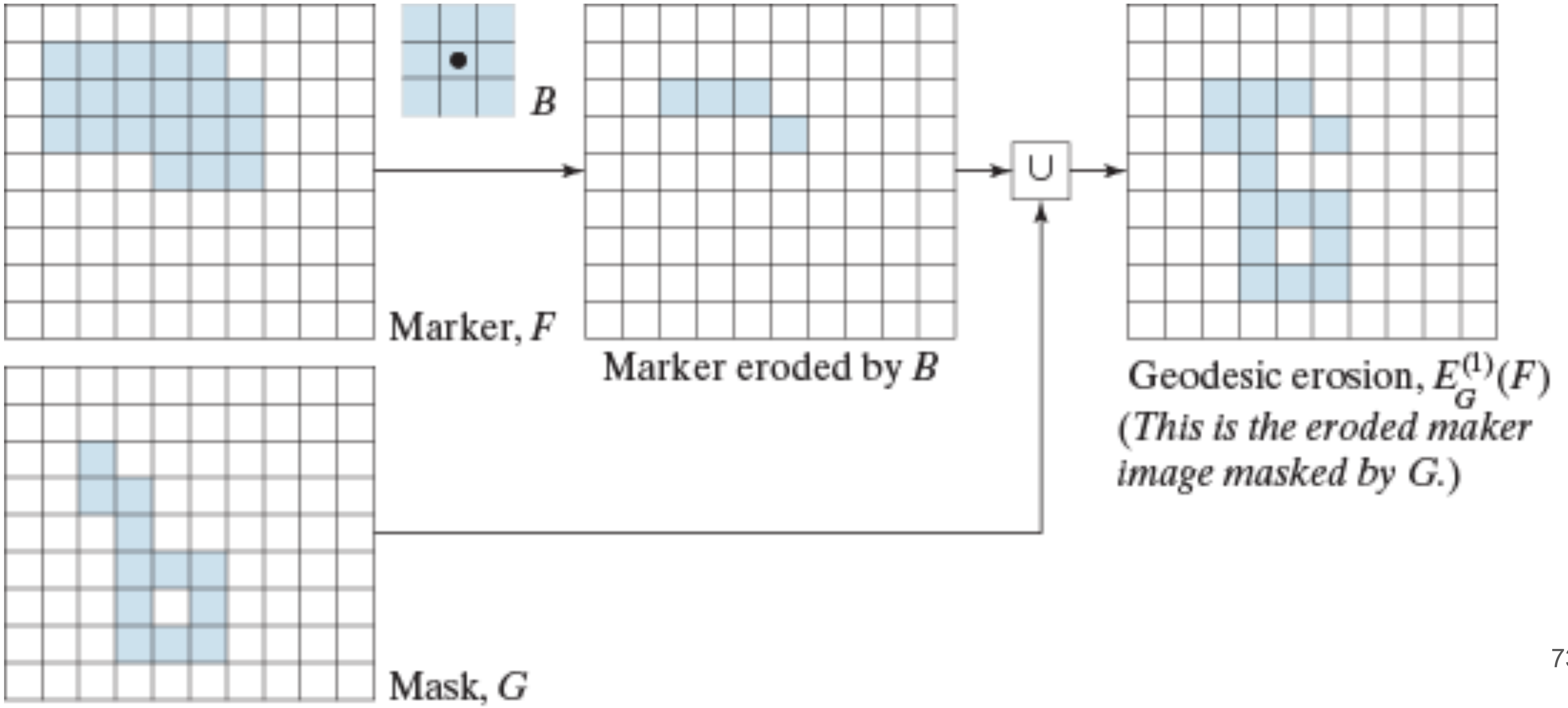
- Dilatation géodésique de taille = 1

$$D_G^{(1)}(F) = (F \oplus B) \cap G$$



Érosion géodésique

$$E_G^{(1)}(F) = (F \ominus B) \cup G$$



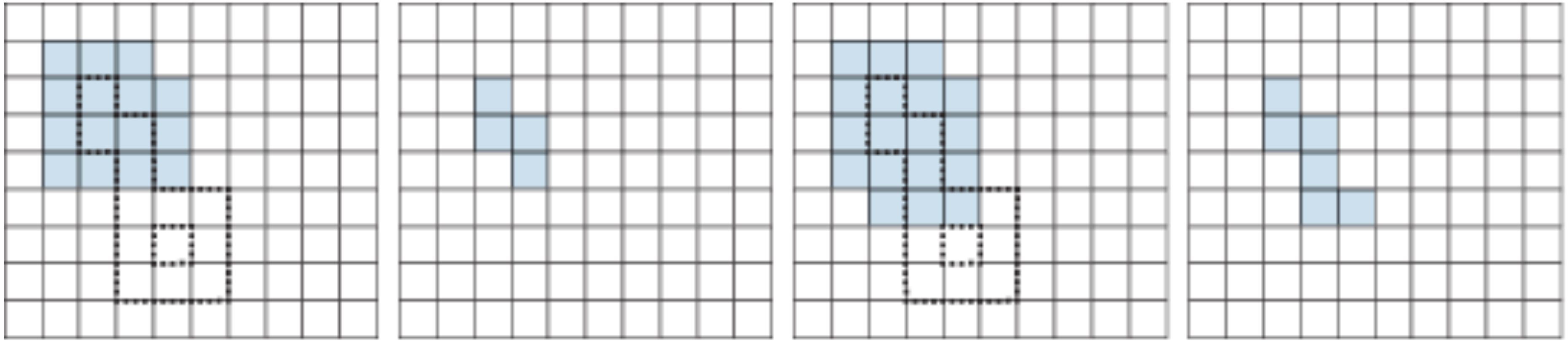
Reconstruction morphologique par dilatation et érosion

- On peut répéter la dilatation ou l'érosion géodésique *n* fois

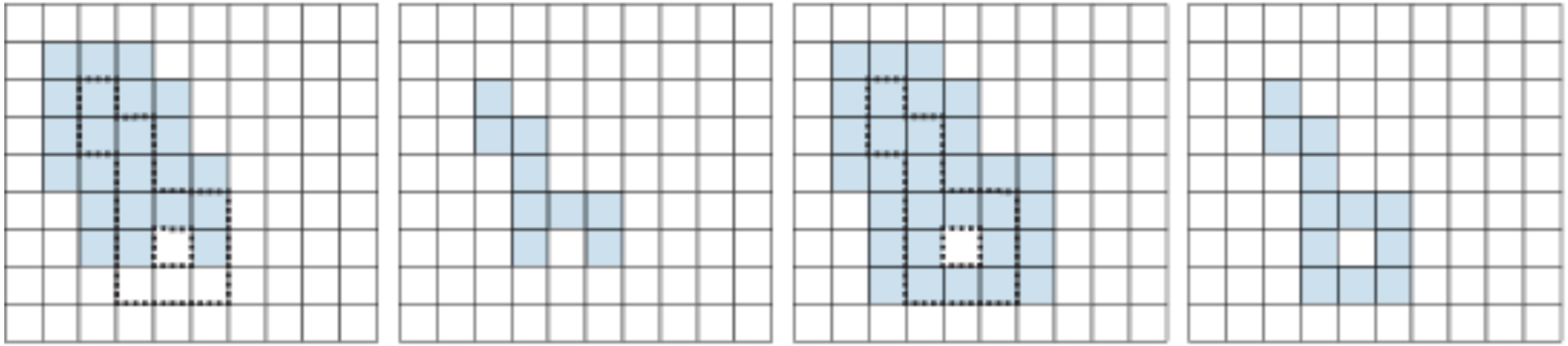
$$D_G^{(n)}(F) = D_G^{(1)}\left(D_G^{(n-1)}(F)\right)$$
$$E_G^{(n)}(F) = E_G^{(1)}\left(E_G^{(n-1)}(F)\right)$$

- Une reconstruction par dilatation / érosion s'arrête lorsqu'il n'y a plus de changements entre 2 itérations.

Exemple : Reconstruction morphologique par dilatation



$D_G^{(1)}(F)$ dilated by B | Result of masking $= D_G^{(2)}(F)$ | $D_G^{(2)}(F)$ dilated by B | Result of masking $= D_G^{(3)}(F)$

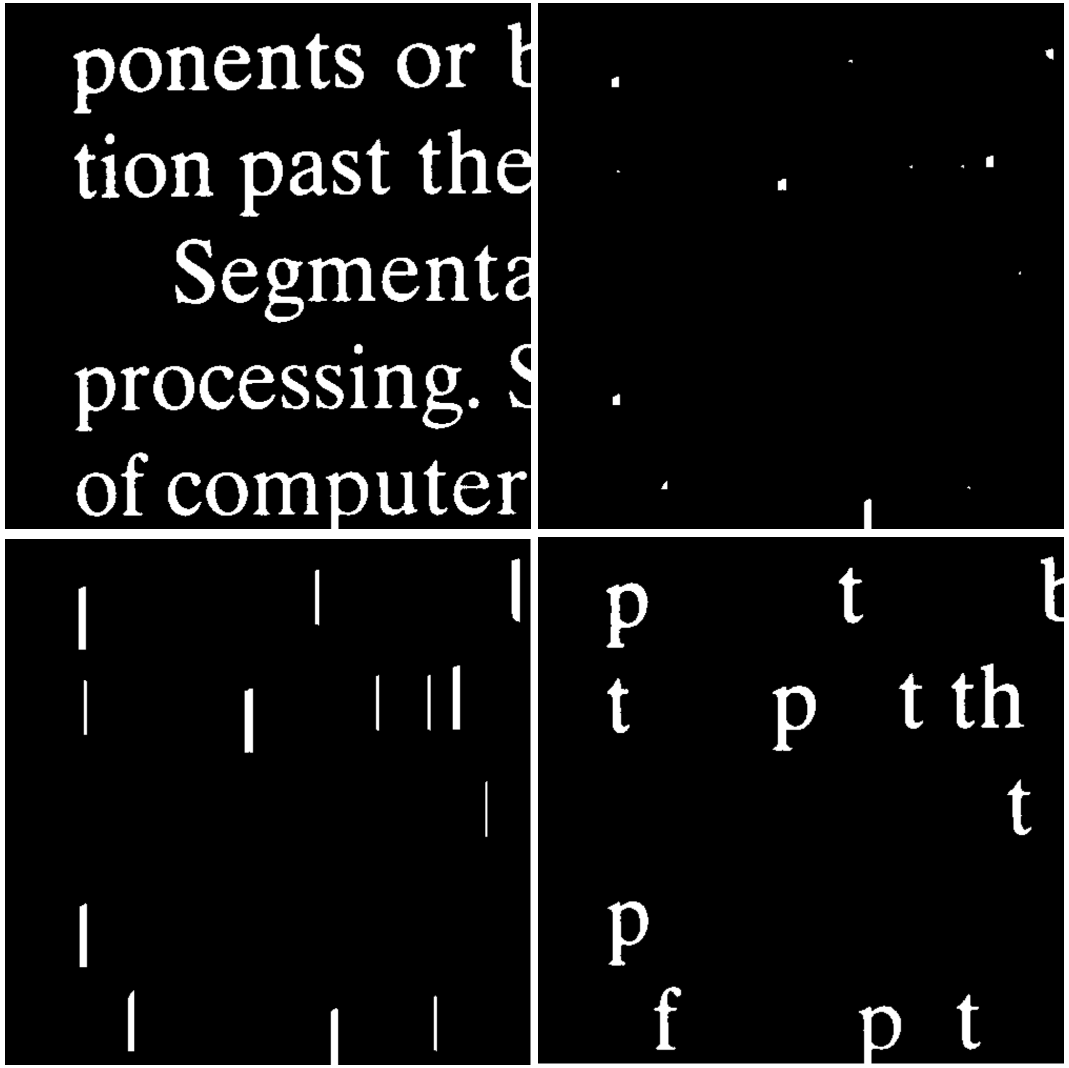


$D_G^{(3)}(F)$ dilated by B | Result of masking $= D_G^{(4)}(F)$ | $D_G^{(4)}(F)$ dilated by B | Result of masking $= D_G^{(5)}(F)$
No changes after this point, so $R_G^D(F) = D_G^{(5)}(F)$

Application : Ouverture par reconstruction

Image

Érosion par une ligne (51px)



Ouverture par dilatation

Ouverture par reconstruction

[Source]

Application : Sélection par marqueurs

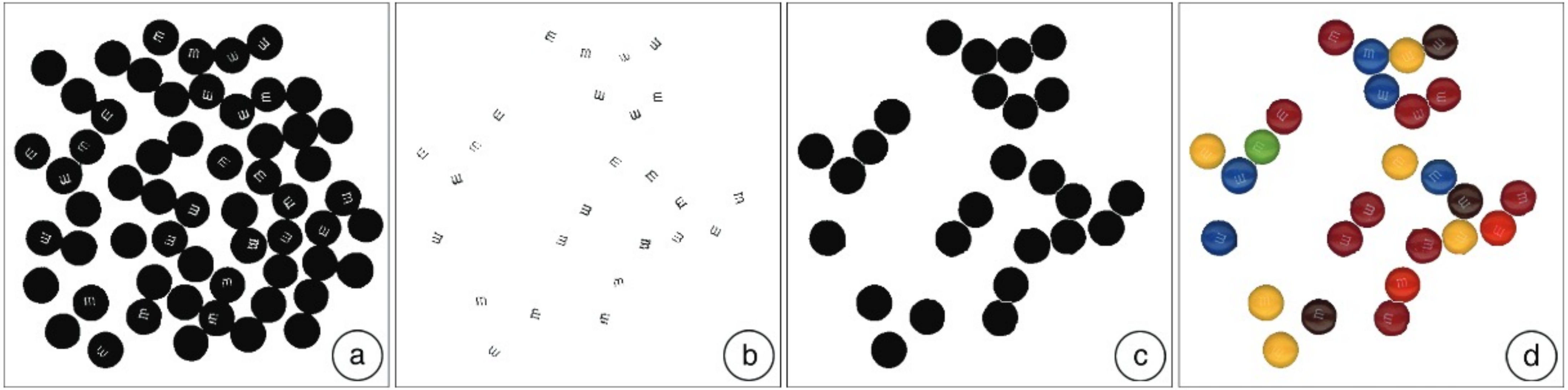


Image segmentée de bonbons

Lettre « m » identifiée en tant que trous

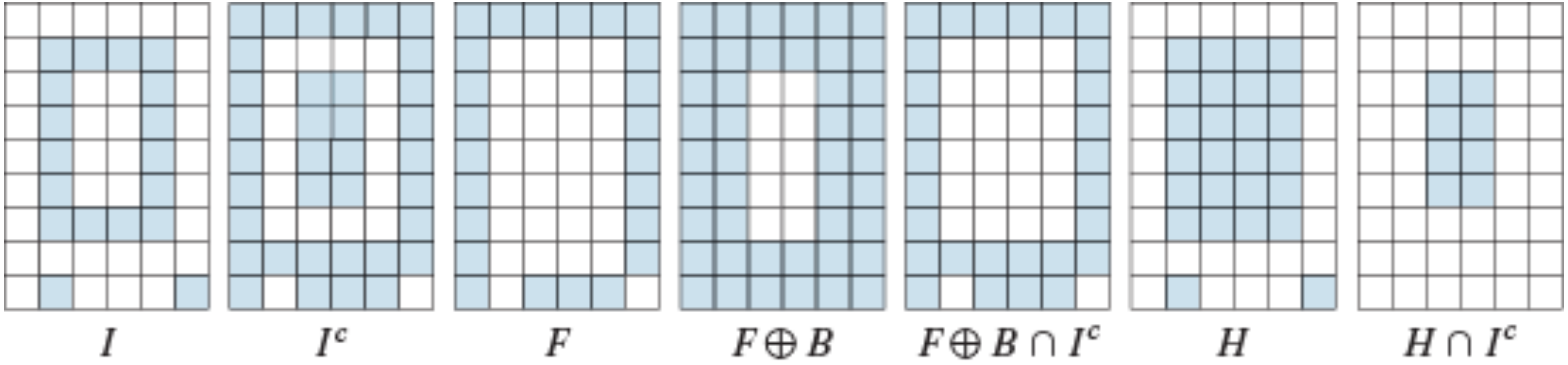
Marqueur en (b) pour sélectionner les bonbons (après hole-filling et watershed)

Masque de l'image originale par (c), qui montrent uniquement les bonbons avec le côté « m » vers le haut!

Source : Russ, Figure 8.27.

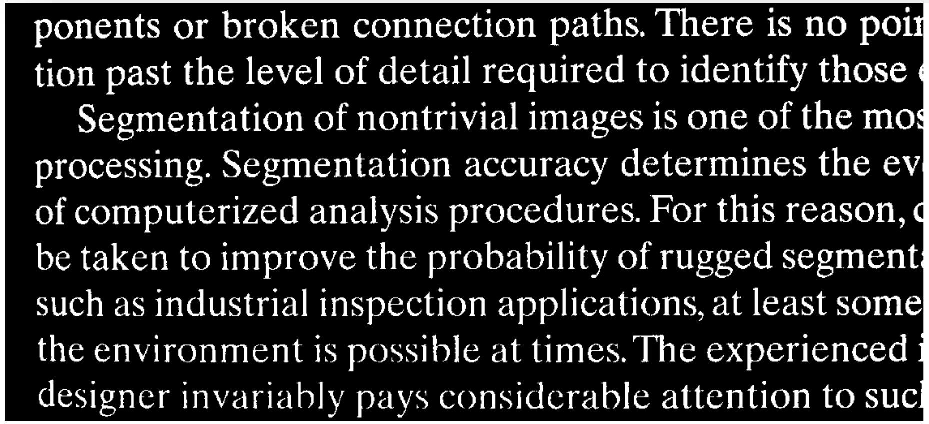
Application : Remplissage de trous automatique

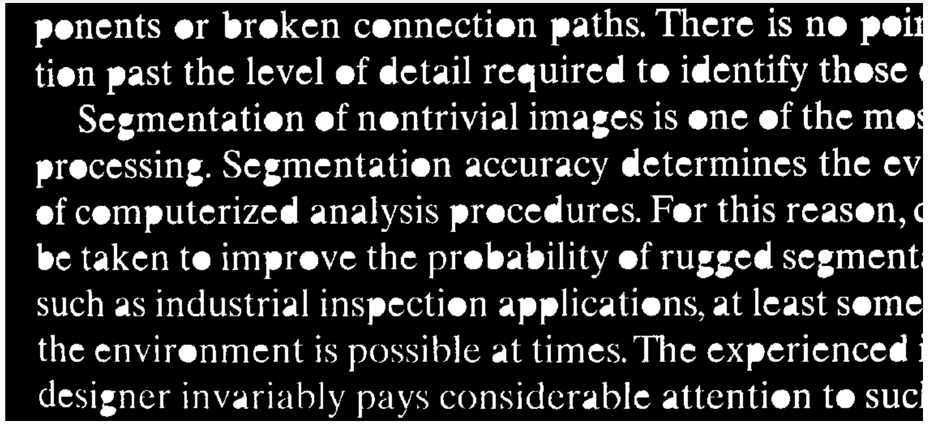
- Marqueur F: Image vide, sauf près de la bordure où on utilise l'inverse de l'image
- Masque : Inverse de l'image I^c
- Image sans trous : $H = \left[R^D_{I^c}(F)\right]^c$



Exemple : Remplissage de trous automatique

```python
img = imageio.imread("../data/text-image.tif")

# Création du marqueur
F = ~img
F[1:-1:, 1:-1:] = False

# Reconstruction morphologique
H = 1 - skimage.morphology.reconstruction(F, ~img, method='dilation')
```

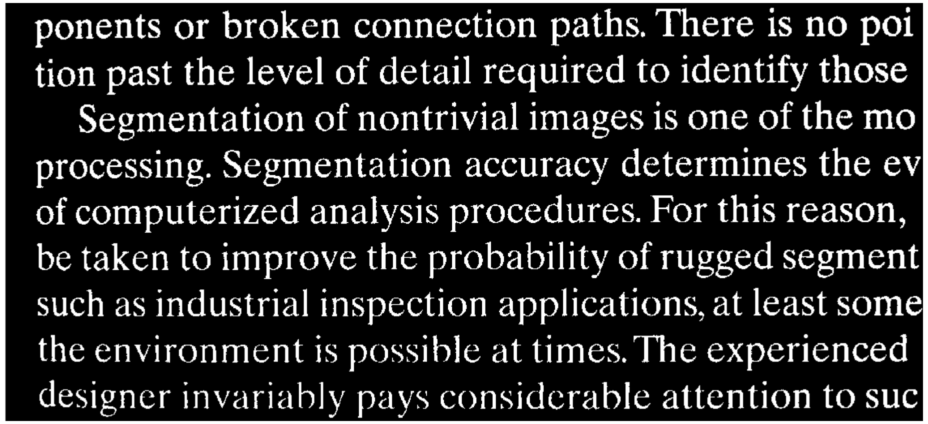




Application : Nettoyage des bordures

- Marqueur F : Image vide, sauf près de la bordure où on utilise la valeur de l'image
- Masque : Image I
- Image sans objets touchant la bordure

$$X = I - R_I^D(F)$$



ponents or broken connection paths. There is no poi
tion past the level of detail required to identify those
Segmentation of nontrivial images is one of the mo
processing. Segmentation accuracy determines the ev
of computerized analysis procedures. For this reason,
be taken to improve the probability of rugged segment
such as industrial inspection applications, at least some
the environment is possible at times. The experienced
designer invariably pays considerable attention to suc

Morphologie en niveaux de gris

Chapitre 6 : Morphologie mathématique

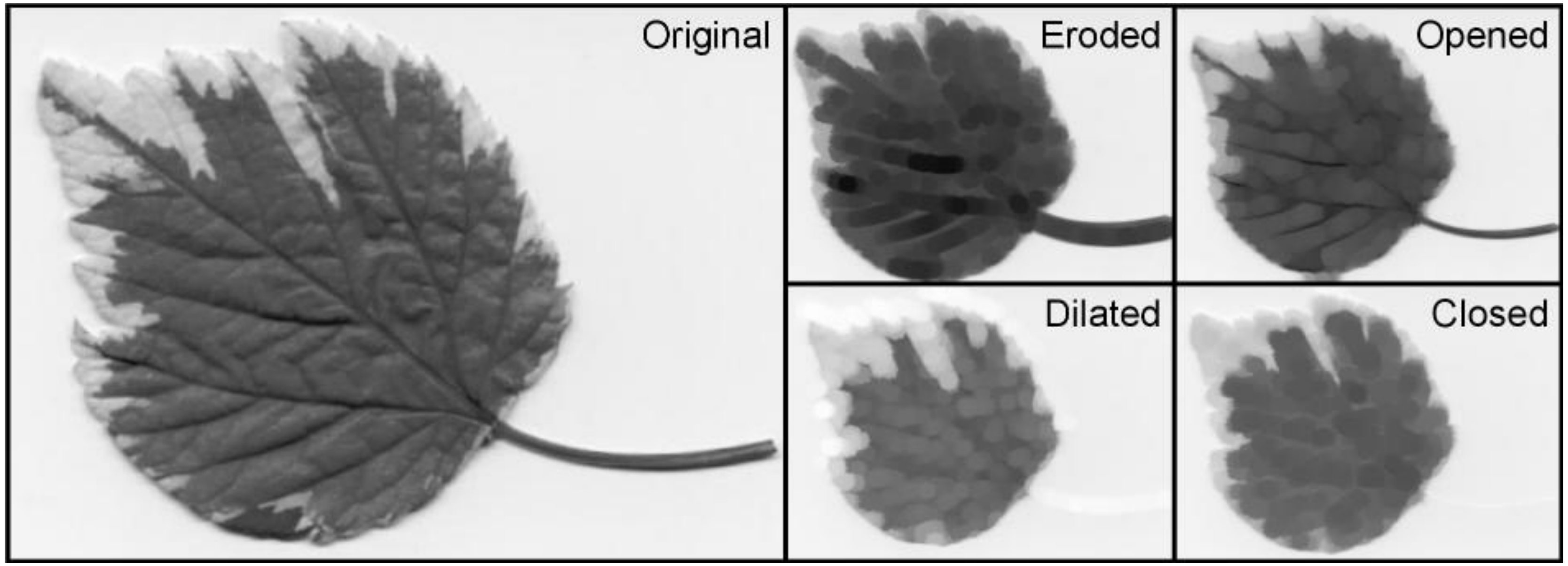
Joël Lefebvre (UQÀM)

INF600F – Traitement d'images

Opérations morphologiques en niveaux de gris

- Généralisation de la morphologie mathématique avec des images binaires.
- Les opérateurs **OR et AND deviennent MAX et MIN**

Source



Élément structurant (gris)

- L'élément structurant est une fonction 2D de **nombres réels**

$$H(i,j) \in \mathbb{R}, \qquad \text{pour } (i,j) \in \mathbb{Z}^2$$

- Contrairement au filtre linéaire (convolution), **une valeur de 0 dans $\boldsymbol{H}$ contribue** au filtre morphologique.
- Il faut distinguer entre les **valeurs nulles (0)** et les **valeurs à ignorer**

$$\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \neq \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & 2 & 1 \\ \hline & 1 & \\ \hline \end{array}$$

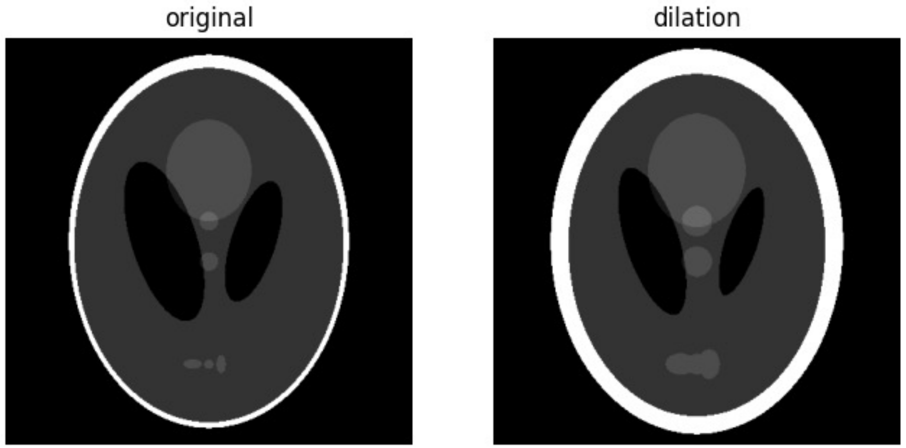
Dilatation morphologique grise

- Maximum des valeurs de H additionnées aux valeurs de la sous-image de I

$$(I \oplus H)(u,v) = \max_{(i,j)\in H} \{I(u+i, v+j) + H(i,j)\}$$

$$\overset{I}{\begin{bmatrix}6&7&3&4\\5&6&6&8\\6&4&5&2\\6&4&2&3\end{bmatrix}} \oplus \overset{H}{\begin{bmatrix}1&1&1\\1&\boxed{2}&1\\1&1&1\end{bmatrix}} = \overset{I\oplus H}{\begin{bmatrix}\ &\ &\ &\ \\ &8&9& \\ &7&9& \\ & & & \end{bmatrix}}$$

$$\overset{I+H}{\begin{bmatrix}7&8&4\\6&\boxed{8}&7\\7&5&6\end{bmatrix}} \quad \text{max}$$



Source

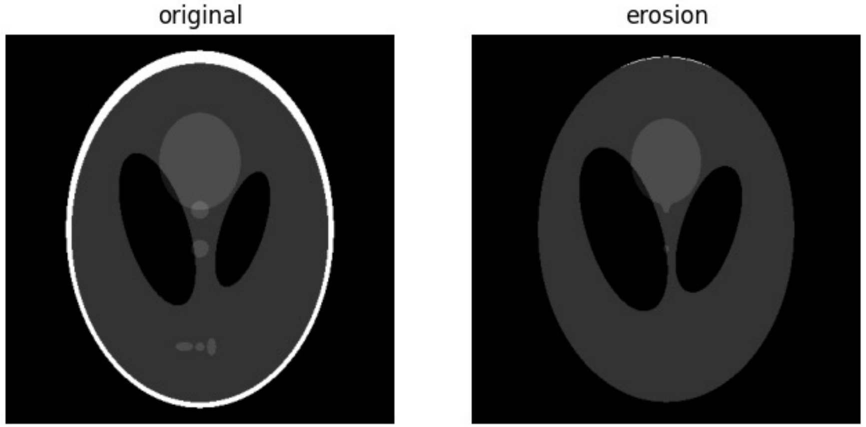
Érosion morphologique grise

- Minimum des valeurs de H soustraites aux valeurs de la sous-image de I

$$(I \ominus H)(u,v) = \min_{(i,j)\in H} \{I(u+i, v+j) - H(i,j)\}$$

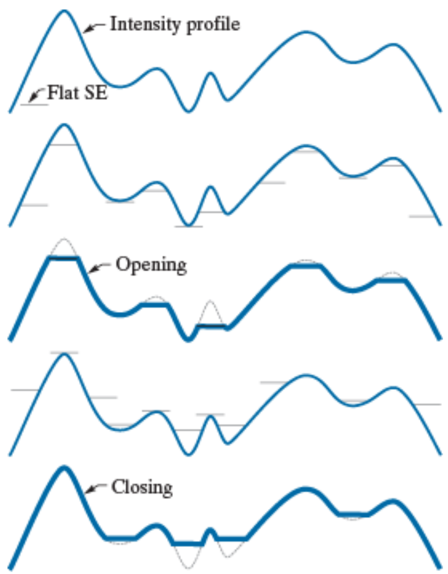
$$\underset{I}{\begin{array}{|c|c|c|c|}\hline 6&7&3&4\\\hline 5&6&6&8\\\hline 6&4&5&2\\\hline 6&4&2&3\\\hline\end{array}} \ominus \underset{H}{\begin{array}{|c|c|c|}\hline 1&1&1\\\hline 1&\boxed{2}&1\\\hline 1&1&1\\\hline\end{array}} = \underset{I \ominus H}{\begin{array}{|c|c|c|c|}\hline \ & \ & \ & \ \\\hline \ &2&1& \ \\\hline \ &1&1& \ \\\hline \ & \ & \ & \ \\\hline\end{array}}$$

$$\underset{I-H}{\begin{array}{|c|c|c|}\hline 5&6&2\\\hline 4&\boxed{4}&5\\\hline 5&3&4\\\hline\end{array}} \quad \text{min}$$

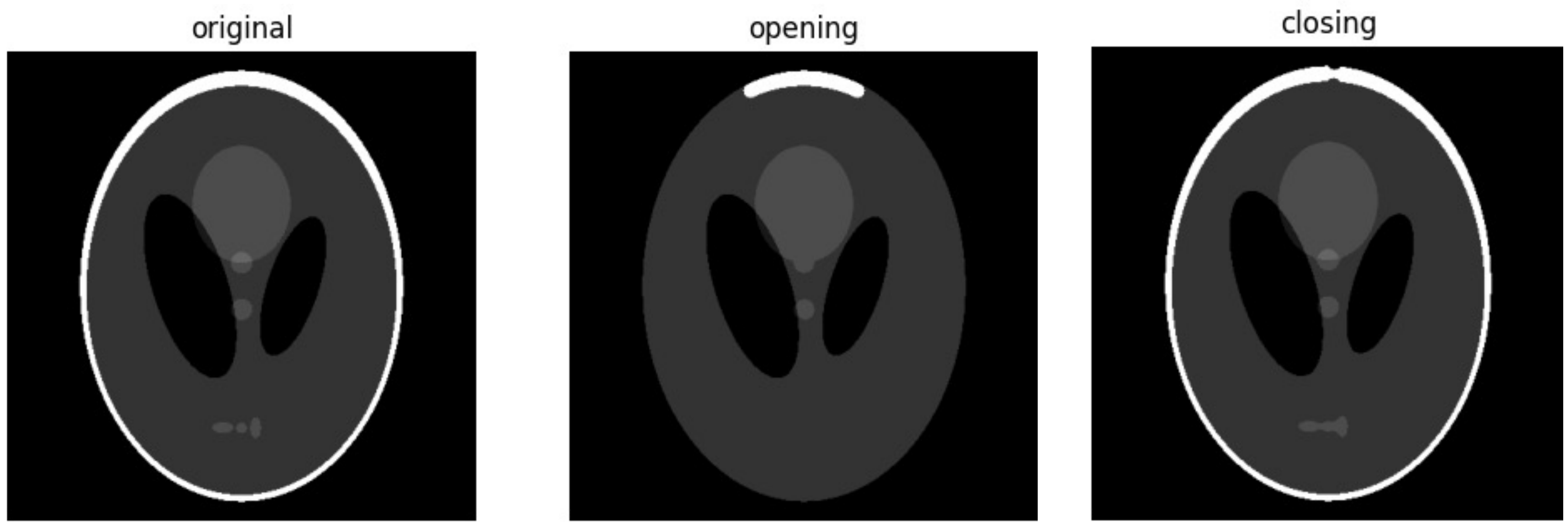


Source

Ouverture et fermeture morphologique en niveaux de gris



- Défini de la même façon que pour les images binaires
- **Ouverture** : Érosion suivie d'une dilatation
- **Fermeture** : Dilatation suivie d'une érosion



Quelques applications de la morphologie mathématique en niveaux de gris

- Lissage morphologique
- Gradient morphologique
- Transformations *Top-Hat* et *Bottom-Hat*
- Granulométrie

Application : Lissage morphologique

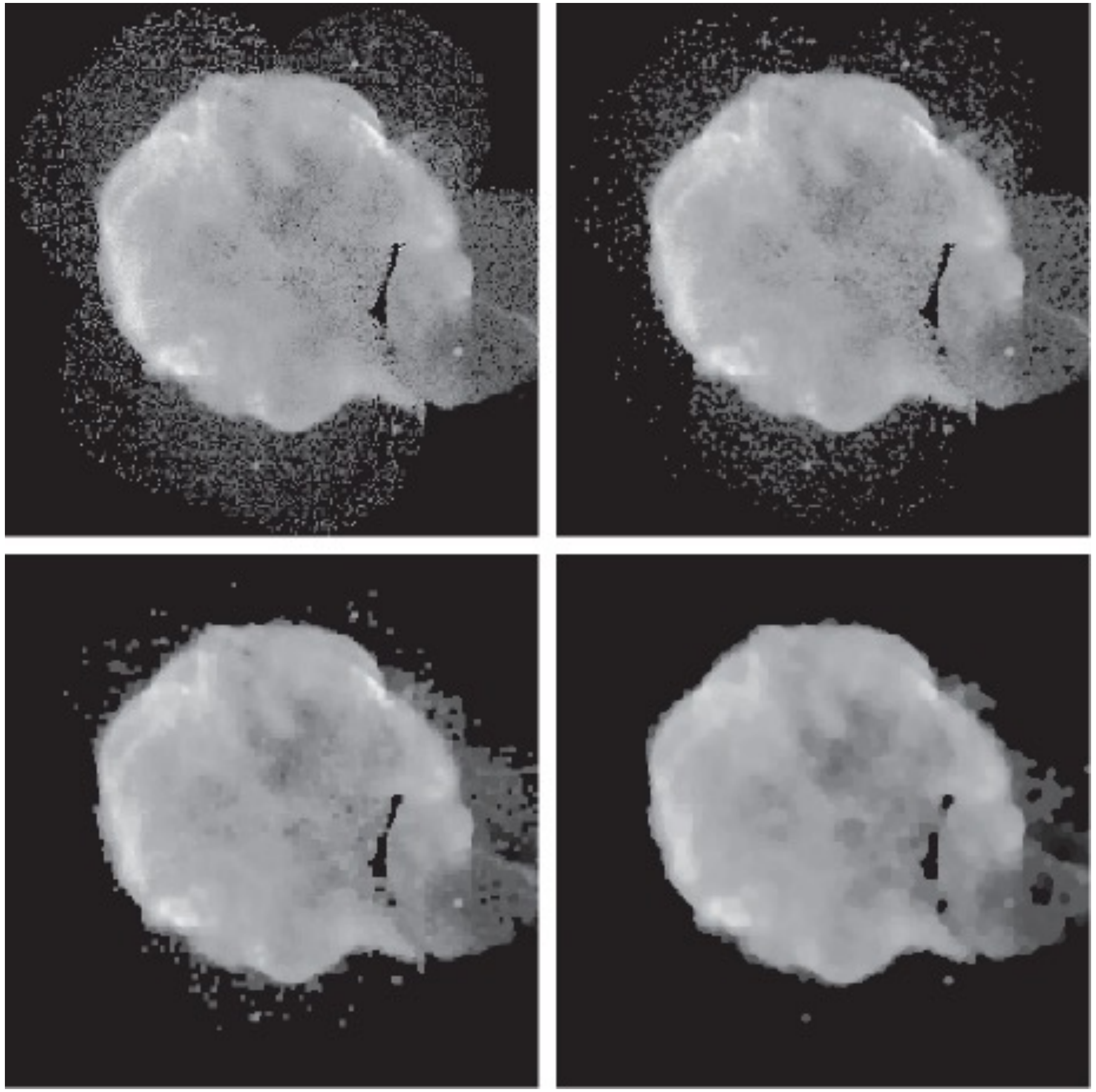


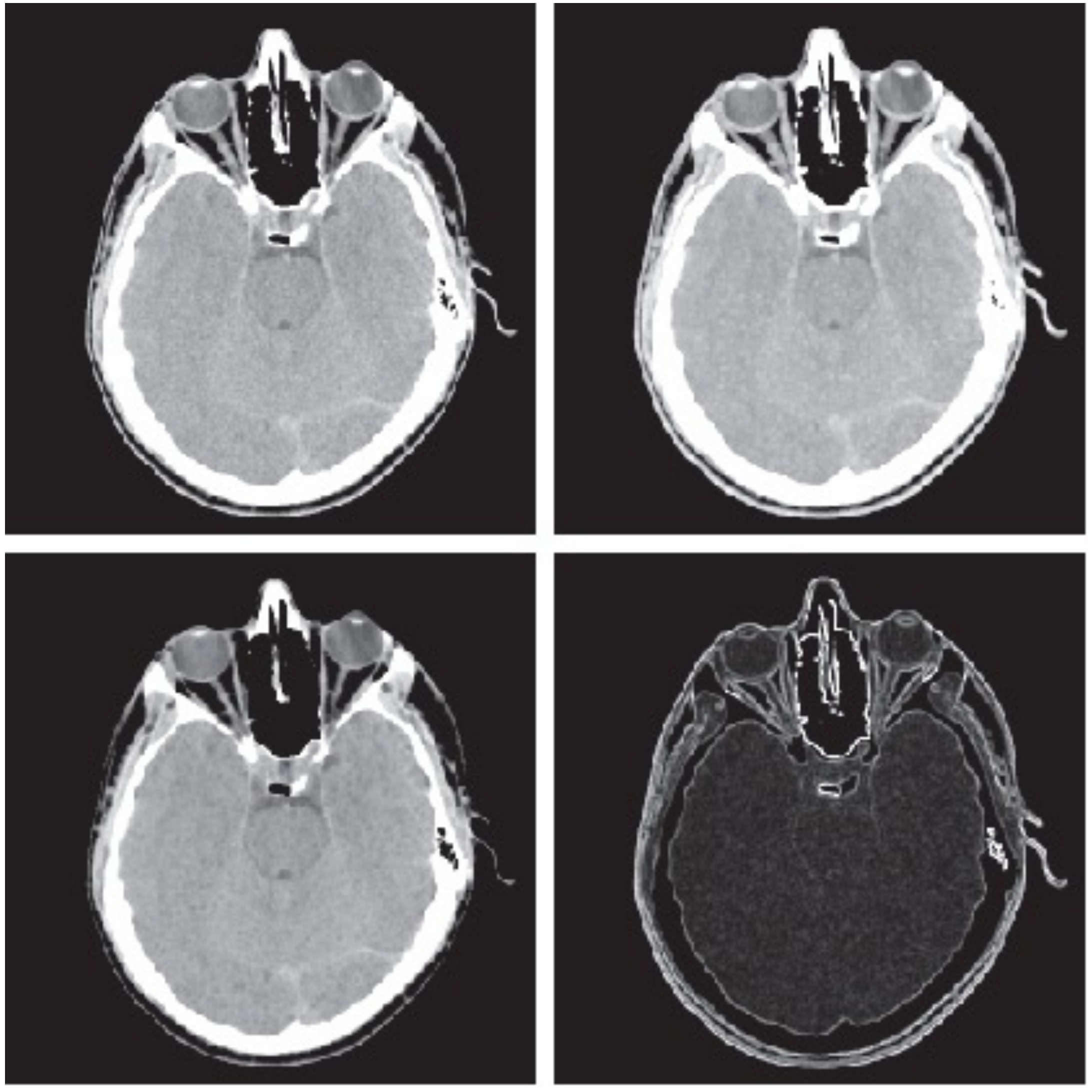
Image originale (Supernova en rayons X)

Ouverture & Fermeture avec un disque de rayon=1

Ouverture & Fermeture avec un disque de rayon=3

Ouverture & Fermeture avec un disque de rayon=5

Application : Gradient morphologique



(A) Scan CT d'un crâne

(B) Dilatation morphologique

(C) Érosion morphologique

(D) Différence entre B et C

(Gonzalez & Woods, Fig9.41)

Filtres *Top-Hat* et *Bottom-Hat* 

- Combinaison de la différence entre l'image et son ouverture morphologique en niveaux de gris

$$T_{hat}(f) = f - (f \circ b)$$

- La **transformation *bottom-hat*** est l'inverse : différence entre la fermeture morphologique et l'image originale

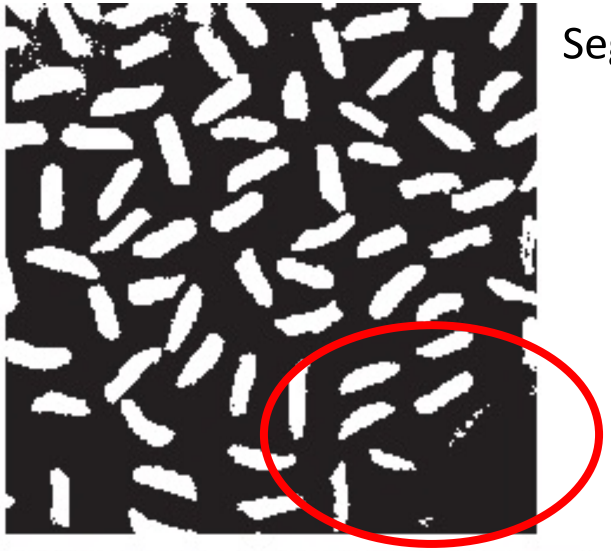
$$T_{bottom}(f) = (f \cdot b) - f$$

- Permet de **retirer des petits objets clairs** sur fond sombre (T_{top}) ou **des petits objets sombres** sur fond clair (T_{bottom})

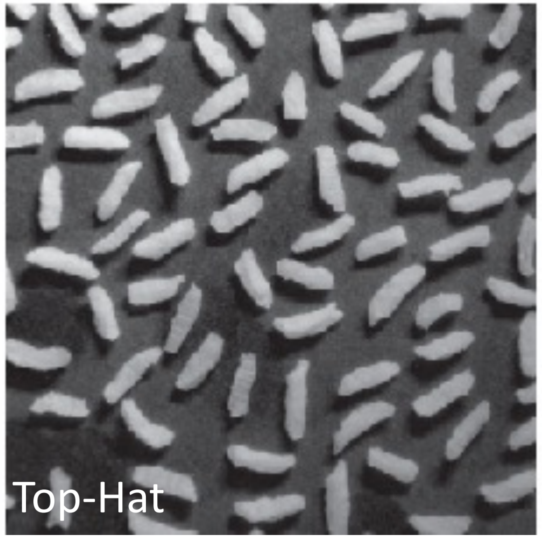
Exemple : Filtre *Top-Hat* pour corriger l’illumination

Image originale

Segmentation



Ouverture morphologique

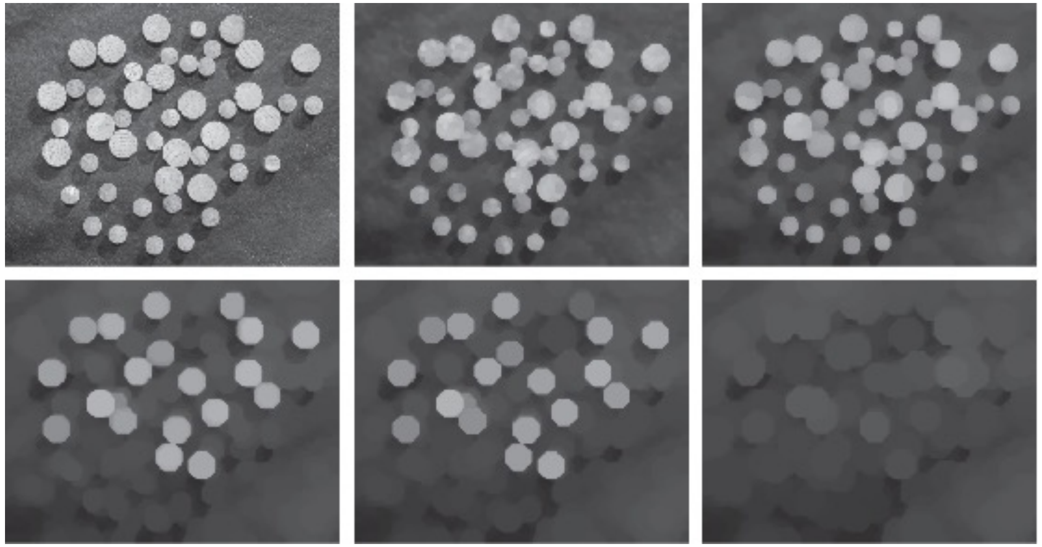


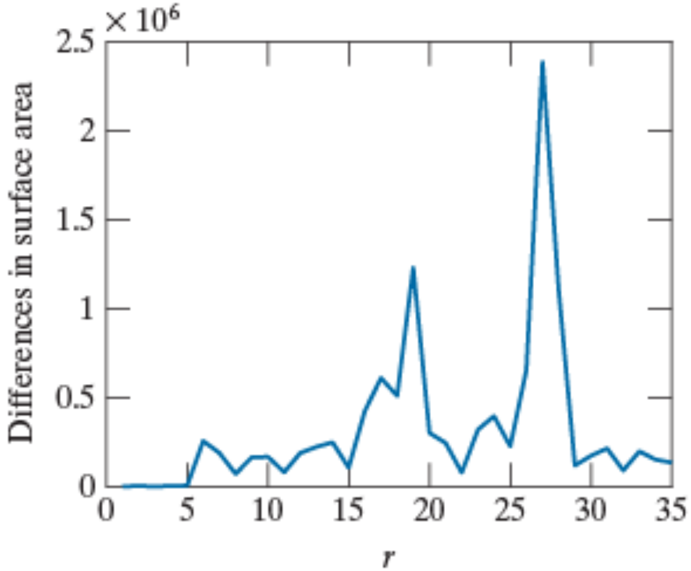
Segmentation

Application : Granulométrie

- Détermination de la **distribution de la taille** des particules
- Application d’ouvertures morphologiques avec un élément structurant de taille croissante.
- Calcul de la **superficie** (***somme des intensités dans l’image***) pour chaque ouverture

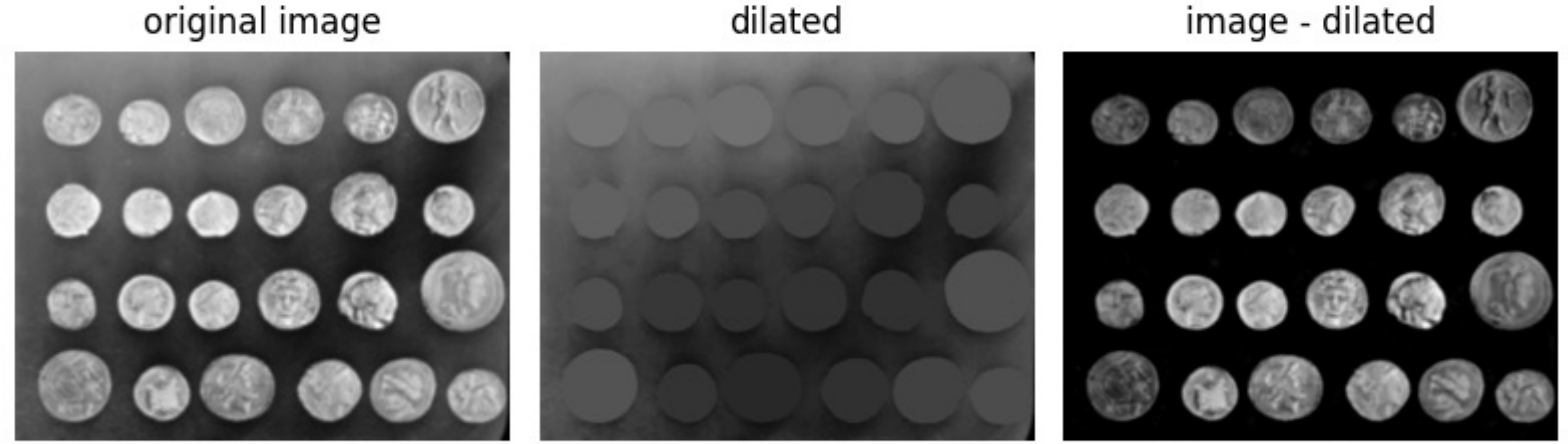
Douilles de bois.
Gonzalez, Fig. 9.43





Reconstruction morphologique en niveaux de gris

- Concept similaire au cas binaire
- Dilatations / Érosions contraintes par un masque et initialisées par un marqueur



Source : Détection des maximums régionaux par reconstruction morphologique de l'arrière-plan