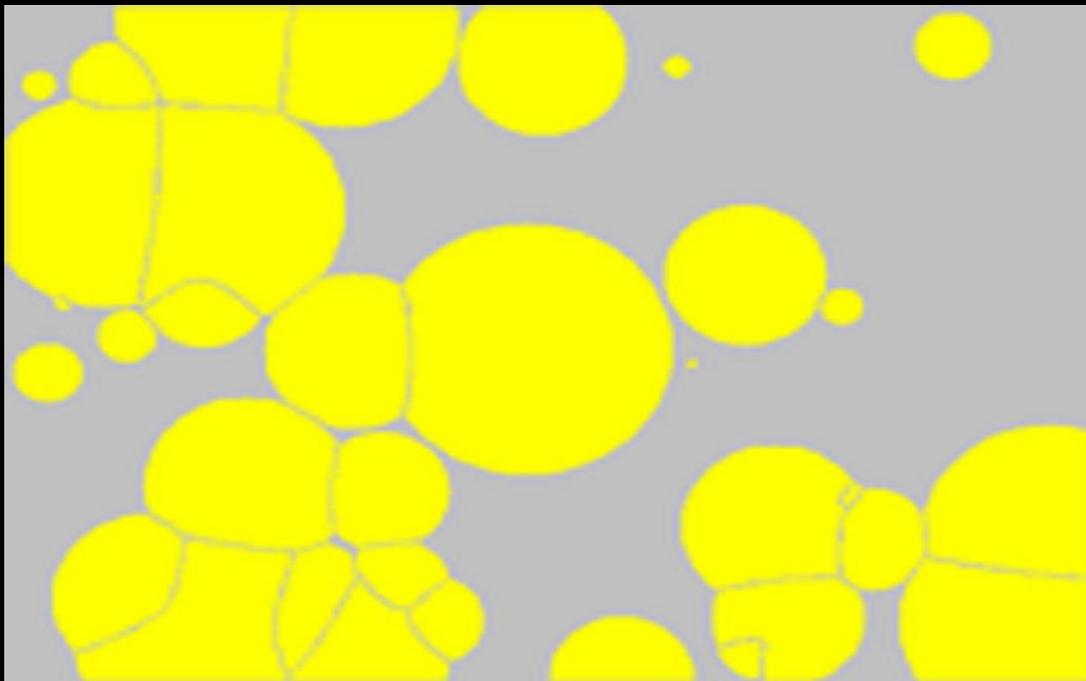


**Spécialité**  
**Automatique et Informatique Industrielle**

**Cours :**  
**IMAGERIE ET VISION INDUSTRIELLE**

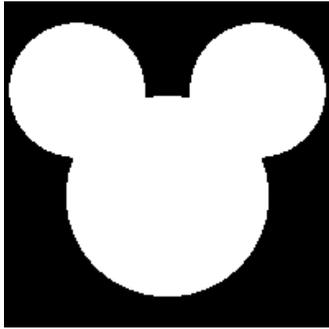
**Chap. 7**  
**Traitement d'Images Binaires**  
**Par Morphologie Mathématique**



# Traitement d'images binaires par Morphologie Mathématique

## Contenu de ce chapitre

1. Introduction
2. Images binaires: premier-plan, arrière-plan et connexité
3. **Opérateurs de base**
  - 3.1 Éléments structurants et Voisinage
  - 3.2 Opérateurs de Logique Binaire
  - 3.3 Opération de Translation dans une image
  - 3.4 Opération de Réflexion
4. **Filtrage ou Opération morphologique**
  - 4.1 Principe d'une opération morphologique
  - 4.2 Dilatation (*dilation*)
  - 4.3 Erosion (*erosion*)
5. **Opérations basées sur la dilatation et l'érosion**
  - 5.1 Ouverture morphologique. (Morphological Opening)
  - 5.2 Fermeture morphologique. (Morphological closing)
  - 5.3 Squelettisation des objets dans une image binaire
  - 5.4 Le périmètre des objets dans une image binaire
  - 5.5 Transformation Tout-ou-Rien (hit-or-miss).
  - 5.6 Transformation morphologique du chapeau-haut (top-hat)
  - 5.7 Transformation morphologique du chapeau-bas (bottom-hat)
6. **Conclusion**



## Traitement d'images binaires par Morphologie Mathématique

### 1. Introduction

Le mot morphologie signifie l'étude de la forme ou de la structure. Dans le traitement d'image, nous utilisons morphologie mathématique comme moyen d'identifier et d'extraire des descripteurs d'images significatifs en fonction des propriétés de forme dans l'image. Les principaux domaines d'application sont la segmentation et l'analyse d'images orientés vers le comptage et l'inspection d'objets automatisés. La morphologie englobe un ensemble puissant et important de méthodes qui peuvent être traitées, mathématiquement, avec précision dans le cadre de la théorie des ensembles. La discipline qui traite de ces problèmes est la morphologie mathématique dont le principal objectif est de mettre en évidence la morphologie (forme) d'un objet. La morphologie mathématique (*mathematical morphology*) analyse et modifie la structure des images.

Les opérations morphologiques peuvent être appliquées à des images de tous types, mais l'utilisation essentielle de la morphologie mathématique est destinée au traitement images binaires et les deux principaux opérateurs morphologiques sont relativement simples appelés dilatation et érosion. Il est en effet possible de montrer que de nombreuses procédures morphologiques plus sophistiquées peuvent être réduites à une séquence de dilatations et d'érosions.

### 2. Images binaires: premier-plan, arrière-plan et connexité

La binarisation d'une image présente plusieurs avantages dont le faible espace mémoire occupé par une image binaire et la simplicité des opérateurs qui lui sont associés. La binarisation est aussi la première étape d'isolement des objets par rapport au fond.

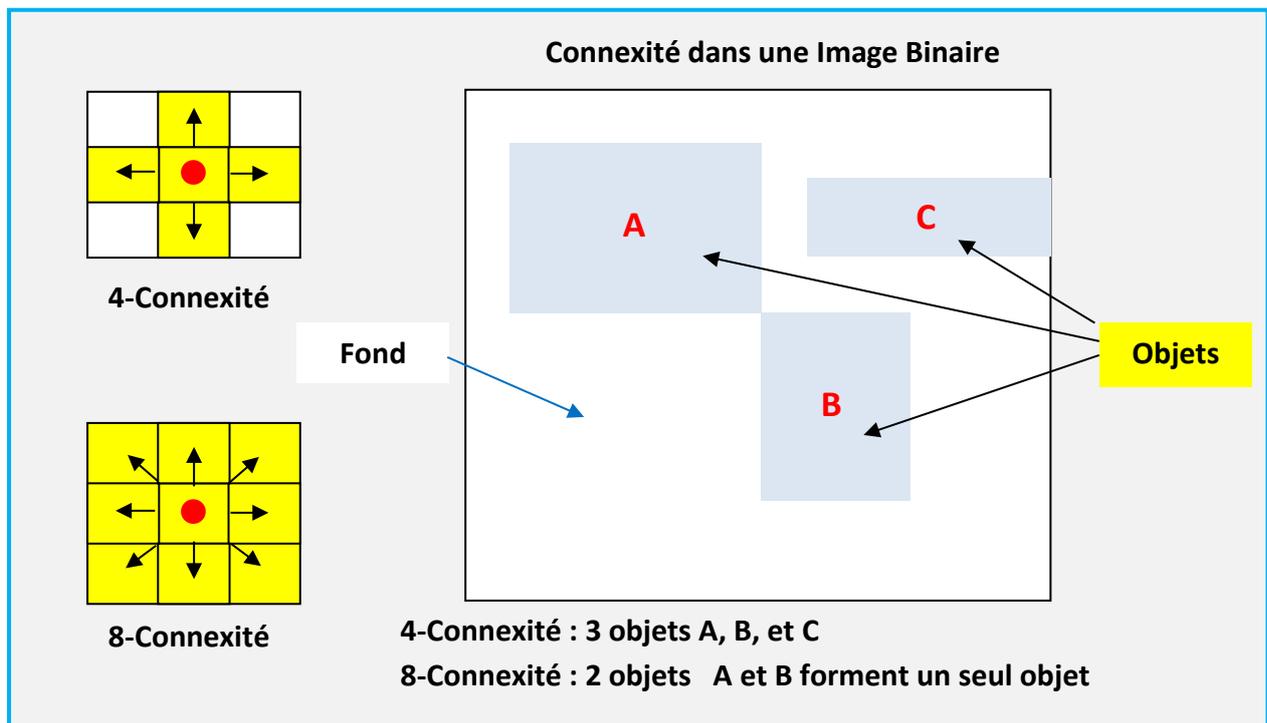
Une image binaire, par exemple issue d'un seuillage, est une image dans laquelle chaque pixel suppose l'un des deux seuls valeurs discrètes possibles, valeurs logiques: 1 ou 0. En traitement d'image, les pixels dans une image

binaire ayant la valeur logique 1 appelés pixels de l'image du premier plan (**pixel d'une forme ou objet**), tandis que les pixels ayant une valeur logique 0 sont appelés pixels de l'arrière-plan (**pixel du fond**) de l'image. Un objet dans une image binaire se compose d'un ensemble de pixels connectés. Deux définitions de la connectivité entre pixels sont couramment utilisées :

**4- connexité** : tout pixel du premier plan doit avoir au moins un pixel de premier plan dans l'une des 4 directions (Nord, Sud, Est, Ouest) comme voisin de lui-même pour être considéré comme faisant partie du même objet.

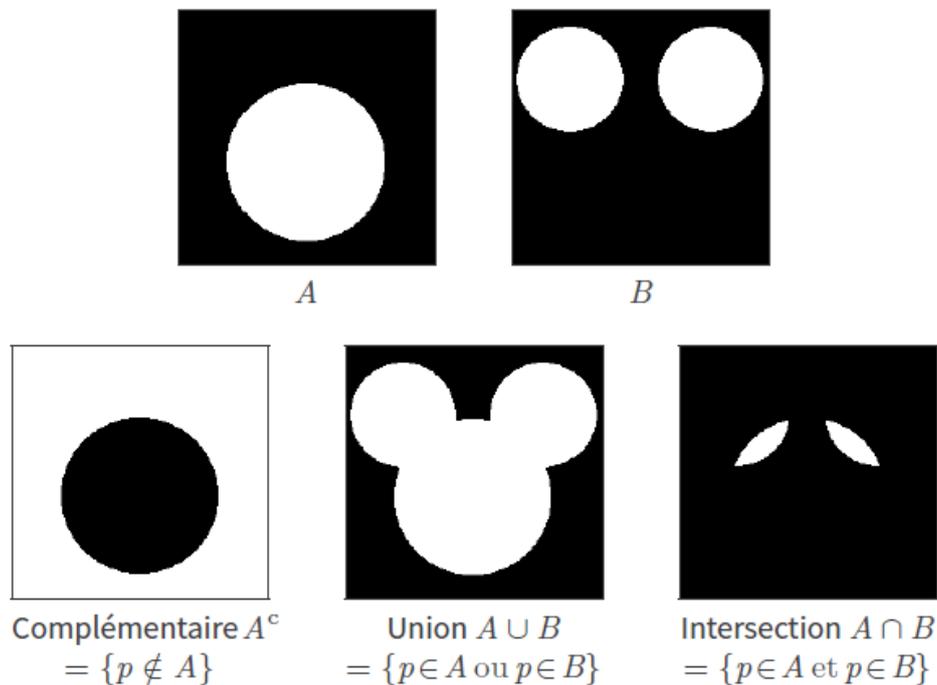
**8- connexité** : tout pixel du premier plan doit avoir au moins un pixel de premier plan dans l'une des 8 directions (Nord, Sud, Est, Ouest, Nord-Est, Nord-Ouest, Sud-Est, Sud-Ouest) comme voisin de lui-même pour être considéré comme faisant partie du même objet.

Ces simples concepts sont illustrés à la figure ci-dessous:



L'image binaire représentée ci-dessus contient deux objets (groupes de pixels connectés) sous 8-connexité mais trois objets si on considère la 4-connexité

## Rappels sur les ensembles



De par leur nature même, les images binaires n'ont pas de contenu textuel (c'est-à-dire échelle de gris ou couleur); ainsi, les seules propriétés d'intérêt dans les images binaires sont la forme, la taille et l'emplacement des objets dans l'image. Les opérations morphologiques peuvent être étendues aux niveaux de gris et des images en couleur, mais il est plus facile, au moins au début, de considérer les opérations morphologiques comme fonctionnant sur une entrée d'image binaire pour produire en sortie une image binaire modifiée. Dans cette perspective, l'effet de tout traitement morphologique se réduit simplement à la détermination des pixels de premier plan qui deviennent en arrière-plan et quels sont les pixels d'arrière-plan qui deviennent au premier plan.

D'une manière assez générale, qu'un pixel de premier plan ou d'arrière-plan donné ait ou non sa valeur modifiée dépend de trois facteurs :

1. **Image binaire à modifier**
2. **Type d'opération morphologique à effectuer**
3. **Élément structurant.**

Les deux premiers facteurs sont assez évidents. Le troisième facteur, appelé l'élément structurant, est un élément clé dans toute opération morphologique. L'élément structurant est l'entité ou motif qui détermine exactement quels sont les pixels entourant le pixel d'avant-plan / d'arrière-plan donné qui doivent être pris en compte afin de rendre la décision de modifier ou non la valeur du pixel en question. Le choix particulier de l'élément structurant est central au traitement morphologique.

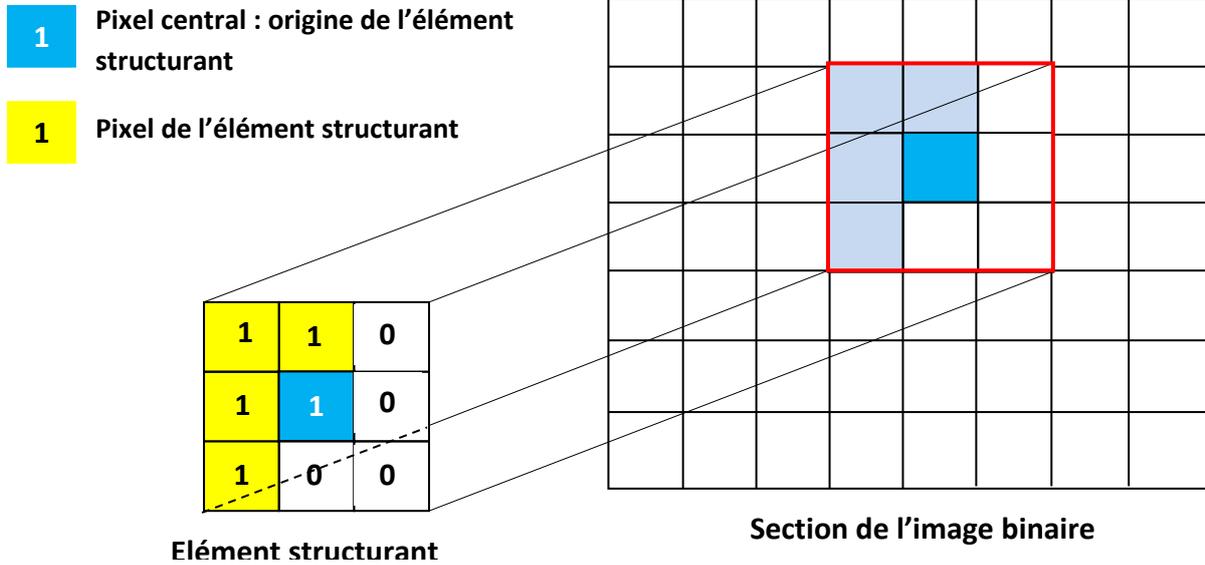
### 3. Opérateurs de base

#### 3.1 Éléments structurants et Voisinage

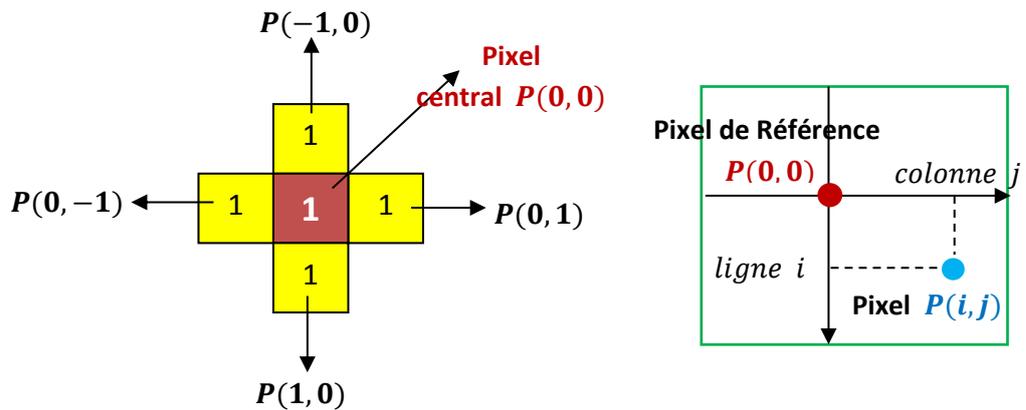
Un élément structurant (*structuring element*)  $E$  est un ensemble de pixels repéré par une origine (qui peut ne pas appartenir à l'élément structurant). L'élément structurant est donc un ensemble de pixels contenant les valeurs 1 ou 0 (semblable à une petite image binaire) formant une structure géométrique bien définie. Les éléments structurants ont un pixel central de coordonnées (0,0) dans cette structure.

Le pixel central de l'élément structurant étant placé directement au-dessus du pixel considéré dans l'image, le voisinage de ce pixel est déterminé par les pixels qui se trouvent superposés avec les pixels ayant la valeur 1 dans l'élément structurant. On note  $E_x$  l'élément structurant centré sur le pixel  $x$ .

Quelques exemples d'éléments structurants morphologiques.  
Le pixel central de chaque l'élément structurant est indiqué en jaune



Le voisinage local défini par un élément structurant. Ceci est donné par les pixels foncés de l'image situés sous les pixels de valeur 1 dans l'élément structurant.



### 3.2 Opérateurs de Logique Binaire

Du fait de leur codage sur deux niveaux, chaque pixel d'une image binaire peut être codé comme un élément logique. Les opérateurs morphologiques sont issus des études sur la morphologie mathématique et reposent sur le concept de transformation géométrique d'une image par un élément structurant. L'élément structurant est un masque de forme quelconque dont les éléments forment un motif. L'application de l'opérateur morphologique consiste à balayer l'image avec ce masque et à effectuer pour chaque pixel une mise en correspondance du pixel et de ses voisins avec le motif du masque, puis d'en effectuer l'opération logique de l'union ou l'intersection ou toute autre combinaison.

## Rappel

On définit  $X, Y, Z$  : trois variables booléennes qui prennent leurs valeurs dans  $\{0, 1\}$

- Opérateur de Négation (ou Complément ation) **NON** : noté  $Z = \bar{X}$

$X$	$Z = \bar{X}$
0	1
1	0

- Opérateur **ET (logical AND)** :  $Z = X \text{ AND } Y$ , noté  $Z = X \cdot Y$

$X$	$Y$	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

- Opérateur **OU (logical OR)** :  $Z = X \text{ OR } Y$ , noté  $Z = X + Y$

$X$	$Y$	$Z = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

### 3.3 Définitions

#### - Opérateur ET logique (AND)

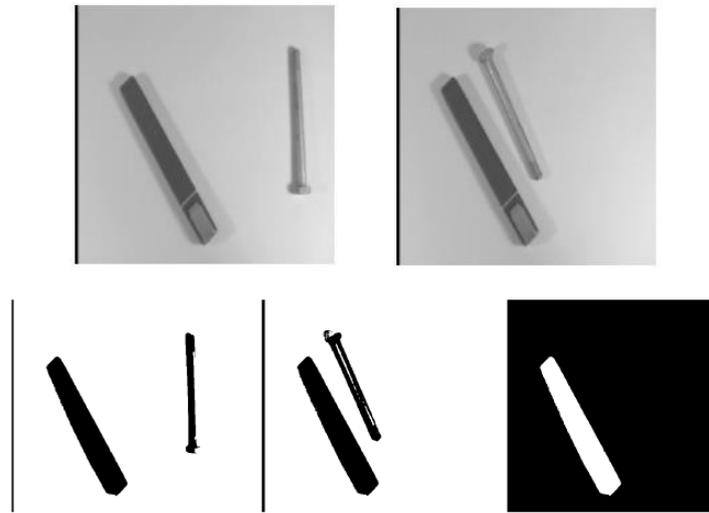
- Deux images binaires de même taille
  - Chaque valeur de pixel de l'image résultat est obtenue par un ET logique sur les pixels
  - Pour les images en niveaux de gris, l'opérateur logique est appliqué sur la représentation binaire des niveaux de gris, en comparant les bits correspondants.
- Exemple : 2 niveaux de gris 47 et 252 codés sur 8bits,

**47** en binaire : **00101111**

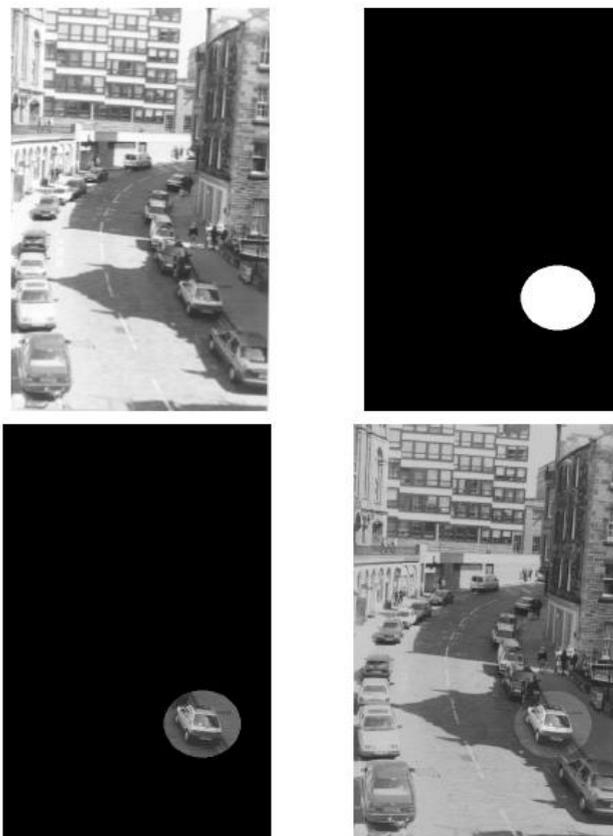
**252** en binaire : **11111100**

Résultat : (**00101111**) ET (**11111100**) = **00101100** soit **44** en niveaux de gris

- Application : Intersection de 2 images.



- Application : Masque pour isoler une région



### - Opérateur **OU** logique (**OR**)

- Entrée : deux images binaires de même taille
- Sortie : une image binaire de même taille
- Chaque valeur de pixel de l'image résultat est obtenue par un **OU** logique sur les pixels.
- Pour les images en niveaux de gris, l'opérateur logique est appliqué sur la représentation binaire des niveaux de gris, en comparant les bits correspondants.

– Exemple : 2 niveaux de gris 47 et 252 codés sur 8bits,

**47** en binaire : **00101111**

**252** en binaire : **11111100**

Résultat : (**00101111**) **OU** (**11111100**) = **11111111** soit **255** en niveaux de gris

- **Application : Fusion de 2 images binaires**



Les objets sont codés 0 et le fond 1. Inverser les images puis faire le 'ou'

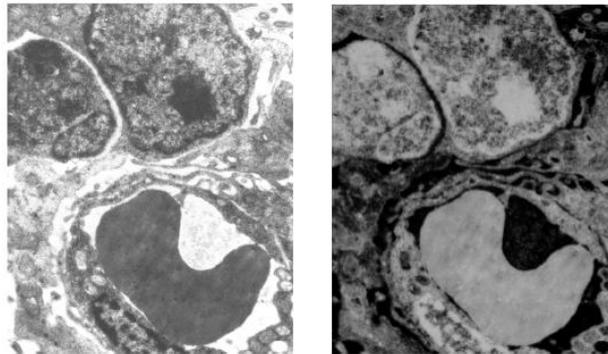
### - Opérateur **NON** logique (**NOT**)

- Aussi appelé opérateur d'inversion ou complémentation
- Pour une image binaire, il change la polarité des pixels
- Pour les images en niveaux de gris,

$$I_2(i,j) = 255 - I_1(i,j)$$

- En niveaux de gris, la négation fait apparaître plus clairement des caractéristiques.

- Les objets apparaissent en blanc sur fond noir ce qui est plus approprié à la vision humaine
- Application : Négatif d'une coupe de tissu biologique

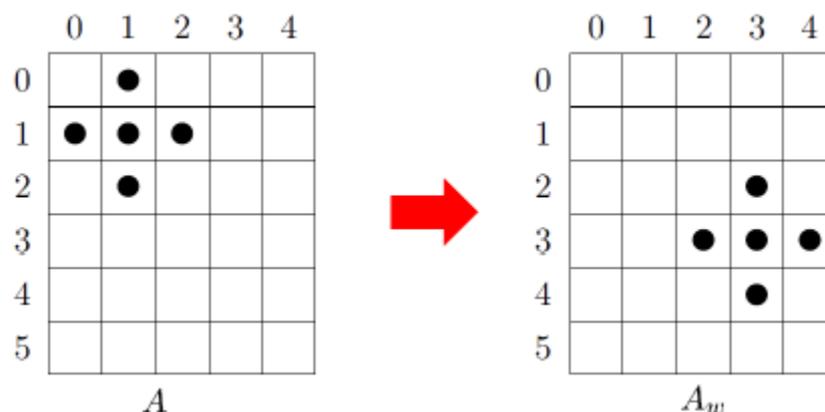


### 3.4 Opération de Translation dans une image

- $A$  un ensemble de pixels dans une image binaire
- $w = (x, y)$  coordonnées d'un point particulier de l'image
- $A_w$  est l'image translatée de  $A$  dans la direction  $(x, y)$

$$A_w = \{(a, b) + (x, y) : (a, b) \in A\}$$

Exemple : translation de  $A$  pour  $w = (2, 2)$

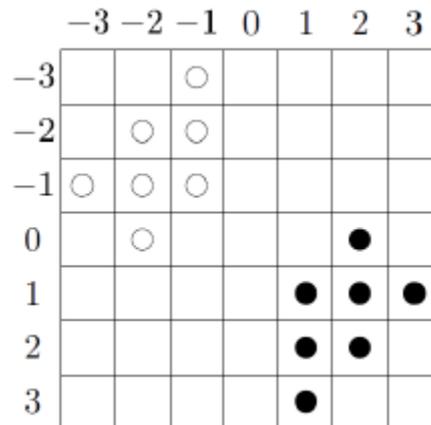


### 3.5 Opération de Réflexion

- A un ensemble de pixels dans une image binaire
- $A_R$  est l'image réfléchie de A définie par :

$$A_R = \{(-a, -b) : (a, b) \in A\}$$

- Exemple d'une réflexion de A



## 4. Filtrage ou Opération morphologique

Le filtrage morphologique repose sur la morphologie mathématique, basé sur une description ensembliste des images. Les opérateurs morphologiques privilégient la notion de forme plutôt que l'information sur l'amplitude des intensités des signaux. Ils s'appliquent aussi bien aux images binaires qu'aux images monochromes (en niveaux de gris) ou couleurs.

On se limitera par la suite au filtrage morphologique sur une image binaire. Ce filtrage non-linéaire fait appel à deux opérateurs de base (***l'érosion et la dilatation***) et à deux opérateurs complémentaires combinant les deux premiers (***l'ouverture et la fermeture***).

Ces opérateurs morphologiques utilisent une forme de référence avec laquelle le signal d'image est comparé localement. Cette forme de référence c'est l'élément structurant.

### 4.1 Principe d'une opération morphologique

L'élément structurant B se déplace sur une image A comme un opérateur de convolution. En chaque position de B, des opérations logiques (essentiellement non linéaire) sont effectuées entre A et B. Les résultats dépendent de la taille et de la forme de l'élément structurant utilisé.

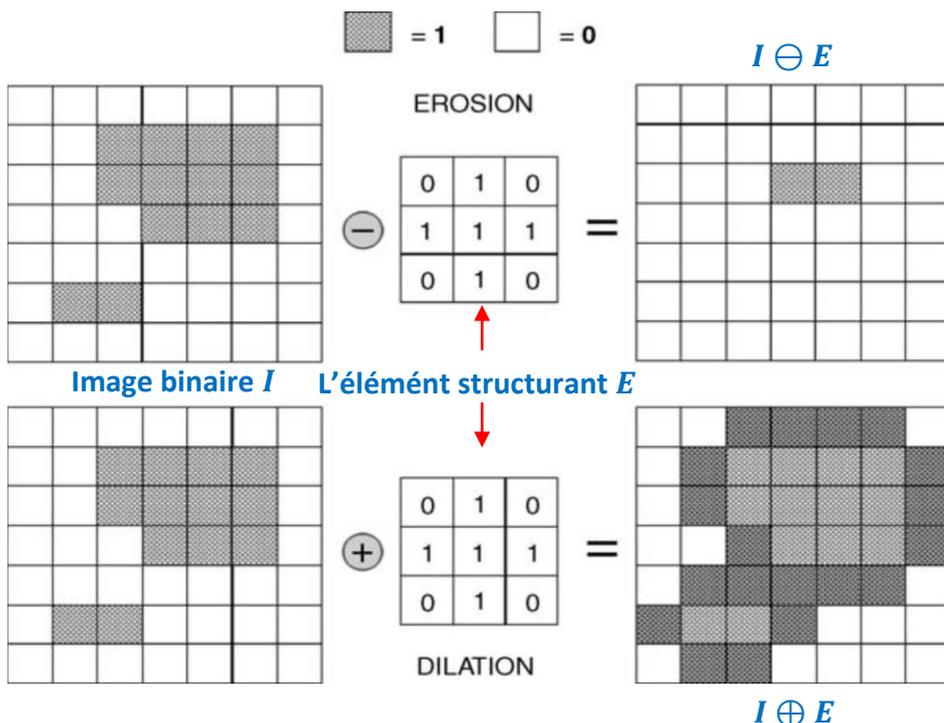
• **Dilatation et érosion**

Les deux opérateurs morphologiques de base les plus importants sont la dilatation et l'érosion. Toutes les autres opérations morphologiques peuvent être définies en fonction de ces opérateurs primitifs. Nous désignons une image générale par  $I$  et un élément structurant arbitraire par  $E$  et on parle de l'érosion / dilatation de  $I$  par  $E$ .

La binarisation d'une image fortement bruitée donne souvent naissance à une image difficile à analyser en termes d'éléments caractéristiques. Les 2 opérateurs morphologiques améliorant les images binaires sont : la dilatation et l'érosion. La dilatation permet d'éliminer les points noirs de l'image et l'érosion d'éliminer les points blancs.

Le principe de la dilatation et de l'érosion fonctionne de manière très similaire à la convolution à noyaux employés dans le filtrage spatial. L'élément structurant est glissé sur l'image afin que son pixel central se trouve successivement au-dessus de chaque pixel de premier plan ou d'arrière-plan selon le cas. La nouvelle valeur de chaque pixel d'image dépend alors des valeurs des pixels dans le voisinage défini par l'élément structurant.

Formellement, la dilatation de l'image  $I$  par l'élément structurant  $E$  est notée  $I \oplus E$  et l'érosion de l'image  $I$  par l'élément structurant  $E$  est notée  $I \ominus E$ .



## 4.2 Dilatation (*dilation*)

La valeur du pixel de sortie est la valeur maximale de tous les pixels du voisinage. Dans une image binaire, un pixel est remis à 1 si l'un des pixels voisins a la valeur 1. La dilatation morphologique rend les objets plus visibles et remplit les petits trous dans les objets.

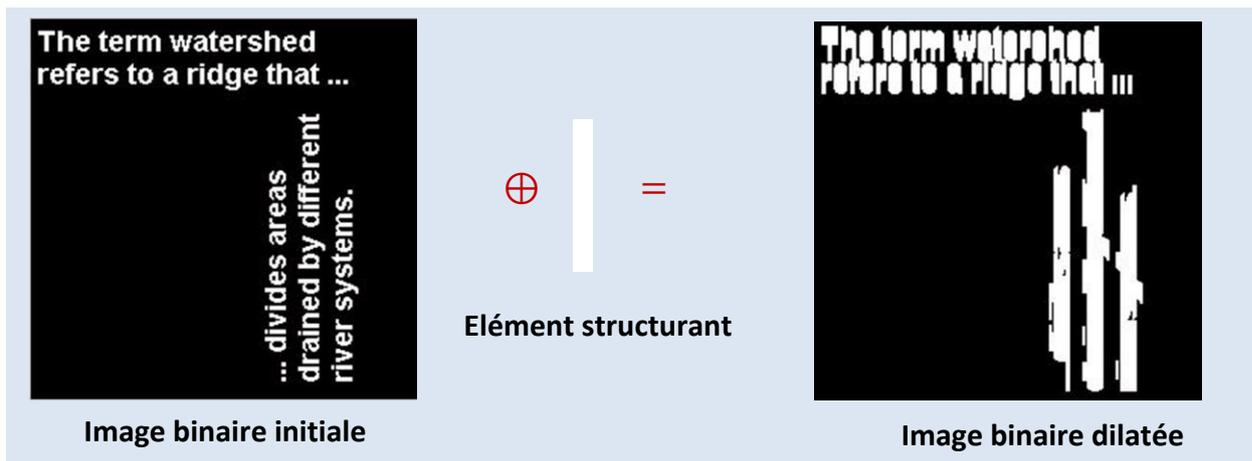
Principe : déplacer l'élément structurant  $E$  sur l'objet  $I$  en positionnant le centre de l'élément structurant sur chaque pixel de l'objet. L'ensemble des pixels des éléments structurants déplacés définit le résultat de la dilatation.

$$I \oplus E = \{x, E_x \cap I \neq \emptyset\}$$

$$I \oplus E = \{E_x \mid x \in I\}$$

### Exemple de Code en Matlab : Dilatation

```
% exemple de dilatation d'une image binaire
clc; clear; close all;
% Lire l'image binaire
bw = imread('text.png');
% Choisir l'élément structurant . ici une ligne verticale
se = strel('line',11,90);
% Dilater l'image avec un élément structurant de ligne verticale et
comparez les résultats.
bw2 = imdilate(bw,se);
imshow(bw), title('Original')
figure, imshow(bw2), title('Dilated')
```



### 4.3 Erosion (*erosion*)

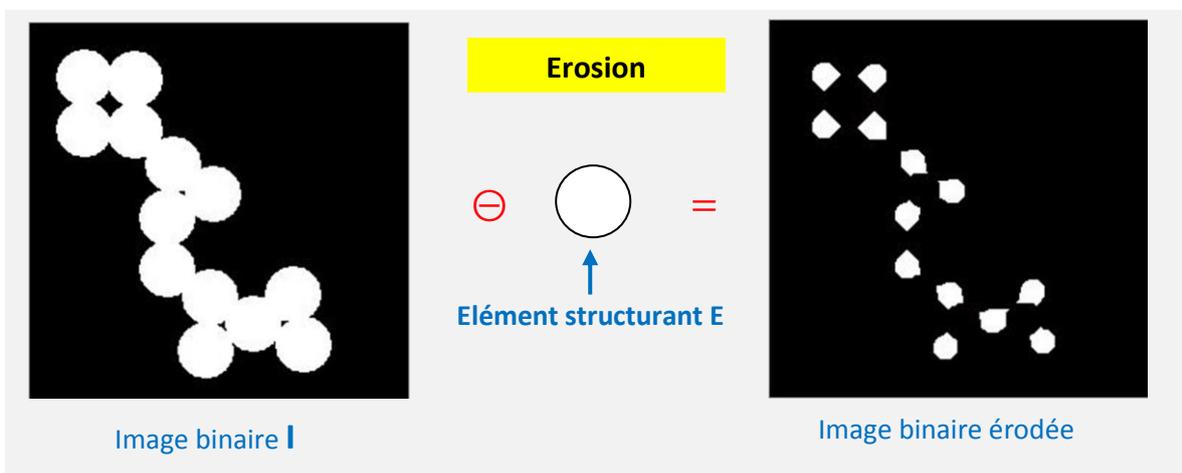
La valeur du pixel de sortie est la valeur minimale de tous les pixels du voisinage. Dans une image binaire, un pixel est remis à 0 si l'un des pixels voisins a la valeur 0. L'érosion morphologique supprime les îles et les petits objets de sorte qu'il ne reste que des objets substantiels.

Principe : déplacer l'élément structurant  $E$  sur l'objet  $I$  sans dépasser de l'objet. Les pixels sur lesquels est passé le centre de l'élément structurant définissent le résultat de l'érosion :

$$I \ominus E = \{x, E_x \subseteq I\}$$

#### Exemple de Code en Matlab : Erosion

```
% exemple de l'érosion d'une image binaire par un élément structurant.
Clc ; clear ; close all ;
% lire l'image binaire
originalBW = imread('circles.png') ;
% choisir l'élément structurant
se = strel('disk',11) ;
% éroder l'image et afficher le résultat
erodedBW = imerode(originalBW,se);
imshow(originalBW), figure, imshow(erodedBW)
```



## Propriétés

### Associativité

L'application de deux dilatations ou deux érosions consécutives peut se faire dans un ordre quelconque :

$$(I \oplus E_1) \oplus E_2 = (I \oplus E_2) \oplus E_1 = I \oplus (E_1 \oplus E_2)$$

$$(I \ominus E_1) \ominus E_2 = (I \ominus E_2) \ominus E_1 = I \ominus (E_1 \oplus E_2)$$

### Monotonie

La dilatation et l'érosion sont des opérations monotones, c'est-à-dire que que les relations d'inclusions sont conservées :

:

$$I_1 \subseteq I_2 \Rightarrow I_1 \oplus E \subseteq I_2 \oplus E$$

$$I_1 \subseteq I_2 \Rightarrow I_1 \ominus E \subseteq I_2 \ominus E$$

### Dualité

La dilatation et l'érosion sont des opérateurs duaux. En travaillant sur le fond plutôt que les objets (considérer le complémentaire de l'image), la dilatation est convertie en l'érosion et vice-versa :

$$I^c \ominus E = (I \oplus E)^c$$

$$I^c \oplus E = (I \ominus E)^c$$

## Comparaison

### Dilatation

- augmente la taille des objets
- bouche les petits trous
- soude les objets proches

### Érosion

- diminue la taille des objets
- élargit les trous
- sépare les objets connectés par un petit pont
- supprime les petits objets

## 5. Opérations basées sur la dilatation et l'érosion

La dilatation et l'érosion sont souvent utilisées en combinaison pour mettre en œuvre des opérations de traitement d'image. Par exemple, la définition d'une ouverture morphologique d'une image est une érosion suivie d'une dilatation, utilisant le même élément structurant pour les deux opérations. Vous pouvez combiner la dilatation et l'érosion pour supprimer les petits objets d'une image et lisser la bordure des gros objets.

## 5.1 Ouverture morphologique (*Morphological Opening*)

L'opération d'ouverture érode une image puis dilate l'image érodée, en utilisant le même élément structurant pour les deux opérations. L'ouverture morphologique est utile pour supprimer de petits objets dans une image tout en préservant la forme et la taille des objets plus gros.

L'érosion est utile pour supprimer de petits objets mais réduit la taille des objets restant. Pour éviter cela, on peut effectuer une dilatation après l'érosion, avec le même élément structurant.

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

Pour un exemple, on utilise l'ouverture morphologique, implémentée par la fonction *imopen* de Matlab, pour extraire des attributs plus significatifs dans une image.

Dans cet exemple, l'ouverture morphologique sur une image d'un circuit imprimé est appliquée pour supprimer toutes les lignes du circuit imprimé. L'image de sortie ne contient que les formes rectangulaires des micropuces et circuits intégrés.

La fonction *imopen* effectue l'érosion et la dilatation en une seule étape.

Code en Matlab :

```
% Ouverture morphologique de l'image en une seule passe
%
clc; clear; close all;
% lecture de l'image en niveaux de gris originale
GI = imread('circuit.tif');
figure
imshow(GI)
% lecture de l'image binaire
BW1 = imread('circbw.tif');
figure
imshow(BW1)
%
% création de l'élément structurant
SE = strel('rectangle', [40 30]);
%
% application de l'ouverture morphologique
BW2 = imopen(BW1, SE);
figure
imshow(BW2);
```

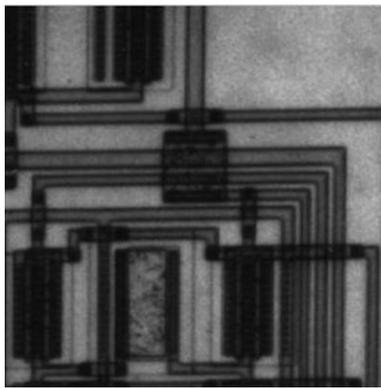
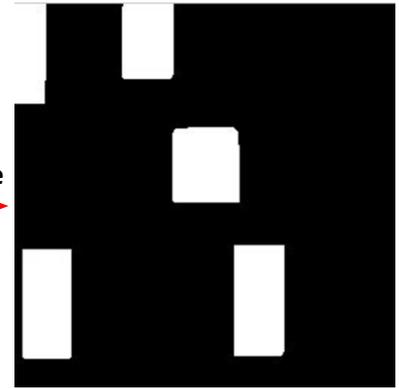


Image à niveaux de gris.  
Circuit imprimé



Image binaire du circuit  
imprimé

Ouverture  
→



Ouverture de l'image binaire.  
Suppression des lignes

**Exemple d'application de l'ouverture morphologique par la fonction *imopen*:  
Suppression des lignes de l'image binaire**

Code pour ouvrir par morphologie une image en effectuant l'érosion puis la dilatation

```
% Ouverture de l'image en deux étapes : érosion suivie par dilatation

% Érode l'image avec l'élément structurant.
% Cela supprime toutes les lignes, mais réduit également les rectangles.
BW3 = imerode(BW1,SE);
imshow(BW3)

% Pour restaurer les rectangles à leurs tailles d'origine, il faut
% dilater l'image érodée en utilisant le même élément structurant, SE.
BW4 = imdilate(BW3,SE);
imshow(BW4)
```

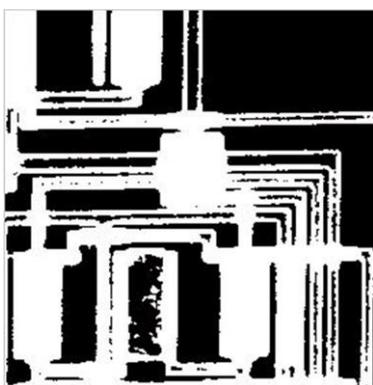


Image binaire d'un circuit  
imprimé

Erosion  
→



Dilatation  
→

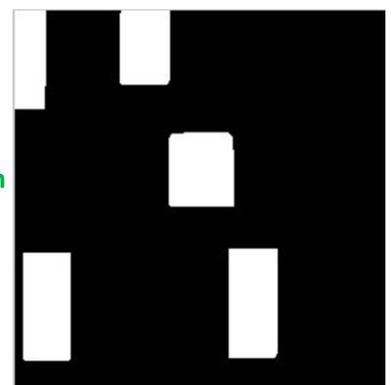


Image binaire érodée puis  
dilatée

**Ouverture morphologique de l'image binaire du  
circuit imprimé en deux étapes : érosion-dilatation**

## 5.2 Fermeture morphologique (*Morphological closing*)

Contrairement à l'érosion, la dilatation agrandit les objets et ferme les trous et fissures. Pour éviter l'élargissement des objets, une érosion peut être appliquée avec le même élément structurant.

L'opération de fermeture dilate une image puis érode l'image dilatée, en utilisant le même élément structurant pour les deux opérations. La fermeture morphologique est utile pour combler les petits trous d'une image tout en préservant la forme et la taille des objets de l'image.

$$I \bullet E = (I \oplus E) \ominus E$$

Pour un exemple, on utilise la fermeture morphologique, implémentée par la fonction `imclose` de Matlab, pour combler les lacunes dans une image. La fonction `imclose` effectue la dilatation suivie de l'érosion en une seule étape

Code en Matlab :

```
% Fermeture morphologique de l'image en une seule passe
%
clc; clear; close all;
% lecture de l'image binaire
originalBW = imread('circles.png');
imshow(originalBW);
%
% Créer un élément structurant en forme de disque pour préserver la %
nature circulaire de l'objet.
% Spécifier un rayon de 10 pixels afin que le plus grand espace soit
%comblé
%
se = strel('disk',10);
% Effectuer une opération de fermeture morphologique sur l'image.
closeBW = imclose(originalBW,se);
figure, imshow(closeBW)
```

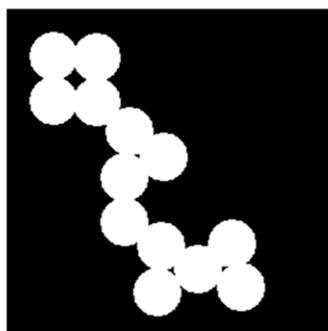


Image binaire initiale

Fermeture  
→



Image binaire fermée

Suppression des lacunes dans l'image binaire  
par fermeture morphologique

## Propriétés

L'ouverture et la fermeture...

6. conservent (souvent) la taille et la forme
7. ne conservent pas (toujours) la topologie
8. sont des opérations idempotentes (une deuxième opération ne modifie rien) :

$$(I \circ E) \circ E = I \circ E$$

$$(I \bullet E) \bullet E = I \bullet E$$

## Comparaison

### Ouverture

- Élimine les petits objets
- Lisse les contours

### Fermeture

- Bouche les petits trous
- Soude les formes proches

### 5.3 Squelettisation des objets dans une image binaire

Le processus de squelettisation érode tous les objets en direction de leurs lignes médianes sans modifier la structure essentielle des objets, tels que la présence des trous et des branches. Le résultat n'est pas une amélioration de l'image mais une mise en évidence de l'information essentielle à la description de son contenu. La squelettisation élimine au maximum l'information redondante afin de diminuer la quantité de données à analyser. C'est une succession d'opérations appelées amincissements (*Thinning*) jusqu'à obtention d'une structure stable ne pouvant plus être amincie.

La fonction  $B = \text{bwskel}(A)$  réduit tous les objets de l'image 2D binaire  $A$  à des lignes courbes de 1 pixel de large, sans modifier la structure essentielle de l'image. Ce processus, appelé squelettisation, extrait la ligne centrale tout en préservant la topologie et le nombre d'Euler (également appelé caractéristique d'Euler) des objets.

Code en Matlab :

```
% Les objets d'intérêt sont des fils sombres sur un fond clair
```

```
I = imread('threads.png');  
imshow(I)
```

```
% prendre le complément de l'image et binariser
```

```
Icomplement = imcomplement(I);  
BW = imbinarize(Icomplement);  
imshow(BW)
```

```
Effectuez la squelettisation de l'image binaire à l'aide de bwskel.
```

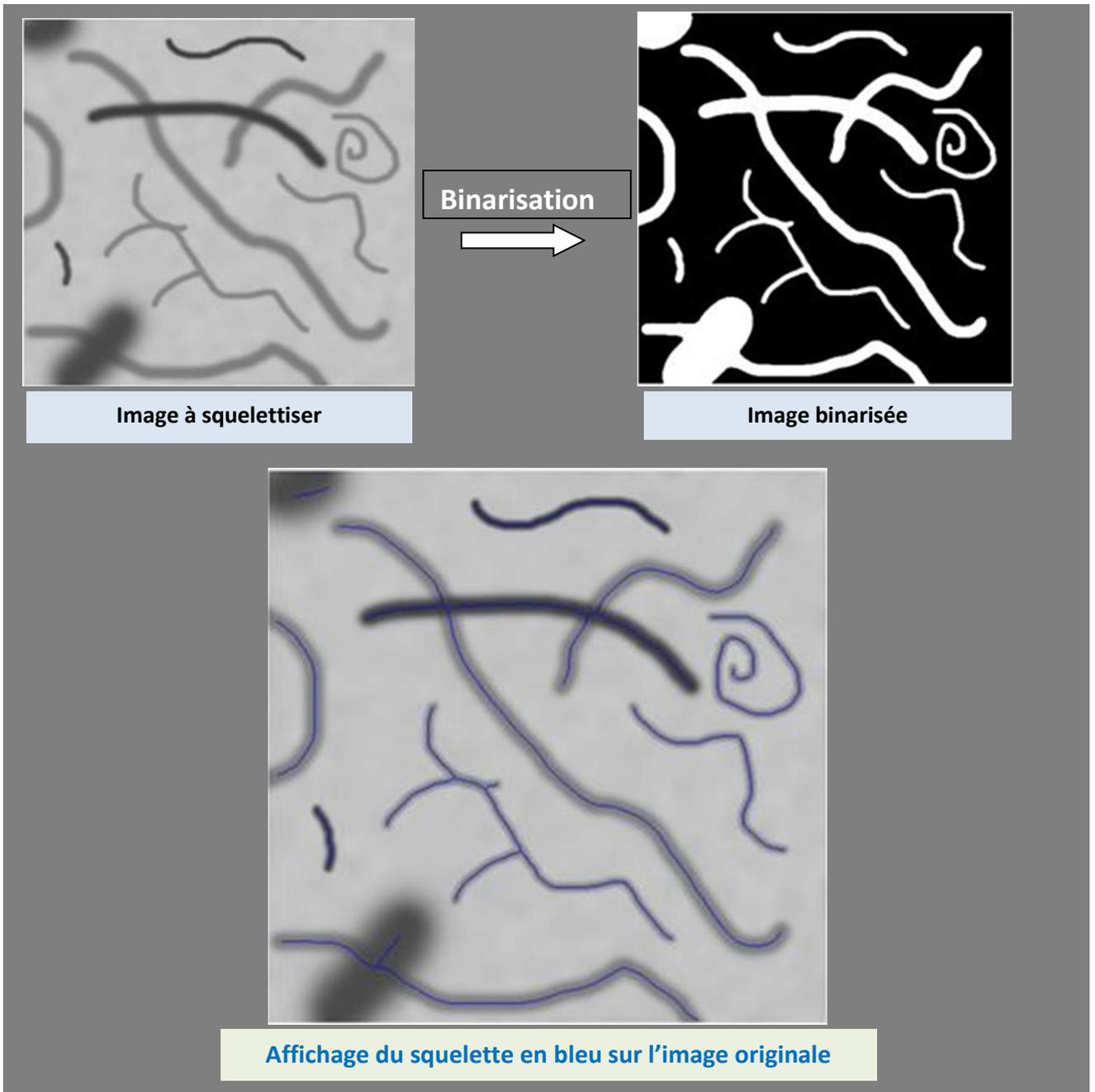
```
out = bwskel(BW);
```

```
% Afficher le squelette sur l'image d'origine sous la forme d'une ligne bleue de 1 pixel % de large sur  
les fils sombres.
```

```
imshow(labeloverlay(I,out, 'Transparency',0))
```

La squelettisation nécessite une image binaire dans laquelle les pixels de premier plan sont 1 (blanc) et l'arrière-plan est 0 (noir). Pour rendre l'image originale adaptée à la squelettisation, on prend le complément de l'image de sorte que les objets soient clairs et que l'arrière-plan soit sombre. Ensuite, on binarise le résultat obtenu par un seuillage.

Le squelette est affiché sur l'image d'origine à l'aide de la fonction de superposition d'étiquettes *labeloverlay*. Le squelette apparaît sous la forme d'une ligne bleue de 1 pixel de large sur les fils sombres.



Exemple d'une image squelette

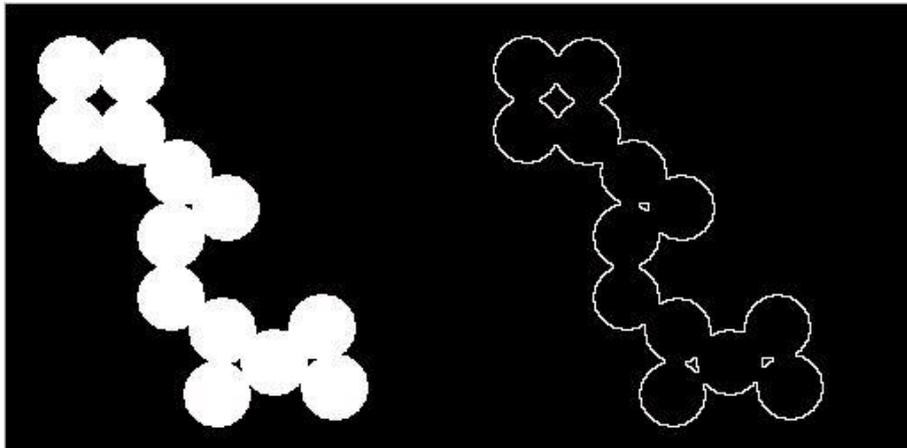
## 5.4 Le périmètre des objets dans une image binaire

Un pixel fait partie du périmètre s'il est différent de zéro et qu'il est connecté à au moins un pixel de valeur nulle.

Le code suivant utilise la fonction `bwperim()` pour détecter les pixels du périmètre d'un objet et affiche une image binaire en affectant la valeur logique 1 aux pixels du périmètre et la valeur logique 0 aux autres pixels.

Code en Matlab

```
% Trouver le périmètre des objets dans une image binaire
clc;clear; close all;
%
% lire l'image binaire
BW = imread('circles.png');
%
% Calculer les périmètres des objets dans l'image.
BW2 = bwperim(BW,8);
%
% affichage de l'image originale et des perimètres côte-à-côte.
imshowpair(BW,BW2, 'montage')
```



### 5.5 Transformation Tout-ou-Rien (hit-or-miss)

La transformation (hit-or-miss) préserve les pixels dans une image binaire dont les voisinages correspondent à la forme d'un élément structurant et ne correspondent pas à la forme d'un deuxième élément structurant disjoint.

La transformation Tout-ou-Rien (hit-or-miss) peut être utilisée pour détecter des motifs d'objets d'une forme particulière dans une image.

Elle correspond à l'intersection des deux ensembles générés par :

- l'érosion par un 1<sup>er</sup> élément structurant :

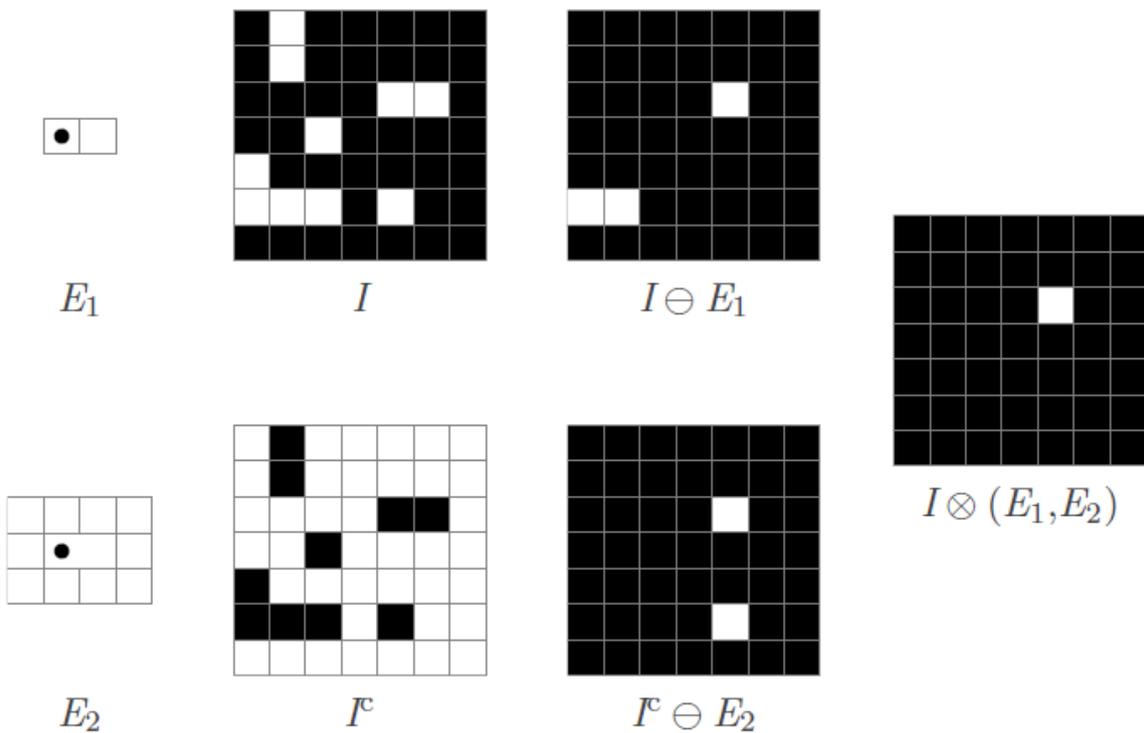
$$E_1 : I \ominus E_1$$

- l'érosion du fond par un 2<sup>e</sup> élément structurant :

$$E_2 : I^c \ominus E_2$$

Avec  $E_1 \cap E_2 = \emptyset$ .

$$I \otimes (E_1, E_2) = (I \ominus E_1) \cap (I^c \ominus E_2) \\ = (I \ominus E_1) \cap (I \oplus E_2)^c$$



## 5.6 Transformation morphologique du chapeau-haut (top-hat)

La transformation chapeau haut ouvre une image, puis soustrait l'image ouverte de l'image d'origine. La transformation top-hat peut être utilisée pour améliorer le contraste dans une image à niveaux de gris avec un éclairage non uniforme. La transformation peut également isoler de petits objets lumineux dans une image.

On peut utiliser le filtrage chapeau pour corriger un éclairage non uniforme lorsque l'arrière-plan est sombre. Cet exemple utilise un filtrage top-hat avec un élément structurant en forme de disque pour supprimer l'éclairage de fond irrégulier d'une image.

### Code en Matlab

```
% le filtrage chapeau pour corriger un éclairage irrégulier
clc;clear; close all;
% lecture de l'image
original = imread('rice.png');
% affichage
figure, imshow(original)
% choix de l'élément structurant
se = strel('disk',12);
% application du filtrage chapeau top-hat
tophatFiltered = imtophat(original,se);
% affichage
figure, imshow(tophatFiltered)
% utiliser la fonction imadjust pour améliorer la visibilité du résultat
contrastAdjusted = imadjust(tophatFiltered);
figure, imshow(contrastAdjusted)
```

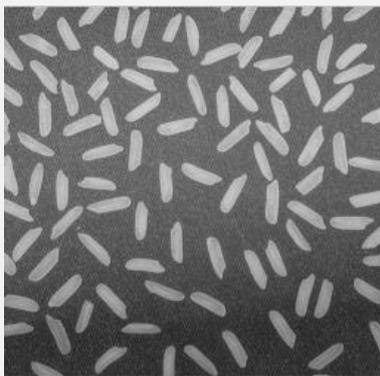


Image originale A

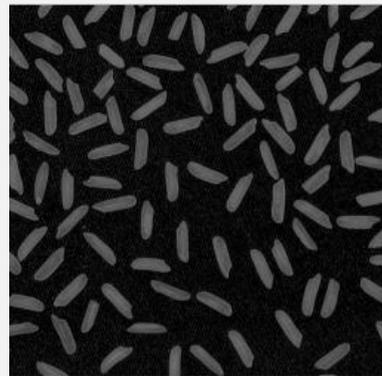


Image filtrée par un filtre top-hat B

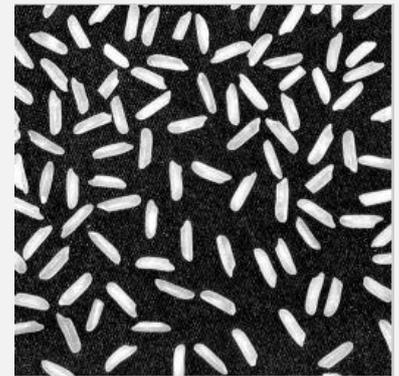


Image plus visible C par ajustage du contraste de B

**Suppression de l'éclairage de fond irrégulier d'une image par un filtrage top-hat.**

## 5.7 Transformation morphologique du chapeau-bas (bottom-hat)

La transformation du chapeau bas ferme une image, puis soustrait l'image d'origine de l'image fermée. La transformation du chapeau bas peut être utilisée pour trouver des creux d'intensité dans une image en niveaux de gris.

Ci-dessous un exemple d'application : amélioration du contraste par combinaison des filtrages top-hat et bottom-hat

Code en Matlab :

```
% exemple pour améliorer le contraste en associant les filtrage top-hat et  
% bottm-hat  
clc; clear; close all;  
% lecture de l'image à traiter  
I = imread('pout.tif');  
imshow(I)  
% choix de l'élément structurant  
se = strel('disk',3);  
% Ajouter l'image d'origine I à l'image filtrée chapeau top-hat,  
% puis soustraire l'image filtrée chapeau bas.  
J = imsubtract(imadd(I,imtophat(I,se)), imbothat(I,se));  
figure, imshow(J)
```



Image originale avec mauvais contraste



Image plus contrastée

**Amélioration du contraste par combinaison des filtrages top-hat et bottom-hat**

## 6. Conclusion

### Extensions de la morphologie mathématique

- Les objets obtenus peuvent être caractérisés par des descripteurs de forme, par exemple :
  - le centroïde : coordonnées du barycentre
  - la surface : nombre de pixels de l'objet
  - la boîte englobante (*bounding box*) : le plus petit rectangle contenant l'objet
  - etc.
- D'autres outils existent :
  - représentation de l'image sous forme d'un arbre
  - extension des outils à des images à niveaux de gris
  - etc.