

Université Paul Sabatier Toulouse

Compte Rendu

Travaux pratiques de Traitement d'Images

M1 EEA

- ✗ Module KEAX7AK1 : Traitement des images
- ✗ Enseignant : A. Herbulet

- ✗ Réalisé par :
 - KABOU Abdeldjalil
 - IAICHE Achour Mehdi Anis

Travaux pratiques de Traitement d'Images

Objectif

Le but est de manipuler les méthodes de morphologie mathématique.

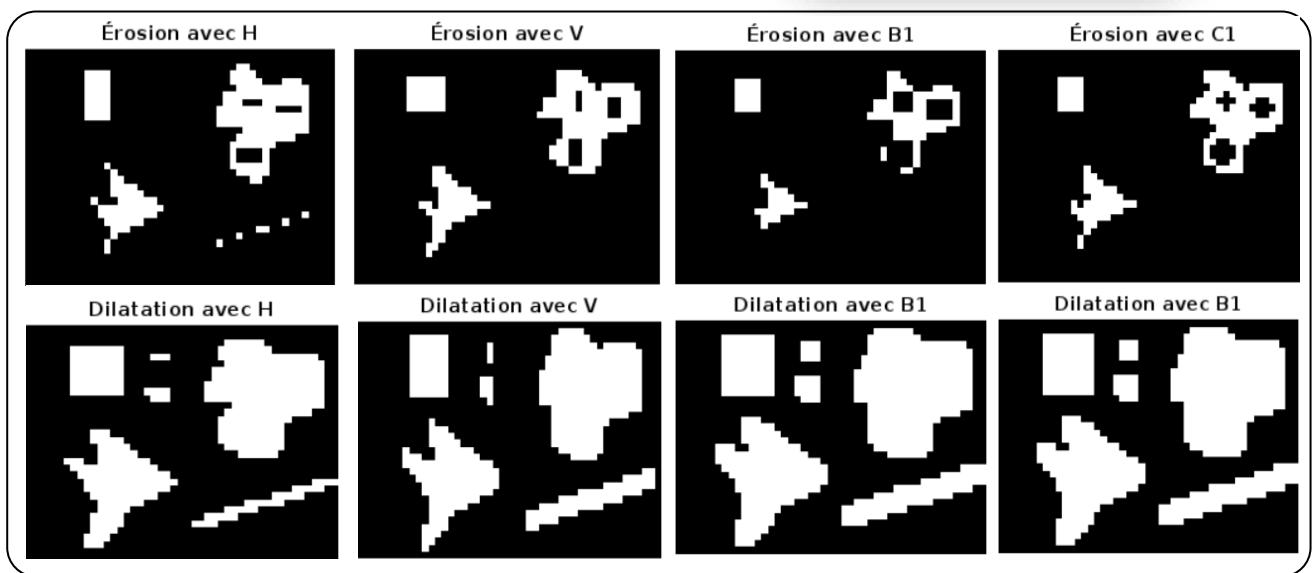
1 Morphologie mathématique

1.1 Erosion et dilatation

- *Opérations d'érosion et de dilatation avec les éléments structurants suivants :*

$$H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad V = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$
$$B_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad C_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Image Originale

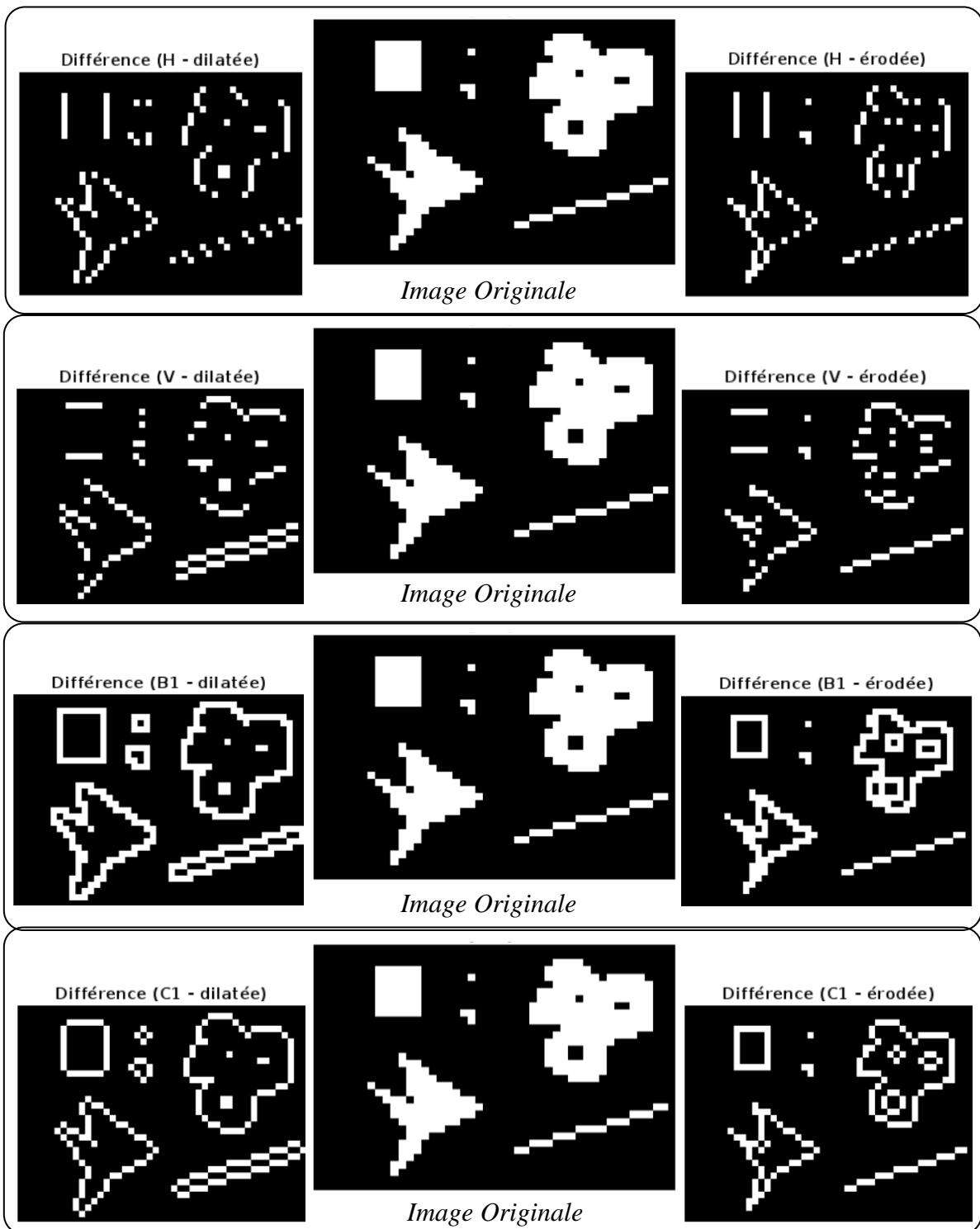


- *Observation :*

Comme nous avons vu dans le cours, avec la **dilatation**, nous comblons les petits trous (nous bouchons les trous plus petits que l'élément structurant) et les objets deviennent plus grands, tandis qu'avec l'**érosion**, nous éliminons les composants plus petits et les objets deviennent plus fins.

- *Différences entre l'image dilatée/érodée et l'image originale :*

Pour afficher les différences, on soustrait l'image originale à l'image dilatée et on soustrait l'image érodée de l'image originale

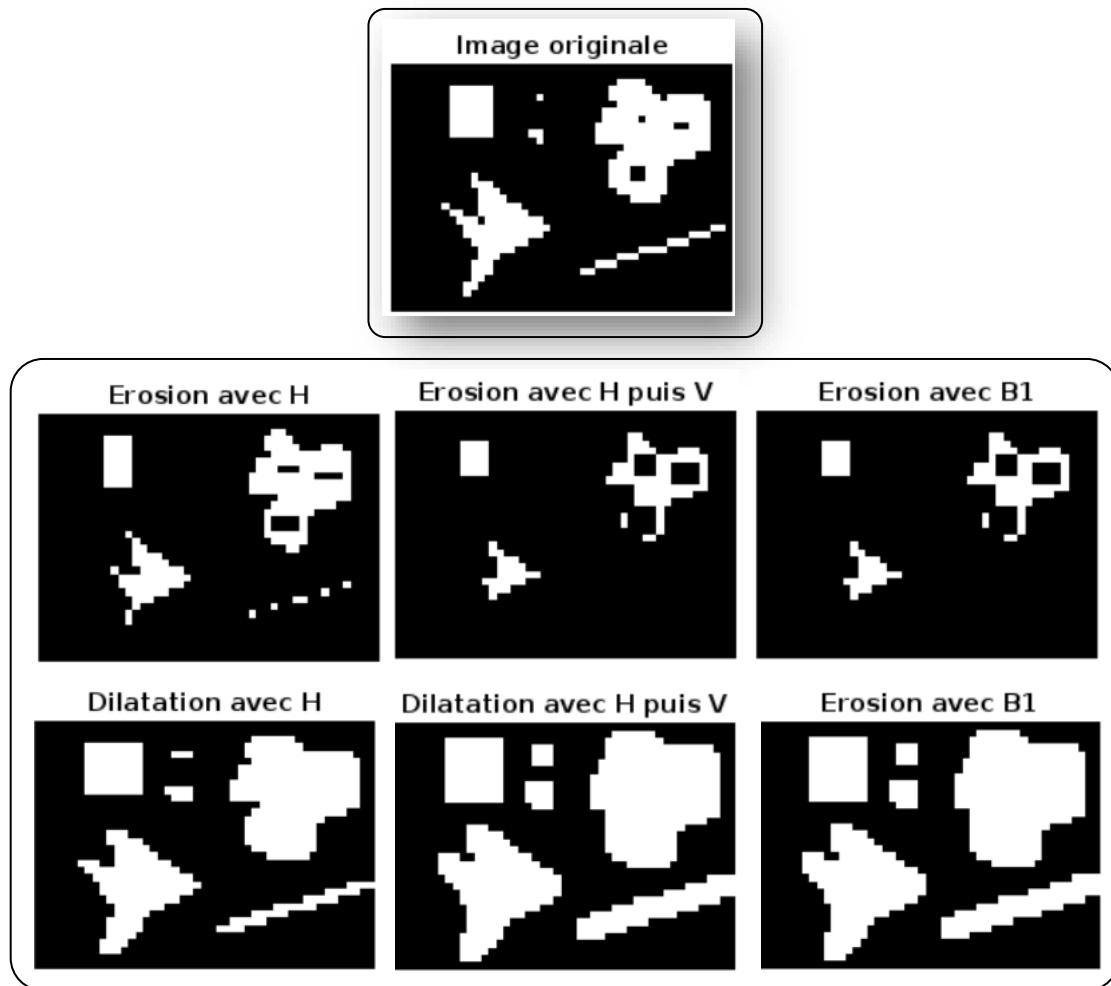


- *Observation :*

Les différences entre l'image originale et l'image dilatée ou l'image érodée confirment ce que nous avons observé, en permettant de visualiser les changements introduits par chaque opération morphologique :

- *La dilatation* montre où de nouveaux pixels ont été ajoutés par rapport à l'original.
- *L'érosion* montre où des pixels ont été retirés

- L'opération d'érosion avec le filtre H puis avec le filtre V



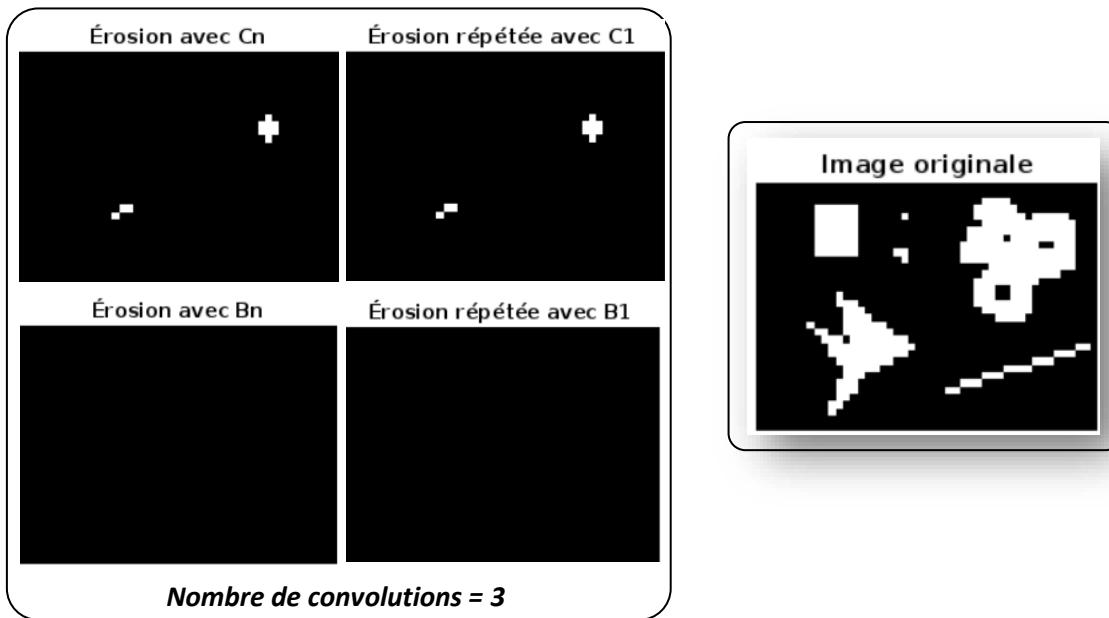
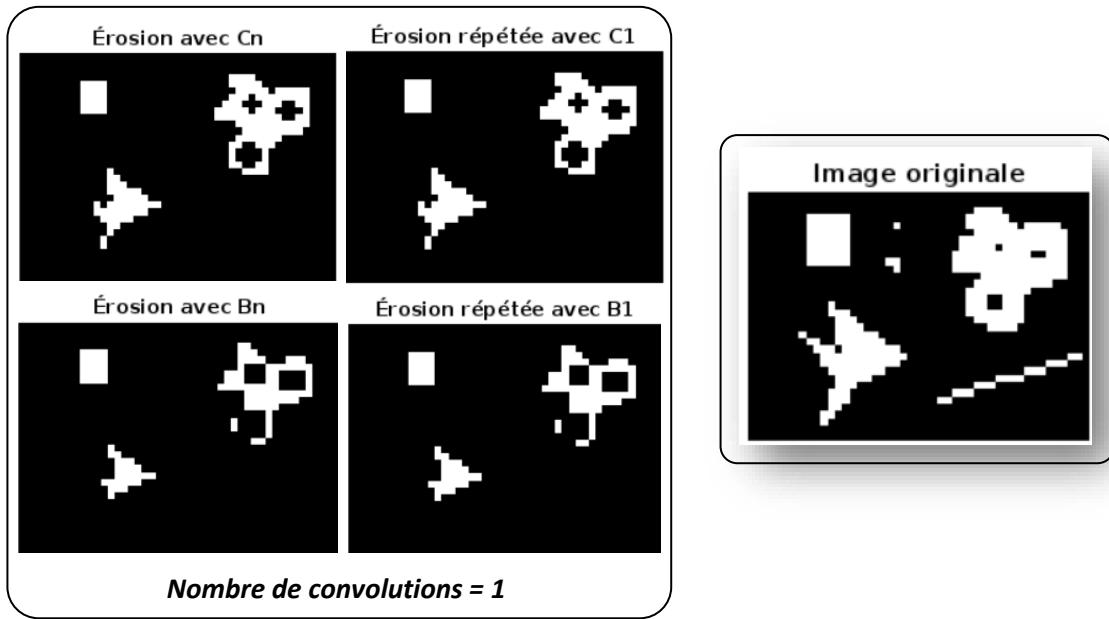
Observation

Nous observons que lorsque nous appliquons l'opération d'érosion ou de dilatation à l'image en utilisant les éléments structurants H et V séparément, et que nous comparons le résultat avec l'opération faite avec B1, nous obtenons un résultat similaire.

Pourquoi ?

Car l'élément structurant B1 effectue des opérations dans toutes les directions (horizontalement, verticalement, et diagonalement), tandis que H et V appliquent l'opération uniquement sur des lignes horizontales et verticales respectivement. Lorsque ces deux opérations sont combinées, l'effet global peut ressembler à l'opération effectuée par B1.

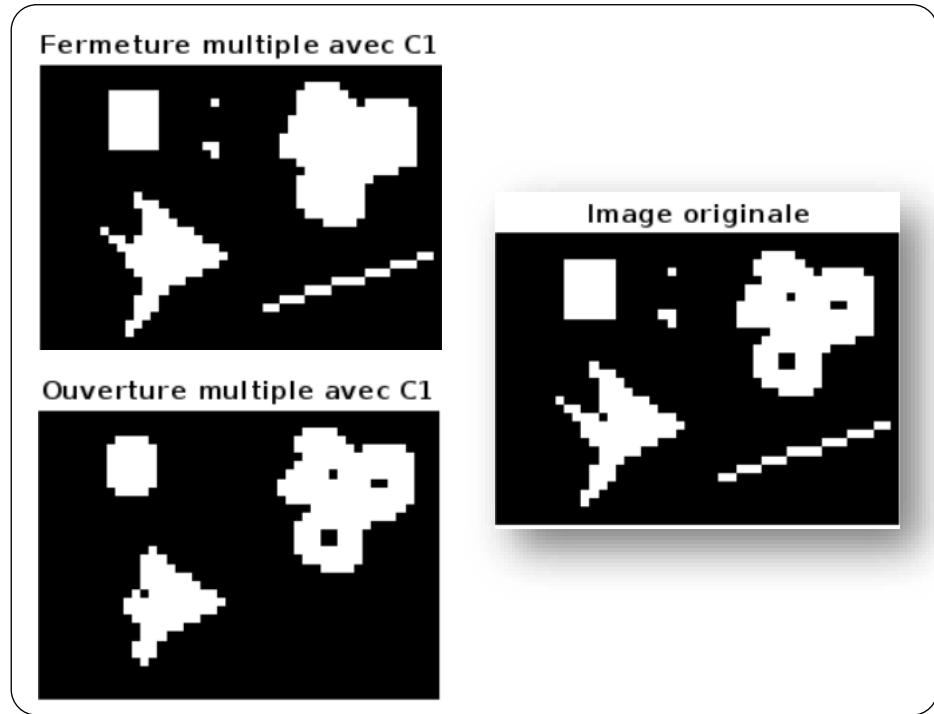
- Vérification qu'en effectuant une érosion avec B_n (resp. C_n), on obtient les mêmes résultats qu'en effectuant une érosion n fois par B_1 (resp. C_1).



Nous remarquons que l'érosion avec l'élément structurant B_n (ou C_n), obtenu par convolution n fois de B_1 (ou C_1), donne le même résultat que l'application de l'érosion n fois de suite avec B_1 (ou C_1).

1.2 Ouverture et Fermeture

- Application de plusieurs opérations d'ouverture et de fermeture d'affilée avec le même élément structurant C1 :



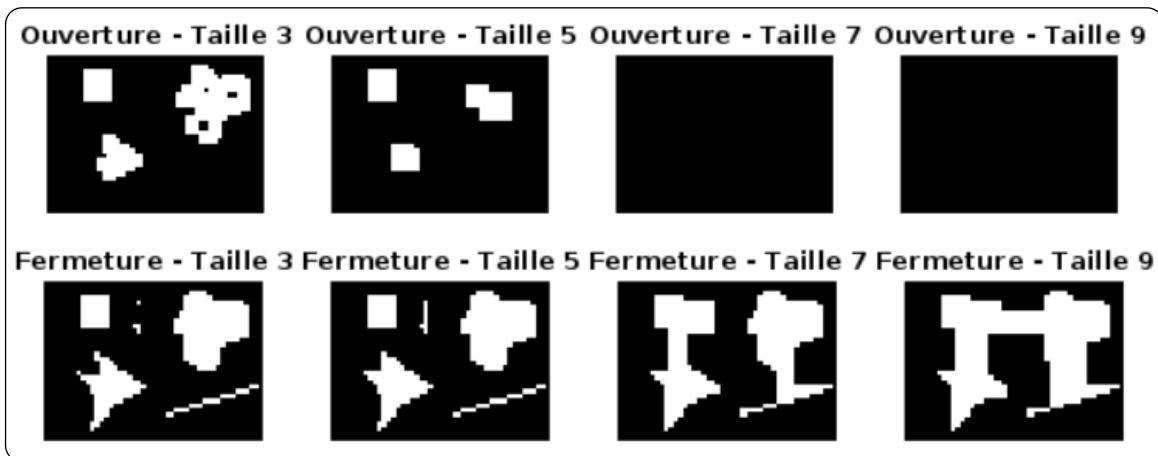
Observation :

On observe qu'avec l'ouverture, l'image est de plus en plus lissée, et les petites structures sont éliminées (supprime les parties de l'ensemble ne contenant pas l'élément structurant).et qu'avec la fermeture les détails fins sont comblés, et les objets deviennent plus réguliers.

Remarque :

Nous avons remarqué que l'application de l'opération plusieurs fois de suite n'apporte plus de changements supplémentaires à l'image. Car les petits objets ont déjà été supprimés, et les structures plus grandes ne sont pas affectées par les itérations suivantes.

- Application des opérations d'ouverture et de fermeture avec des éléments structurants C_n de taille croissante :



Observation :

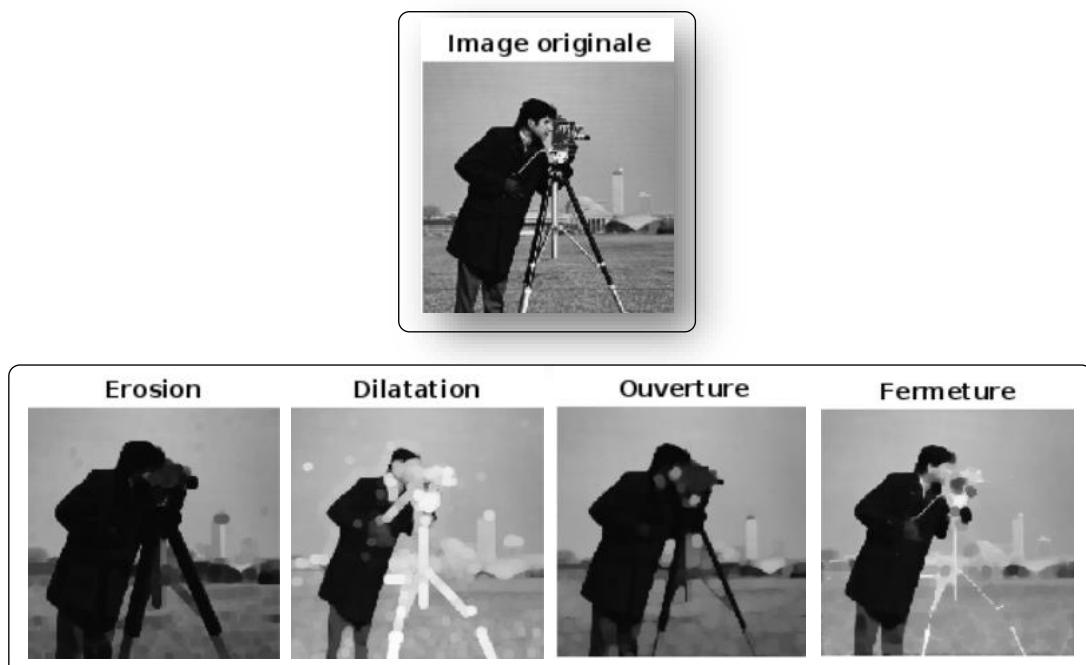
L'ouverture réduit la taille des objets et élimine les petits détails, tandis que la fermeture agrandit et comble les petits trous.

Remarque :

Nous avons remarqué qu'avec l'ouverture, plus la taille de C_n est grande, plus les petits objets sont supprimés, et lorsque dépasse celle des objets présents dans l'image, ceux-ci peuvent être complètement supprimés. Et qu'avec la fermeture plus la taille de C_n est grande, les détails peuvent être complètement comblés

1.3 En niveaux de gris

- Application les opérations d'érosion, de dilatation, d'ouverture et de fermeture sur l'image du « camerama » avec un élément structurant disque de taille 5 :



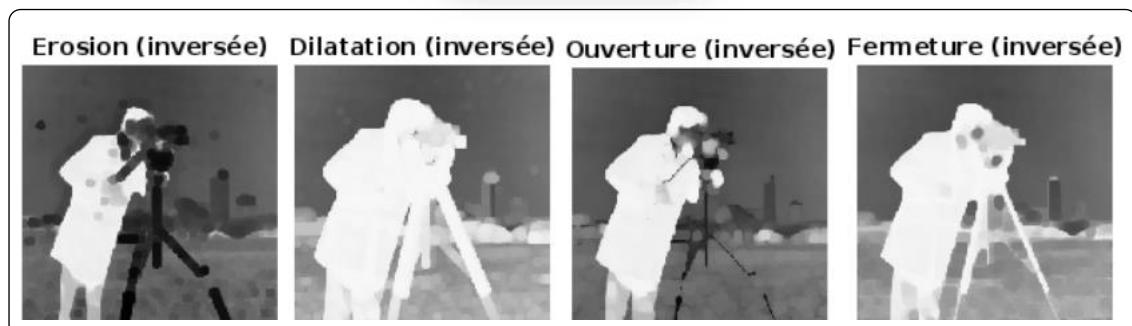
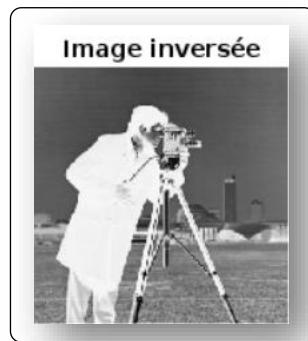
Observation :

L'érosion : tend à réduire les régions claires de l'image et à agrandir les régions sombres.

La dilatation : tend à agrandir les régions claires et à réduire les zones sombres.

L'ouverture : combine l'érosion suivie de la dilatation. Elle lisse les contours de l'image, élimine le bruit et supprime les petites structures brillantes sans modifier la taille globale des objets plus grands.

La fermeture : combine la dilatation suivie de l'érosion. Elle comble les petites dépressions et lisse les contours sans modifier la taille globale des structures sombre



Remarques sur l'image en niveau de gris inverse :

Pour l'image inversée (où les niveaux de gris sont inversés), les résultats des opérations changent car les zones sombres et claires sont échangées :

Érosion : Elle réduira alors les zones sombres et agrandira les zones lumineuses de l'image inversée.

Dilatation : Agrandira les régions sombres et réduira les zones claires de l'image inversée.

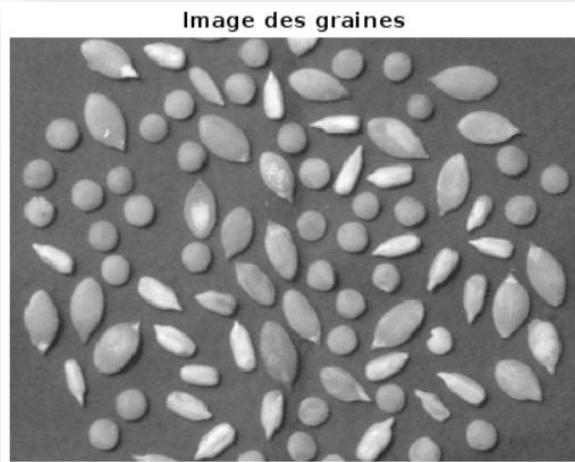
Ouverture et fermeture : Elles produiront des effets similaires à ceux de l'image originale, mais appliqués aux régions opposées (les détails sombres seront lissés ou comblés).

2 Classification de graines

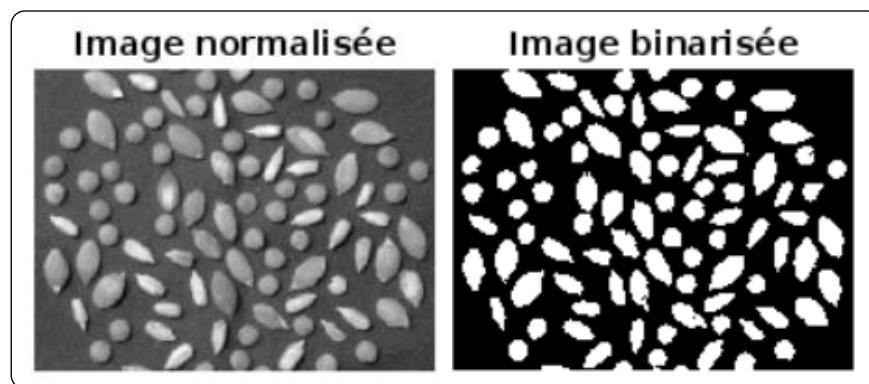
2.1 Prétraitement

- L'image « Seeds.png »

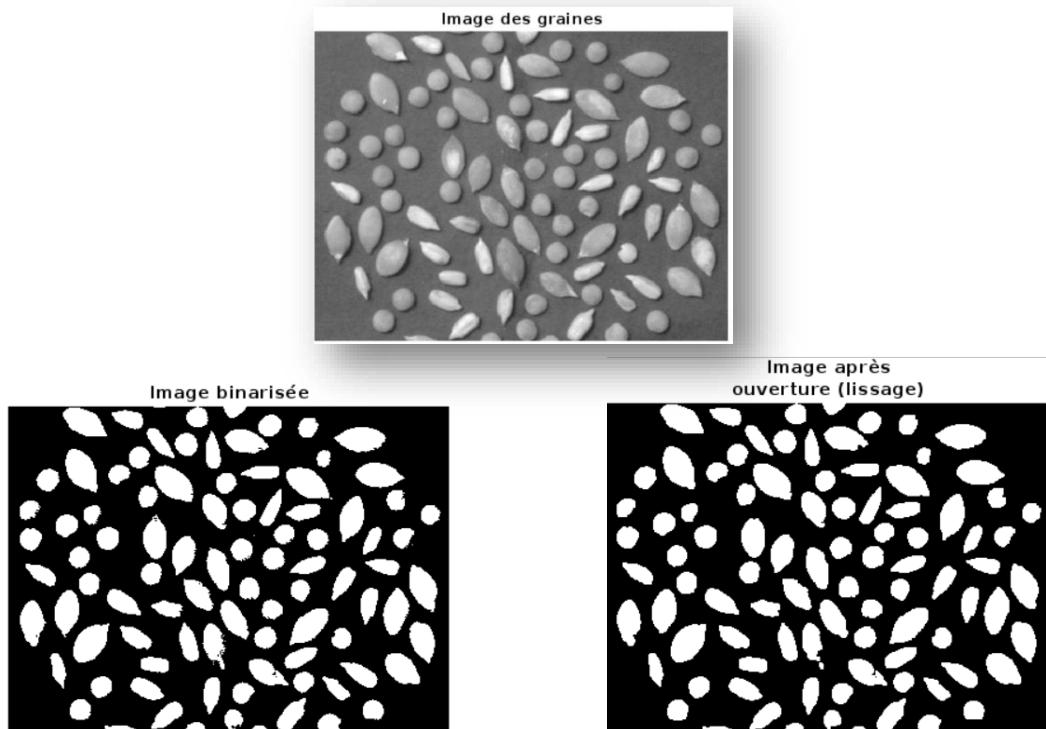
On observe les différences en termes de taille, forme, couleur ou texture.



- Normalisation et Binarisation :



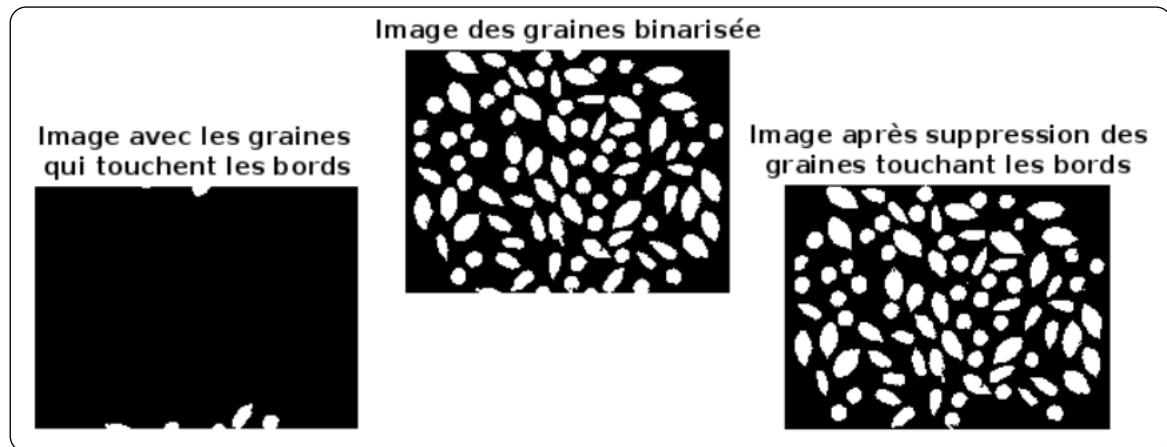
- Lissage l'image en réalisant une ouverture avec un disque de petite taille.



- *La reconstruction entre le bord de l'image et l'image binariser*

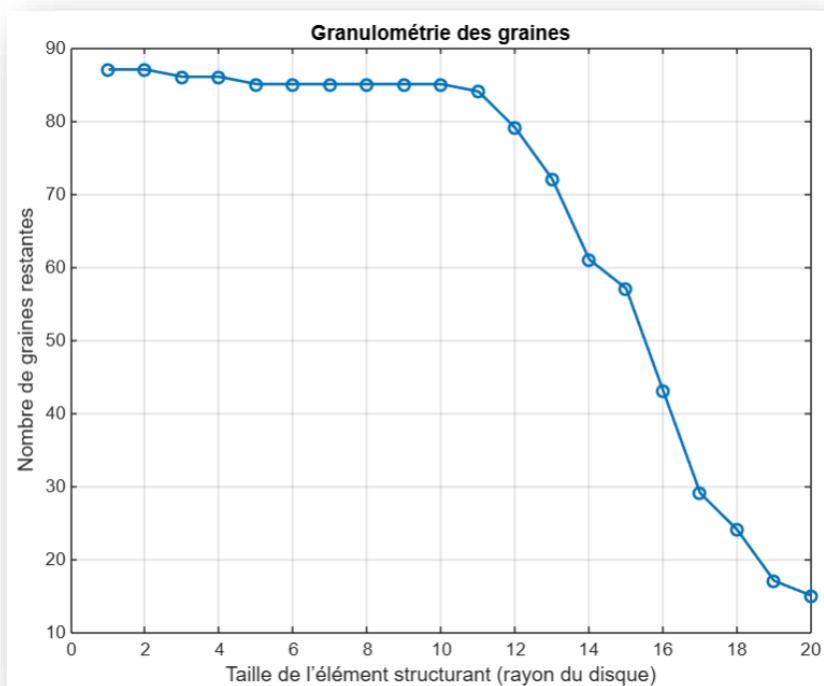
Pour supprimer les graines coupées par le bord de l'image (**border kill**), vous pouvez utiliser une approche basée sur la **reconstruction morphologique**

- Créer une image avec des pixels de valeur **1** sur les bords et **0** ailleurs.
Cela servira de masque pour identifier les graines touchant les bords
- Pour supprimer ces graines, soustrayer l'image reconstruite de l'image binarisée



2.2 Granulométrie

Cette étape (La granulométrie) consiste à appliquer des opérations d'ouverture (éliminer les petits objets et lisser les contours des objets) successives avec des éléments structurants de tailles croissantes (dans notre cas, des disques). Chaque ouverture agit comme un "tamis" qui élimine les graines plus petites que l'élément structurant utilisé.



Observation :

Nous observons que ce graphique nous montre à quelle taille de disque (élément structurant) les graines disparaissent :

- *Graines plus petites* sont éliminées dès les premiers rayons (petits disques).
 - *Graines plus grandes* résistent aux disques de petite taille et ne disparaissent que pour des rayons plus grands.
- *Tailles d'éléments structurant permettant d'obtenir uniquement les "grosses" graines puis les "grosses et moyennes" graines*

Pour les "grosses" graines : à partir de la courbe on remarque que seules les graines plus grandes qui persistent lorsque la taille de disque est grande.

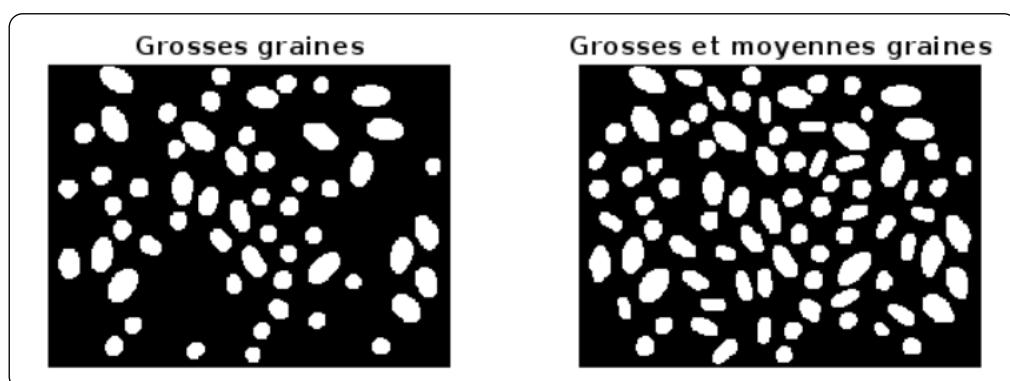
Pour les "grosses et moyennes" graines : à partir de la courbe on remarque qu'avant l'élimination des petites graines (première diminution significative), on trouve les moyennes et grosses graines.

Donc on peut donner un scénario hypothétique basé sur une courbe typique :

- *Taille des disques* : de 1 à 20.
- *Nombre de graines* :
 - ✗ À rayon du disque = 5 : 100 graines.
 - ✗ À rayon du disque = 10 : 60 graines.
 - ✗ À rayon du disque = 15 : 20 graines.
 - ✗ À rayon du disque = 20 : 5 graines.

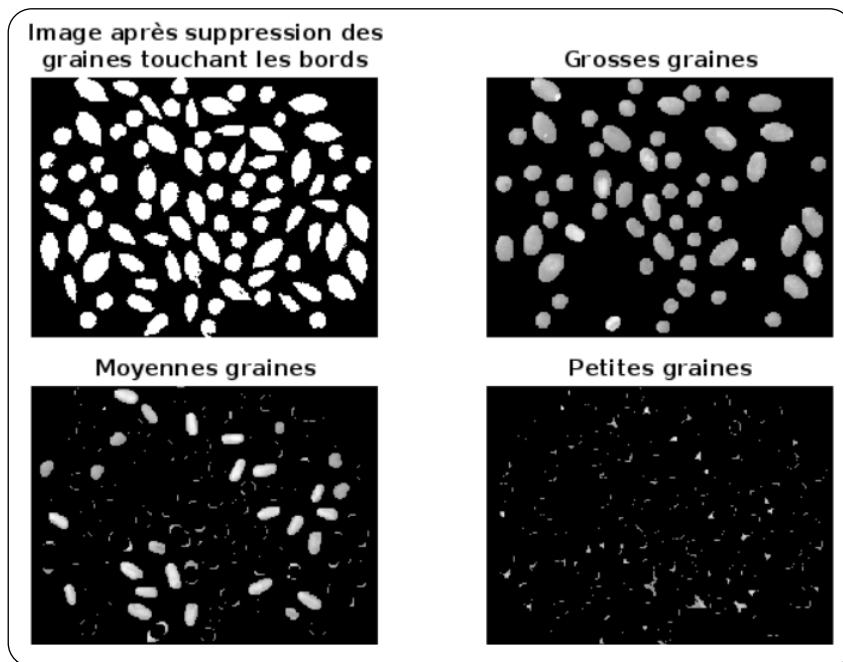
A partir de cela on peut choisir :

- ✗ *Moyennes graines* commencent à disparaître autour de rayon = 10.
- ✗ *Grosses graines* persistent jusqu'à rayon = 15 ou plus.



- Tailles Application des trois différents masques à l'image originale* (plutôt à l'image binaire ou l'image après suppression des graines touchant les bords) pour obtenir les images de chaque type de graines

*: Nous ne pouvons pas travailler directement sur l'image originale, nous devons **binariser l'image** avant de travailler sur les graines. La raison est que l'analyse morphologique des graines, telle que l'extraction des formes ou la granulométrie, repose généralement sur une représentation binaire de l'image, où les objets (les graines) sont séparés du fond. Donc, l'image originale ne peut pas être utilisée directement pour des opérations morphologiques comme l'ouverture ou l'extraction des graines.



Commentaires :

Pour la précision visuelle, est-ce que les graines séparées dans chaque masque correspondent-elles bien aux catégories attendues (petites, moyennes, grosses) ?

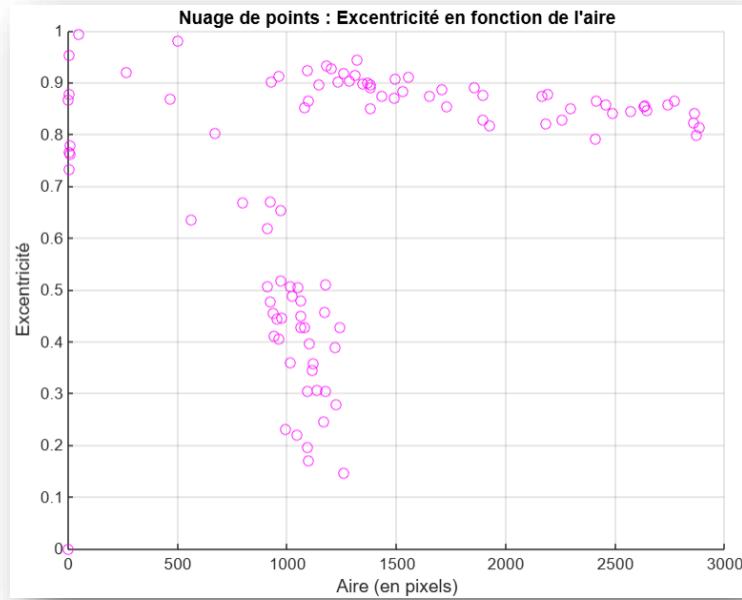
Pour la cohérence de classification, est-ce que les masques sont-ils exclusifs (chaque graine appartient à une seule catégorie) ?

L'efficacité :

En choisissant des tailles d'éléments structurants appropriées, cette méthode permet de classer efficacement les graines en trois catégories, (malgré le fait qu'il puisse y avoir un peu de répétition des graines dans la même catégorie ainsi que quelques erreurs). Cependant, elle reste généralement efficace. Pour améliorer cette classification, plusieurs pistes peuvent être envisagées :

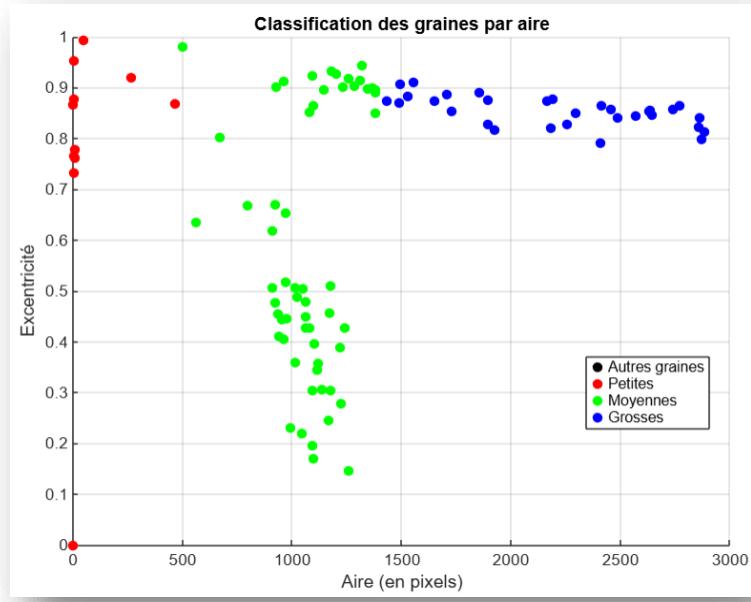
- Ajuster les tailles des éléments structurants (disques) pour affiner la séparation des catégories.
- Explorer d'autres mesures morphologiques, telles que l'excentricité ou le périmètre des graines, afin d'enrichir la caractérisation des différentes tailles de graines.

2.3 Pour aller plus loin



Observation :

- Répartition des graines par catégorie :
 - Les graines de petite taille ont de faibles valeurs d'aire et une excentricité variable.
 - Les graines de taille moyenne ont une aire intermédiaire, avec une excentricité qui varie mais reste dans un certain intervalle.
 - Les graines de grande taille ont une aire élevée et une excentricité potentiellement plus stable (par exemple, proche de 1 si elles sont allongées).
- Pour compter le nombre de graines de chaque type (petites, moyennes, grosses) à partir du graphe, on peut définir des seuils sur les aires et/ou les excentricités pour différencier les catégories. On a choisi manuellement les intervalles correspondant à chaque type de graine en fonction des observations dans le graphe précédent :
 - $[0, 500]$: Petites graines.
 - $[500, 1400]$: Moyennes graines.
 - $[1400, \text{Inf}]$: Grosses graines.



Annexes

1 Morphologie mathématique

Erosion et dilatation

- *Opérations d'érosion et de dilatation*

```
img = double(imread('miniMorpho.png'));
H = strel([1 1 1]);
V = strel([1; 1; 1]);
B1 = strel([1 1 1; 1 1 1; 1 1 1]);
C1 = strel([0 1 0; 1 1 1; 0 1 0]);
Erosion_H = imerode(img, H);
Erosion_V = imerode(img, V);
Erosion_B1 = imerode(img, B1);
Erosion_C1 = imerode(img, C1);
Dilatation_H = imdilate(img, H);
Dilatation_V = imdilate(img, V);
Dilatation_B1 = imdilate(img, B1);
Dilatation_C1 = imdilate(img, C1);
figure;
subplot(3,3,1); imshow(Erosion_H); title('Érosion avec H');
subplot(3,3,2); imshow(Dilatation_H); title('Dilatation avec H');
subplot(3,3,3); imshow(Erosion_V); title('Érosion avec V');
subplot(3,3,4); imshow(Dilatation_V); title('Dilatation avec V');
subplot(3,3,5); imshow(Erosion_B1); title('Érosion avec B1');
subplot(3,3,6); imshow(Dilatation_B1); title('Dilatation avec B1');
subplot(3,3,7.5); imshow(Erosion_C1); title('Érosion avec C1');
subplot(3,3,8.5); imshow(Dilatation_C1); title('Dilatation avec C1');
```

- *Differences entre l'image dilatée/ érodée et l'image originale*

```
img = double(imread('miniMorpho.png'));
H = strel([1 1 1]);
V = strel([1; 1; 1]);
B1 = strel([1 1 1; 1 1 1; 1 1 1]);
C1 = strel([0 1 0; 1 1 1; 0 1 0]);
Erosion_H = imerode(img, H);
Erosion_V = imerode(img, V);
Erosion_B1 = imerode(img, B1);
Erosion_C1 = imerode(img, C1);
Dilatation_H = imdilate(img, H);
Dilatation_V = imdilate(img, V);
Dilatation_B1 = imdilate(img, B1);
Dilatation_C1 = imdilate(img, C1);
% Calculer les différences
diff_Dilatation_H = Dilatation_H - img;
diff_Erosion_H = img - Erosion_H;
diff_Dilatation_V = Dilatation_V - img;
diff_Erosion_V = img - Erosion_V;
diff_Dilatation_B1 = Dilatation_B1 - img;
diff_Erosion_B1 = img - Erosion_B1;
diff_Dilatation_C1 = Dilatation_C1 - img;
diff_Erosion_C1 = img - Erosion_C1;
% Afficher les résultats
figure;
subplot(4, 3, 1), imshow(diff_Dilatation_H), title('Différence (H - dilatée)');
subplot(4, 3, 2), imshow(img), title('Image originale');
subplot(4, 3, 3), imshow(diff_Erosion_H), title('Différence (H - érodée)');
```

```

    subplot(4, 3, 4), imshow(diff_Dilatation_V), title('Différence (V -
dilatée)');
    subplot(4, 3, 5), imshow(img), title('Image originale');
    subplot(4, 3, 6), imshow(diff_Erosion_V), title('Différence (V - érodée)');
    subplot(4, 3, 7), imshow(diff_Dilatation_B1), title('Différence (B1 -
dilatée)');
    subplot(4, 3, 8), imshow(img), title('Image originale');
    subplot(4, 3, 9), imshow(diff_Erosion_B1), title('Différence (B1 - érodée)');
    subplot(4, 3, 10), imshow(diff_Dilatation_C1), title('Différence (C1 -
dilatée)');
    subplot(4, 3, 11), imshow(img), title('Image originale');
    subplot(4, 3, 12), imshow(diff_Erosion_C1), title('Différence (C1 -
érodée)');

```

- L'opération d'érosion avec le filtre H puis avec le filtre V

```

img = double(imread('miniMorpho.png'));
H = strel([1 1 1]);
V = strel([1; 1; 1]);
B1 = strel([1 1 1; 1 1 1; 1 1 1]);
C1 = strel([0 1 0; 1 1 1; 0 1 0]);
Erosion_H = imerode(img, H);
Erosion_V = imerode(img, V);
Erosion_B1 = imerode(img, B1);
Erosion_H_V = imerode(Erosion_H, V);
Dilatation_H = imdilate(img, H);
Dilatation_V = imdilate(img, V);
Dilatation_B1 = imdilate(img, B1);
Dilatation_H_V = imdilate(Dilatation_H, V);
subplot(8,2,1), imshow(img), title('Image originale');
subplot(8,2,2), imshow(Erosion_H), title('Erosion avec H');
subplot(8,2,3), imshow(Erosion_H_V), title('Erosion avec H puis V');
subplot(8,2,4), imshow(Erosion_B1), title('Erosion avec B1');
subplot(8,2,5), imshow(img), title('Image originale');
subplot(8,2,6), imshow(Dilatation_H), title('Dilatation avec H');
subplot(8,2,7), imshow(Dilatation_H_V), title('Dilatation avec H puis V');
subplot(8,2,8), imshow(Dilatation_B1), title('Erosion avec B1');

```

- Vérification qu'en effectuant une érosion avec Bn (resp. Cn), on obtient les mêmes résultats qu'en effectuant une érosion n fois par B1 (resp. C1).

```

img = imread('miniMorpho.png');
img = double(img);
% Définir les éléments structurants
B1 = [1 1 1; 1 1 1; 1 1 1];
C1 = [0 1 0; 1 1 1; 0 1 0];
n = 3; % Nombre de convolutions
Bn = B1;
Cn = C1;
% Convolution pour obtenir Bn et Cn
for i = 1:n-1
    Bn = conv2(Bn, B1);
    Cn = conv2(Cn, C1);
end
% Conversion des éléments structurants en valeurs binaires
Bn = Bn > 0;
Cn = Cn > 0;
% Érosion avec Bn et Cn
Erosion_Bn = imerode(img, strel('arbitrary', Bn));

```

```

Erosion_Cn = imerode(img, strel('arbitrary', Cn));
% Érosion répétée n fois par B1 et C1
Erosion_B1 = img;
Erosion_C1 = img;
for i = 1:n
    Erosion_B1 = imerode(Erosion_B1, strel('arbitrary', B1));
    Erosion_C1 = imerode(Erosion_C1, strel('arbitrary', C1));
end
% Afficher les résultats
figure;
subplot(2,2,1), imshow(img, []), title('Image originale');
subplot(2,2,3), imshow(Erosion_Bn, []), title('Érosion avec Bn');
subplot(2,2,4), imshow(Erosion_B1, []), title('Érosion répétée avec B1');
figure;
subplot(2,2,1), imshow(img, []), title('Image originale');
subplot(2,2,3), imshow(Erosion_Cn, []), title('Érosion avec Cn');
subplot(2,2,4), imshow(Erosion_C1, []), title('Érosion répétée avec C1');

```

2 Ouverture et Fermeture

- Application de plusieurs opérations d'ouverture et de fermeture d'affilée avec le même élément structurant C1 :

```

img = imread('miniMorpho.png');
img = double(img);
C1 = [0 1 0; 1 1 1; 0 1 0];
ouverture_img = img;
fermeture_img = img;
num_iterations = 1;
for i = 1:num_iterations
    ouverture_img = imopen(ouverture_img, C1);
end
for i = 1:num_iterations
    fermeture_img = imclose(fermeture_img, C1);
end
% Afficher le résultat de l'ouverture et de la fermeture multiple
figure;
subplot(1, 2, 1), imshow(img), title('Image originale');
subplot(1, 2, 2), imshow(ouverture_img), title('Ouverture multiple avec C1');
figure;
subplot(1, 2, 1), imshow(img), title('Image originale');
subplot(1, 2, 2), imshow(fermeture_img), title('Fermeture multiple avec C1');

```

- Application des opérations d'ouverture et de fermeture avec des éléments structurants Cn de taille croissante :

```

img = imread('miniMorpho.png');
img = double(img); % Convertir l'image en binaire si elle ne l'est pas
% Définir les tailles croissantes des éléments structurants
sizes = [3, 5, 7, 9];
% Boucle sur les tailles des éléments structurants
figure;
for i = 1:length(sizes)
    se = strel('square', sizes(i)); % Créer un élément structurant carré de
    % taille croissante
    % Appliquer l'ouverture
    ouverture_img = imopen(img, se);
    % Appliquer la fermeture

```

```

fermeture_img = imclose(img, se);
% Afficher les résultats
subplot(2, length(sizes), i);
imshow(ouverture_img);
title(['Ouverture - Taille ' num2str(sizes(i))]);
subplot(2, length(sizes), i + length(sizes));
imshow(fermeture_img);
title(['Fermeture - Taille ' num2str(sizes(i))]);
end

```

En niveaux de gris

```

img = imread('cameraman.tif');
% élément structurant en forme de disque de taille 5
se = strel('disk', 5);
%l'érosion
img_eroded = imerode(img, se);
% la dilatation
img_dilated = imdilate(img, se);
%l'ouverture
img_opened = imopen(img, se);
%la fermeture
img_closed = imclose(img, se);
% Afficher les résultats
figure;
subplot(2,3,1), imshow(img), title('Image originale');
subplot(2,3,2), imshow(img_eroded), title('Erosion');
subplot(2,3,3), imshow(img_dilated), title('Dilatation');
subplot(2,3,4), imshow(img_opened), title('Ouverture');
subplot(2,3,5), imshow(img_closed), title('Fermeture');
% Inverser l'image en niveaux de gris
img_inverted = imcomplement(img);
% Appliquer les mêmes opérations sur l'image inversée
img_inv_eroded = imerode(img_inverted, se);
img_inv_dilated = imdilate(img_inverted, se);
img_inv_opened = imopen(img_inverted, se);
img_inv_closed = imclose(img_inverted, se);
% Afficher les résultats pour l'image inversée
figure;
subplot(2,3,1), imshow(img_inverted), title('Image inversée');
subplot(2,3,2), imshow(img_inv_eroded), title('Erosion (inversée)');
subplot(2,3,3), imshow(img_inv_dilated), title('Dilatation (inversée)');
subplot(2,3,4), imshow(img_inv_opened), title('Ouverture (inversée)');
subplot(2,3,5), imshow(img_inv_closed), title('Fermeture (inversée)');

```

2 Classification de graines

- *Normalisation et Binarisation :*

```

% Charger et afficher l'image
img = imread('Seeds.png');
subplot(2,2,1.5)

```

```

imshow(img);
title('Image des graines');
Img_normalisee = mat2gray(img);
Img_binarisee = imbinarize(img);
subplot(2,2,3)
imshow(Img_normalisee);
title('Image normalisée');
subplot(2,2,4)
imshow(Img_binarisee);
title('Image binarisée');

```

- *Lissage l'image en réalisant une ouverture avec un disque de petite taille.*

```

% Charger et afficher l'image
img = imread('Seeds.png');
subplot(1,3,1)
imshow(img);
title('Image des graines');
Img_binarisee = imbinarize(img);
% Créer un élément structurant en forme de disque
radius = 3; % Taille du disque (à ajuster selon les besoins)
se = strel('disk', radius);
% Appliquer l'ouverture
smoothedImg = imopen(Img_binarisee, se);
% Afficher l'image lissée
subplot(1,3,2)
imshow(Img_binarisee);
title('Image binarisée');
subplot(1,3,3)
imshow(smoothedImg);
title(['l''Image après', newline, 'ouverture (lissage)']);

```

- *La reconstruction entre le bord de l'image et l'image binariser*

```

img = imread('Seeds.png');
Img_binarisee = double(imbinarize(img));
% Créer une bordure de 1
bordure_Mask = zeros(size(Img_binarisee));      % Image de base
bordure_Mask(1, :) = 1; bordure_Mask(end, :) = 1;
bordure_Mask(:, 1) = 1; bordure_Mask(:, end) = 1;
% Reconstruction morphologique à partir de la bordure
Img_reconstruction = imreconstruct(bordure_Mask, Img_binarisee);
% Supprimer les graines touchant les bords
Img_resultat = Img_binarisee & ~ Img_reconstruction;
%affichage
subplot(2,2,1)
imshow(Img_binarisee);
title('Image des graines binarisée');
subplot(2,2,2)
imshow(Img_reconstruction);
title(['Image avec les graines ', newline, 'qui touchent les bords']);
subplot(2,2,3.)
imshow(Img_resultat);
title(['Image après suppression des', newline, 'graines touchant les bords']);

```

2.2 Prétraitement

- *Granulométrie (Traçage)*
- *Tailles d'éléments structurant permettant d'obtenir uniquement les "grosses" graines puis les "grosses et moyennes" graines*

```
img = imread('Seeds.png');
Img_binarisee = double(imbinarize(img));
bordure_Mask = zeros(size(Img_binarisee));
bordure_Mask(1, :) = 1; bordure_Mask(end, :) = 1;
bordure_Mask(:, 1) = 1; bordure_Mask(:, end) = 1;
Img_reconstruction = imreconstruct(bordure_Mask, Img_binarisee);
Img_resultat = Img_binarisee & ~ Img_reconstruction;
% Initialisation des paramètres
disque = 20; % Taille maximale du disque
graines_restantes = zeros(1, disque); % Stocker le nombre de graines restantes
% Boucle sur les tailles de disque
for rad = 1:disque
    % Créer un élément structurant en forme de disque
    se = strel('disk', rad);
    % Appliquer l'ouverture
    ouverture_Img = imopen(Img_resultat, se);
    % Obtenir le tableau de propriétés des graines
    Tab = PropSeed(ouverture_Img);
    % Compter le nombre de graines restantes
    graines_restantes(rad) = size(Tab, 1); % Nombre de lignes dans Tab = nombre de graines
end
% Tracer le résultat
figure;
plot(1:disque, graines_restantes, '-o', 'LineWidth', 1.5);
xlabel('Taille de l'élément structurant (rayon du disque)');
ylabel('Nombre de graines restantes');
title('Granulométrie des graines');
grid on;
% Taille pour grosses graines
grosses = strel('disk', 15);
Img_grosses = imopen(Img_resultat, grosses);
% Taille pour grosses et moyennes graines
grosses_moyennes = strel('disk', 10);
Img_grosses_moyennes = imopen(Img_resultat, grosses_moyennes);
% Afficher les résultats
figure;
subplot(1, 2, 1);
imshow(Img_grosses);
title('Grosses graines');
subplot(1, 2, 2);
imshow(Img_grosses_moyennes);
title('Grosses et moyennes graines');
```

- *Tailles Application des trois différents masques à l'image originale* (plutôt à l'image binaire ou l'image après suppression des graines touchant les bords) pour obtenir les images de chaque type de graines*

```
img = imread('Seeds.png');
Img_binarisee = double(imbinarize(img));
% Suppression des graines sur les bords
```

```

bordure_Mask = zeros(size(Img_binarisee));
bordure_Mask(1, :) = 1; bordure_Mask(end, :) = 1;
bordure_Mask(:, 1) = 1; bordure_Mask(:, end) = 1;
Img_reconstruction = imreconstruct(bordure_Mask, Img_binarisee);
Img_resultat = Img_binarisee & ~Img_reconstruction;
% Taille des éléments structurants
r_moyennes = 10;
r_grosses = 15;
% Éléments structurants
se_moyennes = strel('disk', r_moyennes);
se_grosses = strel('disk', r_grosses);
% Masques pour chaque catégorie
mask_grosses = double(imopen(Img_resultat, se_grosses)); % Masque grosses graines
mask_moyennes_grosses = double(imopen(Img_resultat, se_moyennes)); % Masque moyennes + grosses
mask_moyennes = double(mask_moyennes_grosses & ~mask_grosses); % Masque moyennes seules
mask_petites = double(Img_resultat & ~mask_moyennes_grosses); % Masque petites graines
% Extraire les graines
graines_grosses = mask_grosses .* Img_resultat; % Graines grosses
graines_moyennes = mask_moyennes .* Img_resultat; % Graines moyennes
graines_petites = mask_petites.*Img_resultat; % Graines petites
% Affichage des résultats
figure;
subplot(2, 2, 1);
imshow(Img_resultat);
title(['Image après suppression des', newline, 'graines touchant les bords']);
subplot(2, 2, 2);
imshow(graines_grosses);
title('Grosses graines');
subplot(2, 2, 3);
imshow(graines_moyennes);
title('Moyennes graines');
subplot(2, 2, 4);
imshow(graines_petites);
title('Petites graines');

```

2.3 Pour aller plus loin

- Tracer le nuage de point de l'excentricité en fonction de l'aire

```

img = imread('Seeds.png');
Img_binarisee = imbinarize(img);
% Extraction des propriétés des graines
Tab = PropSeed(Img_binarisee); % Tab contient l'aire (colonne 1) et l'excentricité (colonne 2)
% Extraction de l'aire et de l'excentricité
aire = Tab(:, 1); % Aire des graines (colonne 1)
excentricite = Tab(:, 2); % Excentricité des graines (colonne 2)
% Tracer le nuage de points (excentricité vs aire)
figure;
scatter(aire, excentricite, 'magenta');
xlabel('Aire (en pixels)');
ylabel('Excentricité');
title('Nuage de points : Excentricité en fonction de l''aire');
grid on;

```

- Comptage du nombre de graines de chaque type (petites, moyennes, grosses) à partir du graphe

```

img = imread('Seeds.png');
Img_binarisee = imbinarize(img);
% Extraction des propriétés des graines
Tab = PropSeed(Img_binarisee);
aire = Tab(:, 1); % Aire des graines (colonne 1)
excentricite = Tab(:, 2); % Excentricité des graines (colonne 2)
% Définir les seuils pour les types de graines
seuils_aire = [0, 500, 1400, Inf]; % Modifiez les seuils en fonction des
observations
% Comptage des graines de chaque type
nb_petites = sum(aire > seuils_aire(1) & aire <= seuils_aire(2)); % Petites
graines
nb_moyennes = sum(aire > seuils_aire(2) & aire <= seuils_aire(3)); % Moyennes
graines
nb_grosses = sum(aire > seuils_aire(3) & aire <= seuils_aire(4)); % Grosses
graines
% Afficher les résultats
fprintf('Nombre de petites graines : %d\n', nb_petites);
fprintf('Nombre de moyennes graines : %d\n', nb_moyennes);
fprintf('Nombre de grosses graines : %d\n', nb_grosses);
% Tracer le nuage de points (aire vs excentricité) avec les types de graines
figure;
scatter(aire, excentricite, 'k', 'filled'); % Points en noir
hold on;
scatter(aire(aire > seuils_aire(1) & aire <= seuils_aire(2)), ...
    excentricite(aire > seuils_aire(1) & aire <= seuils_aire(2)), 'r',
'filled'); % Petites graines
scatter(aire(aire > seuils_aire(2) & aire <= seuils_aire(3)), ...
    excentricite(aire > seuils_aire(2) & aire <= seuils_aire(3)), 'g',
'filled'); % Moyennes graines
scatter(aire(aire > seuils_aire(3) & aire <= seuils_aire(4)), ...
    excentricite(aire > seuils_aire(3) & aire <= seuils_aire(4)), 'b',
'filled'); % Grosses graines
xlabel('Aire (en pixels)');
ylabel('Excentricité');
title('Classification des graines par aire');
legend({'Autres graines', 'Petites', 'Moyennes', 'Grosses'}, 'Location',
'best');
grid on;
hold off;

```