

TP TADI - Morphologie mathématique

Marie DIEZ, Lucie JANDET

20 octobre 2021

Mathematical morphology on gray-scale images

1.1 A - Les opérations

Les exemples présentés ci-dessous ont été réalisés avec une forme d'élément structurant de disque, et une taille d'élément structurant de 1.

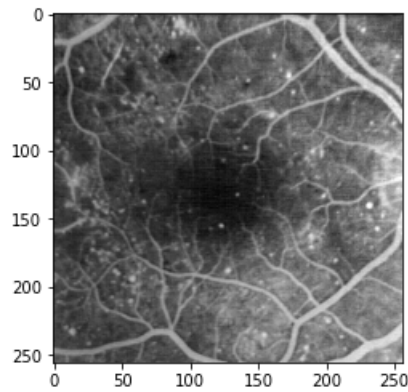


FIGURE 1 – Image initiale : rétine

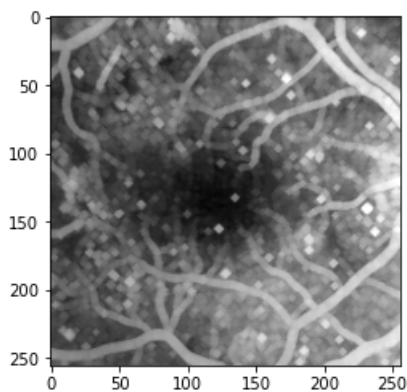


FIGURE 2 – Image dilatée

La dilatation supprime les petites zones sombres relativement à la taille de l'élément structurant, mais élargit également les principales zones claires, une partie de l'information de l'image est perdue.

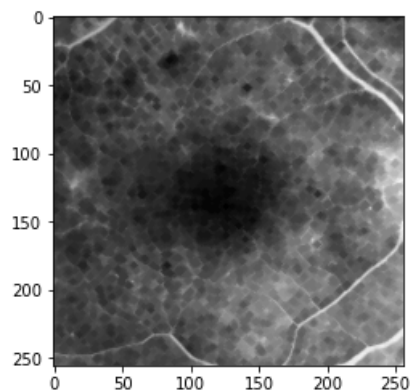


FIGURE 3 – Image érodée

L'érosion supprime les petites zones claires de l'image relativement à la taille de l'élément structurant, mais rétrécit également les formes claires plus importantes, ce qui peut engendrer une perte d'information dans l'image.

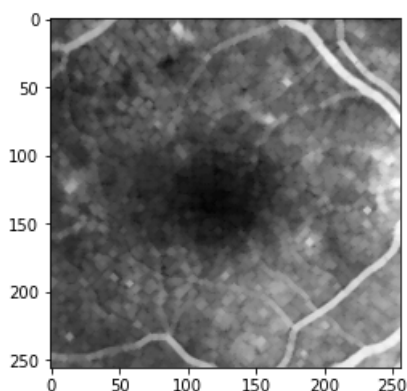


FIGURE 4 – Image ouverte

L'opening est la succession d'une érosion et d'une dilatation. L'érosion permet de supprimer de l'image les petites zones claires et agrandir les zones sombres relativement à la taille de l'élément structurant, ce qui permet de régulariser l'image. La dilatation permet de retrouver l'intensité de l'image initiale. L'ouverture permet donc de régulariser l'image en supprimant les petites zones claires (en fonction de la taille de l'élément) tout en gardant le niveau d'intensité initial.

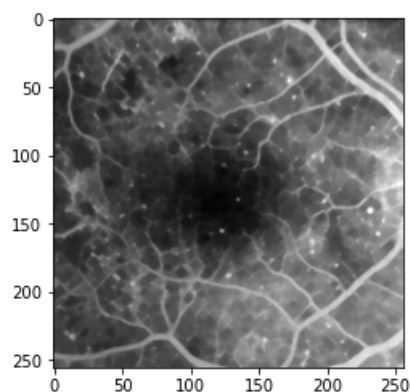


FIGURE 5 – Image fermée

La fermeture est la succession d'une dilatation puis d'une érosion. La dilatation supprime les petites zones sombres relativement à la taille de l'élément structurant et agrandit les zones claires. L'érosion ramène ensuite les zones claires à des tailles presque similaires à celles de l'image initiale, tout en ayant supprimé les petites zones sombres, ce qui permet de régulariser l'image.

B - Taille et forme de l'élément structurant

On peut voir que la taille de l'élément structurant a un rôle important sur les opérations. En effet celui-ci est l'objet référent utilisé, il représente B sur les formules suivantes, sa taille et sa forme impacte donc les résultats :

$$D(X, B) = X \oplus B = \{x + y \mid x \in X, y \in B\}$$

$$E(X, B) = \{x \in \mathbb{R}^n \mid B_x \subset X\} = \{x \in \mathbb{R}^n \mid \forall y \in B, x + y \in X\} = X \otimes \bar{B}$$

Voici les résultats pour différentes tailles et formes d'éléments structurants :

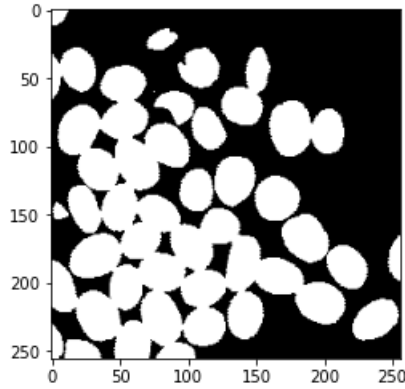


FIGURE 6 – Image initiale : Café

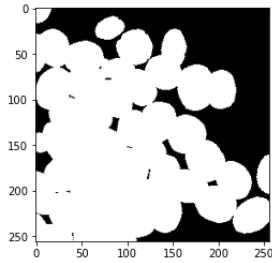


FIGURE 7 – Dilatation - élément=disque, size=5

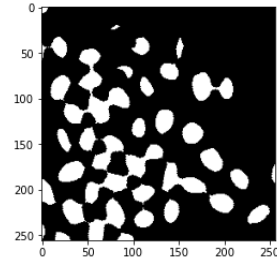


FIGURE 8 – Erosion - élément=disque, size=5

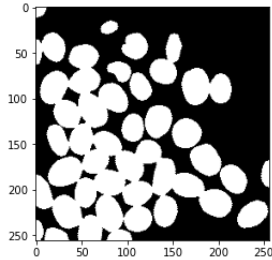


FIGURE 9 – Ouverture - élément=disque, size=5

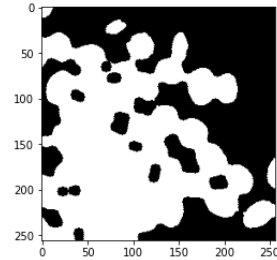


FIGURE 10 – Fermeture - élément=disque, size=5

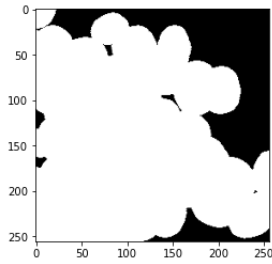


FIGURE 11 – Dilatation - élément=disque, size=10

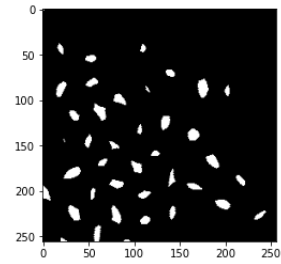


FIGURE 12 – Erosion - élément=disque, size=10

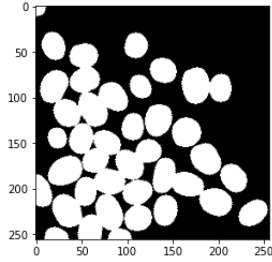


FIGURE 13 – Ouverture - élément=disque, size=10

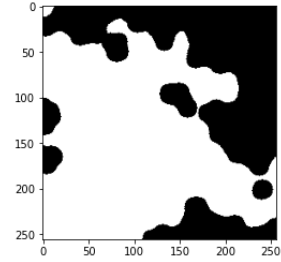


FIGURE 14 – Fermeture - élément=disque, size=10

Plus la taille de l'élément est grand plus la dilatation et respectivement l'érosion (donc ouverture et fermeture aussi) des objets sera importante.

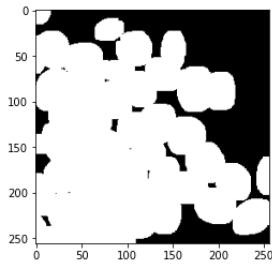


FIGURE 15 – Dilatation - élément=square, size=10

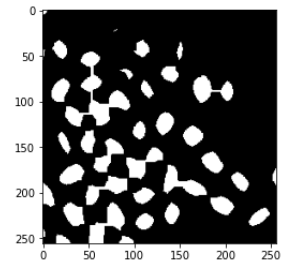


FIGURE 16 – Erosion - élément=square, size=10

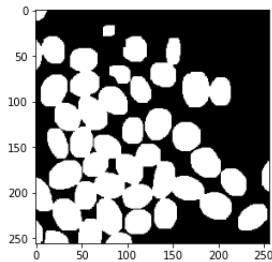


FIGURE 17 – Ouverture - élément=square, size=10

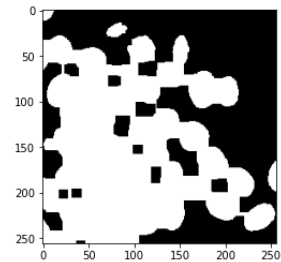


FIGURE 18 – Fermeture - élément=square, size=10

La forme de l'élément structurant a également un impact important, les objets ayant subi les opérations suivent la structure de l'élément. On peut voir ci-dessus la forme carré de l'élément.

1.2 Ces 4 opérations suivent des propriétés particulières :

- Dilatation :
Extensive / Croissante
- Erosion
Anti-Extensive / Croissante
- Ouverture
Filtre morphologique anti-extensif : Croissant + idempotent
- Fermeture
Filtre morphologique extensif : Croissant + idempotent

Illustration pour un élément structurant disque de taille 5 :

Croissance

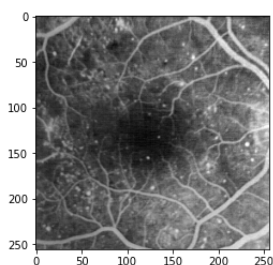


FIGURE 19 – Image original

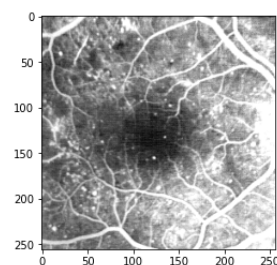


FIGURE 20 – Image éclaircit

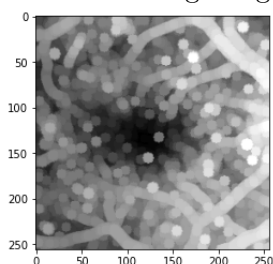


FIGURE 21 – Dilatation sur image original

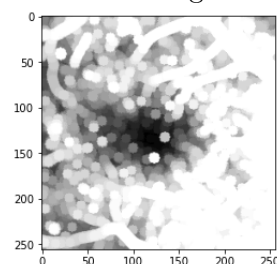


FIGURE 22 – Dilatation sur image éclaircit

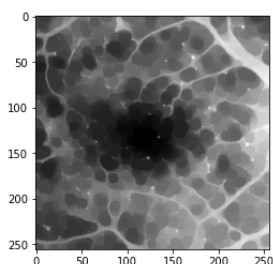


FIGURE 23 – Fermeture sur image original

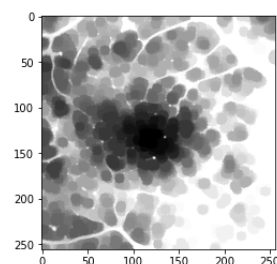


FIGURE 24 – Fermeture sur image éclaircit

La propriété de croissance signifie que si on effectue une opération sur une image et qu'on augmente l'intensité de l'image puis qu'on applique la même opération, le résultat après l'opération

sur la seconde image sera une même augmentation d'intensité. On voit bien à travers cet exemple que les opérations de dilatation et de fermeture sont croissantes, c'est aussi le cas pour l'érosion et l'ouverture.

f croissante $\Leftrightarrow \forall I, J, I \leq J \Rightarrow f(I) \leq f(J)$

Extensivité et Anti-extensivité

La propriété d'extensivité signifie que l'intensité de l'image augmente avec l'application d'une opération et inversement la propriété d'anti-extensivité signifie que l'intensité diminue.

f extensive $\Leftrightarrow \forall I, f(I) \geq I$

f anti-extensive $\Leftrightarrow \forall I, f(I) < I$

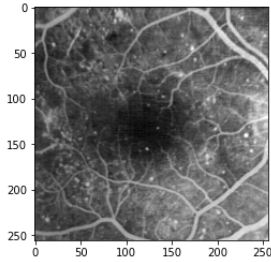


FIGURE 25 – Image original

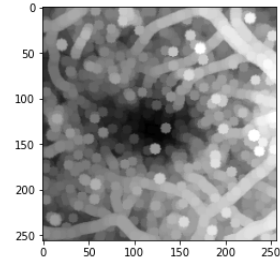


FIGURE 26 – Dilatation sur image originale

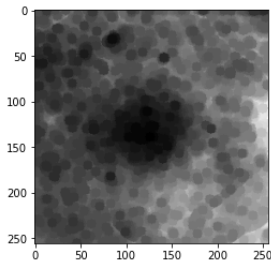


FIGURE 27 – Erosion sur image originale

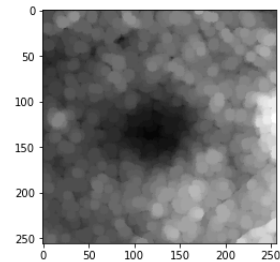


FIGURE 28 – Ouverture sur image originale

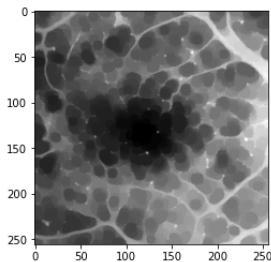


FIGURE 29 – Fermeture sur image originale

On voit bien que les opérations de dilatations et de fermetures sont extensives et que les érosions et les ouvertures sont anti-extensives.

Idempotence

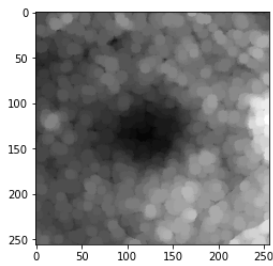


FIGURE 30 – Ouverture $f(I)$

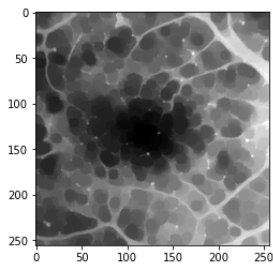


FIGURE 32 – Fermeture $F(I)$

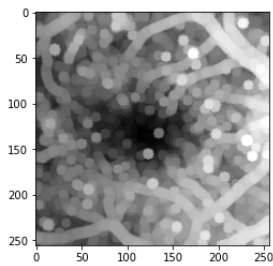


FIGURE 34 – Dilatation $f(I)$

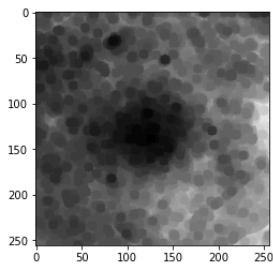


FIGURE 36 – Erosion $f(I)$

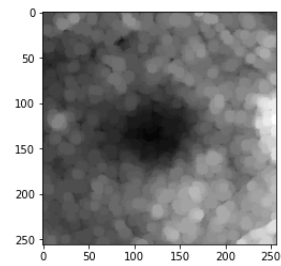


FIGURE 31 – Ouverture $f(f(I))$

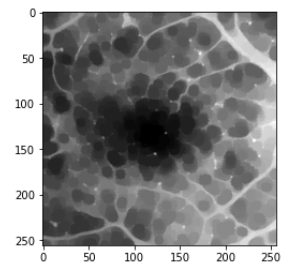


FIGURE 33 – Fermeture $f(f(I))$

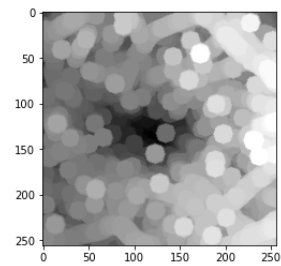


FIGURE 35 – Dilatation $f(f(I))$

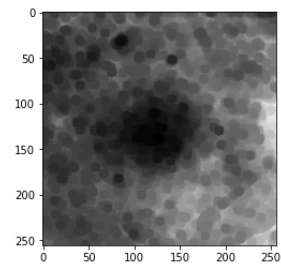


FIGURE 37 – Erosion $f(f(I))$

L'idempotence signifie que l'application successive d'une opération est équivalente à une application unique :

$$f \text{ Idempotent} \Leftrightarrow \forall I, f(f(I)) = f(I)$$

Les opérations d'ouverture et de fermeture sont bien idempotentes, mais pas les opérations de dilatation et d'érosion.

1.3 La dilatation vérifie la propriété d'associativité :

$$D[D(X, B), B'] = D(X, B \oplus B') \quad (1)$$

Pour l'exemple du carré, un carré de taille 3x3 est un élément structurant de taille 1 (une seule couche de points autour du centre) et un carré de taille 5x5 est un élément structurant de taille 2 (deux couches de points autour du centre). La propriété d'associativité indique que l'application de la dilatation avec un carré 3x3 puis un carré 5x5 revient à l'application directe de la dilatation avec leur addition de Minkowski, soit d'un élément structurant carré de taille 3, c'est-à-dire un carré de taille 7x7 (3 couches de points autour du centre). Les figures ci-dessous sont effectivement identiques :

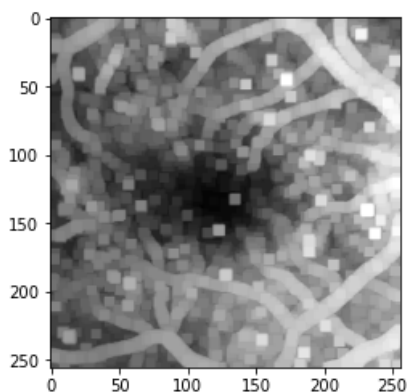


FIGURE 38 – Succession d'une dilatation avec un carré 3x3 puis 5x5

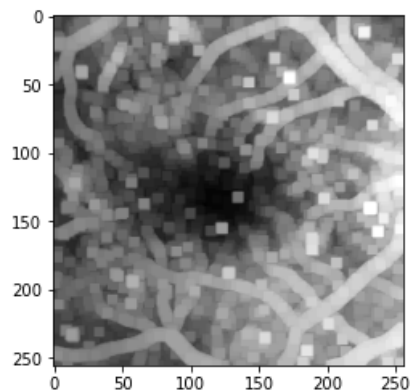


FIGURE 39 – Dilatation avec un carré 7x7

L'ouverture vérifie la propriété suivante :

$$(X_n)_{n'} = X_{\max(n, n')} \quad (2)$$

Par conséquent, la succession de l'ouverture par un carré de taille 3x3 puis par un carré de taille 5x5 donne le même résultat qu'une ouverture par un carré de taille 5x5, comme illustré sur les figures ci-dessous :

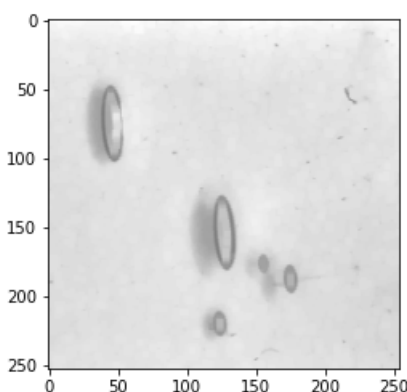


FIGURE 40 – Succession d'une dilatation avec un carré 3x3 puis 5x5

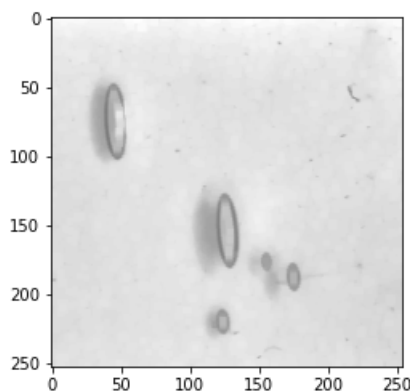


FIGURE 41 – Dilatation avec un carré 5x5

1.4 Le chapeau haut-de-forme $X - X_B$ permet de conserver les objets très contrastés claires. Les quatre figures présentées ci-dessous sont les résultats de l'image rétinée par une transformation chapeau haut-de-forme avec des éléments structurants de différentes formes et tailles. On remarque avec les figures 43 et 44 que plus l'élément structurant est grand, plus la plage de tailles d'objets contrastés conservés est importante, plus il y a de zones claires sur l'image finale. En effet plus l'élément structurant sera grand, plus les pics (zones très claires) seront effacés. La forme de l'élément structurant influence également la forme et le nombre d'éléments que l'on peut retrouver dans l'image finale. La forme ligne est la plus sélective et ne permet de conserver que les éléments fins qui ont la même orientation que la ligne. La forme la moins sélective est le disque figure 45 qui conserve plus de petits éléments contrastés que le carré de même taille.

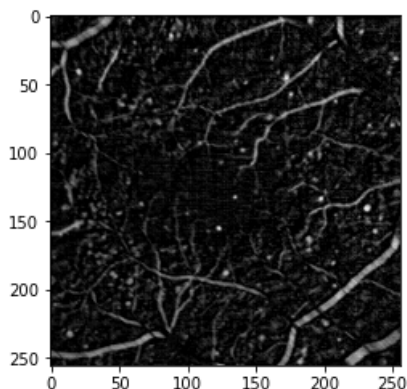


FIGURE 42 – Chapeau haut-de-forme, élément structurant ligne, taille 10, direction - 45°

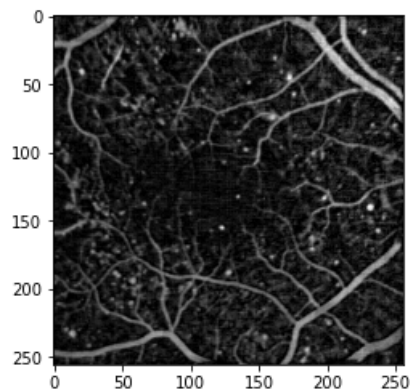


FIGURE 43 – Chapeau haut-de-forme, élément structurant carré, taille 10

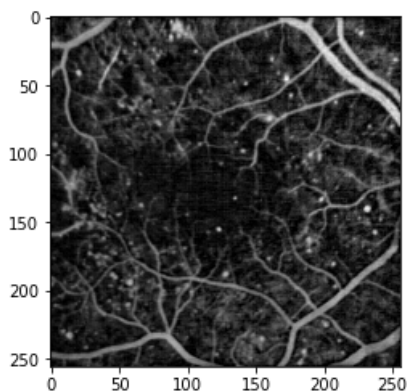


FIGURE 44 – Chapeau haut-de-forme, élément structurant carré, taille 20

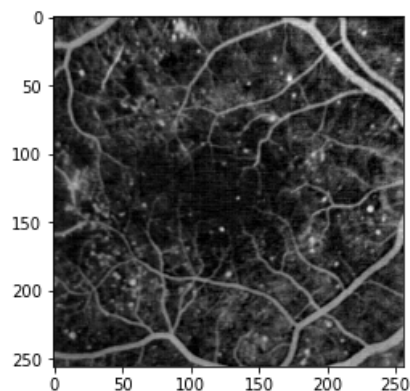


FIGURE 45 – Chapeau haut-de-forme, élément structurant disque, taille 20

L'opération duale du chapeau haut-de-forme est la différence entre l'image fermée et elle-même :

$$X^B - X \quad (3)$$

Cette opération duale permet de récupérer les zones très sombres de l'image (pics sombres), et de les transformer en zones très claires (pics inversés), comme illustré en l'appliquant sur l'image Laiton sur les figures ci-dessous :

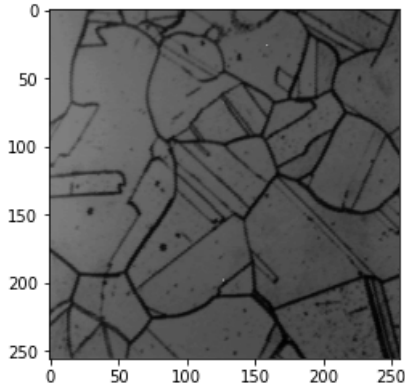


FIGURE 46 – Image laiton initiale

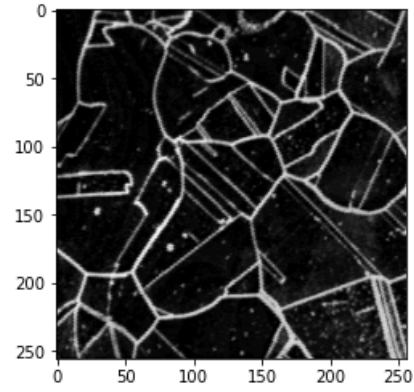
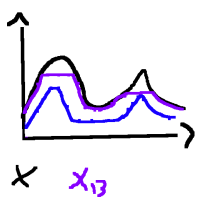
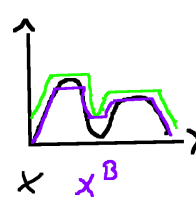


FIGURE 47 – Duale du chapeau haut-de-forme, élément structurant disque, taille 20

Voici des schémas représentant les opérations :



$$X - X_B \rightarrow$$



$$X^B - X \rightarrow$$



FIGURE 48 – Top-hat

FIGURE 49 – Duale Top-hat

1.5 L'application d'une ouverture avec un élément structurant ligne donne des images finales différentes en fonction de la direction de la ligne sélectionnée. Seules les zones blanches ayant une forme rectiligne et dans une direction similaire à la ligne seront maintenues dans l'image finale. Les figures ci-dessous ont été réalisées avec un élément structurant ligne de taille 10, dans les directions 0° , 45° , -45° et 90° . La figure 54 est la combinaison des pixels maximum des quatre figures précédentes. On remarque que cette combinaison a permis de supprimer les petites zones blanches parasites et de sélectionner les formes claires principales dans les directions sélectionnées. En appliquant cette combinaison sur des images ayant été dilatées par un élément structurant ligne de taille 10 et orienté dans 360 directions, l'image finale est semblable à l'image initiale sans les petites zones blanches parasites, comme représenté figure 56. En effet ces petites zones blanches ne sont pas conservées par l'érosion compte tenu du fait que l'élément structurant en forme de ligne de cette taille n'est pas inclus dans ces petites régions circulaires.

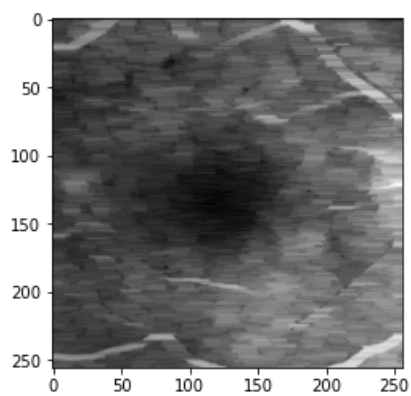


FIGURE 50 – Opening par un élément structurant de taille 10, orientation 0°

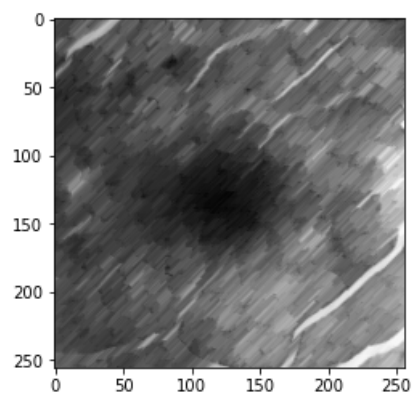


FIGURE 51 – Opening par un élément structurant de taille 10, orientation 45°

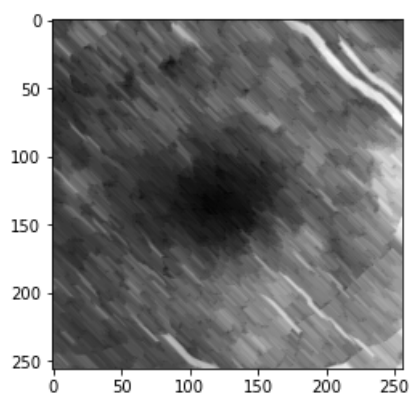


FIGURE 52 – Opening par un élément structurant de taille 10, orientation -45°

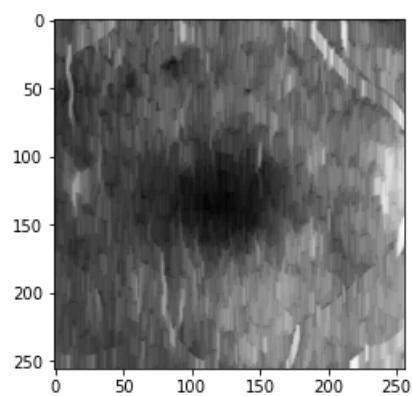


FIGURE 53 – Opening par un élément structurant de taille 10, orientation 90°

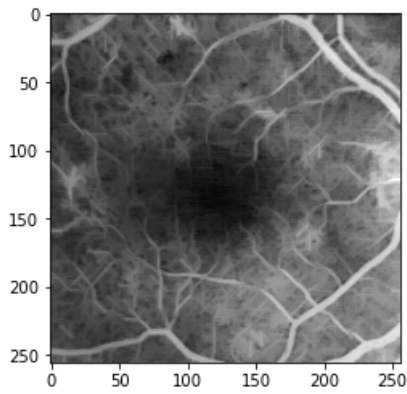


FIGURE 54 – Combinaison des pixels maximum des 4 images précédentes

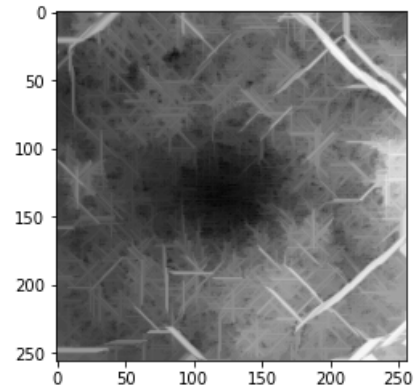


FIGURE 55 – Combinaison des pixels maximum des 4 images précédentes avec un élément de taille 20

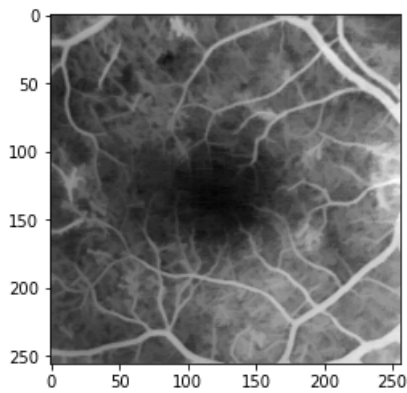


FIGURE 56 – Combinaison des pixels maximum de 360 images qui ont été dilatées à partir de l'image d'origine par des éléments structuraux de forme ligne et de taille 10, orientés à 1° d'écart de 0 à 360 °

Le code pour obtenir l'image figure 56 est le suivant :

```
deg = np.arange(0.0, 360.0, 1.0)
se = [strel('line',10,i) for i in deg]
open = [morpho.opening(im,i) for i in se]
pymax = im
for i in range(im.shape[0]):
    for j in range(im.shape[1]):
        pymax[i,j] = max([el[i,j] for el in open])
plt.imshow(pymax,cmap="gray")
```

Filtres alternée séquentiels

2. Chaque nouvelle étape de filtrage permet d'enlever des petites zones claires grâce à l'ouverture et des petites zones foncées grâce à la fermeture. La taille de la séquence de filtrage dépend de l'effet souhaité. Peu de niveaux seront appliqués si l'objectif est d'enlever les petites zones sombres et claires parasites (les détails). Par exemple les figures 57 et 58 ont reçu respectivement 2 et 3 niveaux de combinaison ouverture/fermeture et on remarque bien que les petits détails sont supprimés. Par ailleurs, si l'objectif est de simplifier grandement l'image, beaucoup de niveaux seront appliqués. Les régions "plates" obtenues peuvent par la suite servir de marqueur dans des algorithmes de segmentation en seuillant la région qui nous intéresse, dans le but de guider la segmentation. Les figures 59 et 60 ont reçu respectivement 10 et 20 niveaux de combinaison ouverture/fermeture. Si on continue avec des éléments de plus en plus grand on tendrait vers une image uniforme en niveau de gris.

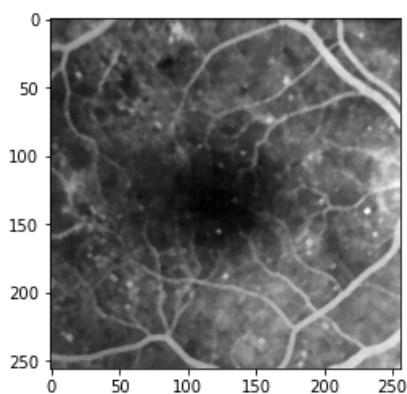


FIGURE 57 – N=2

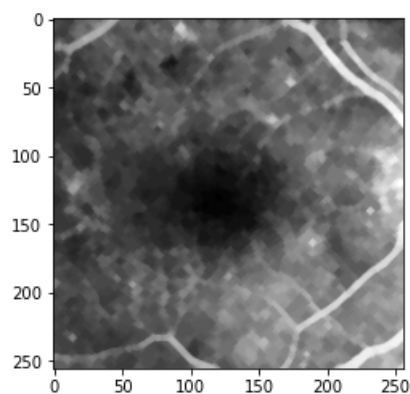


FIGURE 58 – N=3

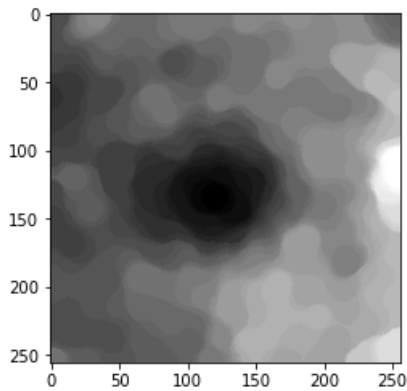


FIGURE 59 – N=10

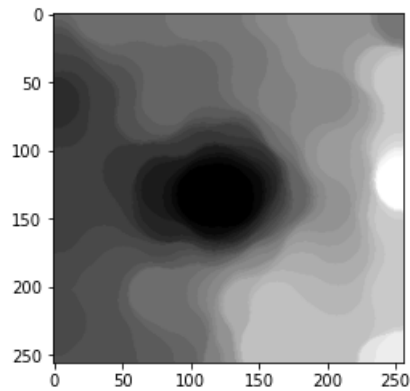


FIGURE 60 – N=20

Reconstruction

3.1 La reconstruction par dilatation $[D(Y, B_1) \cap X]^\infty = D_X^\infty(Y)$ permet à partir du marqueur d'ouverture de récupérer les régions claires de l'image qui n'ont pas été complètement supprimées par l'étape d'érosion. En effet la reconstruction consiste à ajouter l'intersection d'une dilatation Y de l'ouverture avec un élément structurant de taille 1 avec l'image original X jusqu'à convergence.

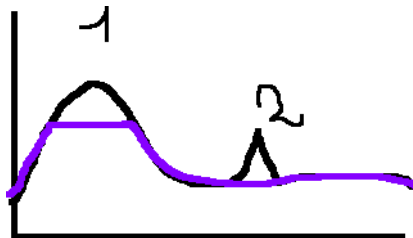


FIGURE 61 – Ouverture

Sur l'exemple ci-dessus, l'étape d'érosion a supprimé le pic 2 mais le pic 1 est toujours présent. La reconstruction par dilatation va permettre de retrouver complètement les informations du pic 1 sans reconstruire le pic 2. On obtiendra donc une image similaire à l'image originale sans les petites zones blanches (en fonction de la taille de l'élément) mais en conservant tous les détails des éléments claires les plus importants.

L'ouverture d'une image permet donc de retirer des petites zones claires parasites d'une image mais entraîne également une perte d'informations de l'image. La reconstruction permet de retrouver cette information perdue sans récupérer les zones parasites. Les figures ci-dessous sont des exemples de reconstruction de l'image de rétine. On remarque en comparant les images à partir d'éléments structurants de taille 2 et 4 (images de gauche VS images de droite) que plus l'élément structurant est petit, plus le nombre de détails restitués de l'image d'origine est important, mais des détails "parasites" plus petits sont maintenus, en effet ils n'ont pas été retirés par l'érosion. On remarque par exemple que les petites tâches blanches sont maintenues sur la figure 62 alors qu'elles disparaissent en augmentant la taille de l'élément structurant figure 63. Par ailleurs, la forme de l'élément structurant semble influencer peu le résultat mais malgré tout elle peut entraîner une variation de la forme des éléments mis en avant sur l'image reconstruite. Ainsi on remarque une tâche légèrement plus contrastée pour l'élément structurant de forme carré en haut de la figure 63 que pour l'élément structurant de forme diamant figure 65.

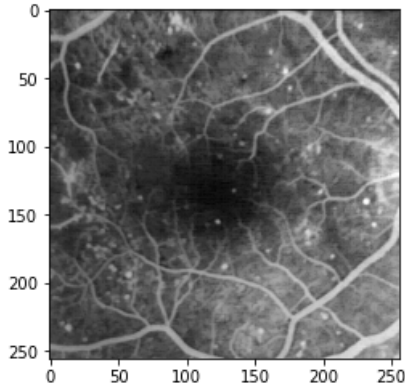


FIGURE 62 – Reconstruction par ouverture, élément structurant taille 2, forme carrée

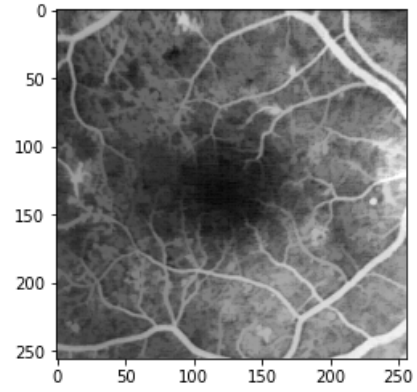


FIGURE 63 – Reconstruction par ouverture, élément structurant taille 4, forme carrée

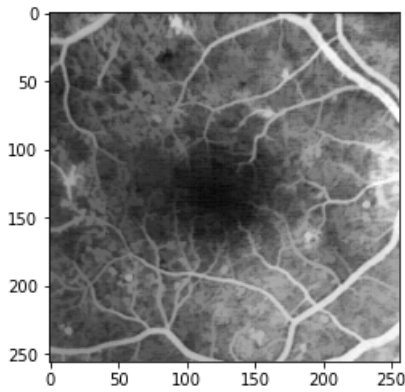


FIGURE 64 – Reconstruction par ouverture, élément structurant taille 2, forme diamant

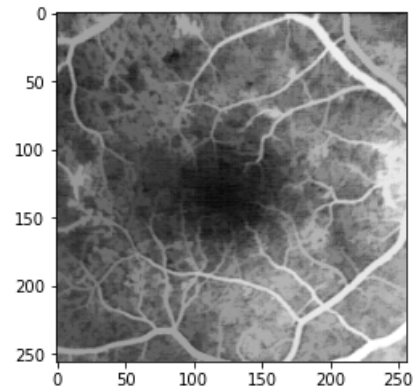


FIGURE 65 – Reconstruction par ouverture, élément structurant taille 4, forme diamant

La reconstruction par fermeture $[E(Y, B1) \cup X]^\infty = E_X^\infty(Y)$ permet à partir du marqueur de fermeture de récupérer les régions sombres de l'image qui n'ont pas été complètement supprimées par l'étape d'ouverture. En effet la reconstruction consiste à prendre l'union d'une érosion de la fermeture avec un élément structurant de taille 1 avec l'image original X jusqu'à convergence.

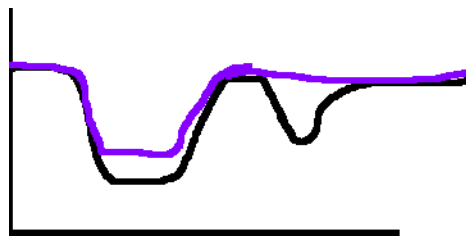


FIGURE 66 – Fermeture

Sur l'exemple ci-dessus, l'étape de dilatation a supprimé la petite vallée mais la plus importante est toujours présente. La reconstruction par érosion va permettre de retrouver complètement les informations de la grande vallée sans reconstruire la seconde. On obtiendra donc une image

similaire à l'image originale sans les petites vallées sombres (en fonction de la taille de l'élément) mais en conservant tous les détails des éléments sombres les plus importants.

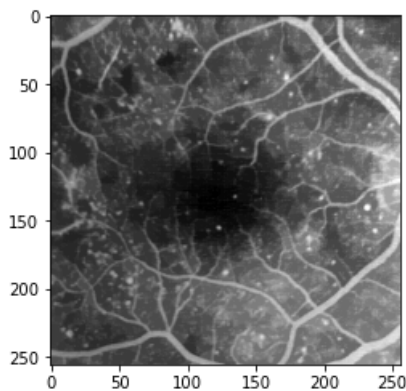


FIGURE 67 – Reconstruction par érosion, élément structurant taille 2, forme disque

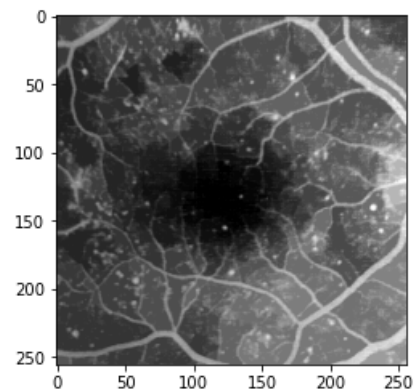


FIGURE 68 – Reconstruction par érosion, élément structurant taille 4, forme disque

Plus l'élément structurant est grand plus on perd les grandes vallées.

3.2 Selon le principe de dualité pour obtenir le même résultat qu'une reconstruction d'image par érosion depuis une fermeture il faut :

- Inverser l'image : $255 - img$
- Faire l'ouverture de l'image : *open*
- Reconstruire l'image par dilatation depuis l'ouverture : *rec*
- Inverser le résultat : $255 - rec$

Voici le résultat d'une fermeture normale et d'une fermeture obtenue par principe de dualité, pour un élément structurant de taille 4 et une forme de disque :

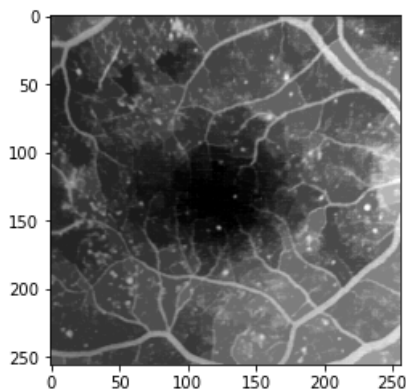


FIGURE 69 – Reconstruction par érosion normale

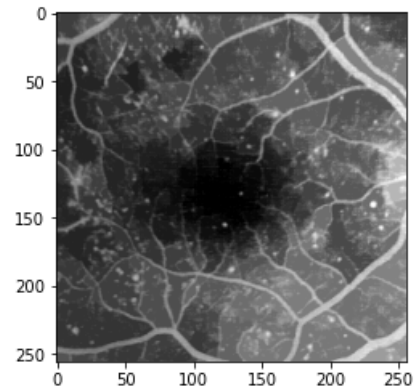


FIGURE 70 – Reconstruction par érosion en utilisant le principe de dualité

3.3 Les filtres alternés séquentiels permettent comme nous l'avons vu précédemment de supprimer des détails de l'images, cependant pendant cette opération on dégrade aussi les éléments

plus important. On ajoutant des étapes de reconstruction nous pouvons supprimer les détails et retrouver les informations des éléments importants. Voici les résultats avec et sans reconstruction avec un élément structurant de taille 4 en forme de disque :

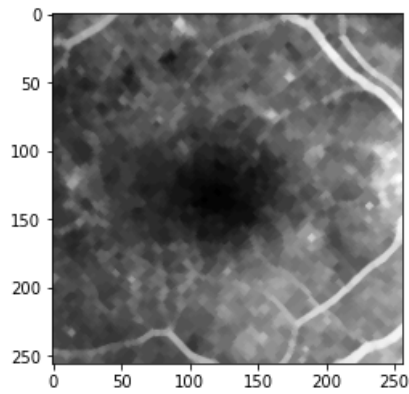


FIGURE 71 – Asf sans reconstruction

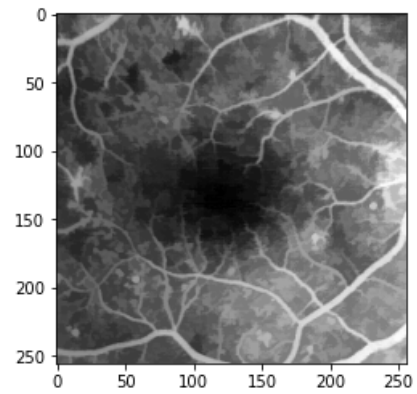


FIGURE 72 – Asf avec reconstruction

Cela permet aux filtres alternés séquentiels d'être plus robustes en reconstruisant les éléments qui n'ont pas été supprimés par les étapes d'érosions et de dilations des ouvertures et fermetures.

Segmentation

4.1 Le gradient morphologique permet de détecter les contours de l'objet de l'image, en mettant en avant les zones de jonction entre les zones claires et les zones foncées, comme illustré ci-dessous.

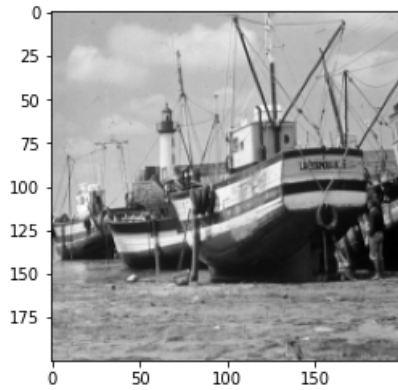


FIGURE 73 – Image initiale du bateau

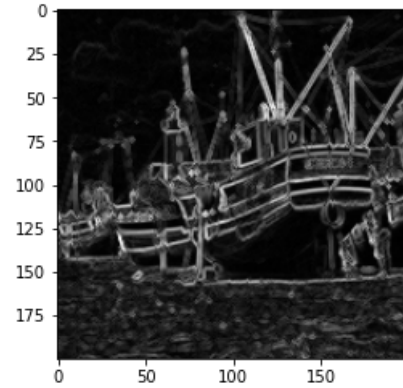


FIGURE 74 – Gradient morphologique de l'image du bateau

4.2 La ligne de partage des eaux permet de réaliser une segmentation de l'image étudiée. L'algorithme prend en entrée une image et des marqueurs. Les marqueurs sont le point de départ de l'étape d'inondation de l'algorithme. Lorsque 2 bassins versant se croisent, une ligne est mise en place à leur jonction, c'est la ligne de partage des eaux et elle permet de créer les régions de la segmentation. Les marqueurs utilisés ici sont les minimums locaux du gradient du gradient morphologique, la segmentation s'effectue sur la gradient lui-même.

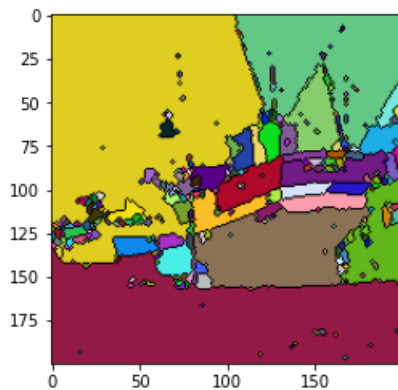


FIGURE 75 – Image du bateau segmentée : les lignes de partage des eaux séparent les zones segmentées de l'image représentées par des couleurs différentes

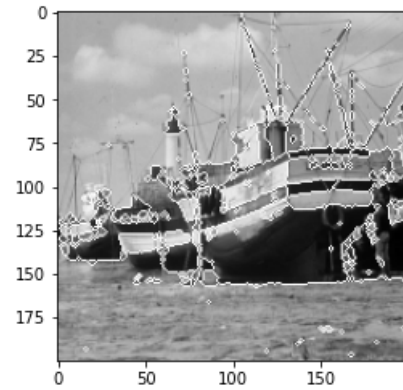


FIGURE 76 – Image du bateau segmentée : les lignes de partage des eaux sont superposées à l'image pour une visualisation de la segmentation de l'image initiale

La segmentation obtenue est relativement peu intéressante dans le sens où on retrouve de nombreuses petites régions qui n'apportent aucune information pertinente. Cela est dû aux choix simplistes des marqueurs.

4.3 L'application d'une fermeture sur l'image du gradient morphologique permet de réduire le nombre de zones sombres parasites. Le résultat de la segmentation sur la fermeture du gradient morphologique est présenté figure 77 et 78.

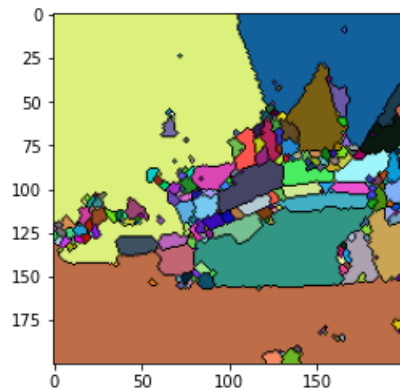


FIGURE 77 – Image du bateau segmentée après application de la ligne de partage des eaux sur la fermeture du gradient morphologique de l'image initiale

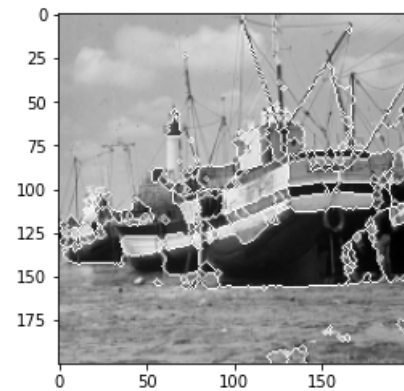


FIGURE 78 – Image du bateau segmentée après application de la ligne de partage des eaux sur la fermeture du gradient morphologique de l'image initiale

En effet la fermeture va permettre de régulariser l'image, ce qui va de limiter le nombre de minimum locaux et donc le nombre de régions indésirables segmentées :

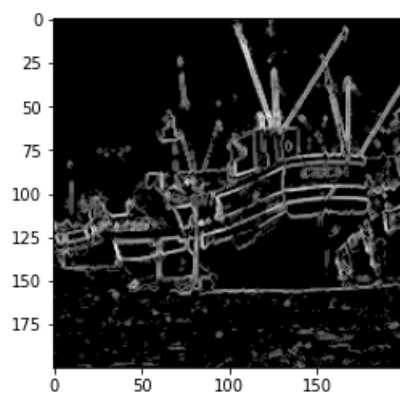


FIGURE 79 – Gradient morphologique

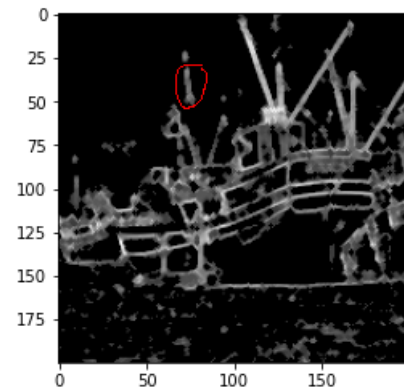


FIGURE 80 – Fermeture du gradient morphologique

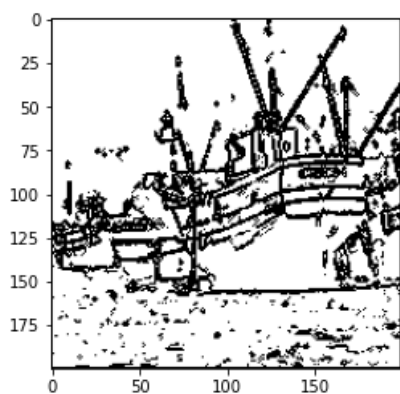


FIGURE 81 – Minimum locaux

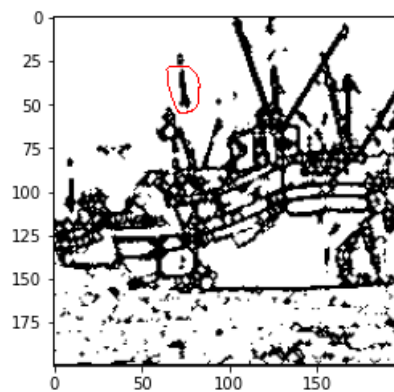


FIGURE 82 – Minimum locaux avec fermeture du gradient

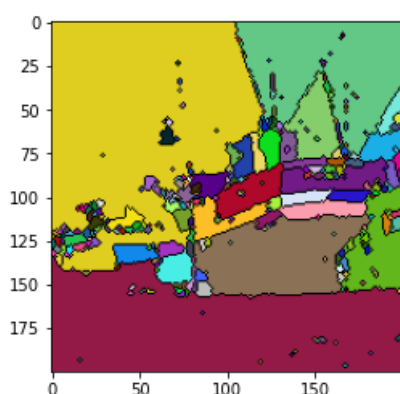


FIGURE 83 – Segmentation

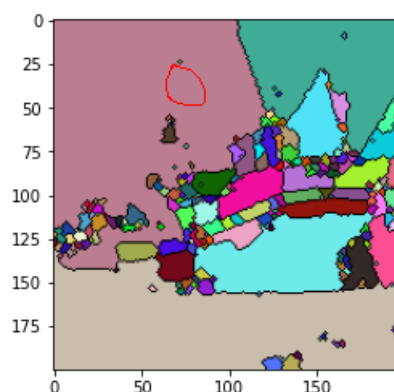


FIGURE 84 – Segmentation avec fermeture du gradient

L'image ainsi obtenue contient encore des régions de segmentation parasites. Pour une segmentation plus nette, nous avons essayé de traiter l'image avec une reconstruction par dilatation avant l'application du gradient morphologique, avec un élément structurant de forme disque et de taille 2. L'idée est de simplifier l'image pour retirer les détails de l'image qui mène la création de marqueurs non importants. Les résultats de segmentation obtenus sont présentés figures 85 et 86. La segmentation obtenue contient moins de détails mais a permis de supprimer une partie des zones parasites non souhaitées dans la segmentation.

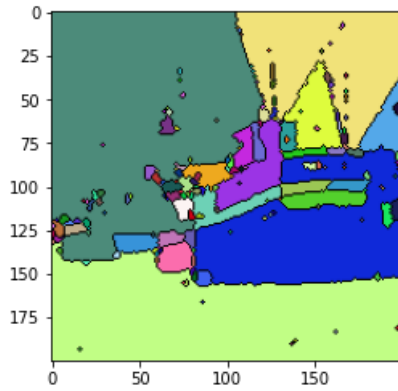


FIGURE 85 – Image du bateau segmentée après application de la ligne de partage des eaux sur la fermeture du gradient morphologique de l'image initiale

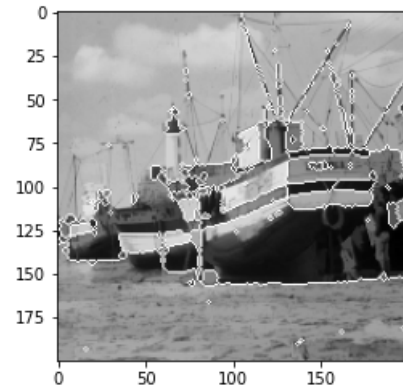


FIGURE 86 – Image du bateau segmentée après application de la ligne de partage des eaux sur la fermeture du gradient morphologique de l'image initiale

4.4 Nous avons éliminer les minimums locaux, mais nous n'avons pas d'idée pour l'élimination des minimums régionaux.

Pour limiter le nombre de minimum locaux on peut seuiller l'image du gradient avant de faire le watershed. On obtiendra donc moins de régions segmentées en fonction du seuil. C'est ce qui a déjà été réalisé dans le code :

```
grad=np.int32(grad>40)*grad
```

Voici les résultats avec et sans seuillage des minimums du gradient morphologique :

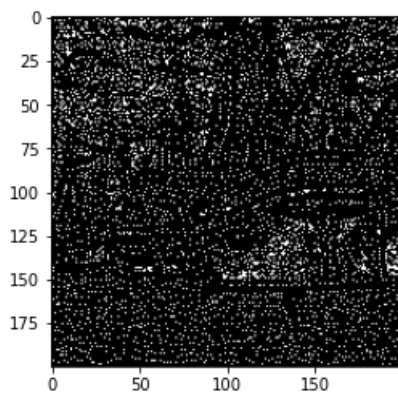


FIGURE 87 – Minimum locaux sans seuillage

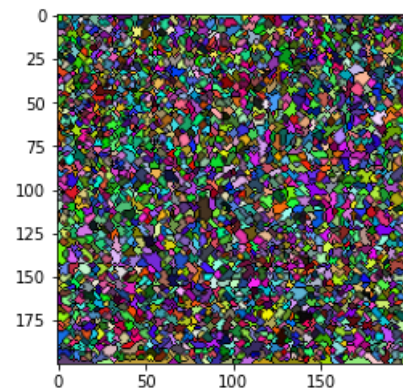


FIGURE 88 – Watershed depuis les minimums locaux sans seuillage

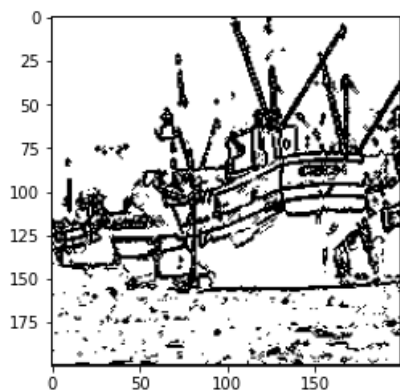


FIGURE 89 – Minimum locaux avec seuillage

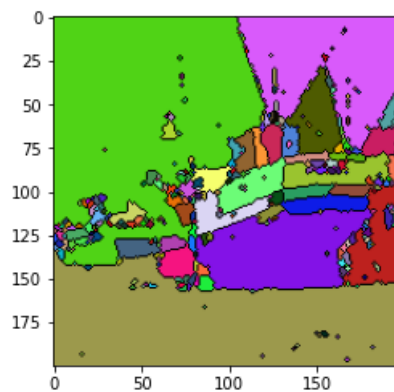


FIGURE 90 – Watershed depuis les minimums locaux avec seuillage

4.4/4.5 On peut aussi tester en tant que marqueurs un seuillage des filtres alternés séquentiels comme nous l'avons mentionné auparavant. (Code en Annexes)

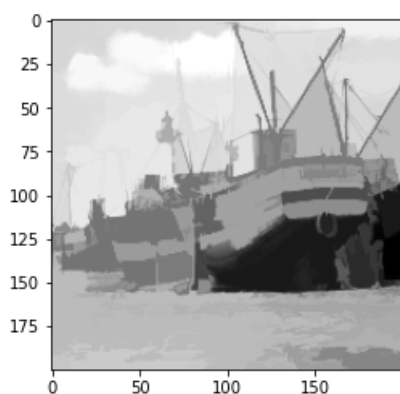


FIGURE 91 – Asf

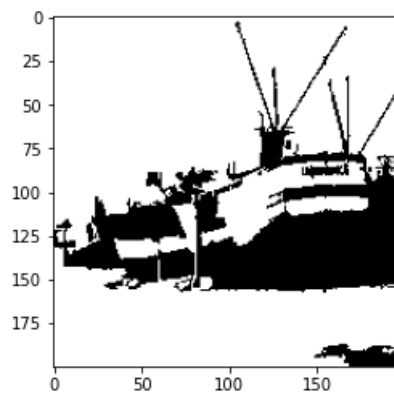


FIGURE 92 – Asf avec seuillage $T=120$

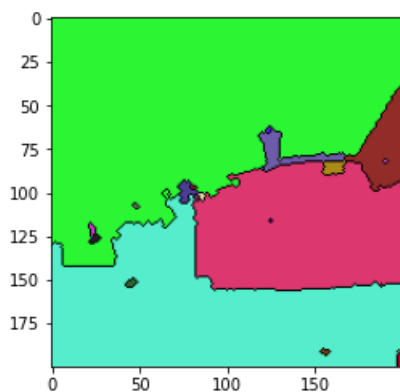


FIGURE 93 – Segmentation

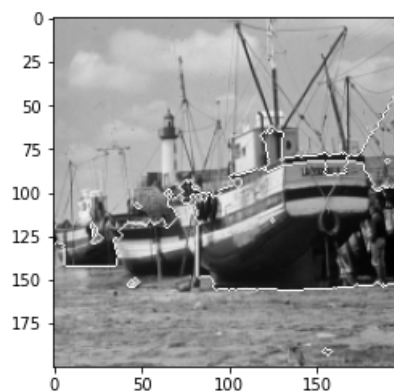


FIGURE 94 – Visualisation de la segmentation

Le seuillage nous permet de choisir quel objet nous voulons conserver pour la segmentation, ici nous avons choisi de segmenter le bateau, le résultat est intéressant dans le sens où il y a peu de régions parasites, cependant le bateau n'est pas parfaitement segmenté, il faudrait mettre en

place des tests pour trouver le seuil et le niveau N de reconstruction pour les filtres alternés séquentiels idéals en fonction de l'objet à segmenter.

4.5 L'inconvénient de l'algorithme de ligne de partage des eaux est la sur-segmentation. Pour régler ce problème on peut définir des marqueurs. On va mettre en place une reconstruction par érosion de l'image $\min(\text{marqueurs}, \text{image gradient})$ $E_{f \wedge g}(g, B_\infty)$ ce qui va permettre que les minimums soient uniquement ceux qui ont été définis par les marqueurs. On applique par la suite l'algorithme du watershed sur cette image reconstruite, cela permet d'assurer que les bassins versants correspondent soit à l'objet à segmenter soit au fond, on obtient un contour fermé autour de l'objet. (Code en Annexes)

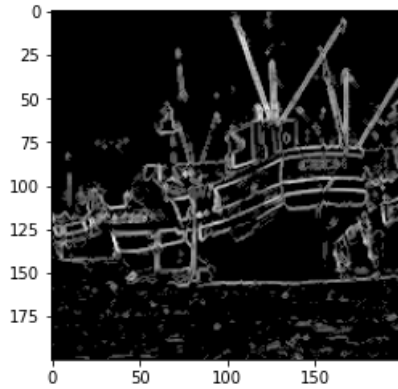


FIGURE 95 – Gradient morphologique

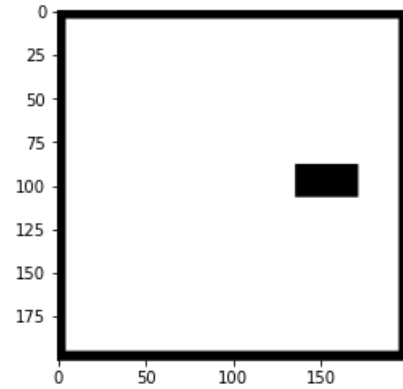


FIGURE 96 – Masque - Marqueur

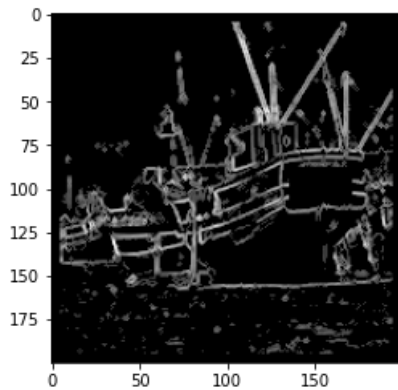


FIGURE 97 – minimum entre gradient et marqueurs

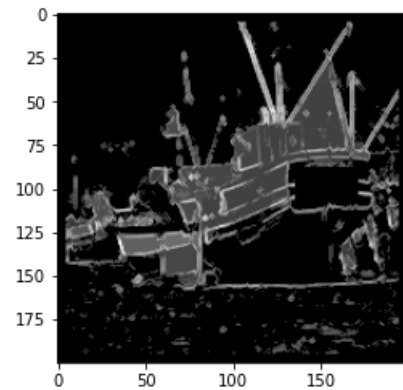


FIGURE 98 – Reconstruction par érosion de la figure 97

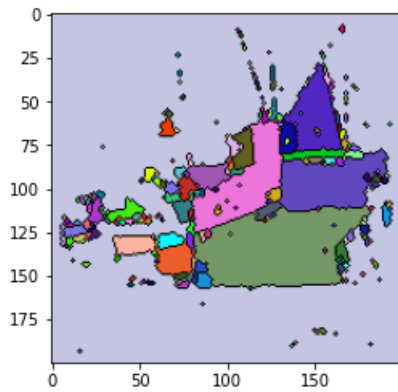


FIGURE 99 – Segmentation

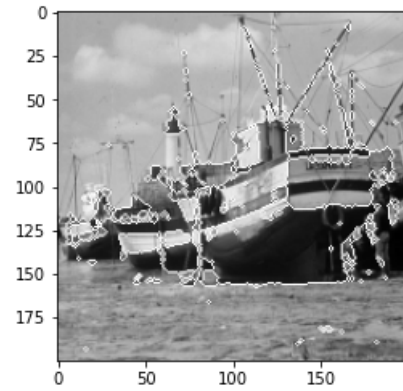


FIGURE 100 – Visualisation de la segmentation

4.6

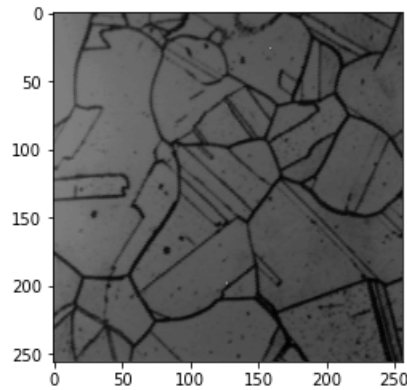


FIGURE 101 – Image initiale : Laiton

Nous avons pensé à 2 façons de faire :

- Avec des marqueurs définis manuellement
- Avec des marqueurs définis automatiquement

Marqueur définis manuellement

Nous avons utilisé le même principe que la question précédente, où l'on définit manuellement des marqueurs à travers un masque, puis on applique une reconstruction par érosion sur l'image $\min(\text{masque}, \text{gradient})$ pour que les minimums soient uniquement ceux qui ont été définis par les marqueurs. Les bassins versants vont donc partir de ces minimums et s'innoder jusqu'à l'intersection d'un nouveau bassin, cela va nous donner la segmentation des cellules et donc des lignes noires. (Code en Annexes)

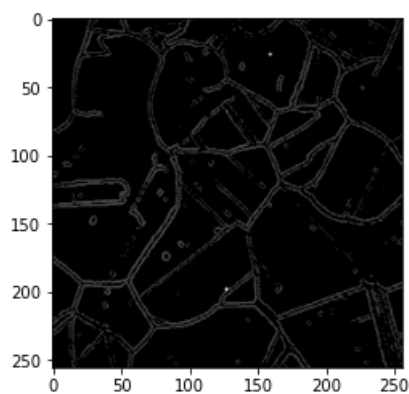


FIGURE 102 – Gradient morphologique

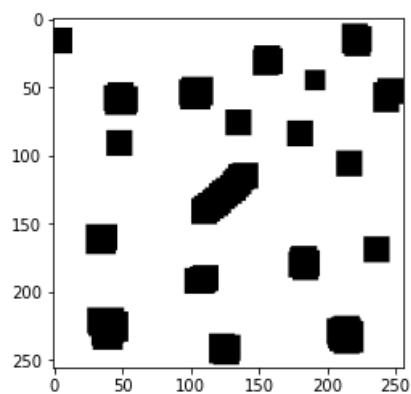


FIGURE 103 – Masque

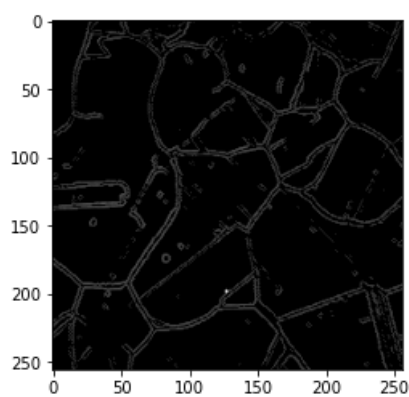


FIGURE 104 – $\min(\text{masque}, \text{gradient})$

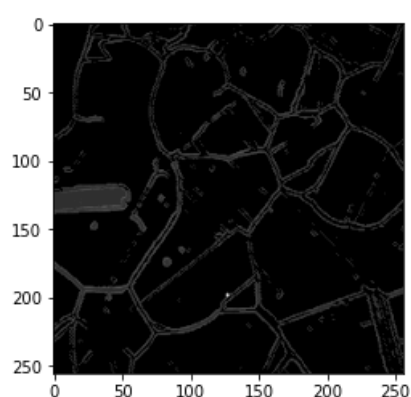


FIGURE 105 – Reconstruction

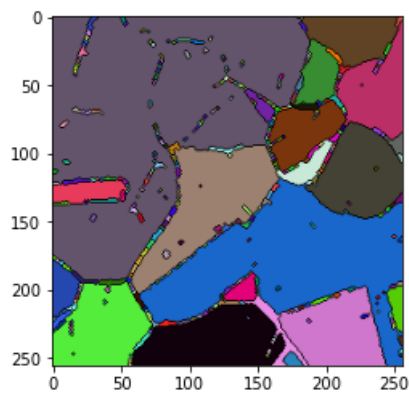


FIGURE 106 – Segmentation

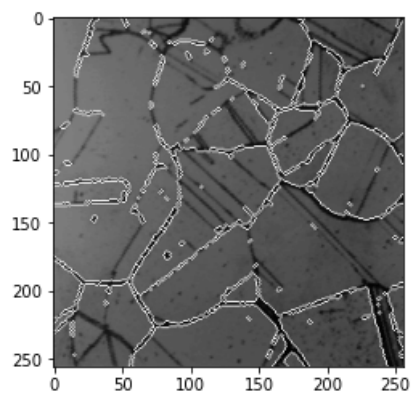


FIGURE 107 – Visualisation de la segmentation

Les résultats sont très intéressants malgré quelques artefacts (on pourrait utiliser un pré-traitement pour enlever les imperfections) mais cependant il est coûteux de définir des marqueurs manuellement, on peut donc penser à une approche automatique.

Marqueur définis automatiquement

Nous sommes parties de l'idée de vouloir inonder chaque bassin versant en associant un bassin versant à chaque cellule, de cette manière nous pourrions obtenir une segmentation des lignes noires.

Voici les étapes que nous avons mis en place :

- Asf pour supprimer les artéfacts
- Calculer le gradient morphologique
- Mettre en place le masque de marqueurs en faisant une dilatation pour remplir les lignes
- Calculer le watershed sur l'image du gradient en utilisant comme marqueur les minimums locaux du masque précédent.

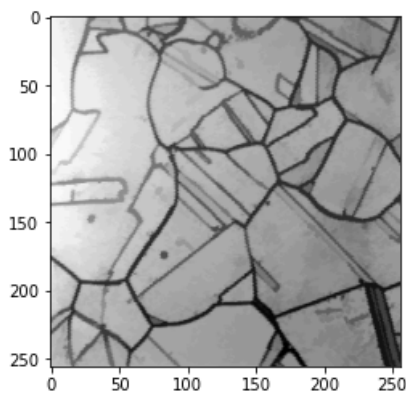


FIGURE 108 – Asf

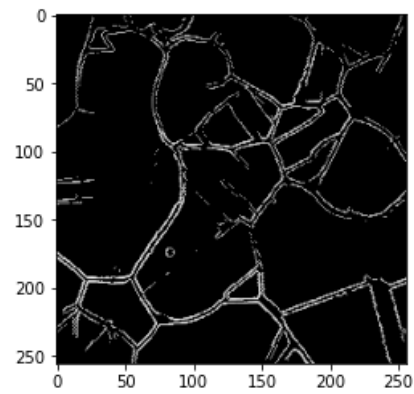


FIGURE 109 – Gradient morphologique

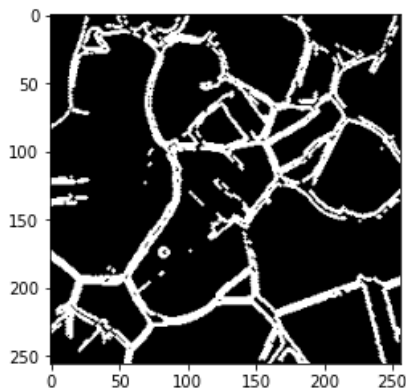


FIGURE 110 – Gradient morphologique dilaté

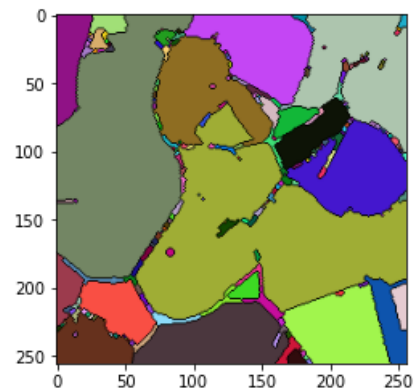


FIGURE 111 – Segmentation

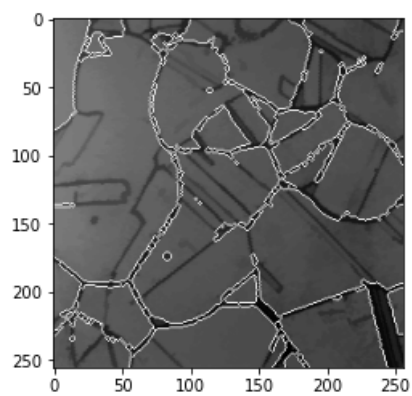


FIGURE 112 – Visualisation de la segmentation

Annexes - code

Code de la question 4.4/4.5 - ASF et seuillage

```
im=skio.imread('Images/bat200.bmp')
se=morpho.selem.disk(1)
plt.imshow(im, cmap='gray')
plt.show()

grad=morpho.dilation(im,se)-morpho.erosion(im,se)
grad=np.int32(grad>40)*grad
plt.imshow(grad, cmap="gray")
plt.show()

# Reconstruction ASF
markers=im.copy()
N=10
for k in range(N):
    se=strel('disk',k)
    open=morpho.opening(markers,se)
    recopen = morpho.reconstruction(open, markers)
    clos=morpho.closing(recopen,se)
    markers = morpho.reconstruction(clos, recopen, 'erosion')
plt.imshow(markers, cmap="gray")
plt.show()
# Seuillage
markers = markers > 120
plt.imshow(markers, cmap="gray")
plt.show()
markers = ndi.label(markers)[0]

# Watershed
labels = morpho.watershed(grad, markers, watershed_line=True)
plt.imshow(couleurs_alea(labels))
plt.show()

# visualization of the result
segm=labels.copy()
for i in range(segm.shape[0]):
    for j in range(segm.shape[1]):
        if segm[i,j] == 0:
            segm[i,j]=255
        else:
            segm[i,j]=0
#superimposition of segmentation contours on the original image
contourSup=np.maximum(segm,im)
plt.imshow(contourSup, cmap="gray")
plt.show()
```

Code de la question 4.5 - Masque et reconstruction du gradient

```
im=skio.imread('Images/bat200.bmp')
markers=skio.imread('Images/bat200Markers.bmp')
se=morpho.selem.disk(1)

# Gradient Morphologique
se=morpho.selem.disk(1)
grad=morpho.dilation(im,se)-morpho.erosion(im,se)
grad=np.int32(grad>40)*grad
plt.imshow(grad,cmap="gray")
plt.show()

# on rends me masque binaire
markers = markers < 100
markers = np.where(markers == True, 0,255)
plt.imshow(markers, cmap='gray')
plt.show()

# min(gradient, markers)
min = np.zeros((grad.shape))
for i in range(min.shape[0]):
    for j in range(min.shape[1]):
        min[i,j] = np.min([markers[i,j], grad[i,j]])
plt.imshow(min, cmap='gray')
plt.show()

# reconstruction de l'image markers depuis l'image min
rec=morpho.reconstruction(markers, min, 'erosion');
plt.imshow(rec, cmap='gray')
plt.show()

# Selection des minimum locaux depuis l'image des marqueurs reconstruite
local_mini = skf.peak_local_max(255-rec, #il n'y a pas de fonction local_min...
                                indices=False)
markers1 = ndi.label(local_mini)[0]

# watershed sur l'image reconstruite
labels = morpho.watershed(rec, markers1, watershed_line=True)
plt.imshow(couleurs_alea(labels))
plt.show()
# viewimage_color(couleurs_alea(labels)) - Utilisable si gimp 'est install

# visualization of the result
segm=labels.copy()
for i in range(segm.shape[0]):
    for j in range(segm.shape[1]):
        if segm[i,j] == 0:
            segm[i,j]=255
        else:
            segm[i,j]=0
#superimposition of segmentation contours on the original image
contourSup=np.maximum(segm,im)
plt.imshow(contourSup,cmap="gray")
plt.show()
```

Code de la question 4.6 - Version 1 - Sélection manuelle des marqueurs

```
im=skio.imread('Images/laiton.bmp')
plt.imshow(im, cmap='gray')
plt.show()
markers=skio.imread('Images/laitonMarkers.bmp')
plt.imshow(markers, cmap="gray")
plt.show()

# Gradient morphologique
se=morpho.selem.disk(1)
grad=morpho.dilation(im,se)-morpho.erosion(im,se)
grad=np.int32(grad>40)*grad
plt.imshow(grad, cmap="gray")
plt.show()

# min(gradient, markers)
min = np.zeros((grad.shape))
for i in range(min.shape[0]):
    for j in range(min.shape[1]):
        min[i,j] = np.min([markers[i,j], grad[i,j]])

plt.imshow(min, cmap='gray')
plt.show()

# reconstruction de l'image des markers depuis l'image min
rec=morpho.reconstruction(markers, min, 'erosion');
plt.imshow(rec, cmap='gray')
plt.show()

# selection des minmums locaux
local_mini = skf.peak_local_max(255-rec, #il n'y a pas de fonction local_min...
                                indices=False)
markers1 = ndi.label(local_mini)[0]

# watershed sur l'image du gradient
labels = morpho.watershed(grad, markers1, watershed_line=True)
plt.imshow(couleurs_alea(labels))
plt.show()
# viewimage_color(couleurs_alea(labels)) - Utilisable si gimp 'est install

# visualization of the result
segm=labels.copy()
for i in range(segm.shape[0]):
    for j in range(segm.shape[1]):
        if segm[i,j] == 0:
            segm[i,j]=255
        else:
            segm[i,j]=0
#superimposition of segmentation contours on the original image
contourSup=np.maximum(segm,im)
plt.imshow(contourSup, cmap="gray")
plt.show()
```

Code de la question 4.6 - Version 2 - Sélection automatique des marqueurs

```
im=skio.imread('Images/laiton.bmp')
plt.imshow(im, cmap='gray')
plt.show()

# Filtre alternatif séquentiel
im1=im.copy()
N=3
for k in range(N):
    se=strel('disk',k)
    open=morpho.opening(im1,se)
    recopen = morpho.reconstruction(open, im1)
    clos=morpho.closing(recopen,se)
    im1 = morpho.reconstruction(clos, recopen, 'erosion')
plt.imshow(im1,cmap="gray")
plt.show()
im=im1

# dilatation du gradient morphologique
se=morpho.selem.disk(1)
grad=morpho.dilation(im,se)-morpho.erosion(im,se)
grad=np.int32(grad>40)*grad
plt.imshow(grad,cmap="gray")
plt.show()
mask=grad>20
se1=strel('disk',1)
mask=morpho.dilation(mask,se1)
plt.imshow(mask,cmap="gray")
plt.show()

# Selection des minimum locaux
local_mini = skf.peak_local_max(255-mask, #il n'y a pas de fonction local_min...
                                indices=False)
markers1 = ndi.label(local_mini)[0]

labels = morpho.watershed(grad, markers1, watershed_line=True)
plt.imshow(couleurs_alea(labels))
plt.show()
# viewimage_color(couleurs_alea(labels)) - Utilisable si gimp 'est install

# visualization of the result
segm=labels.copy()
for i in range(segm.shape[0]):
    for j in range(segm.shape[1]):
        if segm[i,j] == 0:
            segm[i,j]=255
        else:
            segm[i,j]=0
#superimposition of segmentation contours on the original image
contourSup=np.maximum(segm,im)
plt.imshow(contourSup,cmap="gray")
plt.show()
```