



Traitement d'images

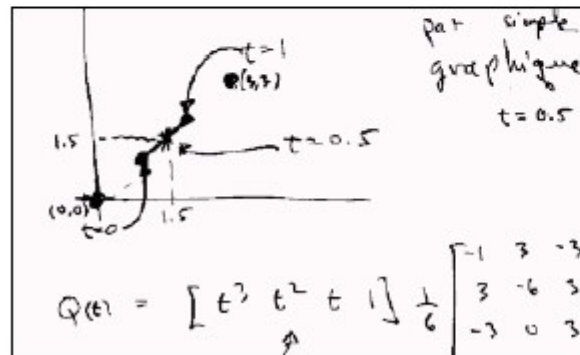
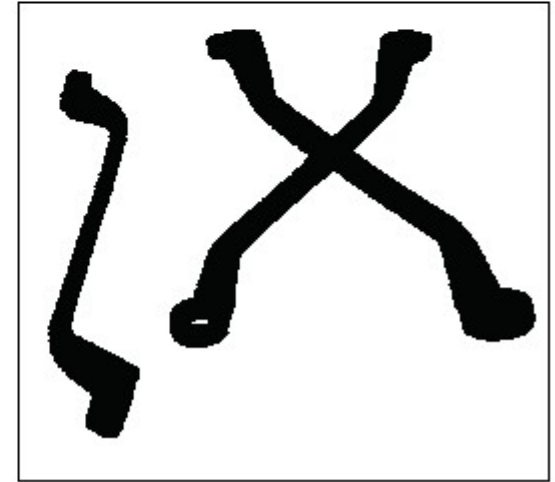
Images binaires

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

Images binaires

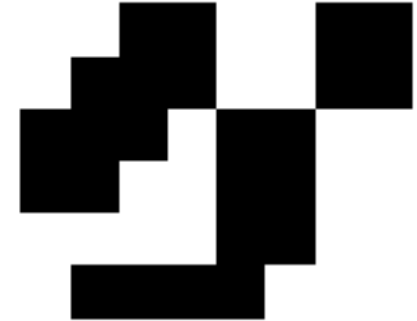
Plusieurs techniques particulières permettent de manipuler les images binaires (0:1)

Utile, par exemple, pour traiter les résultats de segmentation



Connexité des pixels

Combien d'objets voyez-vous ici ?



Connexité-4

p et q sont connexe-4 si q est dans $N_4(p)$

1	1	1
1	1	1
1	1	1

Connexité-8

p et q sont connexe-8 si q est dans $N_8(p)$

1	1	1
1	1	1
1	1	1

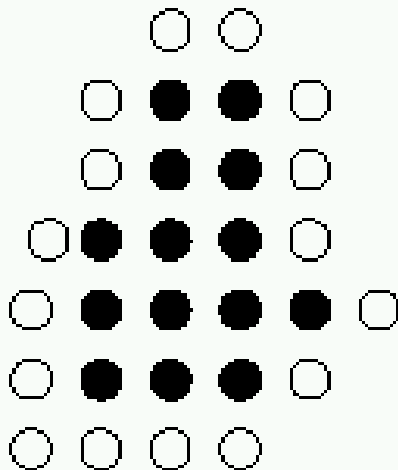
Connexité 4 ou 8

Conseils :

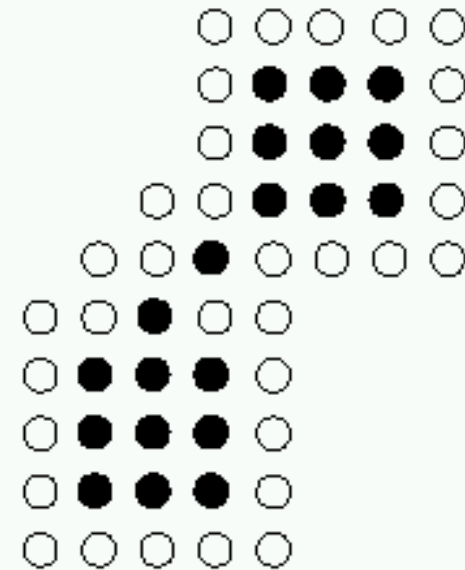
Connexités différentes pour les contours et régions

Connexité-4 pour les régions

Connexité-8 pour les contours



Région : 4-connexes
Contour : 8-connexes



Région : 8-connexes
Contour : 4-connexes



Comment mesurer les distances ?

Quelle est la distance entre A et B ?

				B
		A		



Distance discrète entre pixels

Distance D_4 (distance de Manhattan)

$$D_4(p,q) = |x-s| + |y-t|$$

Forme un diamant centré sur (x,y)

Ex: $D_4 \leq 2$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		



Distance discrète entre pixels

Distance D_8 (distance de l'échiquier)

$$D_8(p,q) = \text{maximum} (|x-s|, |y-t|)$$

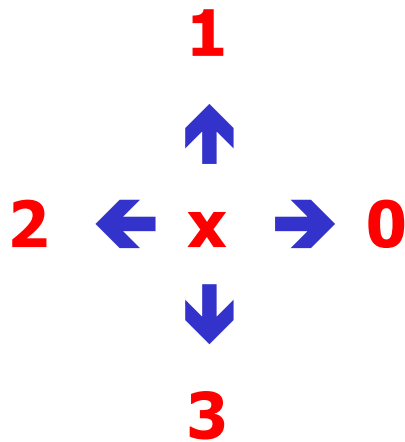
Forme un carré centré sur (x,y)

Ex: $D_8 \leq 2$

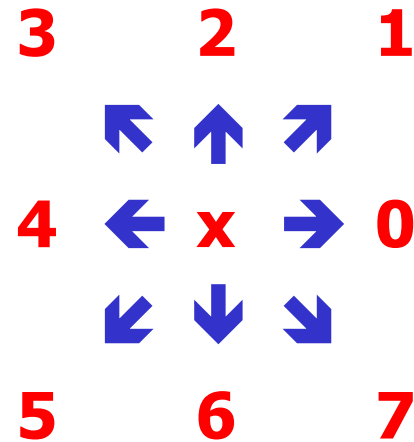
2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Codage de Freeman

- Pour coder les directions dans une image, on utilise le codage de Freeman :

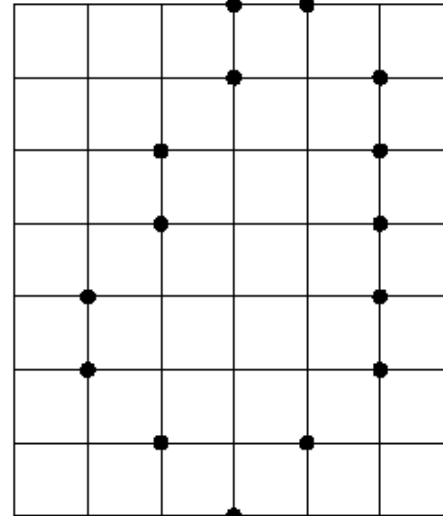


*Codage de Freeman en
connexité 4 (peu utilisé)*

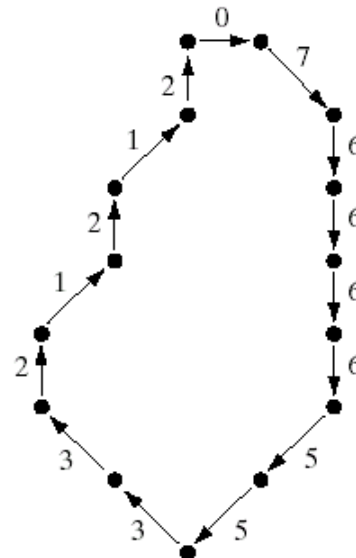
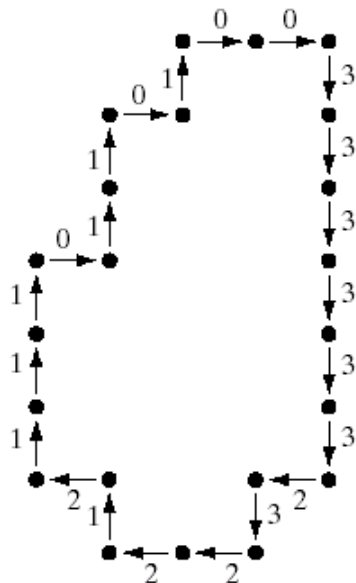


*Codage de Freeman en
connexité 8*

- On peut ainsi coder les contours en partant d'un pixel (en haut à gauche) et en tournant dans le sens des aiguilles d'une montre



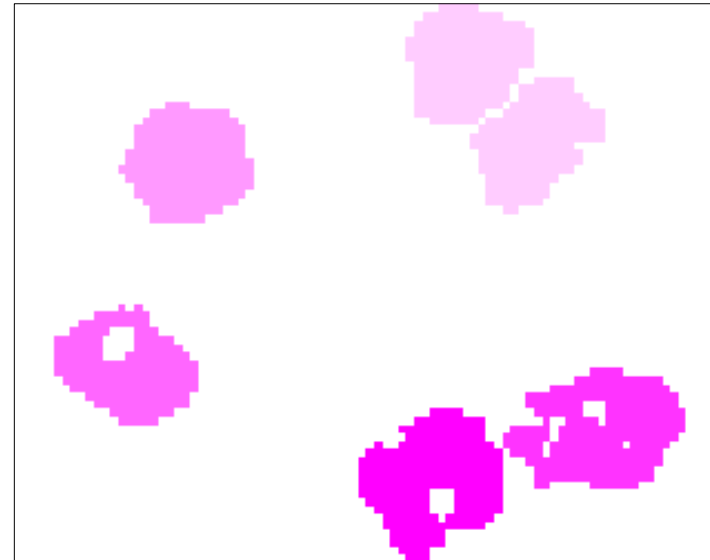
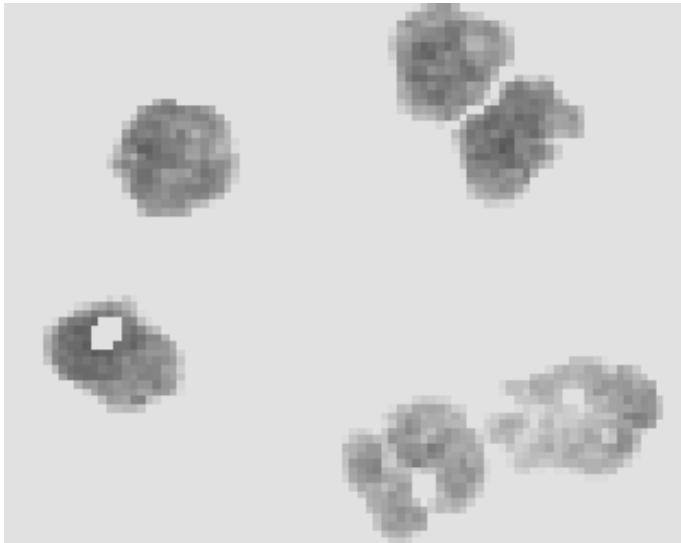
(a) Digital boundary with resampling grid superimposed.
(b) Result of resampling.
(c) 4-directional chain code.
(d) 8-directional chain code.



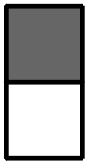
Etiquetages de composantes connexes

- **Une image segmentée** n'est qu'une succession de pixels
- **Composante connexe** = ensemble de pixels connexes (voisins) appartenant à une même entité
 - On désire donner une **valeur commune** pour les pixels d'**une région** ou d'**un contour**
 - On désire avoir une valeur différente pour chaque région/contour

Utilisé en post-segmentation



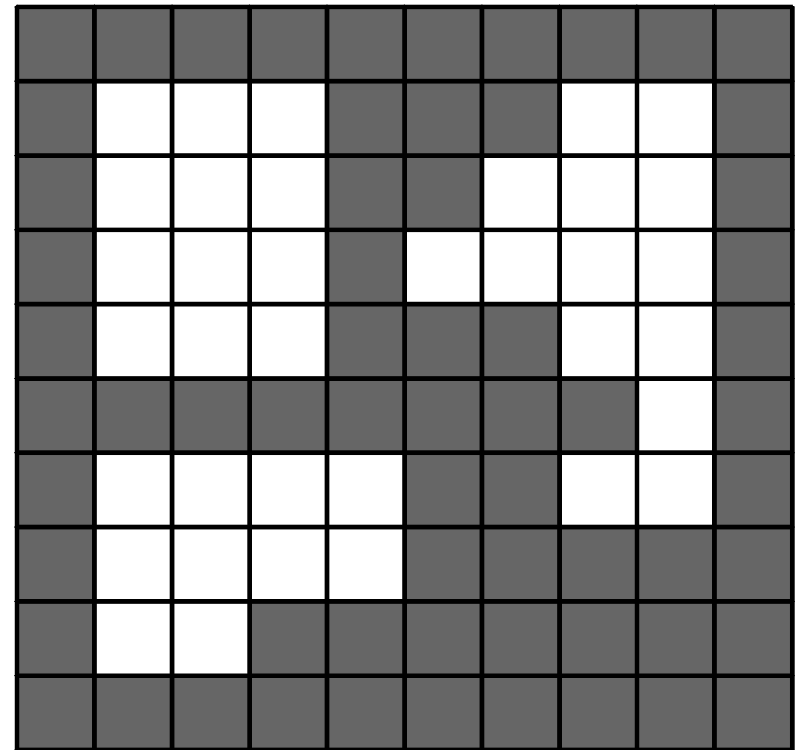
Etiquetages de composantes connexes



← *Fond*

← *Objets segmentés*

- Nous allons effectuer un parcours de l'image pour affecter un numéro unique (étiquette) pour chaque région
- Tous les pixels d'une même région doivent avoir le même numéro (étiquette)

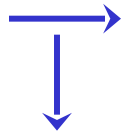


Etiquetages de composantes connexes

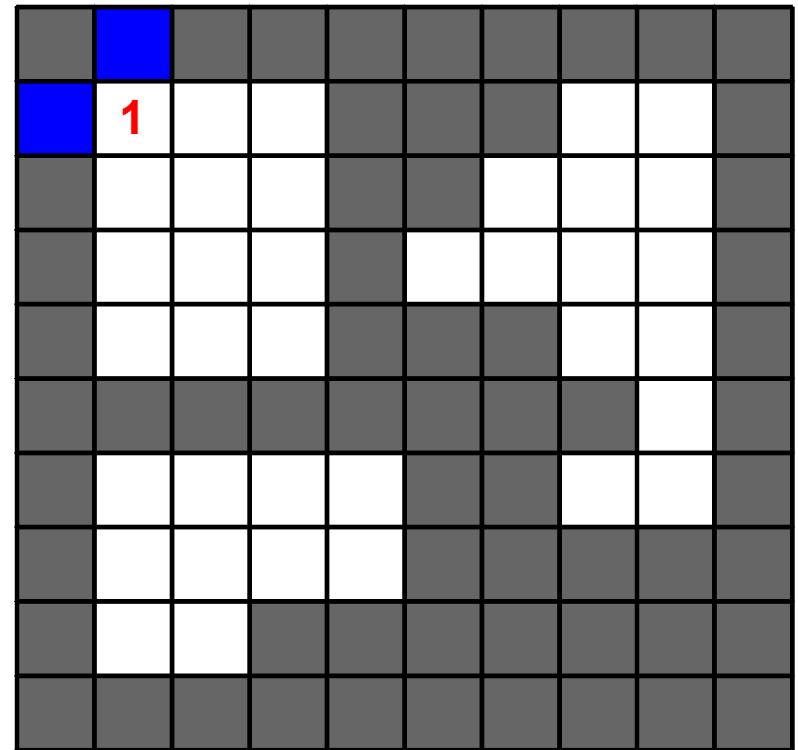
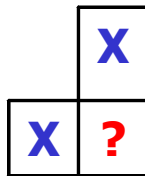
Premier parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - soit la plus petite étiquette parmi ses voisins **haut** et **gauche**
 - soit une nouvelle étiquette.

Parcours



Voisinage

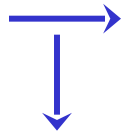


Etiquetages de composantes connexes

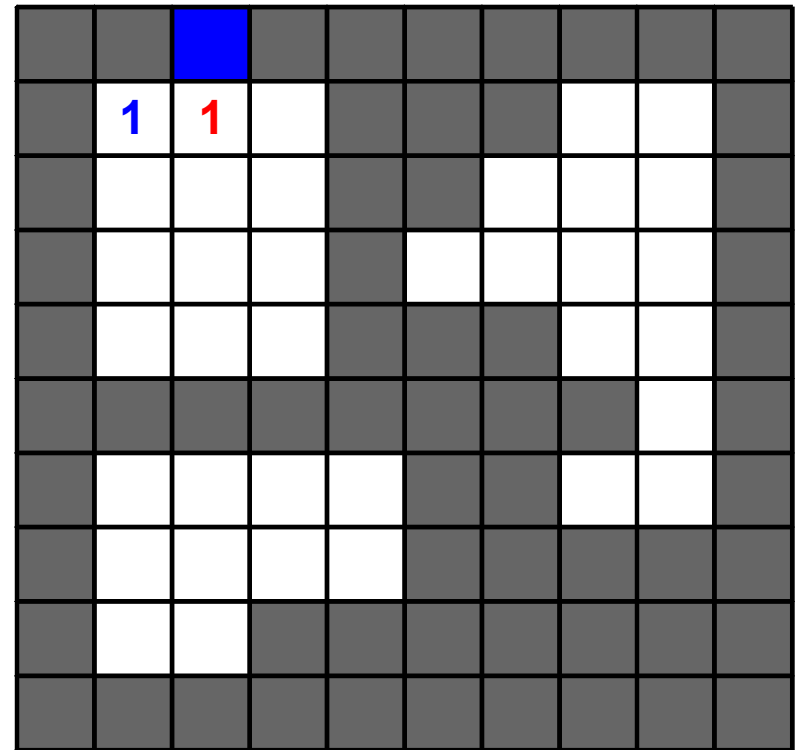
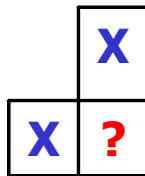
Premier parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - soit la plus petite étiquette parmi ses voisins **haut** et **gauche**
 - soit une nouvelle étiquette.

Parcours



Voisinage

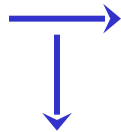


Etiquetages de composantes connexes

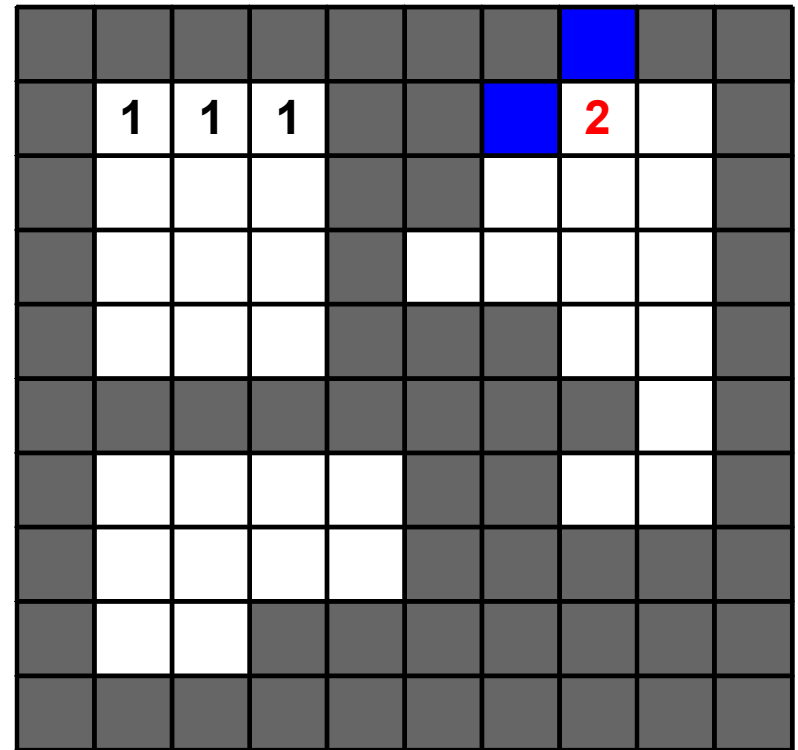
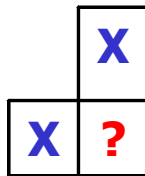
Premier parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - soit la plus petite étiquette parmi ses voisins **haut** et **gauche**
 - soit une nouvelle étiquette.

Parcours



Voisinage

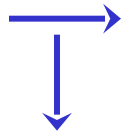


Etiquetages de composantes connexes

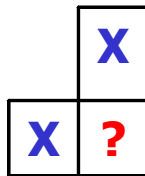
Premier parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - soit la plus petite étiquette parmi ses voisins **haut** et **gauche**
 - soit une nouvelle étiquette.

Parcours



Voisinage



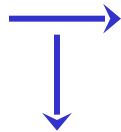
	1	1	1				2	2	
	1	1	1			3			

Etiquetages de composantes connexes

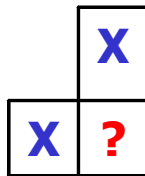
Premier parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - soit la plus petite étiquette parmi ses voisins **haut** et **gauche**
 - soit une nouvelle étiquette.

Parcours



Voisinage



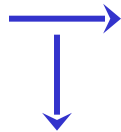
	1	1	1				2	2	
	1	1	1			3	2		

Etiquetages de composantes connexes

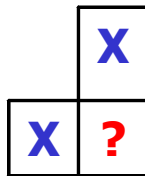
Premier parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - soit la plus petite étiquette parmi ses voisins **haut** et **gauche**
 - soit une nouvelle étiquette.

Parcours



Voisinage



	1	1	1				2	2	
	1	1	1			3	2	2	
	1	1	1		4	3	2	2	
	1	1	1				2	2	
								2	
	5	5	5	5			6	2	
	5	5	5	5					
	5	5							

Etiquetages de composantes connexes

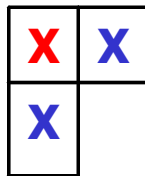
Deuxième parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - la plus petite étiquette parmi la sienne et celles ses voisins bas et droite

Parcours



Voisinage



	1	1	1				2	2	
	1	1	1			3	2	2	
	1	1	1		4	3	2	2	
	1	1	1				2	2	
								2	
	5	5	5	5			6	2	
	5	5	5	5					
	5	5							

Etiquetages de composantes connexes

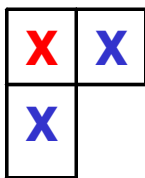
Deuxième parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - la plus petite étiquette parmi la sienne et celles ses voisins bas et droite

Parcours



Voisinage



	1	1	1				2	2	
	1	1	1			3	2	2	
	1	1	1		4	3	2	2	
	1	1	1				2	2	
								2	
	5	5	5	5			2	2	
	5	5	5	5					
	5	5							

Etiquetages de composantes connexes

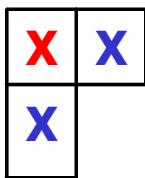
Deuxième parcours de l'image

- Pour chaque pixel d'une région, on lui affecte
 - la plus petite étiquette parmi la sienne et celles ses voisins bas et droite

Parcours



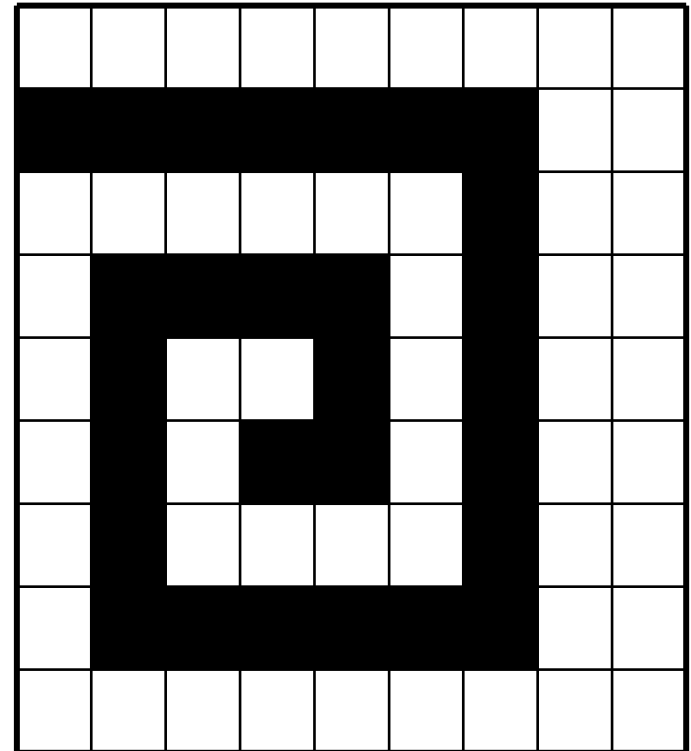
Voisinage



	1	1	1				2	2	
	1	1	1			2	2	2	
	1	1	1		2	2	2	2	
	1	1	1				2	2	
								2	
	5	5	5	5			2	2	
	5	5	5	5					
	5	5							

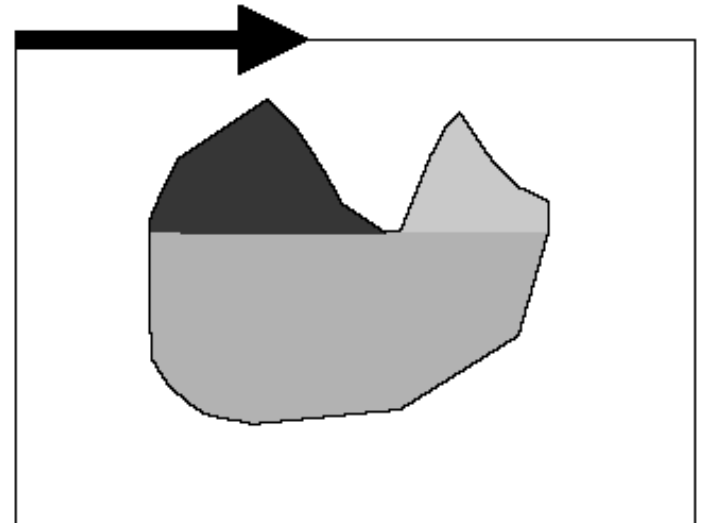
Etiquetages de composantes connexes

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

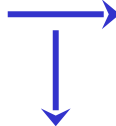


Etiquetages de composantes connexes

- Il est possible de ne faire qu'un seul parcours
 - Gestion d'une **table d'équivalence d'étiquettes**
 - **Mise à jour récursive** des étiquettes lorsque 2 étiquettes se « rencontrent »



Etiquetages de contours

- Pour étiqueter des contours, on peut utiliser le même algorithme
- On procède de la même façon, sauf qu'on utilise la 8-connexité
 - *on regarde 4 voisins au lieu de 2* 
- **A faire attention :** normalement l'épaisseur d'un contour doit être d'un seul pixel
 - *supprimer les pixels redondants (non-maxima)*
- Il existe d'autres algorithmes plus performants pour les contours

1	2	3
4	?	



Fermeture des contours

- Du gradient aux contours
 - *Les méthodes de gradient vus en classe ne donnent pas forcément des contours fermés*
 - *Il y a des trous et plusieurs morceaux isolés*
- On doit alors « fermer les contours » pour avoir une segmentation avec les contours équivalentes aux régions
- Plusieurs méthodes existent, mais rien de facile



Fermeture des contours

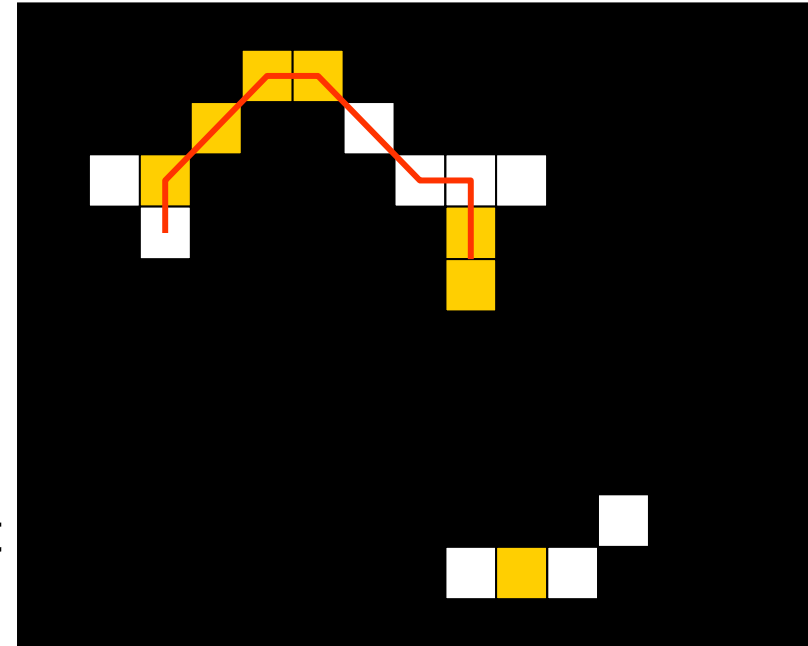
Épaisseur des contours

- Condition importante : un contour doit avoir une épaisseur de **1 pixel partout**
- Si le contour est trop épais, où si le contour présente plusieurs chemins, on doit **conserver les pixels de gradient le plus fort**

Fermeture des contours

Calcul et suivi du meilleur chemin dans le gradient

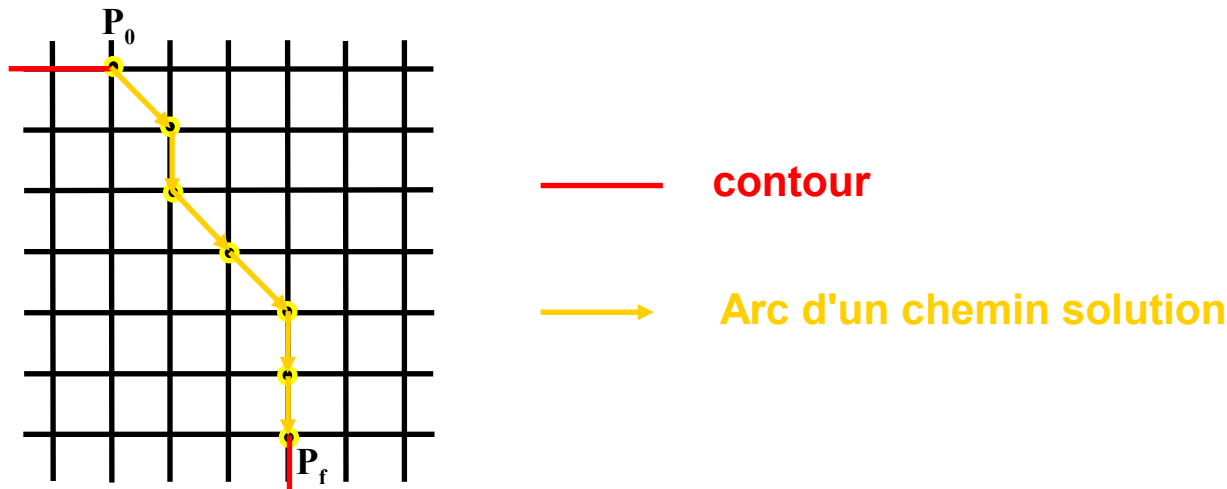
- On **identifie les extrémités** des chaînes des points de contour
- Analyse de la configuration de voisinage
- On cherche à **propager** ces extrémités :
 - Dans la **direction locale du gradient** (à condition que leur amplitude soit supérieure à s_b)
 - Vers celui des voisins immédiats ayant la **plus forte amplitude de gradient**



Fermeture des contours

Calcul et suivi du meilleur chemin dans le gradient

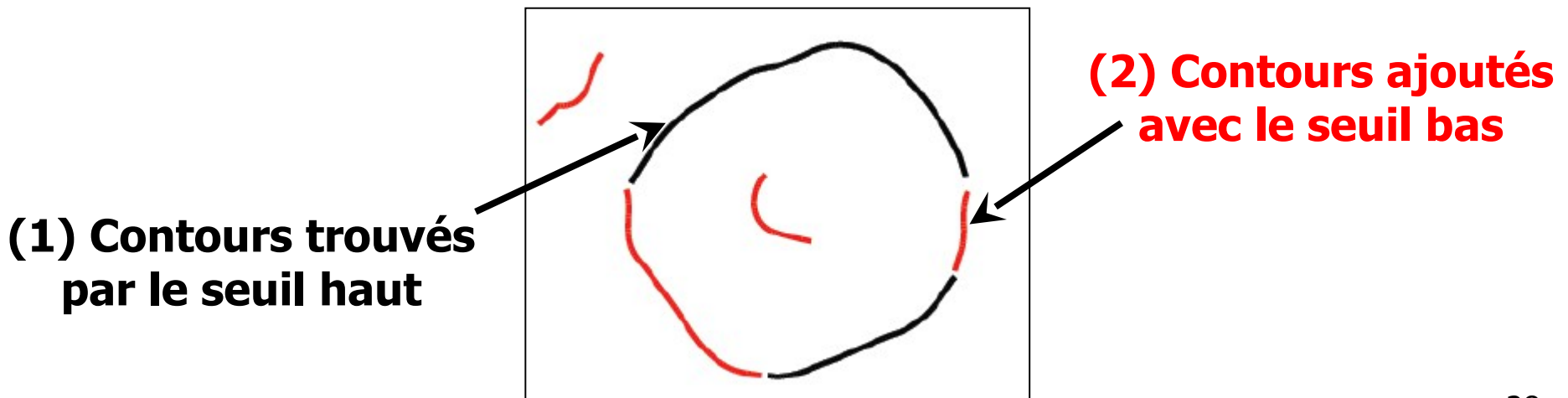
- Au bout des contours, on cherche le chemin permettant de rejoindre une autre extrémité de contour
 - *Suivi du chemin de gradient le plus fort*



Fermeture des contours

Seuillage par hystérésis

- Définir deux seuils
 - *seuil bas : minimum pour être contour*
 - *seuil haut : minimum pour être contour d'un objet*
- (1) Seuiller avec le seuil haut
- (2) Ajouter les contours connectés supérieur au seuil bas





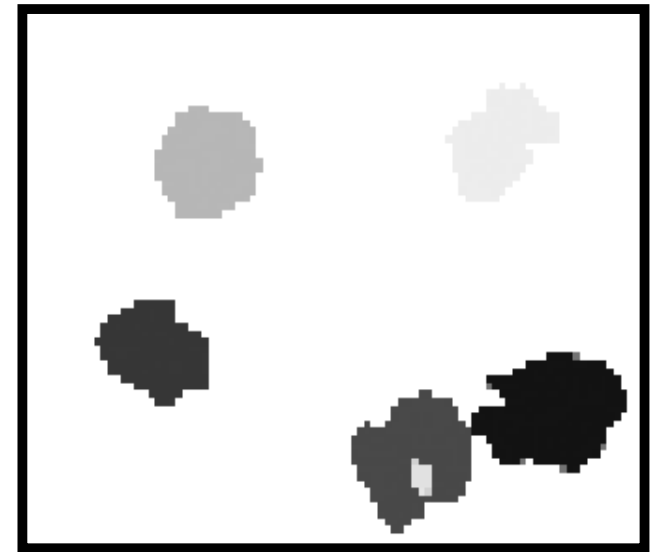
Opérateurs morphologiques

Opérateurs morphologiques

- Très utilisés sur les images binaires (images de masques)
 - *mais aussi sur les images en niveaux de gris*
- Permettent de modifier la morphologie des objets
 - Pour nettoyer le résultat de la segmentation
 - Remplir les trous, éliminer le bruit
 - Pour lisser le résultat de la segmentation

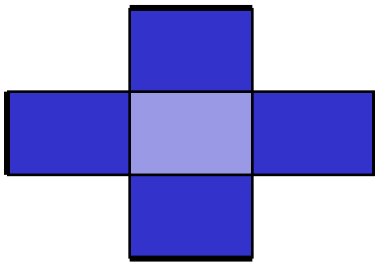
Utilisé en post-segmentation

- Caractérisés par
 - un élément structurant
 - des transformations
 - érosion, dilatation,
 - ouverture (érosion & dilatation), fermeture (dilatation & érosion)

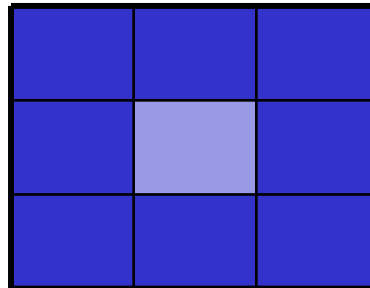


Opérateurs morphologiques

- Soit une image binaire avec **fond=0** et **objet=1**
- L'**élément structurant** "glisse" sur les bords (intérieurs, extérieurs) des objets et transforme sur son passage :
 - des pixels d'**objet** en pixels de **fond** (érosion)
 - des pixels de **fond** en pixels d'**objet** (dilatation)
- Exemple d'éléments structurants :



Connexité-4



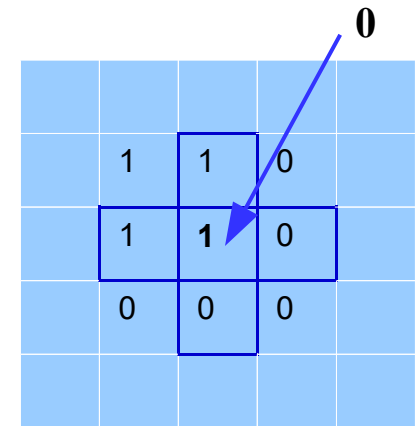
Connexité-8

Il existe d'autres formes d'éléments structurants, pas forcément symétriques

Opérateurs morphologiques

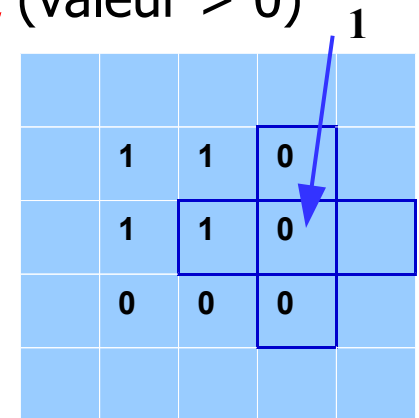
■ Erosion

- Si un des pixels du masque est *fond* (valeur 0) alors le pixel central devient *fond*



■ Dilatation

- Si un des pixels du masque fait partie de l'*objet* (valeur > 0) alors le pixel central devient *objet*



■ Ouverture :

- Erosion puis dilatation

■ Fermeture

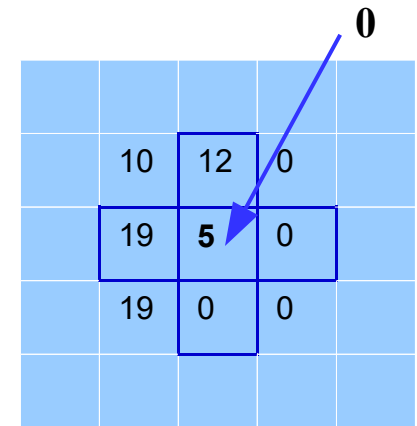
- Dilatation puis érosion

Opérateurs morphologiques

pour image en niveaux de gris

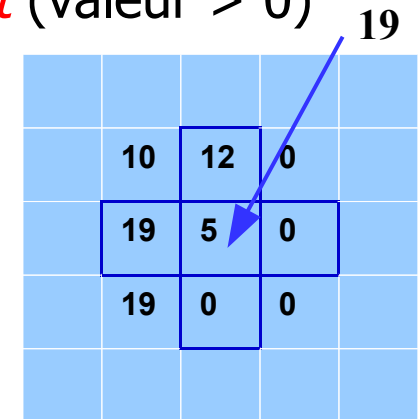
■ Erosion

- Si un des pixels du masque est *fond* (valeur 0) alors le pixel central devient *fond*
=> Image en niveaux de gris : filtre minimum



■ Dilatation

- Si un des pixels du masque fait partie de l'*objet* (valeur > 0) alors le pixel central devient *objet*
=> Image en niveaux de gris : filtre maximum



■ Ouverture :

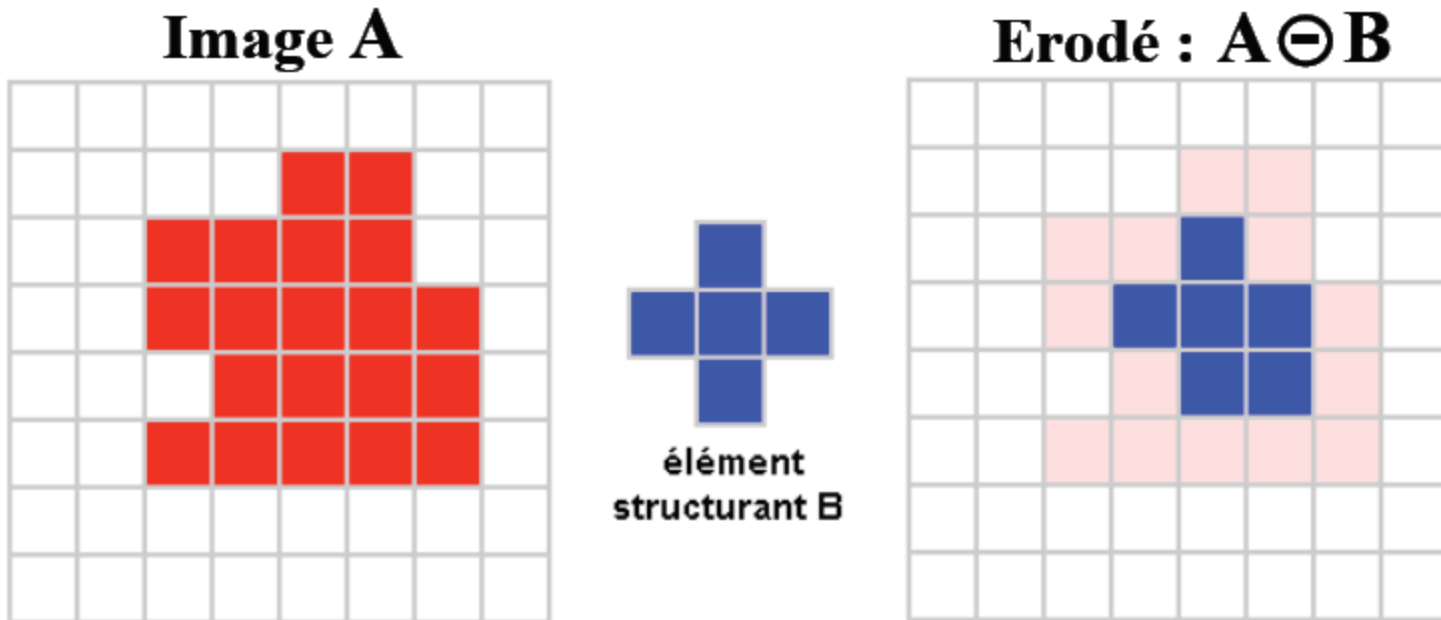
- Erosion puis dilatation

■ Fermeture

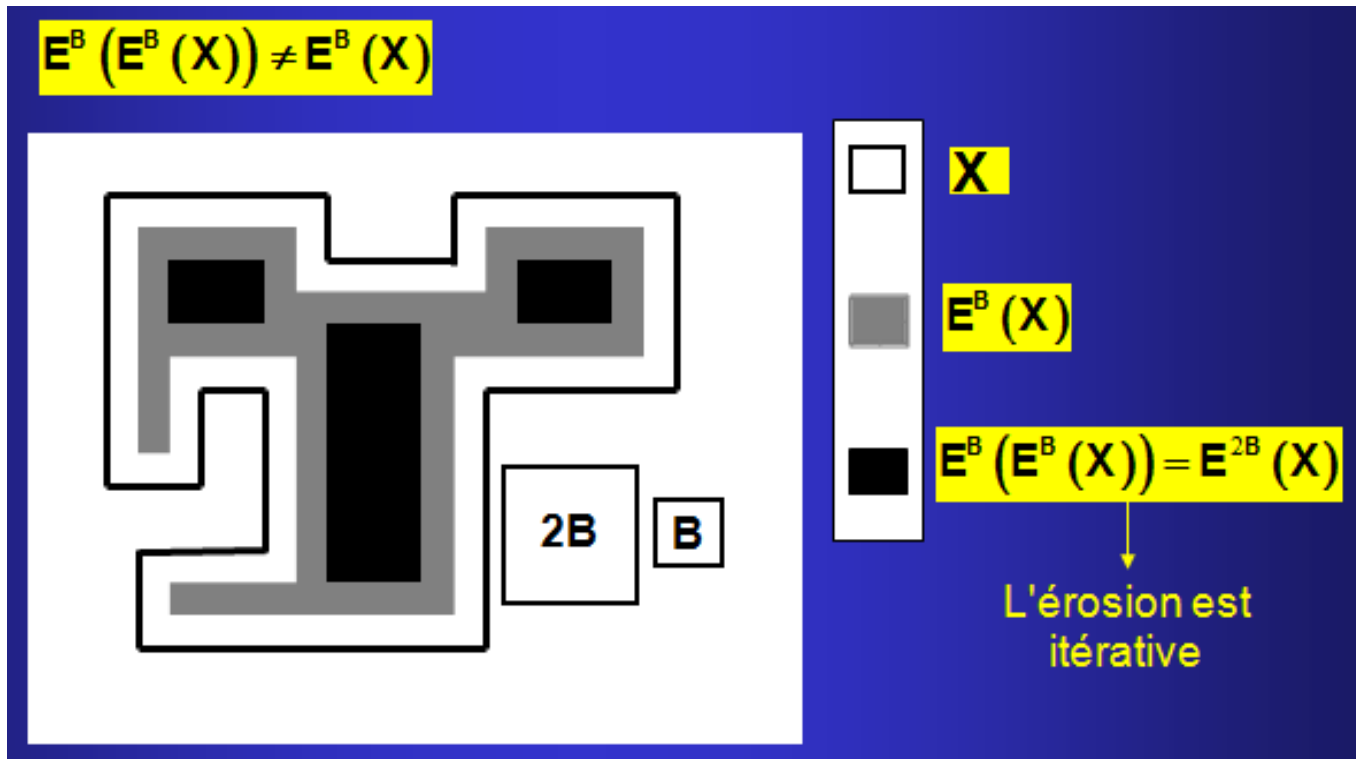
- Dilatation puis érosion

Erosion

- On positionne l'origine de B en chaque pixel x de l'objet A
- Si tous les pixels de B font partie de l'objet A, alors l'origine de B appartient à l'érodé



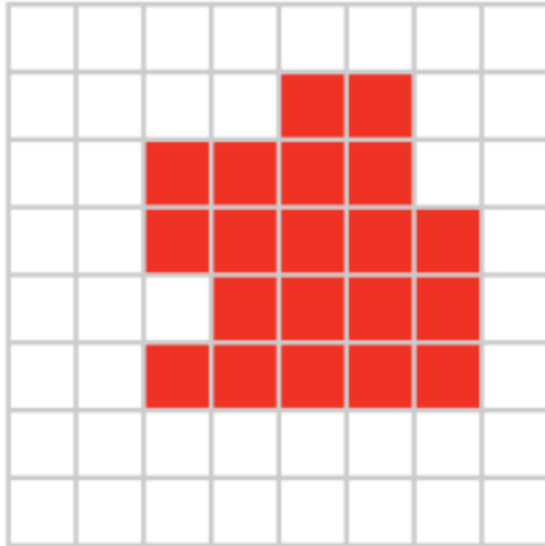
Erosion



Dilatation

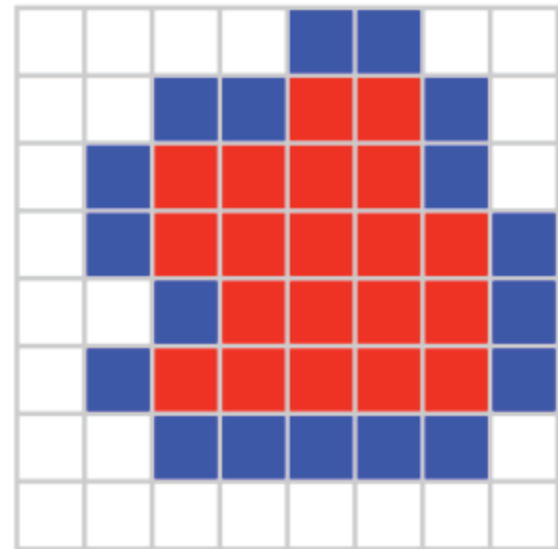
- Pour chaque position de B, est-ce que l'intersection entre B et l'objet A est non vide ?
 - Si oui, **x l'origine de B** appartient à l'image dilatée

Image A

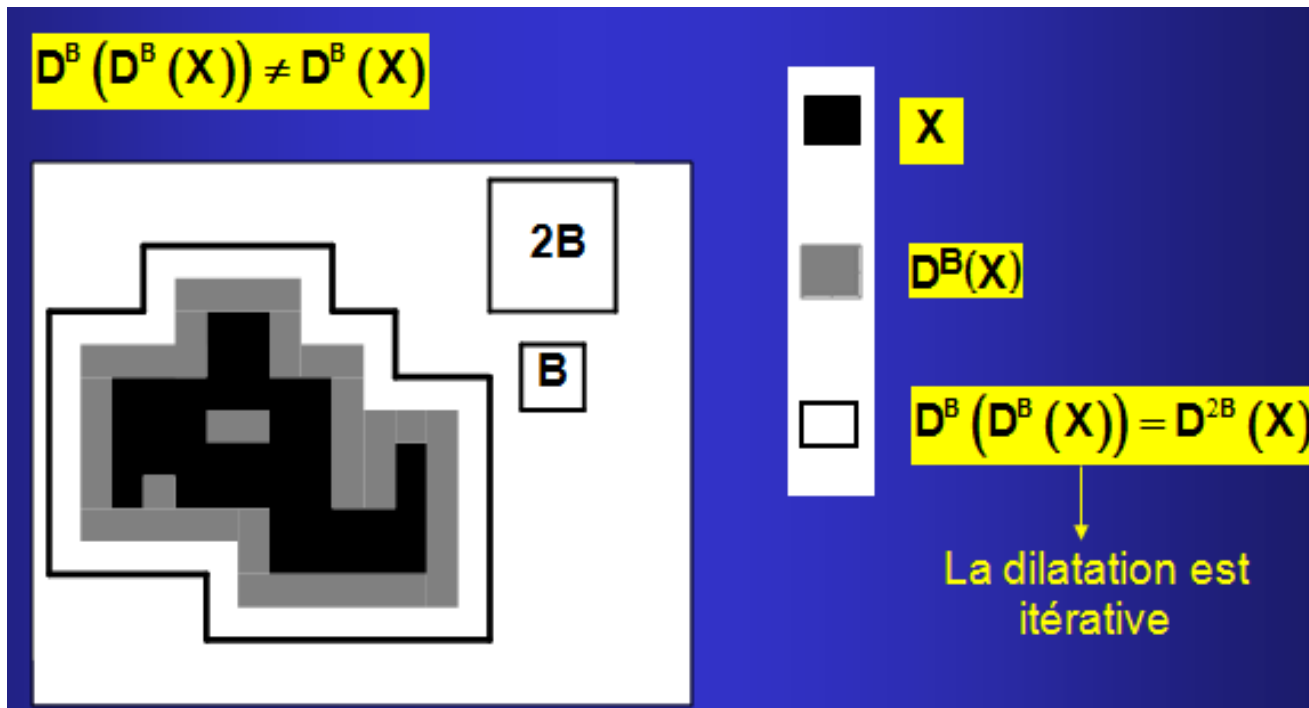


**élément
structurant B**

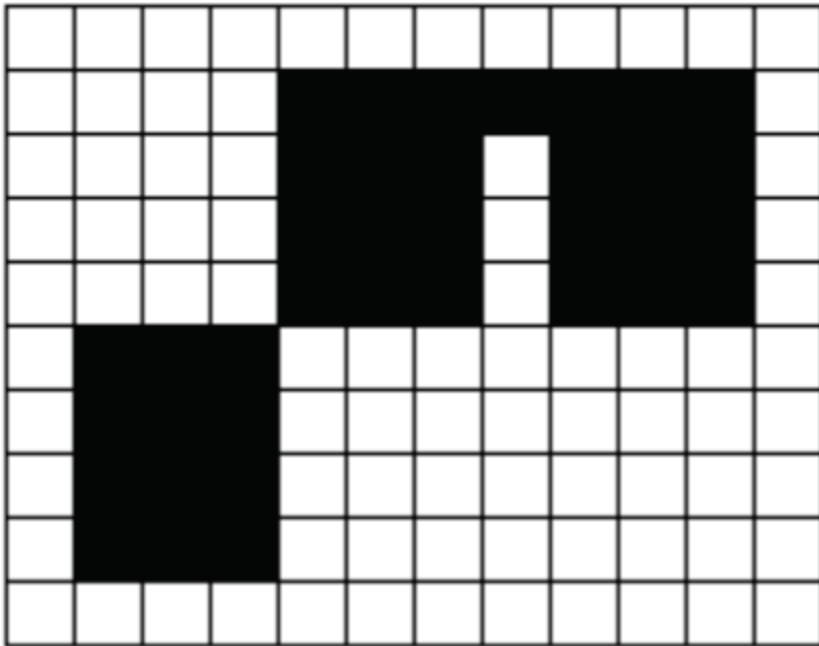
Dilaté : $A \oplus B$



Dilatation



Combien d'objets ?



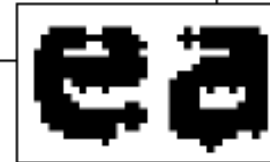
- Combien d'objets ?
 - 1 ou 2 ?
- Et après une érosion ?
- Et après une dilatation ?
- *Risque de fusion ou séparation d'objets...*

Exemple de dilatation

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



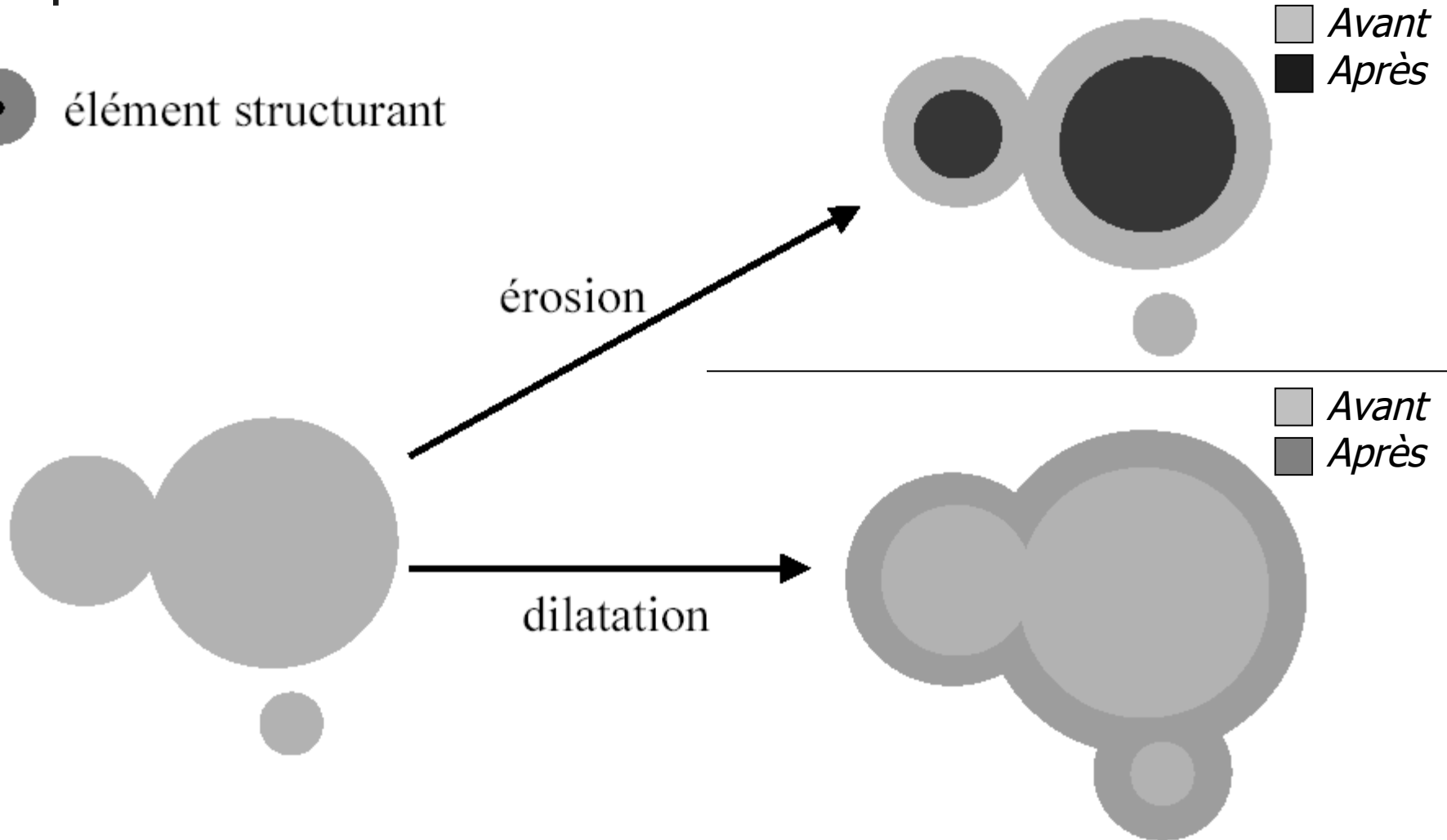
0	1	0
1	1	1
0	1	0

a c
b

FIGURE 9.5
(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

Erosion - dilatation

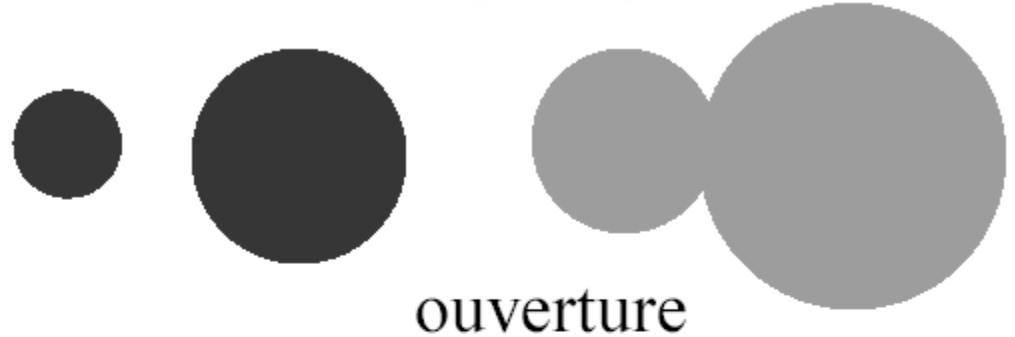
● élément structurant



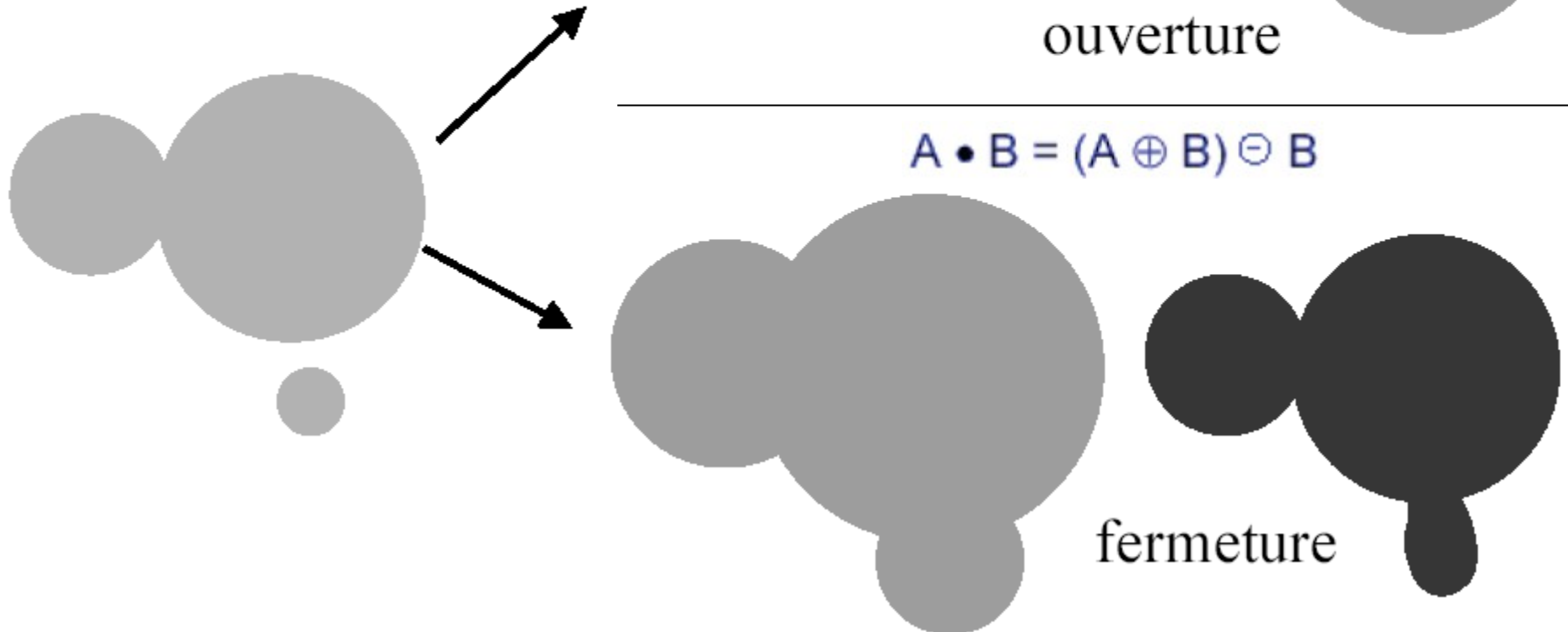
Ouverture - fermeture

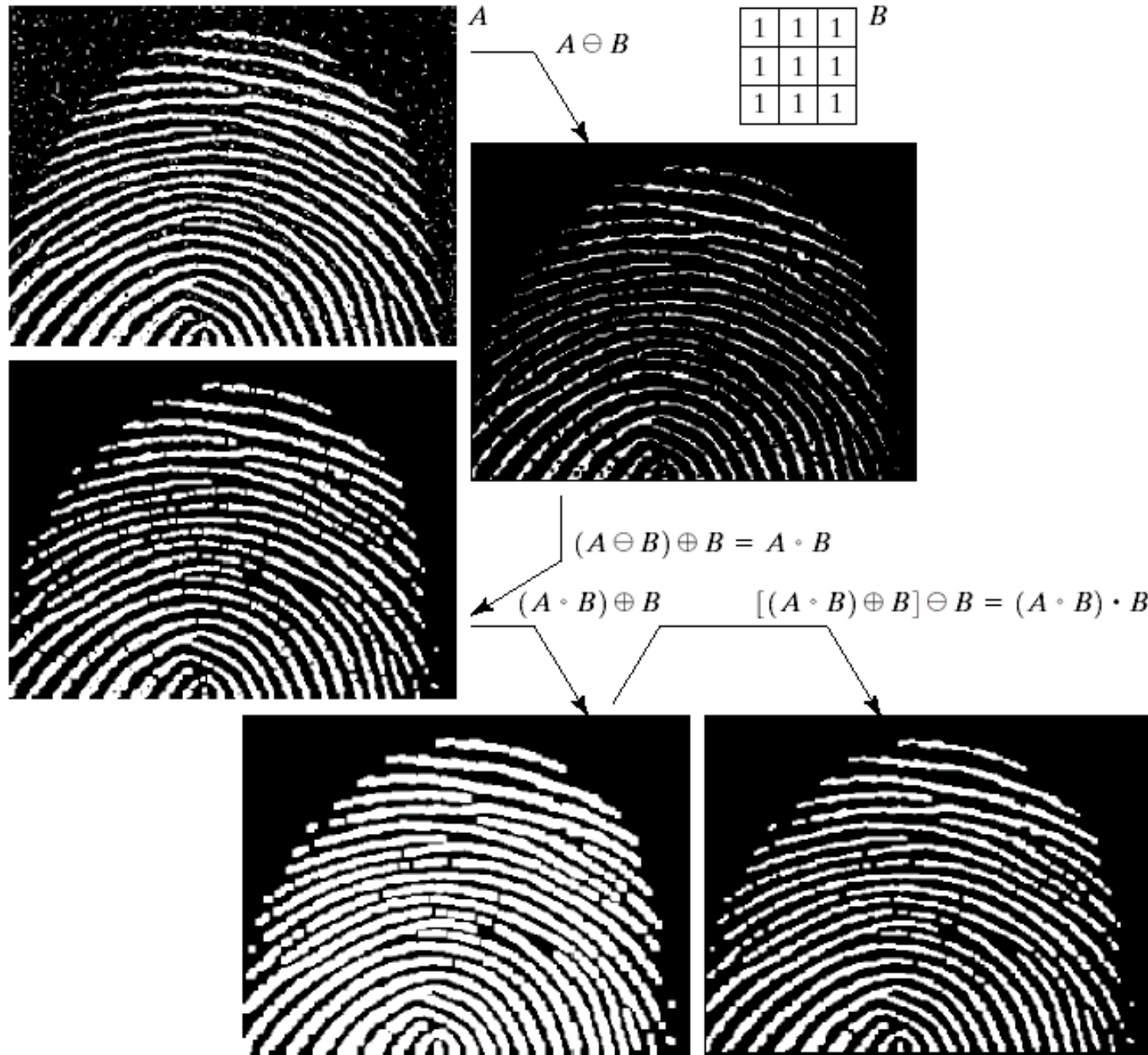
• élément structurant

$$A \circ B = (A \ominus B) \oplus B$$



$$A \bullet B = (A \oplus B) \ominus B$$





a b
d c
e f

FIGURE 9.11
 (a) Noisy image.
 (c) Eroded image.
 (d) Opening of A .
 (d) Dilation of the opening.
 (e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

Influence de l'élément structurant



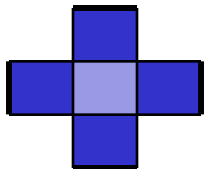
Image originale



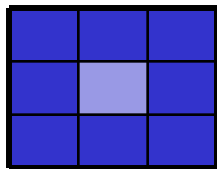
Dilatation 4-voisins



Dilatation 8-voisins



4-voisins



8-voisins



Erosion 4-voisins

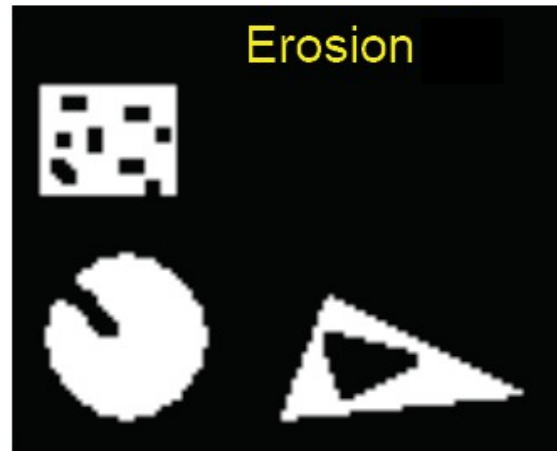


Erosion 8-voisins

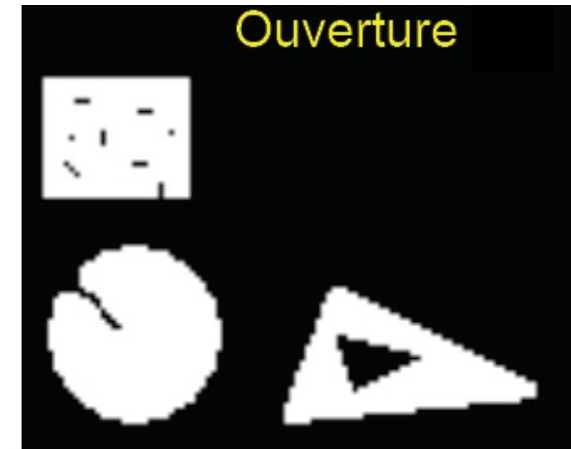
Exemple récapitulatif



Image originale



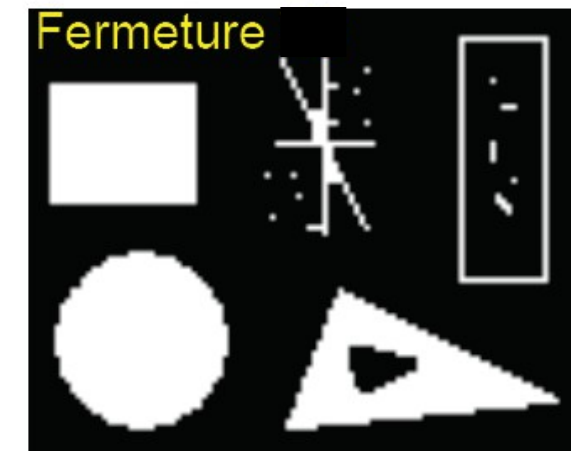
Erosion



Ouverture



Dilatation



Fermeture

Gradient interne, externe

Gradient interne : contour intérieur



Image originale

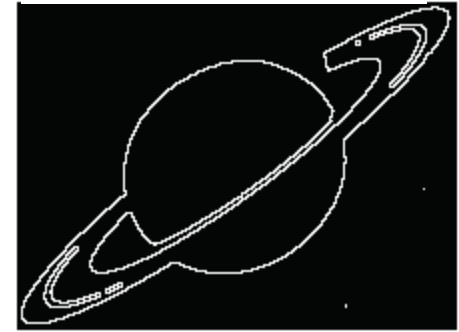
–



Image érodée

=

$$A - (A \ominus B)$$



Contour intérieur

Gradient externe : contour extérieur

$$(A \oplus B) - A$$



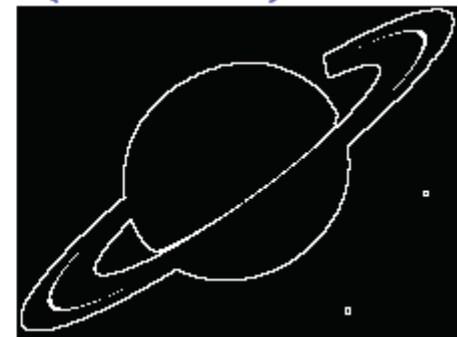
Image dilatée

–



Image originale

=



Contour extérieur

Gradient morphologique

$$(A \oplus B) - (A \ominus B)$$



Image dilatée



Image érodée



Gradient morphologique



Références

■ Livres:

- Introduction au Traitement d'Images (Lingrand), chapitre 7, section 8.4
- Digital Image Processing 2ed (Gonzalez & Woods), chapitre 9, sections 10.1 et 10.2, chapitre 11

- Caroline Rougier. Cours de Traitement d'images (IFT2730). Université de Montréal (Canada)

<http://www-etud.iro.umontreal.ca/~rougierc/ift2730/>

Chap11 : Morphologie mathématique :

http://www-etud.iro.umontreal.ca/~rougierc/ift2730/cours/Cours11_IFT2730_2008_2.pdf

- Cours de segmentation. Site lagaule.org (France)

http://lagaule.org/downloads/TC3_03_segmentation.ppt

- Jérôme Vicente. Cours de Traitement d'images. Polytech'Marseille (France)

<http://iusti.polytech.univ-mrs.fr/~vicente/enseignement.html>

<http://iusti.polytech.univ-mrs.fr/~vicente/supportcours/vision4-2006.pdf>