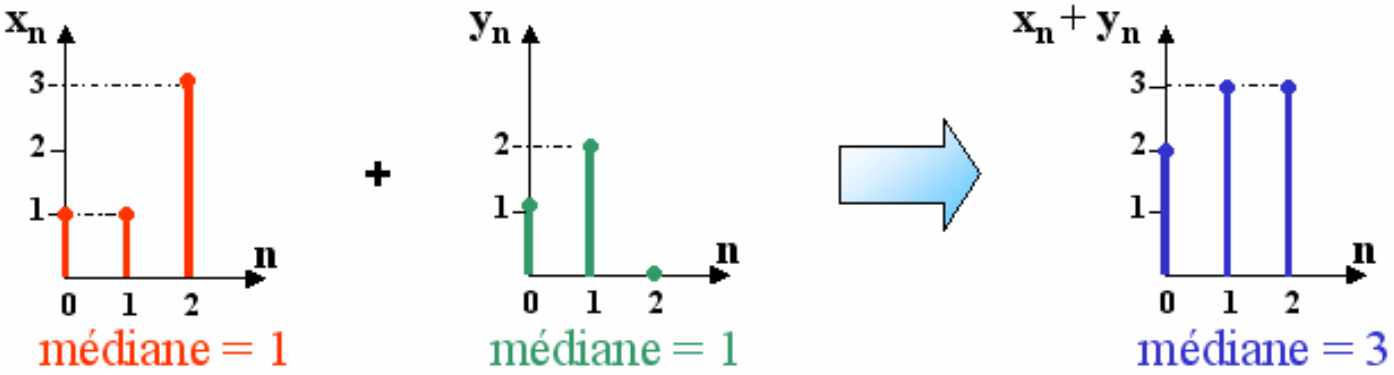
Exemple de filtrage non-linéaire : le filtrage médian

- Le filtrage médian est une opération non-linéaire :

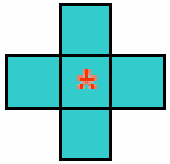
  médiane { $x_m + y_m$ } ≠ médiane { x_m } + médiane { y_m } sauf exception

  - exemple sur des séquences de signaux de longueur 3:



- En traitement d'image, les tailles des fenêtres utilisées pour le filtrage médian sont généralement impaires : ***3 ×3 ; 5 ×5 ; 7 ×7***

  - exemple : fenêtre de 5 pixels en croix



- Pour les fenêtres de taille paire (2 K valeurs) : après ordonnancement croissant des valeurs, prendre la moyenne des 2 valeurs centrales :

$$\text{valeur de sortie} = \frac{(\ K^{\text{ième}} \text{ valeur ordonnée} + (K+1)^{\text{ième}} \text{ valeur ordonnée}\ )}{2}$$

Exemple d'application d'un filtrage médian

- Image de référence (taille 3 × 3) :

91	**55**	**90**
77	**68**	**95**
115	**151**	**210**

➢ Filtrage médian avec une fenêtre de taille 3 × 3 :

↳ on liste les valeurs de l'image de référence sur la fenêtre 3 × 3 :

55 , 68 , 77 , 90 , **91** , 95 , 115 , 151 , 210

↓

valeur médiane = 91

➢ Filtrage médian sur une fenêtre de taille 3 × 3 en croix :

	55	
77	**68**	**95**
	151	

↳ on liste les valeurs de l'image de référence sur la fenêtre :

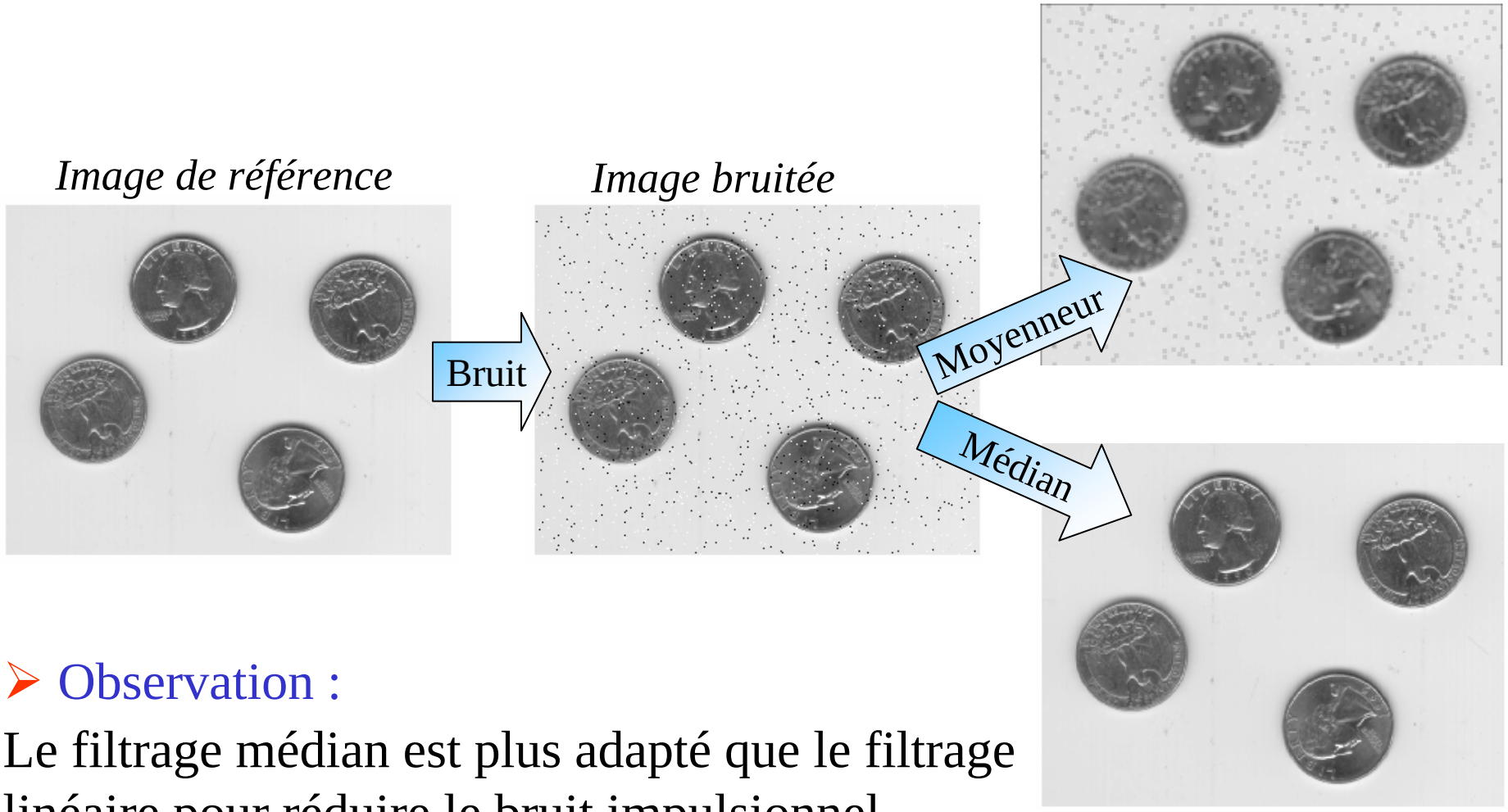
55 , 68 , **77** , 95 , 151

↓

valeur médiane = 77

Comparaison : filtre médian et filtre moyenneur

- Image « *Pièces* » de référence, bruitée (bruit de type impulsionnel) puis filtrée avec un filtre moyenneur (3×3) et un filtre médian (3×3).



➢ Observation :

Le filtrage médian est plus adapté que le filtrage linéaire pour réduire le bruit impulsionnel

Exercice Chapitre 4 – Comparaison : filtres non-linéaires *vs.* filtres linéaires

Contrairement au filtrage par convolution (filtrage linéaire), le filtrage non-linéaire fait intervenir les pixels voisins suivant une loi non-linéaire. Le filtre médian (cas particulier du filtrage d'ordre), utilisé dans cet exercice, est un exemple classique de ces filtres. À l'instar du filtrage par convolution, les filtres non-linéaires opèrent sur un voisinage donné.

1. Création d'une image bruitée

Chargez l'image *BOATS_LUMI.BMP*. Mettez à jour la liste des chemins dans le path browser. Le but de l'exercice est de comparer les effets linéaires et non-linéaires de deux filtrages sur une image bruitée.
La fonction ***imnoise*** de Matlab permet de créer différents types de bruit. À l'aide de cette fonction, créez l'image bruitée de *BOATS_LUMI* avec un bruit de type « poivre-et-sel » (« *salt-and-pepper* »). Affichez l'image bruitée et expliquez comment peut-on créer ce bruit ?

2. Application d'un filtre linéaire

On souhaite réduire le bruit dans l'image. Dans un premier temps, on considère un filtre moyenneur (3×3) pour réduire le bruit dans l'image. Son noyau de convolution est :

$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Appliquez ce filtre (utilisez la fonction ***imfilter***), et observez l'image bruitée. Interprétez le résultat obtenu.

3. Application d'un filtre non-linéaire

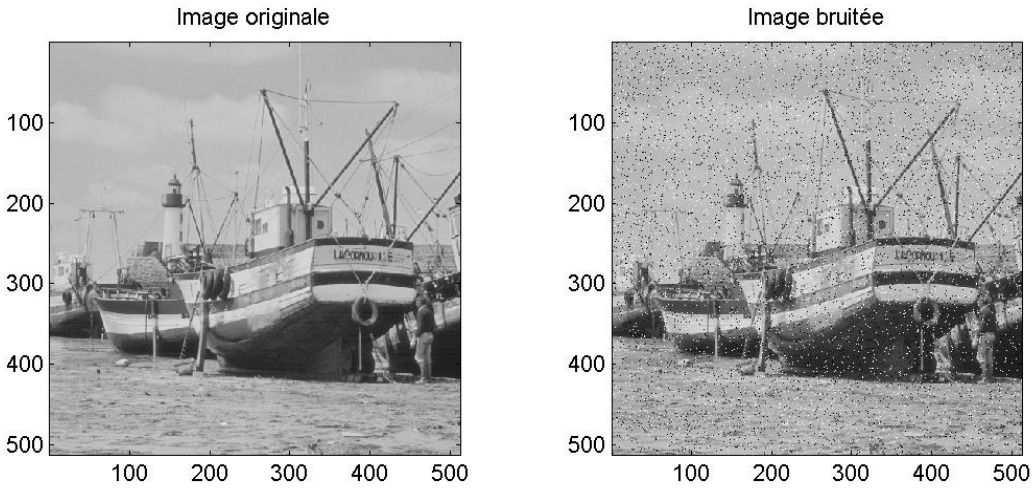
On souhaite à présent réduire le bruit avec un filtre médian (3×3). Sous Matlab, le filtrage médian peut être réalisé à l'aide de la fonction ***medfilt2***. Expliquez ce qu'effectue cette fonction. Appliquez ce filtre à l'image bruitée et observez les résultats. Expliquez les différences avec le filtre moyenneur précédemment utilisé.

Correction de l'exercice : Filtrage linéaire vs. Filtrage non-linéaire

1 – Voici les commandes à entrer pour créer et observer l'image bruitée de *BOATS_LUMI* :

```
I=imread('BOATS_LUMI.BMP') ; % image bateau en niveaux de gris
IB = imnoise(I,'salt & pepper'); % image bruitée
figure(1)
subplot(1,2,1)
subimage(I)
title('Image originale')
subplot(1,2,2)
subimage(IB)
title('Image bruitée')
```

Voici les résultats obtenus :

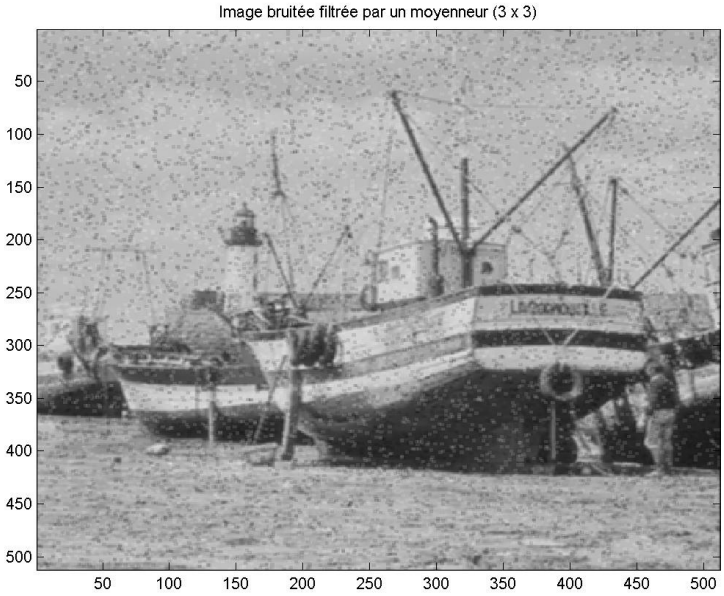


Le bruit « poivre-et-sel » utilisé ici consiste à mettre, aléatoirement, plusieurs pixels aux valeurs 255 ou 0 (valeurs extrêmes de l'intervalle des niveaux de gris). Ce type de bruit impulsionnel peut apparaître par numérisation d'une image ou au cours d'une transmission.

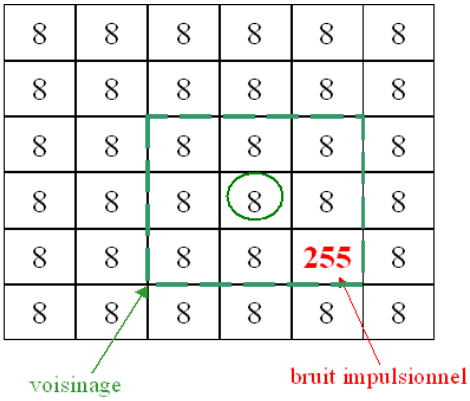
2 – Voici les commandes pour appliquer un filtre moyenneur à l'image bruitée :

```
% Filtre moyenneur
N = ones(3)/9 ; % noyau de convolution du filtre
If1 = imfilter(IB,N) ;
figure(2)
image(If1)
title('Image bruitée filtrée par un moyenneur (3 x 3)')
v=0:1/255:1; colormap([v' v' v']); % LUT pour afficher en niveaux de gris
```

Voici l'image obtenue :



Le bruit « *salt-and-pepper* » est faiblement réduit. On distingue encore nettement les grains dans l'image. Le filtre moyenneur réalise en effet la moyenne pour chaque pixel sur un voisinage (3 × 3) et le bruit des impulsions (à 0 ou à 255) participe à cette moyenne :

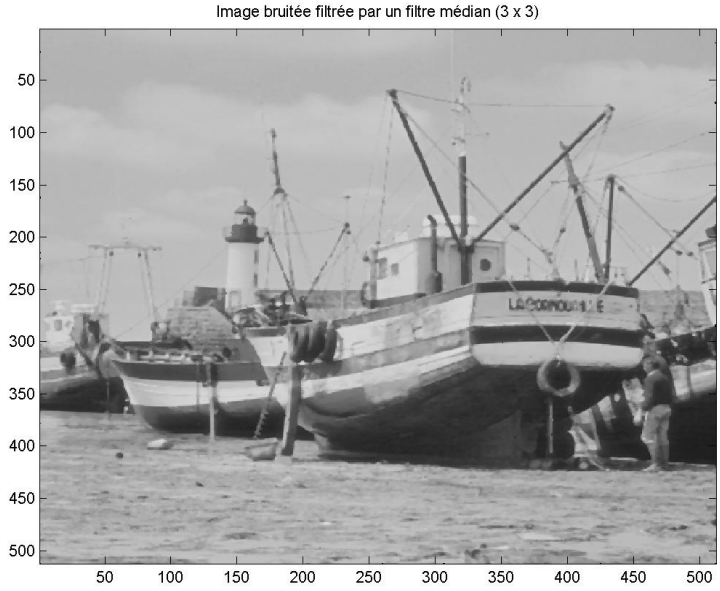


Le pixel cerclé a le niveau de gris 8 ainsi que toute l'image, à l'exception d'un pixel bruité à 255. En sortie du filtrage moyenneur, la valeur de sortie du pixel cerclé (et de tout pixel ayant la valeur 255 dans son voisinage) sera : $(8\times8+255)/9 \approx 35$. La valeur du pixel après filtrage n'est donc pas représentative du voisinage de ce pixel, le bruit impulsionnel est trop faiblement réduit. Ce filtre linéaire n'est donc pas adapté.

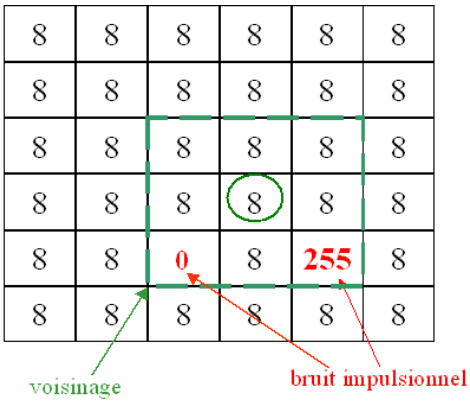
3 – Voici les commandes pour appliquer un filtre médian à l'image bruitée :

```
% Filtre médian
If2 = medfilt2(IB,[3 3]) ; % filtrage médian avec un voisinage 3 x 3
figure(3)
image(If2)
title('Image bruitée filtrée par un filtre médian (3 x 3)')
v=0:1/255:1; colormap([v' v' v']); % LUT pour afficher en niveaux de gris
```

Voici l'image obtenue :



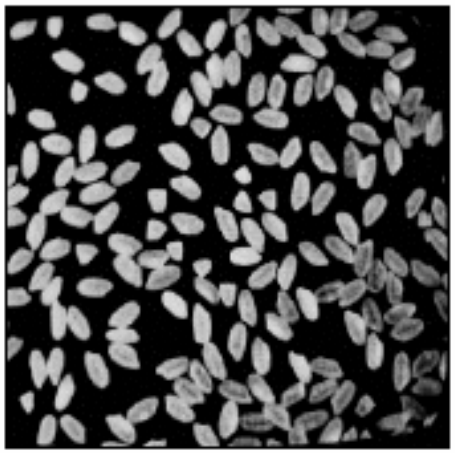
Le bruit impulsionnel est visiblement réduit. Le filtrage médian d'un pixel P, sur un voisinage V(P) de taille (M × N), ordonne les valeurs des pixels de V(P) par ordre croissant, et attribue en sortie la valeur médiane sur ce voisinage au pixel P (opération non-linéaire) :



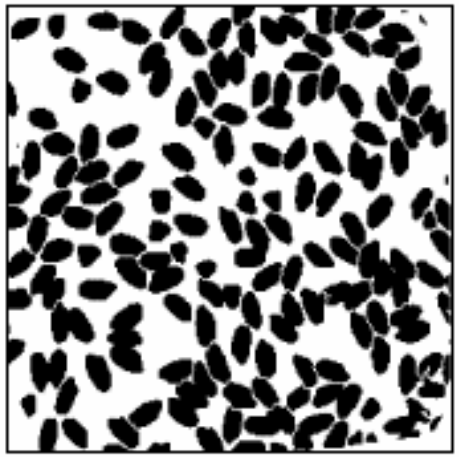
Reprenons l'exemple précédent : les valeurs des pixels sont ordonnées par ordre croissant : 0, 8, 8, 8, **8**, 8, 8, 8, 255. La valeur médiane est donc 8. **Pour cette opération non-linéaire, les impulsions 0 et 225 n'ont pas d'influence sur la valeur médiane**. Le filtrage médian est donc adapté à la réduction du bruit impulsionnel.

Exemple d'application du filtrage morphologique, grains de riz (1/2)

- a : Image de référence (grains de riz)
  - **Pré-traitement :**
- b : Binarisation et régularisation des formes
- c : Suppression des objets incomplets (grains sur les bords)



a : image initiale



b : segmentation de a

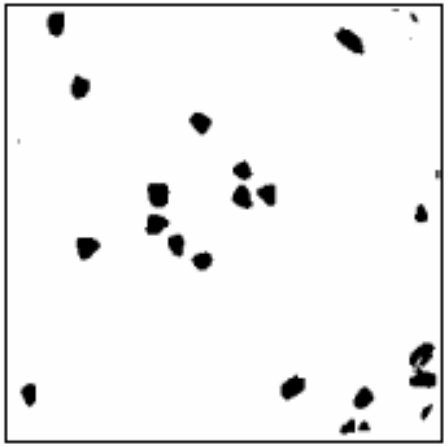
c : grains internes

Exemple d'application du filtrage morphologique, grains de riz (2/2)

- **Tri des formes et répartition en classes de formes par morphologie :**

- d : Classe des grains entiers
- e : Classe des grains brisés
- f : Classe des grains à plusieurs composantes connexes (paquets de grains)

d : grains entiers

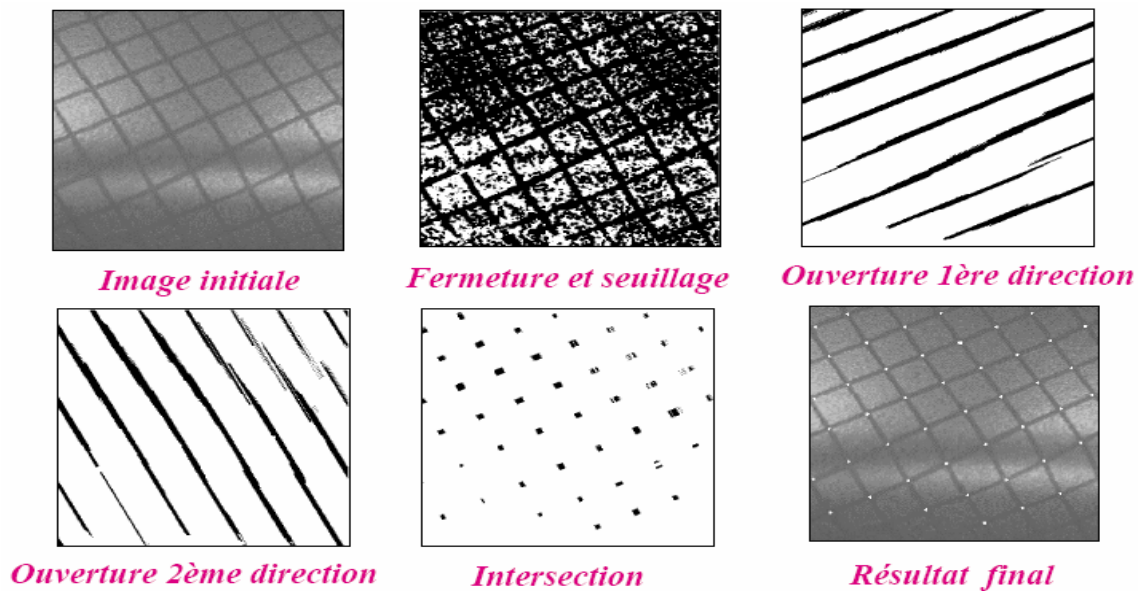


e : grains brisés

f : paquets de grains

Possibilité de dénombrer et mesurer des grains de riz

Exemple d'application du filtrage morphologique, feuilles métalliques



- Fermeture pour adoucir les contours et régulariser les formes, puis binarisation de l'image par seuillage
- Ouverture dans 2 directions diagonales connues pour détecter les lignes
- Détection des intersections, puis localisation sur l'image de référence

Possibilité de détecter un mouvement, de mesurer une déformation de la feuille, …

Filtrages morphologiques

♦ Objectifs :

- Obtention d'un ensemble de points connectés ayant une forme simple (pour une image binaire)
- Obtention des principales composantes connexes (en faible nombre)
- Régularisation des formes du signal d'image

♦ Applications :

- Filtrage de taille
- Dénombrement d'objets
- Mesure d'objets

Chapitre 4

FILTRAGE NON-LINEAIRE

Filtrage par morphologie

Cas des images images binaires

Rappel : Opérateurs binaires classiques

X , Y , Z : variables booléennes ⇒ états possibles : '0' ou '1'

• opérateur **ET** : Z = X ET Y (noté Z = X . Y)

X	Y	Z
0	0	0
1	0	0
0	1	0
1	1	1

• opérateur **OU** : Z = X OU Y (noté Z = X + Y)

X	Y	Z
0	0	0
1	0	1
0	1	1
1	1	1

• opérateur **NON** : $Z = \overline{X}$

X	Z
0	1
1	0

Une variable qui ne peut prendre que deux états (vrai ou faux, allumé ou éteint, en haut ou en bas, positif ou négatif, noir ou blanc, …), est appelée une ***variable booléenne***. Typiquement, on attribue la valeur '1' à l'un des deux états possibles et '0' à l'autre.
On définit pour ces variables booléennes trois opérateurs de base :

- le 'ET', la sortie est à 'vrai' uniquement si les deux entrées sont à 'vrai' ;
- le 'OR', la sortie est à 'vrai' si l'une des deux entrées au moins est à 'vrai' ;
- le 'NON', la sortie est à l'état inverse de l'entrée.

Les « ***tables de vérité*** », qui caractérisent ces différents opérateurs, sont données sur la figure ci-dessus. On note également « A . B » pour « A et B », et « A + B » pour « A ou B ».
Notons que d'autres opérateurs binaires sont construits à partir de combinaisons de ces trois opérateurs de base : le 'NOR' (i.e. 'non ou'), le 'NAND' (i.e. 'non et'), le 'XOR' (i.e. 'ou exclusif'), …

Filtrage morphologique

➢ Concept :

- S'inscrit dans la théorie de description des images.
- Prise en compte de la forme des composants structurés de l'image.
- Applications basiques : traitement d'images binaires, extension au traitement d'images monochromes.

➢ Opérateurs :

- 2 opérateurs basiques ⇒ "EROSION" et "DILATATION"
- Combinaison de ces 2 opérateurs

  ⇒ 2 opérateurs complémentaires :

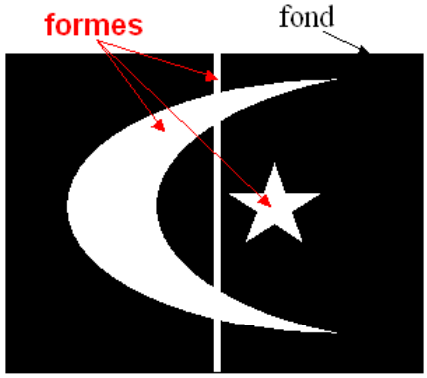
  "OUVERTURE" et "FERMETURE"
- Ces opérateurs dépendent d'un élément structurant.

Le filtrage morphologique repose sur la morphologie mathématique, basée sur une description ensembliste des images. Les opérateurs morphologiques privilégient la notion de forme plutôt que l'information sur l'amplitude des signaux. Ils s'appliquent aussi bien aux images binaires (deux niveaux : blanc ou noir) qu'aux images monochromes (en niveaux de gris).
Dans cette ressource, nous nous limitons 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 est appelée l'***élément structurant***. Nous allons maintenant expliquer en détail le sens de ces termes.

Image binaire

- L(m,n) ∈ {0, 1} ∀(m,n) ∈ S support de l'image
  L = 0 pixel de l'arrière plan (ou du fond)
  L = 1 pixel de l'objet (ou de la forme)
  ↳ 2 catégories : « fond », « forme »

- Forme(s) : X
  X = { p ∈ S / L(p) = 1 }



Une image binaire est une image pour laquelle les pixels (m, n) n'ont que deux valeurs de luminance L(m, n) possibles, notées conventionnellement 0 (fond) et 1 (formes). On définit donc les formes 'X' comme étant l'ensemble des points 'P' d'affixe 'p', appartenant au support 'S' de l'image, tel que la luminance en ces points soit égale à 1 :

$$\mathbf{X = \{ p \in S / L(p) = 1 \}}$$

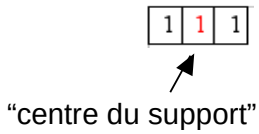
Une image binaire peut être obtenue par une numérisation dont la quantification ne comporte que deux niveaux de reconstruction, ou par une binarisation d'une image monochrome, notamment en utilisant son histogramme pour choisir un seuil adéquat (cf. exercice « Binarisation » du chapitre 2).

Éléments Structurants (images binaires)

➢ ***Élément structurant:***

un ensemble de pixels à 1 sur un support dans l'image avec une origine ayant comme coordonnées (0, 0) (pixel marqué en rouge dans les figures)

❖ 3 exemples typiques:



- a -
élément structurant 1-D

- b -
éléments structurants 2-D

- c –
éléments structurants 2-D

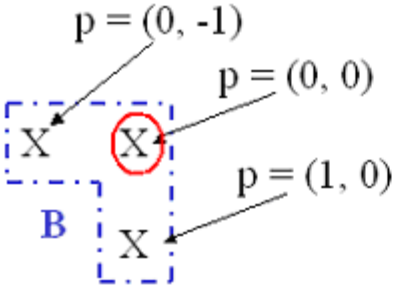
$$B = \{p \text{ de sorte que } L(p) = 1\}$$

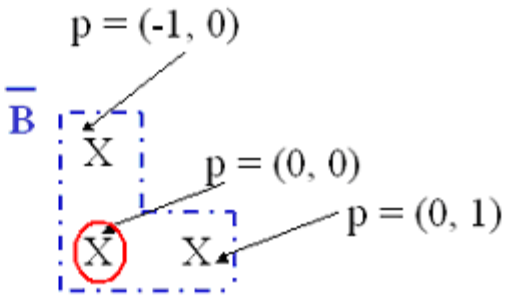
↳ *élément structurant*

symétrique: $\overline{B} = \{p \in B \Rightarrow -p \in \overline{B}\}$

L'**élément structurant**, B, est un ensemble de pixels à 1. Le point O de coordonnées (0, 0) fait généralement partie de B, mais pas obligatoirement.
Soit B un élément structurant composé d'un ensemble de points P d'affixe p. On définit alors $\overline{B}$, l'élément structurant symétrique de B, comme étant l'ensemble des points d'affixe '-p'.

Exemple :





Erosion Morphologique (par B)

Notation $X \ominus B$

Définition $X \ominus B = \{ p \in S \text{ tel que } B_p \subseteq X\}$

où B_p est la translation de B par p

- $X \ominus B$: ensemble des pixels d'affixe p de S tel que pour chaque p, B_p est complètement inclus dans l'objet X

Exemple :

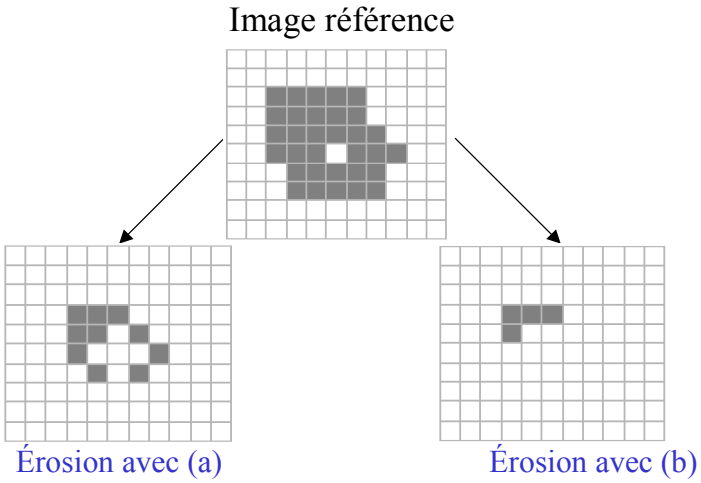
Élément Structurant

0	1	0
1	1	1
0	1	0

(a) élément structurant à 4-connexité

1	1	1
1	1	1
1	1	1

(b) élément structurant à 8-connexité



L'érosion d'une forme X par un élément structurant B est notée « X⊖B ». Elle est définie par : **$X \ominus B = \{ p \in S$, tel que $B_p \subseteq X\}$**. Il s'agit donc de l'ensemble des pixels P d'affixe p du support S de l'image, qui vérifient $B_p \subseteq X$, lorsqu'ils sont pris comme centre de l'élément structurant B (i.e. translation de B par p).

La figure ci-dessus présente deux cas d'érosion. Les deux érosions sont réalisées sur la même image de départ, mais avec deux éléments structurant différents :

- cas a : L'élément structurant est à 4-connexité (origine à 4 voisins). Chaque pixel du support qui a la valeur 0, ou qui a l'un de ses 4 voisins à la valeur 0 est mis à la valeur 0 après filtrage.

- cas b : L'élément structurant est à 8-connexité (origine à 8 voisins). Chaque pixel du support qui a la valeur 0, ou qui a l'un de ses 8 voisins à la valeur 0 est mis à la valeur 0 après filtrage.

Dans les deux cas, on observe qu'une érosion élimine les pixels isolés sur le fond et érode le contour des objets.

Dilatation Morphologique (par B)

Notation : $X \oplus B$

Définition: $X \oplus B = \{ p \in S \text{ tel que } \overline{B}_p \cap X \neq \varnothing \}$

où $\overline{B}_p$ est le symétrique de B translaté par p

• $X \oplus B$: ensemble de pixels d'affixe p dans S tel que pour chaque p, $\overline{B}_p$ n'a pas d'intersection nulle avec X

Exemple :

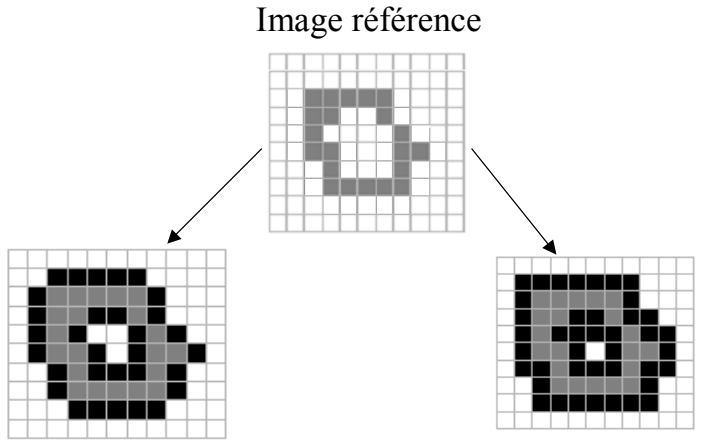
Élément Structurant

0	1	0
1	1	1
0	1	0

(a) élément structurant à 4-connexité

1	1	1
1	1	1
1	1	1

(b) élément structurant à 8-connexité



La dilatation d'une forme X par un élément structurant B est notée « $X \oplus B$ ». Elle est définie par : **$X \oplus B = \{ p \in S$, tel que $\overline{B_p} \cap X \neq \varnothing \}$**. Il s'agit donc de l'ensemble des pixels P d'affixe p, tel que le translaté $\overline{Bp}$, de l'élément structurant symétrique $\overline{B}$, ait une intersection non vide avec X.

La figure ci-dessus présente deux cas de dilatation.

- cas a : L'élément structurant est à 4-connexité. Chaque pixel du support qui est égal à la valeur 1, ou qui a l'un de ses 4 voisins à la valeur 1 est mis à la valeur 1 après filtrage.

- cas b : L'élément structurant est à 8-connexité. Chaque pixel du support qui est égal à la valeur 1, ou qui a l'un de ses 8 voisins à la valeur 1 est mis à la valeur 1 après filtrage.

Dans les deux cas, on observe qu'une dilatation élimine les trous isolés dans les objets et dilate le contour des objets en tenant compte de l'élément structurant.

Propriété : L'érosion par B de l'ensemble X^C complémentaire de X par rapport à S, est équivalente à la dilatation de X par B. On dit alors que l'érosion et la dilatation sont duales par rapport à la complémentation : **$X \oplus B = (X^C \ominus B)^C$**

Filtrage par Érosion et Reconstruction

(par dilatations contraintes)

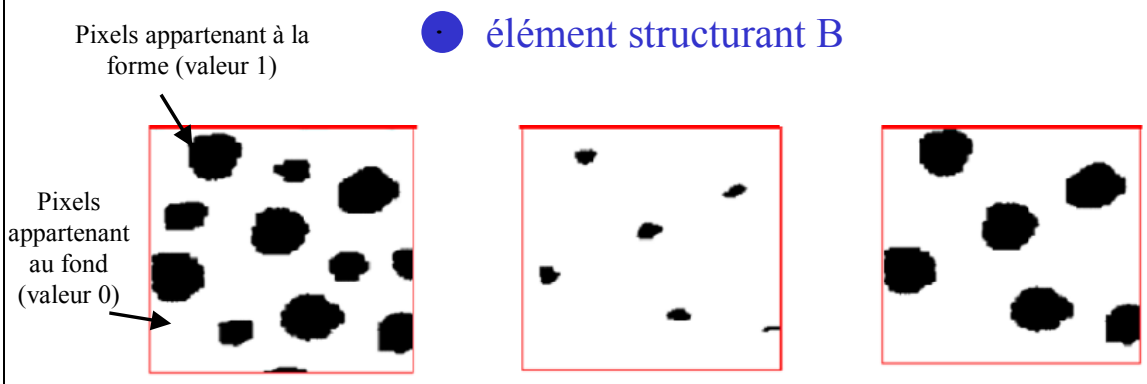


Image de référence I_a | **Image I_b : Érosion de I_a par élément structurant B** | **I_c: Reconstruction à partir de I_b et de I_a**

$$I_c = (\ldots(((I_b \oplus B) \text{ ET } I_a) \oplus B) \text{ ET } I_a)\ldots \quad \ldots\ldots)$$

jusqu'à l'idempotence de I_c

L'image I_b sert de point de départ pour la première dilatation dans la série de dilatations

La figure ci-dessus présente un exemple de reconstruction d'une image avec une succession de transformations T basées sur une première érosion par l'élément structurant B, puis une suite de dilatations par ce même élément structurant. La transformation T est définie par : « $T(X) = (X \oplus B) \; ET \; I_a$ ». L'image reconstruite I_c est obtenue par répétitions de la transformation T, jusqu'à l'idempotence de I_c. On a donc : I_c=ToTo…oToT(I_b). Dans le cas de l'image I_a, cette reconstruction permet de ne conserver que les tâches noires étendues de l'image, les tâches isolées de petites tailles sont supprimées.

Remarque : une transformation est idempotente si, après transformation, le résultat est invariant par la transformation i.e. ToT(X) = T(X). C'est le cas ici dans la mesure où l'on compare le résultat de la dilatation avec l'image de référence I_a (via l'opérateur binaire classique ET). Cette propriété assure la stabilité du filtre morphologique.

Ouverture Morphologique (par B)

Notation : X o B

Définition $X \circ B = (X \ominus B) \oplus B$

Érosion puis Dilatation

Effets : lissage de forme

- Suppression des petits détails de la bordure de l'objet
- Découpage des isthmes étroits

Fermeture Morphologique (par B)

Notation : X ● B

Définition $X \bullet B = (X \oplus B) \ominus B$

Dilatation puis Érosion

Effets: lissage de forme

- Le remplissage des canaux étroits et des petits trous
- Composants connectés

Propriété : Idempotence

$(X \circ B) \circ B = X \circ B$ $\quad$ $(X \bullet B) \bullet B = X \bullet B$

À partir des deux opérateurs morphologiques de base, que sont l'érosion et la dilatation, on peut, en les associant, engendrer deux autres transformations morphologiques qui sont l'ouverture et la fermeture morphologique (transformations idempotentes). On définit :

- L'**Ouverture** de X par B, notée X o B.

C'est l'opération correspondant à l'érosion par B suivie de la dilatation par B. Soit :

$$\mathbf{X \circ B = (X \ominus B) \oplus B}$$

L'ouverture d'une image binaire par un élément structurant circulaire adoucit les bords des formes en supprimant les détails finis de bord et coupe les isthmes étroits.

- La **Fermeture** de X par B, notée X • B.

De manière duale à l'ouverture, la fermeture correspond à la dilatation de X par B suivie de l'érosion par B :

$$\mathbf{X \bullet B = (X \oplus B) \ominus B}$$

La fermeture adoucit également les bords des formes X, bouche les canaux étroits, fusionne les objets proches les uns des autres et bouche les trous de petite taille.
À titre de résumé de la ressource, les différents filtrages morphologiques sont appliqués sur une image binaire de référence, avec un élément structurant à 8-connexité :

Image binaire de référence

Image érodée

Image dilatée

Ouverture

Fermeture

Les ***formes*** sur l'image binaire de référence correspondent aux zones blanches du visage, de la chevelure, du col de chemise, et de la végétation derrière le personnage.
L'***érosion*** érode les contours et supprime les pixels isolés. À l'inverse, la ***dilatation*** élimine les trous et dilate les contours. Ces deux phénomènes sont nettement visibles, notamment, sur la végétation, et les zones claires de la chevelure du personnage.
L'***ouverture*** et la ***fermeture*** adoucissent les contours des formes (visage, végétation, …). Pour une ouverture, une érosion est d'abord utilisée. Cette dernière supprime certaines parties des formes qui ne pourront donc pas être ensuite dilatées. Sur l'image obtenue après ouverture,

certaines formes sont donc moins grandes que sur l'image obtenue après fermeture (chevelure, …)

Exercice Chapitre 4 – Traitement d'images binaires par Morphologie Mathématique

L'analyse par morphologie mathématique vise à modifier la structure et la forme des objets de l'image, par exemple, pour séparer les objets, les discriminer en fonction de leur taille, remplir les trous, …

- ♦ Rappels :

La transformation morphologique modifie la valeur d'un pixel de l'image en fonction de la valeur de ses voisins. Pour cela, on utilise un ***élément structurant***, qui est un masque binaire. Il permet de prendre en compte le voisinage du pixel.
Exemple d'un élément structurant à 4-connexité (le centre est marqué par un cercle) :

0	1	0
1	**1**	1
0	1	0

Les traitements morphologiques sont définis à partir de deux opérations de base qui sont l'***érosion*** et la ***dilatation***.

Définition :

Soient « S » le support de l'image, et « p » l'affixe d'un point P de cette image :

- L'érosion d'une forme X par un élément structurant B est notée « $\mathbf{X \ominus B}$ ». Elle est définie par : $\mathbf{X \ominus B = \{ p \in S,}$ **tel que** $\mathbf{B_p \subseteq X\}}$;
- La dilatation d'une forme X par un élément structurant B est notée « $\mathbf{X \oplus B}$ ». Elle est définie par : $\mathbf{X \oplus B = \{ p \in S,}$ **tel que** $\mathbf{\overline{B_p} \cap X \neq \varnothing\}}$.
- L'érosion et la dilatation sont duales par rapport à la complémentation :

$$X \oplus B = (X^C \ominus B)^C$$

Avec :
- X^C complémentaire de X ;
- $\overline{B}$ élément symétrique de B (i.e. $\overline{B}(i, j) = B(-i, -j)$) ;
- B_p élément transposé de B par le vecteur d'affixe p.

Vous allez utiliser Matlab pour réaliser des filtrages morphologiques. Avant de commencer, ouvrez le fenêtre d'aide de Matlab (commande ***helpwin***) et accédez à la ***Toolbox Image Processing***. Vous aurez ainsi une description détaillée pour chacune des fonctions utilisées dans cet exercice.

1. Erosion et dilatation d'une image binaire

Mettez à jour la liste des chemins dans le ***path browser***. Ouvrez un nouveau fichier « ***M-File*** » dans lequel vous allez saisir vos commandes et dont chaque ligne (se terminant par un « *;* ») sera ensuite interprétée. Chargez l'image *CIRCUIT.TIF* avec ***imread***.

1.1 – Binarisez l'image en utilisant la méthode de votre choix (cf. exercice binarisation du chapitre 2).

1.2 – Par convention le « fond » de l'image binaire doit être noir (à « 0 »), et la forme de l'objet doit être blanche (à « 1 »). Donc, vous pouvez inverser l'image binaire avec l'opérateur « ~ » (**I = ~ I**).

1.3 – Choisissez un élément structurant. Vous pouvez l'écrire directement sous la forme d'une matrice binaire (ex. SE = ***ones***(3), SE = [0 1 0;1 1 1;0 1 0], …). Erodez les formes de l'image binaire avec ***imerode***. Définissez, avec « l'aide de Matlab » (commande ***helpwin***), cette opération.

1.4 – De même, dilatez les formes de l'image binaire avec ***imdilate*** et définissez cette opération avec l'aide de Matlab.

1.5 – Erodez ou dilatez l'image binaire en jouant avec la forme de l'élément structurant. Observez notamment les résultats lorsque l'élément structurant n'est pas symétrique par rapport à son origine.

1.6 – On veut observer la relation de dualité entre l'érosion et la dilatation. Pour cela, créez une image binaire résultat de l'érosion du fond, et une seconde résultat de la dilatation des formes. Comparez ces deux images et concluez.

2. Ouverture (érosion puis dilatation) d'une image binaire bruitée

2.1 – Chargez l'image *CIRCUIT.TIF* et ajoutez lui du bruit avec la fonction ***randn***. Cette fonction permet de générer une matrice d'un bruit gaussien N(0, 1) que vous pouvez amplifier et à ajouter à votre image originale

Remarques :
- les dimensions de la matrice du bruit et de l'image originale doivent être les mêmes ;
- pensez à convertir vos données pour les opérations effectuées (fonctions ***double*** et ***uint8***).

2.2 – Binarisez l'image originale et l'image bruitée, puis comparez les.

2.3 – Choisissez un élément structurant symétrique. Faites l'érosion puis la dilatation de l'image binaire bruitée. La composée de ces deux fonctions s'appelle l'***ouverture*** de l'image. Quel est l'intérêt ?

2.4 – Que ce passe t-il si, cette fois, l'élément structurant n'est pas symétrique ? Proposez et testez une méthode qui garantisse alors, la reconstruction des objets à l'identique. Une piste : il faut penser à faire l'érosion et la dilatation avec un élément structurant différent.

3. Fermeture puis ouverture d'une image binaire

3.1 – Chargez l'image en niveaux de gris *PEARLITE.TIF*. Binarisez et inversez l'image.

3.2 – Faites la fermeture (***imclose***) de cette image binaire inverse. Définissez cette opération (cf. aide Matlab). Pour générer l'ES vous pouvez utiliser la fonction ***strel*** (cette fonction génère des « objets élément structurant »).

3.3 – Effectuez l'ouverture (***imopen***) de l'image binaire précédente. Définissez cette opération.

3.4 – Analysez la chaîne complète des opérations.

Correction de l'exercice : Morphologie Mathématique pour images binaires

1.1 – Voici un exemple de solution pour binariser l'image *CIRCUIT.TIF* :

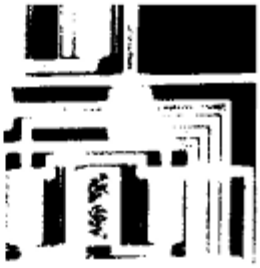
```
I = imread('CIRCUIT.TIF');
seuil = graythresh(I) % recherche du seuil avec la méthode d'Otsu
Ib = im2bw(I,seuil); % binarisation
```

1.2 – Sous Matlab, l'opérateur « ~ » correspond au « non logique ». Vous pouvez donc inverser une image binaire avec la commande : Ibi=~Ib. Pour observer l'opération effectuée, tapez :

```
figure(1)
imshow(Ib)
Ibi =~ Ib;
figure(2)
imshow(Ibi)
```

Ib

Ibi

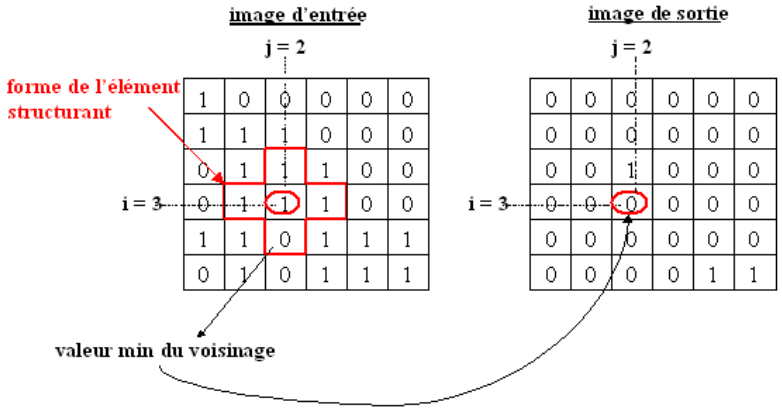


1.3 – on considère, par exemple, l'élément structurant suivant : $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

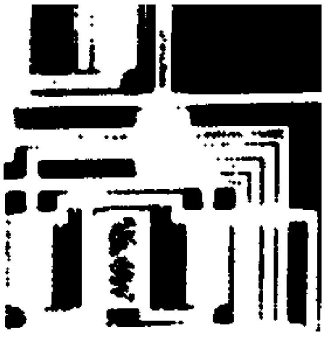
Pour éroder l'image binaire Ibi et afficher le résultat :

```
SE = [0 1 0;1 1 1;0 1 0] ; % élément structurant
Ier = imerode(Ibi,SE) ;
figure(3)
imshow(Ier)
```

L'élément structurant permet de définir un voisinage pour chacun des pixels de l'image d'origine : les pixels voisins sont ceux à 1 au sein de l'élément structurant. On balaie ensuite l'ensemble des pixels de l'image en leur appliquant l'élément structurant. La valeur d'un pixel après érosion est alors définie comme étant la valeur minimale de tous les pixels dans son voisinage. Pour une image binaire, si l'un des pixels du voisinage est à 0, la valeur de sortie du pixel est alors automatiquement 0.
La figure ci-dessous présente un exemple d'érosion, on détaille le cas du pixel d'affixe (3, 2), avec un élément structurant à 4-connexité.



Voici l'image obtenue après érosion :

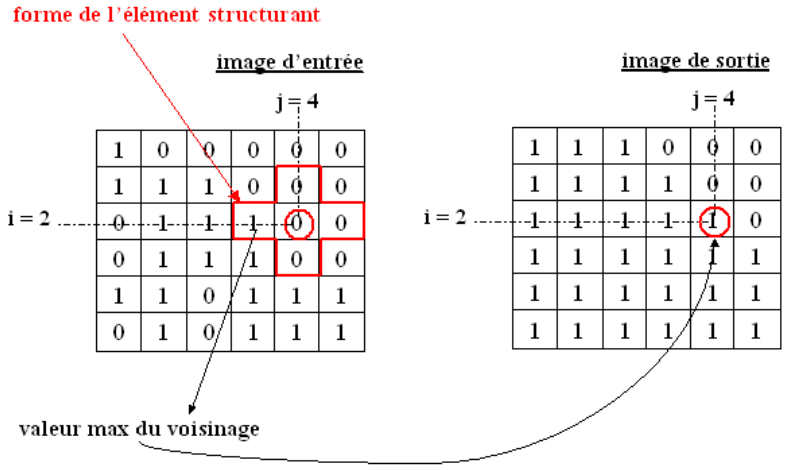


L'érosion a éliminé les pixels isolés sur le fond et a érodé le contour des objets de l'image *CIRCUIT*.

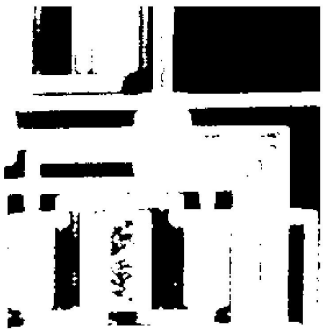
1.4 – Pour dilater l'image binaire Ibi et afficher le résultat :

```
Idi = imdilate(Ibi,SE) ;
figure(4)
imshow(Idi)
```

La valeur de sortie d'un pixel est la valeur maximale parmi tous les pixels compris dans le voisinage. Pour une image binaire, si l'un des pixel du voisinage est à 1, la valeur de sortie du pixel est 1. La figure ci-dessous présente un exemple de dilatation, et détaille le cas du pixel d'affixe (2,4), avec un élément structurant à 4-connexité.



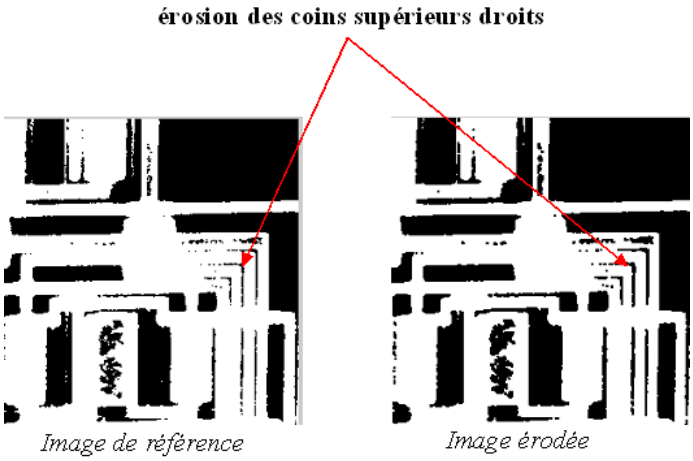
Voici l'image obtenue après dilatation :



La dilatation élimine les trous isolés dans les objets et dilate les bords des objets.

1.5 – On érode l'image avec l'élément structurant suivant : $\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

Dans ce cas, l'élément structurant n'est pas symétrique. Cette érosion tronque les coins supérieurs droits des objets.



Les formes érodées et dilatées dans les objets dépendent donc fortement de la forme de l'élément structurant.

1.6 – On veut montrer que : « dilater la forme c'est éroder le fond puis inverser », soit que : **$Ibi \oplus SE = (Ibi^c \ominus SE)^c$** :

- on créé Idi = $Ibi \oplus SE$:
  Idi = imdilate(Ibi,SE) ;

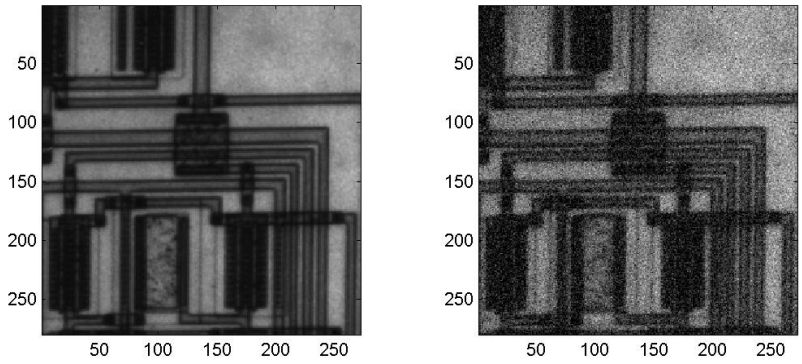
- on créé Idi2= $(Ibi^c \ominus SE)^c$:
  Idi2 = ~imerode(~Ibi,SE) ;

Les deux images obtenues sont identiques. On peut vérifier ce résultat en entrant la commande : ***isequal***(Idi,Idi2). La fonction ***isequal*** retourne 1 si les deux matrices sont identiques, 0 sinon.

2.1 – Voici la liste de commandes pour chargez l'image *CIRCUIT.TIF* et lui ajouter un bruit gaussien N(0, 1) :

```
I = imread('circuit.tif');
% Ajout de bruit :
[nb_lig, nb_col] = size( I ); % taille de l'image
Bruit = 25 * randn(nb_lig, nb_col);
IB = double( I ) + Bruit;
% Conversion
IB = uint8( IB );
```

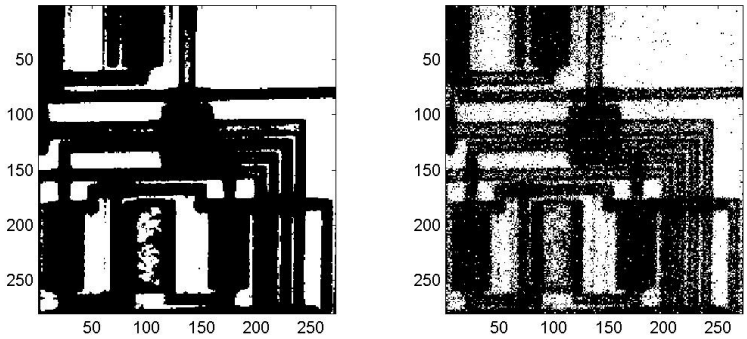
Voici l’image Circuit avant (à gauche) et après (à droite) ajout du bruit :



2.2 – Voici un exemple de script pour construire l’image binaire et l’image binaire bruitée :

```
seuil1 = graythresh(I) ;
Ib = im2bw(I,seuil1);
seuil2=graythresh(IB) ;
IBb = im2bw(IB,seuil2);
subplot(1,2,1)
subimage(Ib)
subplot(1,2,2)
subimage(IBb)
```

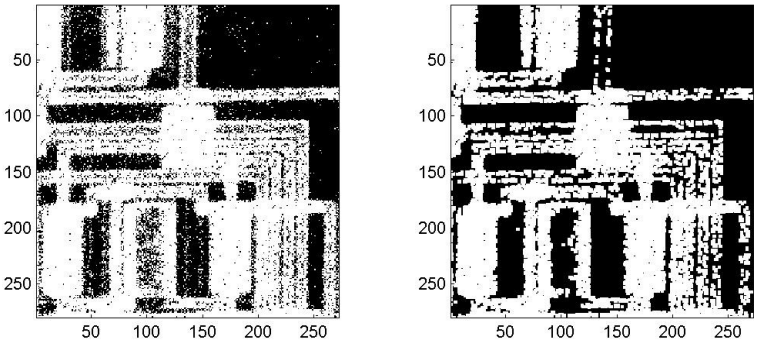
Voici les résultats obtenus:



2.3 – On considère, par exemple, un élément structurant à 8-connexité. Pour réaliser l’ouverture de l’image bruitée, on inverse l’image (afin d’avoir les formes en blanc et le fond en noir), puis on écrit :

```
SE = [1 1 1 ;1 1 1 ;1 1 1] ;
Iouv = imdilate(imerode(~IBb,SE),SE) ;
subplot(1,2,1)
subimage(~IBb)
subplot(1,2,2)
subimage(Iouv)
```

Voici les images obtenues:



Dans le cas présent, l'intérêt d'une telle transformation est d'éroder dans un premier temps les formes de l'image pour supprimer les pixels isolés qui correspondent au bruit, puis de dilater les formes de l'image afin de leur rendre une proportion proche de celle qu'elles avaient avant l'érosion. Le bruit est ainsi atténué.

2.4 – Si l'élément structurant n'est pas symétrique, l'érosion va bien atténuer le bruit de l'image (pixels isolés sur le fond). Pour reconstruire les formes initiales, on applique une dilatation avec le même élément structurant.

- ♦ Mise en garde :

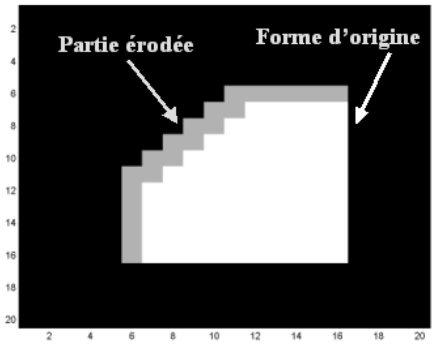
Il faut prendre garde à bien comprendre la définition d'une dilatation :

$$X \oplus B = \{ p \in S, \text{ tel que } \overline{B_p} \cap X \neq \emptyset \}$$

Le résultat de la dilatation est l'ensemble des points, tels que si on leur applique l'élément structurant **symétrique**, l'intersection avec la forme X n'est pas l'ensemble vide.

On considère, par exemple, un élément structurant non symétrique : $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

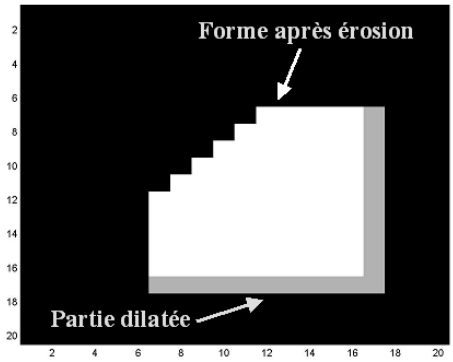
Le coin supérieur gauche est donc tronqué par dilatation:



Pour reconstruire la forme, on veut utiliser une dilatation.
Considérons que la définition de la dilation **ne tienne pas** compte de la symétrie : $X \oplus B = \{ p \in S, \text{ tel que } B_p \cap X \neq \emptyset \}$.

Attention cette définition est erronée et a pour unique but de vous présenter une erreur classique à ne pas commettre avec un élément structurant non symétrique.

Avec une mauvaise définition de la dilatation, on s'aperçoit que la reconstruction avec le même élément structurant n'aboutit pas :

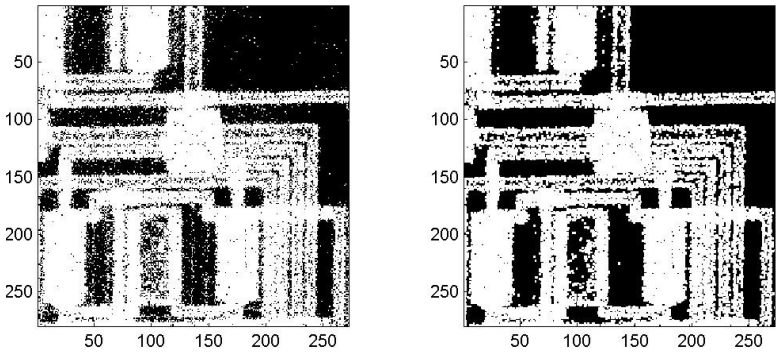


Pour reconstruire la partie grisée, il faut utiliser l'élément structurant suivant : $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ qui est le symétrique de l'élément structurant de départ.
Afin de créer des ouvertures et des fermetures qui garantissent cette relation de symétrie, et donc une reconstruction cohérente des formes, **la définition de la dilatation est caractérisée par l'élément structurant symétrique de celui utilisé lors de l'érosion**.
Dans le cas d'une ouverture avec un élément structurant symétrique, le problème ne se pose évidemment pas.

Voici un exemple de script pour réaliser l'ouverture de l'image bruitée binarisée *Circuit* avec un élément structurant non symétrique :

```
SE1 = [1 1 0 ;1 1 0 ;0 0 0]
SE2 = [0 0 0 ;0 1 1 ;0 1 1]
Iouv = imdilate(imerode(~IBb,SE1),SE2) ;
```

Voici les résultats obtenus :



Le bruit est atténué et les formes sur l'image de sortie gardent les mêmes proportions que sur l'image d'entrée.

3.1 – Voici la commande pour charger l'image *Pearlite.tif*, la binariser et l'inverser :

```
I = imread('pearlite.tif');
L = graythresh(I)
I = ~im2bw(I,L);
```

3.2 – Pour réaliser la fermeture de cette image on créé un élément structurant avec la fonction ***strel*** de Matlab :

```
SE = strel('disk', 6)
```

Dans la console on a alors :

```
SE =

Flat STREL object containing 109 neighbors.
Decomposition: 6 STREL objects containing a total of 22 neighbors

Neighborhood:
     0     0     1     1     1     1     1     1     1     0     0
     0     1     1     1     1     1     1     1     1     1     0
     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1
     0     1     1     1     1     1     1     1     1     1     0
     0     0     1     1     1     1     1     1     1     0     0
```

l'image de la fermeture est alors obtenue par la commande :

```
Ifer = imclose(I,SE);
```

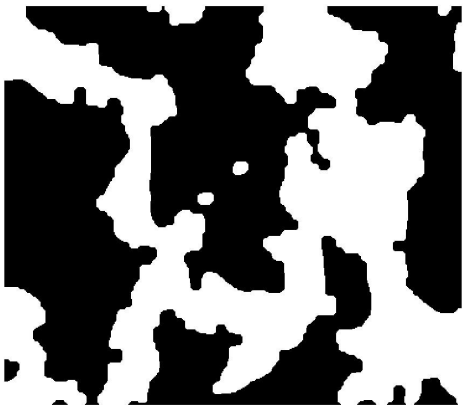
Voici l'image obtenue:

La fermeture adoucit les bords des formes, bouche les canaux étroits, fusionne les objets proches les uns des autres et bouche les trous de petite taille.

3.3 – Pour réaliser l’ouverture de l’image précédente, on utilise la fonction ***imopen*** de Matlab :

```
Im = imopen(Ifer,SE);
```

On obtient alors l’image suivante:



L’ouverture d’une image binaire adoucit les bords des formes en supprimant les détails.

3.5 – La succession de la fermeture puis de l’ouverture nous a permis d’extraire les objets larges de l’image binaire, et de réaliser une segmentation. Ces deux traitements sont typiquement utilisés pour adoucir les bords, compléter, ou pour enlever des objets dans une image binaire. Le choix de l’élément structurant dépend de la taille et de la forme des objets à modifier.

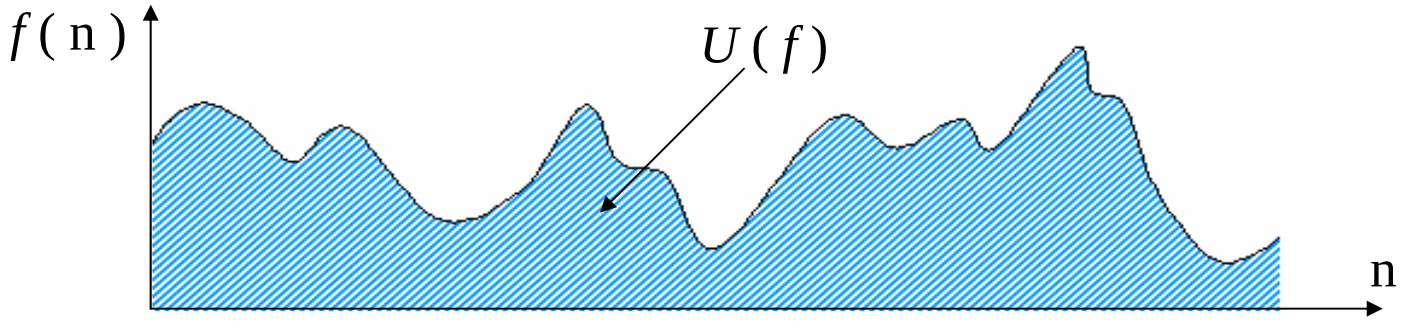
Filtrage morphologique des images monochromes

- Extension de la morphologie binaire à travers la notion d'***ombre*** (ou ***sous-graphe***) de la fonction image

- Définitions :
  - niveau de gris ***« f »*** d'un pixel P : f (m, n) où (m, n) est l'affixe de P
  - sous-graphe ***« U (f) »*** : $U(f) = \{ (m, n, l), l \in R$, tel que : $l \leq f(m, n) \}$
  - reconstruction **T** de l'image « f » à partir de son sous-graphe $U(f)$:

$$f(m, n) = T[U(f)] = \mathrm{Sup}_l \{ l \text{ tel que } (m, n, l) \in U(f) \}$$

  - exemple (cas d'un signal 1-D) :



Érosion d'une image en niveaux de gris

➢ L'érosion (notée ⊖) d'une image monochrome « *f* » par un élément structurant « B » est définie en terme de sous-graphe :

- $Y = U(f) \ominus U(B)$
- $f \ominus B = T[Y] = Sup_l \{ l \text{ tel que } (m, n, l) \in Y \}$

➢ Classiquement, on considère un élément structurant plan de valeur nulle sur son support

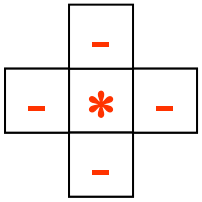
$(f \ominus B)(P) = \text{Min} \{ \text{valeurs des pixels du voisinage du pixel P} \}$

▪ exemple :

m \ n			
	115	91	77
	95	**68**	90
	55	151	210

Image de référence

⊖



Élément structurant (4-connexité)

liste ordonnée des valeurs du voisinage :

68 , 90 , 91 , 95 , 151

min

$(f \ominus B)(P) = 68$

Dilatation d'une image en niveaux de gris

➢ Comme pour l'érosion, la dilatation (notée $\oplus$) est définie en terme de sous-graphe :

- $Y = U (f) \oplus U (B)$
- $f \oplus B = T [Y] = Sup_l \{ l \text{ tel que } (m, n, l) \in Y \}$

➢ En considérant un élément structurant plan de valeur nulle sur son support :

$$(f \oplus B) (P) = \text{Max} \{ \text{valeurs des pixels du voisinage du pixel P} \}$$

▪ exemple :

m \ n			
	115	91	77
	95	68	90
	55	151	210

Image de référence

$\oplus$

-	-	-
-	*	-
-	-	-

Élément structurant (8-connexité)

liste ordonnée des valeurs du voisinage :

55 , 68 , 77 , 90 , 91 , 95 , 115 , 151 , **210**

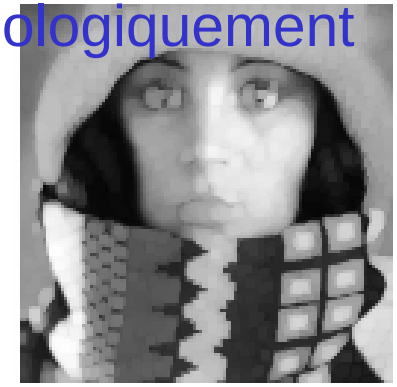
max

$(f \oplus B) (P) = 210$

Exemples de filtrages morphologiques
(élément structurant 3×3 complet)

- Image de réf

- Image dilatée morphologiquement



➢ Observation :

La **dilatation** en niveaux de gris accroît la luminance des pixels entourés de voisins plus lumineux

- Image é

➢ Observation :

L'**érosion** en niveaux de gris réduit la luminance des pixels qui sont entourés de voisins de moindre intensité

Exercice Chapitre 4 – Filtrage morphologique pour des images monochromes

Dans cet exercice, il s'agit d'étendre aux images monochromes, les traitements morphologiques qui ont été vus pour des images binaires.
Chargez l'image en niveaux de gris *CAMERAMAN.TIF* et mettez à jour la liste des chemins dans le « path browser ».

♦ Remarque : dans le cas des images binaires, les formes étaient à 1 et le fond à 0. Pour obtenir une telle représentation, nous avons vu qu'il faut inverser l'image avec l'opérateur « ~ » pour la plupart des images naturelles. Dans le cas des images naturelles en niveaux de gris, les objets (formes) sont plutôt sombres et le fond clair. On réalise donc le même type d'opération d' « inverse vidéo » grâce à la fonction ***imcomplement*** (tapez *help imcomplement* pour plus d'informations). Ainsi, les formes sont claires et le fond est sombre.

Filtrage morphologique d'images monochromes

1 – Erodez l'image *Cameraman.tif* (***imerode***) avec l'élément structurant ES défini par : ES=***strel***('ball',5,5).
Affichez le résultat et comparez le avec l'image avant traitement. A l'aide des définitions données dans la ***toolbox Image Processing*** de Matlab, donnez une définition de l'érosion pour les images en niveaux de gris.

2 – De même, Dilatez l'image *Cameraman.tif* (***imdilate***) avec le même élément structurant. Affichez le résultat et comparez le avec l'image avant traitement. Toujours à l'aide des définitions données dans la *toolbox Image Processing*, donnez une définition de la dilatation pour les images en niveaux de gris.

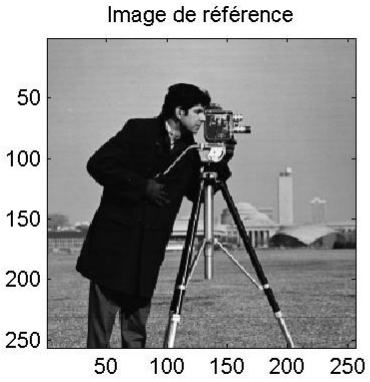
3 – Réalisez une ouverture et une fermeture de l'image *Cameraman.tif*. Observez et interprétez les résultats obtenus.

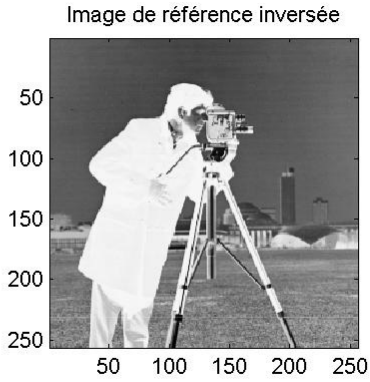
Correction de l'exercice : Filtrage morphologique d'images monochromes

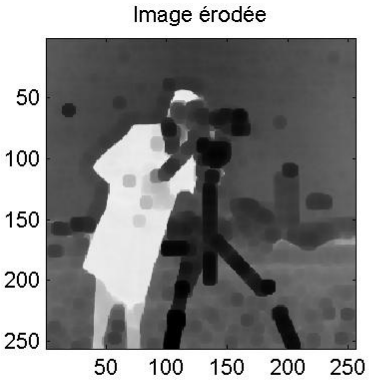
1 – Voici les commandes pour éroder l'image et comparer le résultat avec l'image d'origine :

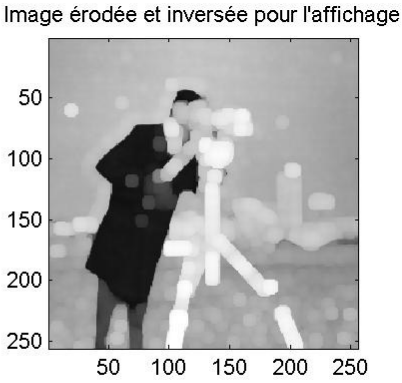
```
% Chargement de l'image
I = imread('cameraman.tif') ;
Ic = imcomplement(I); % inverse vidéo de l'image de départ
figure(1)
subplot(1,2,1)
subimage(Ic)

% Erosion
SE = strel('ball',5,5) % définition de l'élément structurant
Ierod = imerode(Ic,SE);
subplot(1,2,2)
subimage(Ierod)

% Affichage des images en niveaux de gris non-inversés
figure(2)
subplot(1,2,1)
subimage(I)
subplot(1,2,2)
subimage(imcomplement(Ierod))
```

Voici les images obtenues avant et après érosion :







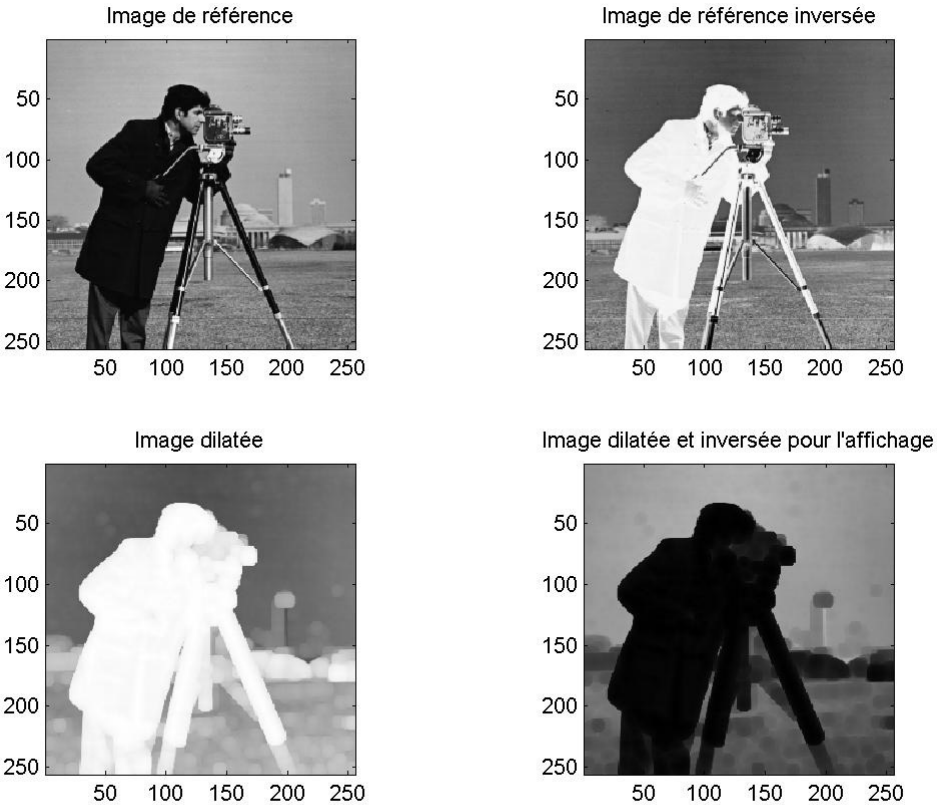


Une érosion en niveaux de gris réduit l'intensité lumineuse des pixels qui sont entourés de voisins de moindre intensité (visible sur les images inversées). Ce voisinage est défini par l'élément structurant. L'ensemble des pixels de l'image est balayé en appliquant l'élément structurant. **La valeur d'un pixel après érosion est alors définie comme étant la valeur minimale de tous les pixels de son voisinage**.

2 – Voici les commandes pour dilater l'image et comparer le résultat avec l'image d'origine :

```
% Dilatation
figure(3)
Idilat = imdilate(Ic,SE);
subplot(1,2,1)
subimage(Idilat)
subplot(1,2,2)
subimage(imcomplement(Idilat))
```

Voici les images obtenues avant et après dilatation :

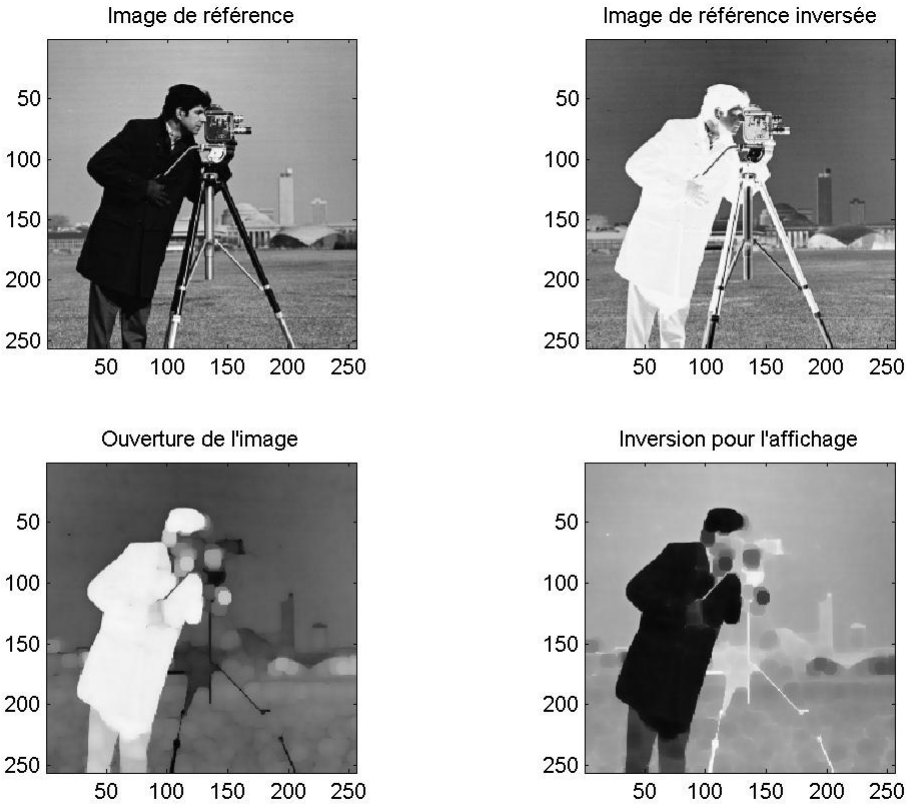


Une dilatation en niveaux de gris accroît l'intensité lumineuse d'un pixel entouré de voisins plus lumineux (visible sur les images inversées). L'ensemble des pixels de l'image est balayé en appliquant l'élément structurant. **La valeur d'un pixel après dilatation est alors définie comme étant la valeur maximale de tous les pixels compris dans le voisinage**.

3 – Voici un exemple de script pour réaliser l'ouverture de l'image et comparer le résultat avec l'image d'origine :

```
% Ouverture
figure(4)
Iouv = imopen(Ic,SE);
subplot(1,2,1)
subimage(Iouv)
subplot(1,2,2)
subimage(imcomplement(Iouv))
```

Voici les images obtenues avant et après l'ouverture :

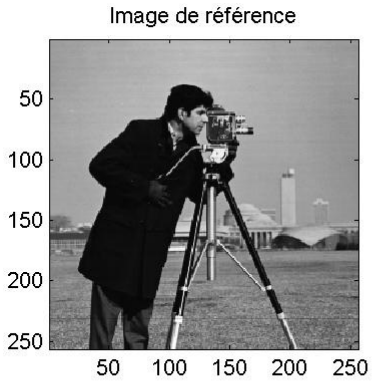


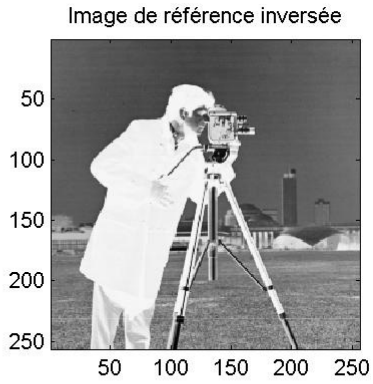
Une ouverture en niveau de gris est une érosion suivie d'une dilatation en niveaux de gris. Elle élimine les points clairs isolés et lisse les contours. On observe également une segmentation des différentes formes de l'image.

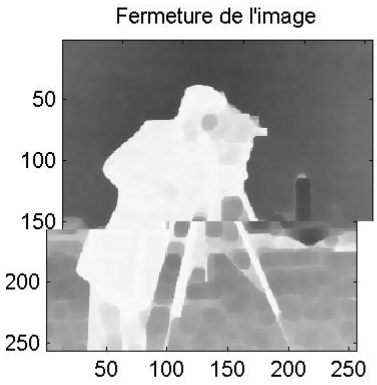
Voici un exemple de script pour réaliser la fermeture :

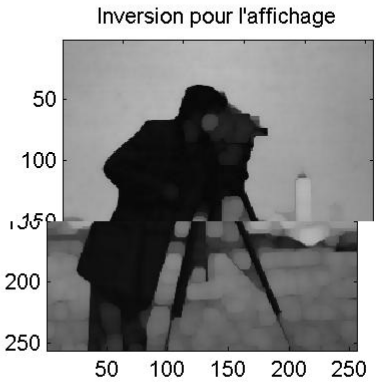
```
% Fermeture
figure(5)
Iferm = imclose(Ic,SE);
subplot(1,2,1)
subimage(Iferm)
subplot(1,2,2)
subimage(imcomplement(Iferm))
```

Voici les images obtenues avant et après la fermeture :









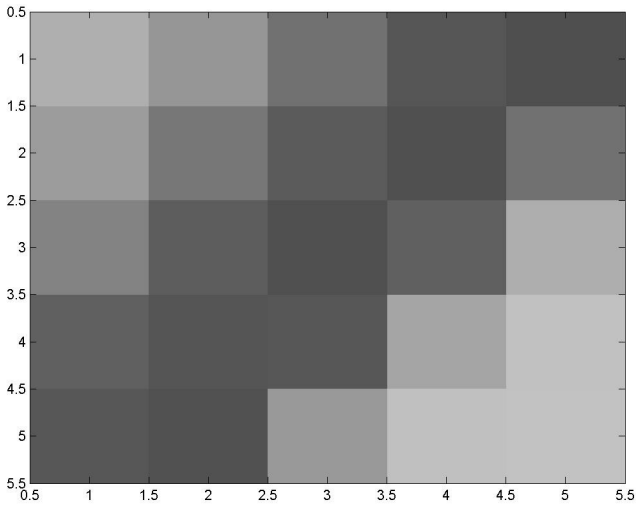
Une fermeture en niveaux de gris est une dilatation suivie par une érosion en niveaux de gris. Elle élimine les points sombres isolés et lisse les contours.

Chapitre 4 – Filtrage non-linéaire

TEST

Partie 1 :

On souhaite utiliser différents filtres pour traiter la portion d'image 5×5 suivante :



Cette portion d'image est extraite de l'image « Bateaux » et contient une partie du filin de pêche. Voici la matrice qui donne les valeurs numériques de luminance des 3×3 pixels :

175	150	114	86	79
156	119	91	80	113
132	93	80	96	174
96	85	87	165	193
87	82	153	192	194

On ignore ici les effets de bords, et on s'intéresse donc uniquement aux résultats obtenus après filtrage pour les 3×3 pixels centraux. :

X	X	X	X	X
X	?	?	?	X
X	?	?	?	X
X	?	?	?	X
X	X	X	X	X

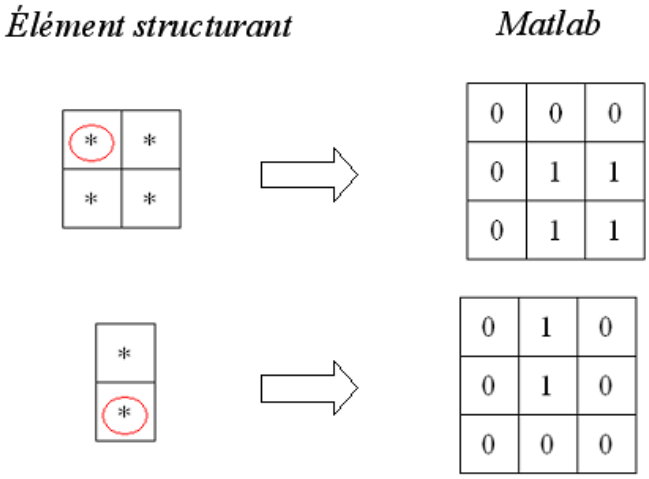
Complétez cette matrice dans chacun des cas suivants :

- filtrage médian 3×3 complet ;
- filtrage médian 3×3 en croix ;
- érosion avec un élément structurant 3×3 complet ;
- dilatation avec un élément structurant 3×3 complet ;
- fermeture avec un élément structurant 3×3 en croix.

Partie 2 :

Développez respectivement pour l'érosion et la dilatation une fonction sous Matlab (ne pas traiter les effets de bord). Ces deux fonctions ont donc en paramètres d'entrée : l'image à filtrer et l'élément structurant. L'élément structurant considéré s'inscrira dans un support de taille 3×3.

- Exemples :



À partir de vos deux fonctions précédentes, créez les fonctions pour la fermeture et l'ouverture d'une image.

Testez vos quatre fonctions sur l'image « Bateaux ». Comparez les résultats (par exemple avec la fonction ***isequal***) obtenus avec ceux obtenus en utilisant directement les fonctions ***imdilate***, ***imerode***, ***imclose***, et ***imopen*** de la toolbox « *Image Processing* » de Matlab.