

# Travaux pratiques n°3: Traitement des Images

Etudiants: Dogan Ozgur et Melissaratos Konstantinos

Encadrant: M. Girard Félix

## 1 Morphologie mathématique

### 1.1 Érosion & dilatation

#### 1.1.1 Opérations simples avec les éléments structurants

Les opérations morphologiques sont effectuées sur l'image *miniMorpho.png*, présente ci-dessous:

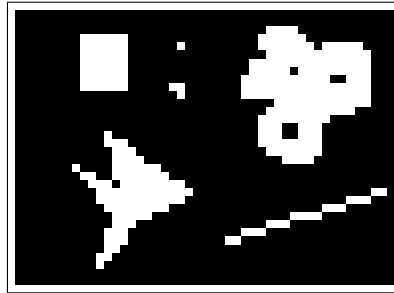


Figure 1: Image *miniMorpho.png*

Nous avons à disposition les éléments structurants suivants:

$$H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$V = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Les images après avoir effectuer les opération d'érosion et de dilatation sont les images qui suivent.

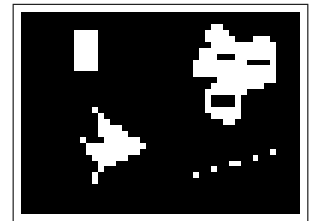
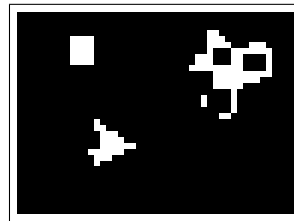
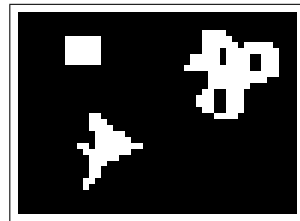
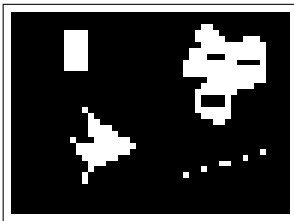


Figure 2: Image érodée avec  $H$

Figure 3: Image érodée avec  $V$

Figure 4: Image érodée avec  $B_1$

Figure 5: Image érodée avec  $C_1$

On peut voir qu'en effectuant une érosion avec différents éléments structurant, il est possible de supprimer des parties appartenant aux objet constituant l'image, et même de les supprimer complètement en fonction de la taille et de la disposition des valeurs non nulles dans l'élément structurant.

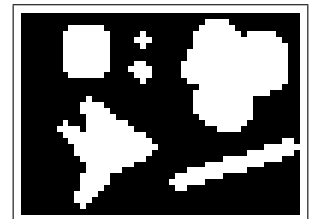
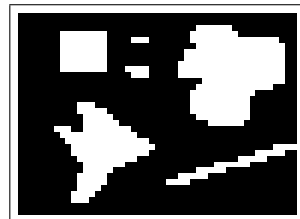
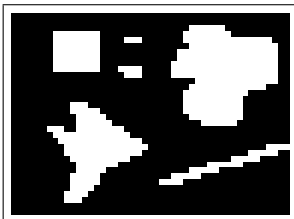


Figure 6: Image dilatée avec  $H$

Figure 7: Image dilatée avec  $V$

Figure 8: Image dilatée avec  $B_1$

Figure 9: Image dilatée avec  $C_1$

On peut voir qu'avec la dilatation, les trous qui étaient présents dans les objets de l'image ont été supprimés. De plus ces objets sont devenus plus larges, se sont étalés sur l'image.

### 1.1.2 différence

Les images obtenues après avoir effectué la différence image dilatée - image originale sont les suivantes:

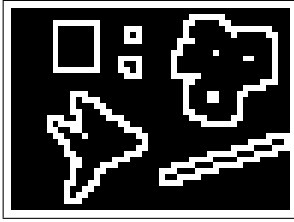


Figure 10: Image dilatée avec  $H$  - image d'origine

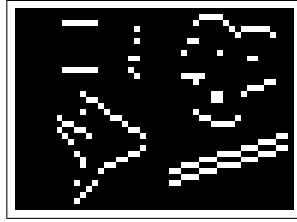


Figure 11: Image dilatée avec  $V$  - image d'origine

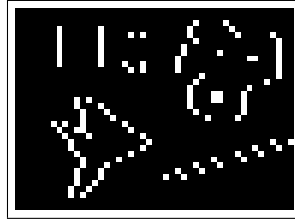


Figure 12: Image dilatée avec  $B_1$  - image d'origine

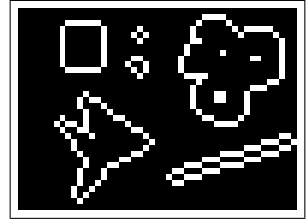


Figure 13: Image dilatée avec  $C_1$  - image d'origine

On observe que la différence de l'image dilatée par l'image d'origine nous donne les contours des objets de l'image dilatée. Les différences entre ces contours sont dues aux natures différentes des éléments structuraux utilisés lors de la dilatation.

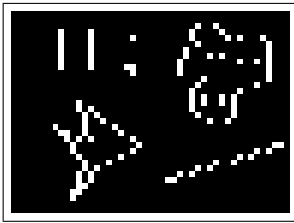


Figure 14: Image d'origine - image érodée avec  $H$

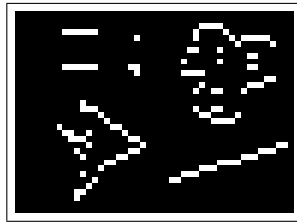


Figure 15: Image d'origine - image érodée avec  $V$

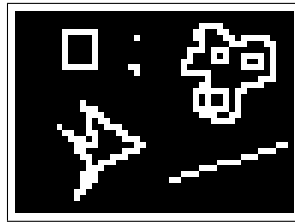


Figure 16: Image d'origine - image érodée avec  $B_1$

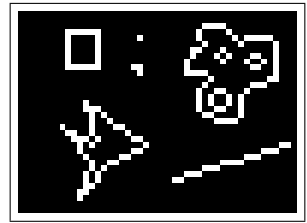


Figure 17: Image d'origine - image érodée avec  $C_1$

De même, nous obtenons les contours des objets de l'image, ainsi que les contours de trous les composant. Les différences sont dues ici aussi aux différences des éléments structuraux.

### 1.1.3 Opérations successives

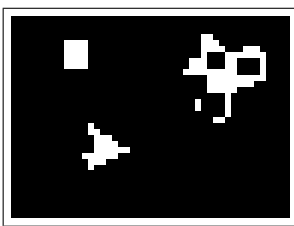


Figure 18: Image érodée avec le  $H$  puis avec  $V$

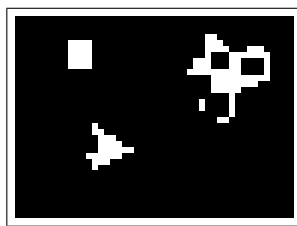


Figure 19: Image érodée avec  $B_1$



Figure 20: Image dilatée avec  $H$  puis avec  $V$

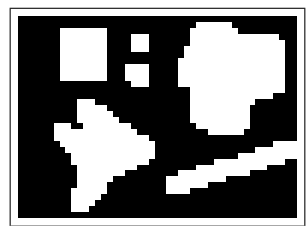


Figure 21: Image dilatée avec  $B_1$

Nous pouvons constater que l'érosion avec  $H$  puis avec  $V$  est l'érosion avec  $B_1$  sont les mêmes opérations. Cela semble logique étant donné qu'avec  $H$  nous faisons une érosion horizontale et avec  $V$  une érosion verticale alors qu'avec  $B_1$  prend en compte les voisins se situant verticalement et horizontalement par rapport au pixel en question. Nous pouvons faire le même constat pour la dilatation.

#### 1.1.4 Convolution des éléments structurants

Nous avons créé un nouvel éléments structurant  $B_2 = B_1 * B_1$  et nous avons fait de même pour obtenir  $C_2$ . Nous procédons par la suite à des érosions pour obtenir les résultats suivants.

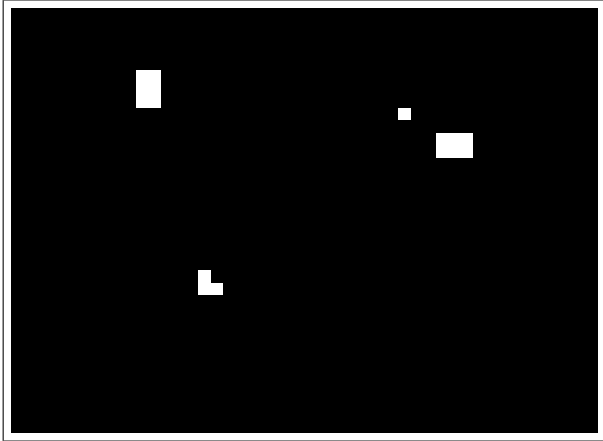


Figure 22: Image érodée deux fois successives avec  $B_1$

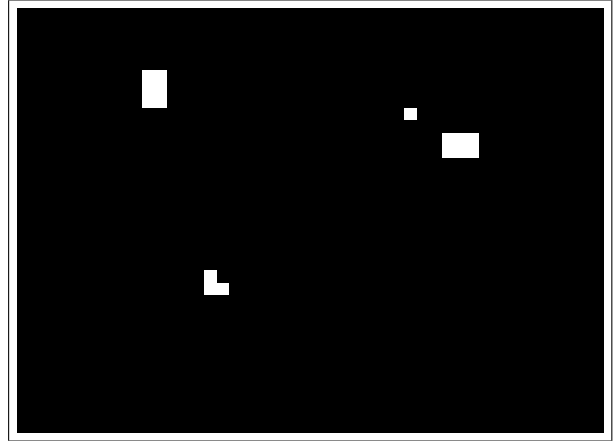


Figure 23: Image érodée avec  $B_2$

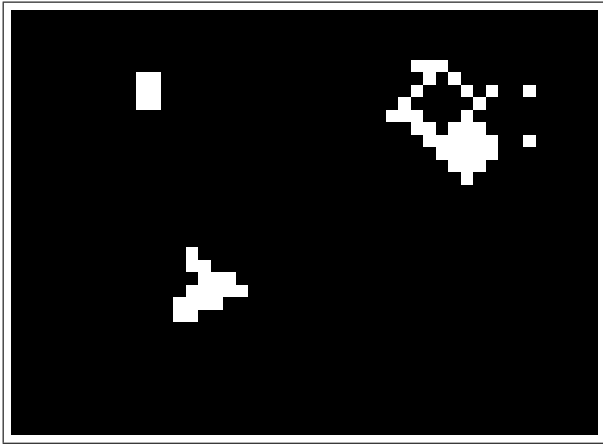


Figure 24: Image érodée deux fois successives avec  $C_1$

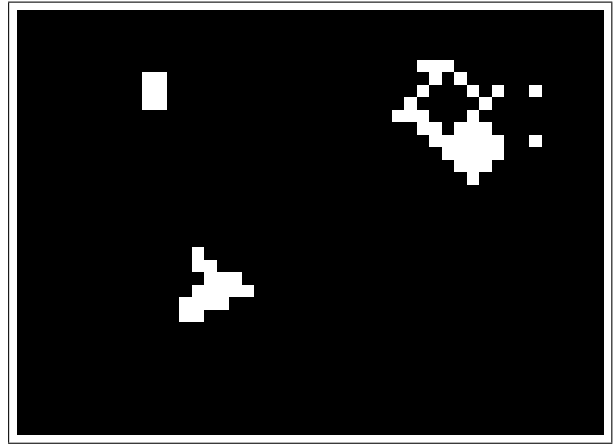


Figure 25: Image érodée avec  $C_2$

Nous pouvons constater que les résultats des figures 21 et 22 sont identiques. Nous pouvons faire la même observation pour les figures 23 et 24.

Au lieu de répéter une érosion ou une dilatation plusieurs fois avec un même élément structurant, on peut faire la convolution de cet élément structurant avec lui-même pour en obtenir un nouveau qui correspond. Ainsi on peut diminuer le nombre d'opération à effectuer en parcourant une seule fois notre image, et donc diminuer la complexité en temps de l'algorithme.

## 1.2 Ouverture & fermeture

L'ouverture est l'opération qui consiste à faire une érosion suivie d'une dilatation. Le but de cette opération est de supprimer des petits objets qui sont négligeables par rapport à l'objet principal de l'image. Elle peut aussi être utilisée pour séparer des objets connectés entre eux.

La fermeture est une dilatation suivie d'une érosion. Son but est de boucher les trous présents dans les objets constituant l'image.

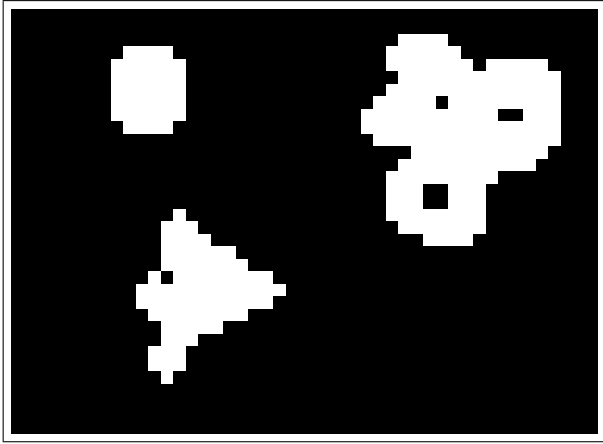


Figure 26: Image obtenue après 1, 2 et 3 ouvertures successives avec  $C_1$

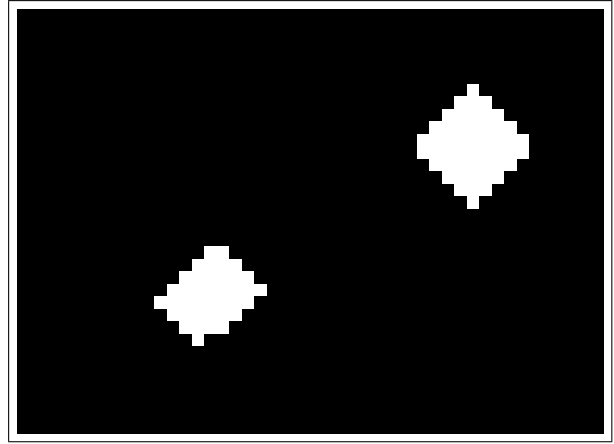


Figure 27: Image obtenue après une seule ouverture avec  $C_3$

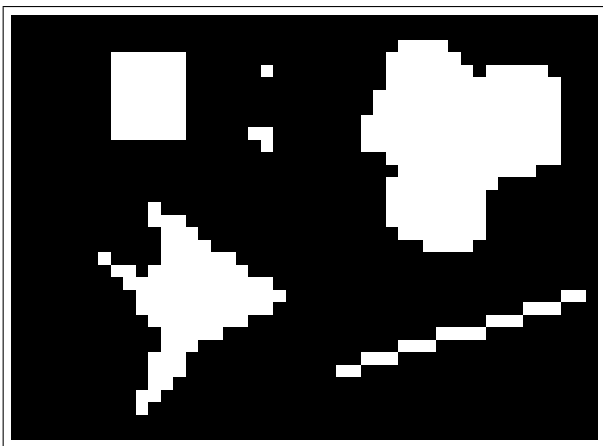


Figure 28: Image obtenue après 1, 2 et 3 fermetures successives avec  $C_1$

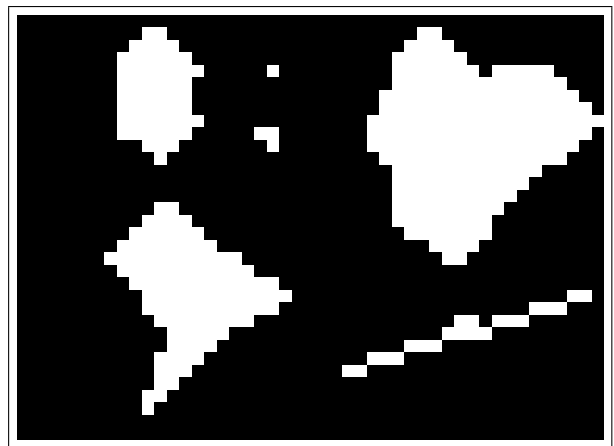


Figure 29: Image obtenue après une seule fermeture avec  $C_3$

Nous pouvons voir que les résultats de l'ouverture et de la fermeture sont différents entre eux. On peut alors dire que l'ordre des opérations d'érosion et de dilatation sont importants. Nous pouvons voir aussi que les ouvertures ou fermetures successives donnent le même résultat. Cependant nous pouvons voir que l'ouverture avec l'élément structurant convolué avec lui-même ne donne pas le même résultat que les opérations successives avec ce dernier, ce qui n'est pas le cas avec l'érosion et la dilatation comme vu précédemment.

### 1.3 En niveaux de gris

Pour les opération sur une image en niveaux de gris nous utilisons le image *cameraman.pgm* et sont image inversée:



Figure 30: Image *cameraman.pgm*



Figure 31: Image *cameraman.pgm* inversée

Les opération sont effectuée avec un disque de rayon 5 obtenu grâce à la fonction MatLab `strel` dont les paramètres utilisés sont la forme de l'élément structurant et la taille, le rayon dans notre cas. Ainsi les résultats obtenus sont les suivants:



Figure 32: Image après érosion avec le disque



Figure 33: Image après dilatation avec le disque



Figure 34: Image après ouverture avec le disque



Figure 35: Image après fermeture avec le disque

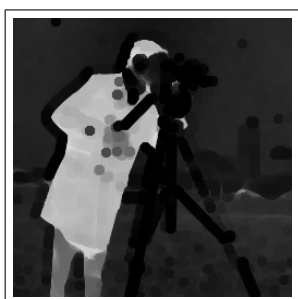


Figure 36: Image après érosion avec le disque



Figure 37: Image après dilatation avec le disque



Figure 38: Image après ouverture avec le disque

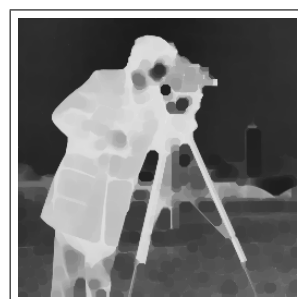


Figure 39: Image après fermeture avec le disque

Nous pouvons voir que pour chacune des opérations, l'image obtenue est plus simplifiée, avec moins de détails, faisant un effet *cartoon*. Nous pouvons aussi voir que le résultat entre l'image d'origine et l'inverse donne, pour

chaque opération, l'image inverse du résultat.

## 2 Classification de graines

### 2.1 Pré-traitement de l'image

Dans cette partie nous avons l'image ci-dessous dont différents types de graines sont présentes. Le but est de classer ces graines en fonction de leurs caractéristiques morphologiques différentes.

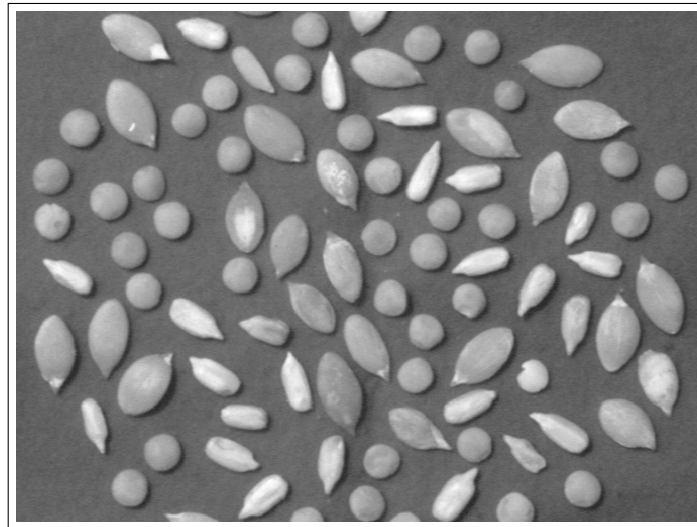


Figure 40: Image *Seeds.png*

Sur cette image nous pouvons repérer trois différentes caractéristiques morphologiques: la couleur, la forme et la taille des graines. Cependant, les classer automatiquement n'est pas simple.

Nous avons tout d'abord normalisé l'image, c'est-à-dire la transformer tel que sa dynamique soit égale à 255, le nombre de niveaux de gris différents, afin de la binariser par la suite. Le résultats sont les suivants:

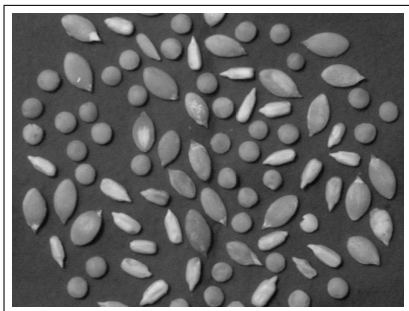


Figure 41: Image *Seeds.png* normalisée

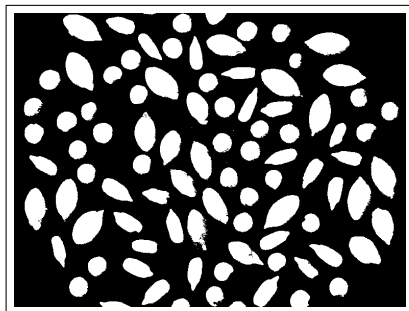


Figure 42: Image *Seeds.pgm* binarisée

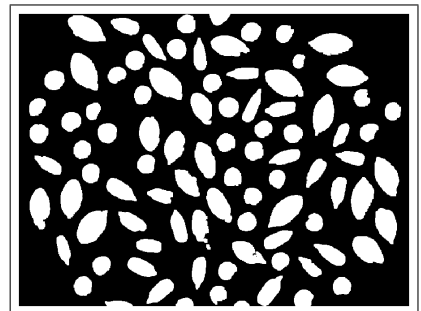


Figure 43: Image binarisée après ouverture

Nous pouvons voir que l'image binariée a certains défauts. Ce derniers sont supprimées grâce à une ouverture avec un disque de taille 3, dont le résultat est sur la figure 43.

Nous pouvons voir aussi que certaines graines sur les bords de l'image sont coupées. À l'oeil il est simple de les distinguer, mais pour une machine la tâche est plus compliquée. C'est pourquoi nous procédons à un *border kill* pour les supprimer. L'image finale que nous allons utiliser pour la classification est l'image de la figure 45:

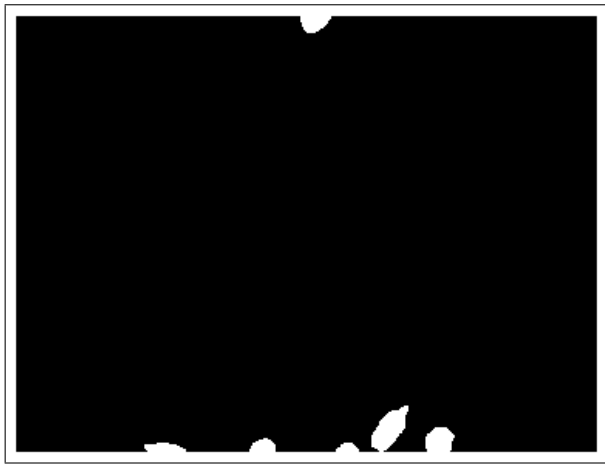


Figure 44: Image des graines touchant la bordure

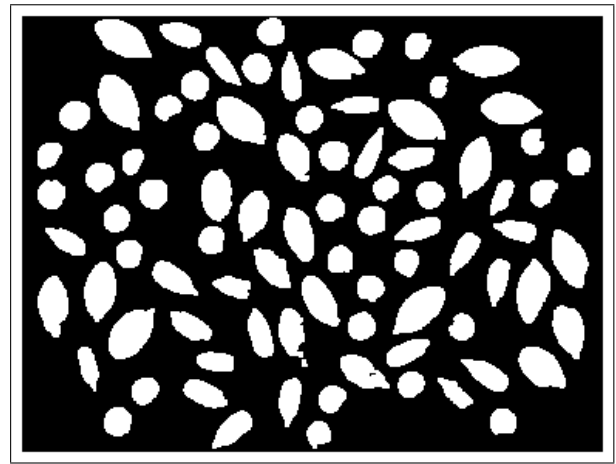


Figure 45: Image sans les graines coupées

## 2.2 Granulométrie

À l'aide de la fonction `PropSeed` nous pouvons avoir l'aire et l'excentricité des graines sur une image. Ainsi en faisant des ouvertures successives les graines disparaîtront au fur et à mesure, commençant par les petites graines. Voici le graphe montrant le nombre de graines en fonction du nombre d'ouvertures:

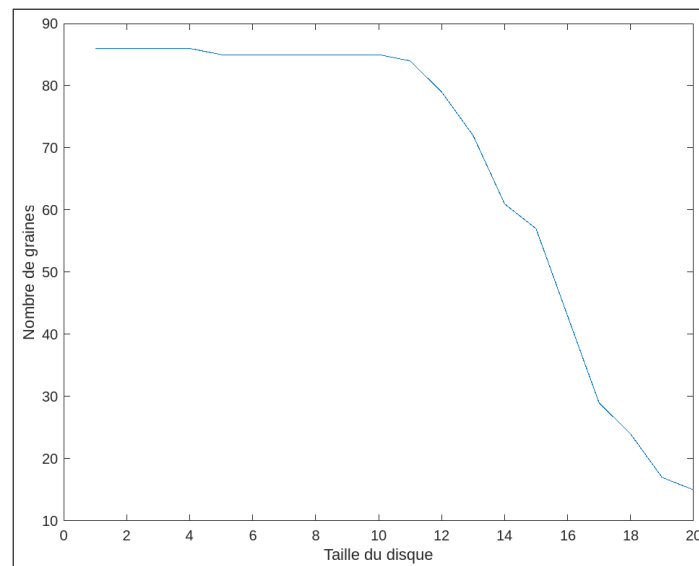


Figure 46: Graphe montrant le nombre de graines en fonction du nombre d'ouvertures successives

Grâce à ce graphe nous pouvons définir des seuils afin d'obtenir à l'aide de soustractions les images de petites, moyennes et grandes graines.

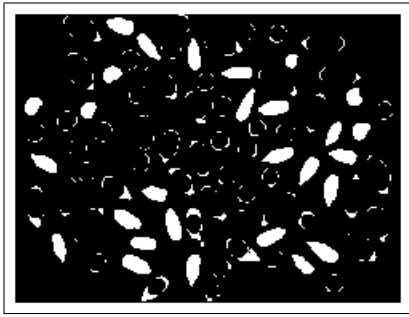


Figure 47: Image des graines de petite taille

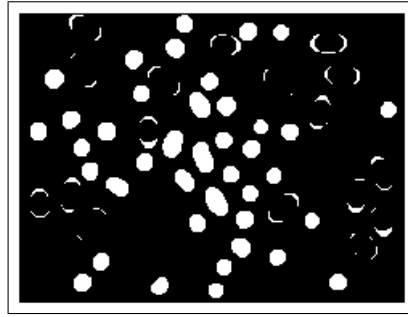


Figure 48: Image des graines de taille moyenne

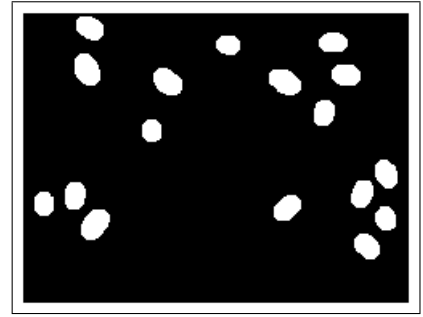


Figure 49: Image des graines de grande taille

Nous pouvons voir sur les images des petites et moyennes graines qu'il y a du bruit. Ce bruit est dû à la soustraction par des images dont les objets sont plus petits à cause des ouvertures. Pour enlever ce bruit il serait possible d'utiliser différentes méthodes, comme par exemple procéder à des érosions ou bien à de fermetures.

Cette méthode est assez efficace mais nous ne considérons qu'une seule caractéristique morphologique des graines, tandis que la fonction `PropSeed` nous en donne une autre, leur excentricité.

## 2.3 Pour aller plus loin

Grâce aux données obtenus par la fonction `PropSeed` nous pouvons tracer le nuage des points de l'excentricité des graines en fonction de leur aire.

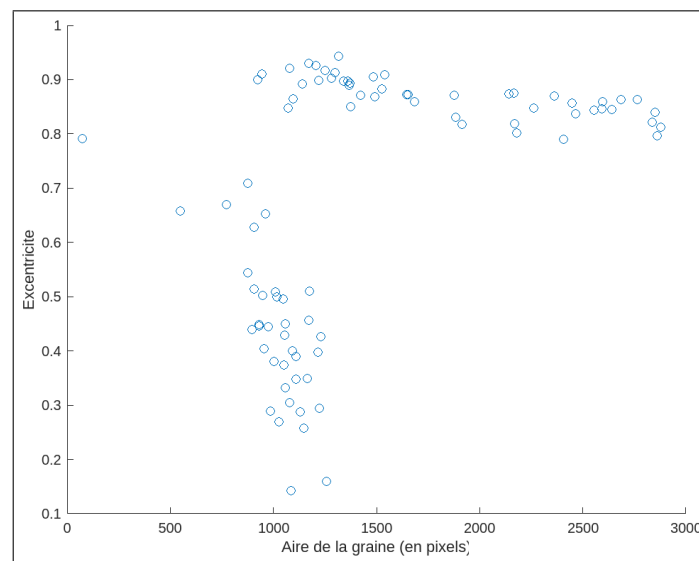


Figure 50: Nuage des points de l'excentricité en fonction de l'aire des graines

Sur le nuage des points nous pouvons distinguer trois groupes des points correspondant à nos trois types de graines. Ainsi l'algorithme pour les compter est simple, il suffit de définir les intervalles de taille et d'excentricité pour chaque classe et évaluer pour chaque graine quels intervalle lui correspondent pour la placer dans la classe correspondante.