

## TP de morphologie mathématique

Le but de ces travaux pratiques est d'expérimenter les principales opérations de morphologie mathématique pour en comprendre les effets.

### Avant de commencer le TP

Il faut exécuter les commandes

```
export PATH=/cal/softs/anaconda/anaconda3/bin:$PATH
source /tsi/tp/TP/Init/debut_tp_morpho_python.sh
spyder programmation/tp_morpho.py &
```

La première ligne vous permet d'utiliser python en version 3 distribué sous forme anaconda. Vous pouvez l'ajouter dans vos fichiers de configuration afin d'utiliser toujours cette version de python.

La seconde ligne crée un répertoire TPIMAMORPHO dans lequel se trouvera un lien symbolique vers le répertoire images. Vous serez automatiquement mis dans ce répertoire. Les images se trouvent dans le (lien symbolique vers le) répertoire images (dans lequel vous ne pouvez pas créer ou sauvegarder des images, créez-les dans votre répertoire de travail). Vous pouvez aussi bien sûr travailler sur les images de votre choix.

Vous trouverez dans le répertoire programmation un fichier tp\_morpho.py qui contient quelques commandes pour commencer sous python. Il faut d'emblée exécuter les SECTIONS 1 et 2 afin d'inclure les packages nécessaires au TP. La SECTION 2 définit des fonctions utiles pour ce TP. Ensuite la SECTION 3 contient les commandes correspondant aux différentes questions de ce TP. Ce sont des exemples pour vous aider à mener à bien le TP.

Vous pouvez utiliser autre chose que spyder, assurez-vous seulement que vous êtes dans le bon répertoire lorsque vous voulez accéder aux images. Les commandes python `pwd` et `cd` sont utiles pour cela.

### Elément structurant

L'élément structurant est défini par la fonction `strel`. Il est possible de choisir la forme : 'diamond', 'square', 'disk', 'line', et la taille. Par exemple

```
se = strel('square',11)
```

crée un élément structurant carré de taille  $11 \times 11$  pixels.

## 1 Morphologie mathématique sur des images à niveaux de gris

Les images suivantes peuvent être utilisées :

- bat200.bmp
- bulles.bmp
- cailloux.png
- cailloux2.png

— laiton.bmp  
— retina2.gif

Une image peut être lue et visualisée avec les commandes :

```
im=skio.imread('bat200.bmp')  
viewimage(im)
```

1. Tester les de dilatation, érosion, ouverture et fermeture avec différentes formes et tailles d'éléments structurants (fonctions `morpho.dilation`, `erosion`, `opening` ou `closing`<sup>1</sup>). Quelle est l'influence de la taille et de la forme de l'élément structurant ?
2. Vérifier les propriétés des quatre opérations sur des exemples.
3. A quoi est égale la succession d'une dilatation par un carré de taille  $3 \times 3$  et d'une dilatation par un carré de taille  $5 \times 5$  ? Même question pour l'ouverture. Quelles sont les propriétés de ces deux opérations qui expliquent ces résultats ?
4. Effectuer une transformation du chapeau haut-de-forme sur une image à niveaux de gris (image originale moins ouverture), par exemple sur l'image `retina2.gif`. Commenter le résultat pour différentes tailles et différentes formes d'éléments structurants. Quelle serait l'opération duale (illustrer sur l'image `laiton.bmp` par exemple) ?
5. Définir des segments dans plusieurs directions comme éléments structurants, et calculer le max des ouvertures obtenues pour chacun. Conclusion ?

## 2 Filtres alternés séquentiels

Réaliser un filtre alterné séquentiel, par exemple en utilisant la suite de commandes suivante :

```
se1=strel('disk',1)  
se2=strel('disk',2)  
se3=strel('disk',3)  
se4=strel('disk',4)  
se5=strel('disk',5)...  
fas1=morpho.closing(morpho.opening(im,se1),se1)  
fas2=morpho.closing(morpho.opening(fas1,se2),se2)  
fas3=morpho.closing(morpho.opening(fas2,se3),se3)  
fas4=morpho.closing(morpho.opening(fas3,se3),se3)  
fas5=morpho.closing(morpho.opening(fas4,se3),se3)...
```

Les tests doivent permettre de comprendre le type de résultat obtenu en fonction de la forme de l'élément structurant et de la taille maximale de l'élément structurant utilisé.

Quel comportement peut-on attendre si l'on poursuit la séquence avec un élément structurant de plus en plus grand ?

---

1. Nous utilisons les fonctions de morphologie mathématique incluses dans `skimage.morphology` que nous renomons `morpho` pour plus de clarté.

### 3 Reconstruction

La reconstruction permet de rendre les opérations robustes. La reconstruction par dilatation est obtenue par la séquence suivante ( $m$  est le marqueur,  $I$  l'image à reconstruire, ou masque, et  $B_1$  est un élément structurant élémentaire, de taille 1) :

$$\begin{aligned} m_0 &= m \wedge I \\ m_1 &= D(m_0, B_1) \wedge I \\ &\dots \\ m_i &= D(m_{i-1}, B_1) \wedge I \\ &\dots \end{aligned}$$

cette séquence étant itérée jusqu'à convergence (lorsque  $m_n = m_{n-1}$ ).

Par exemple la reconstruction d'une image à partir d'une ouverture permet de récupérer les petites parties des objets en partie préservés par l'ouverture. Tester par exemple (en changeant éventuellement l'élément structurant) :

```
im=skio.imread('retina2.gif');
se4 = strel('disk',4);
open4 = morpho.opening(im,se4);
reco=morpho.reconstruction(open4,im);
```

Commenter les résultats.

Ajouter une reconstruction à chaque étape du filtre alterné séquentiel (reconstruction par dilatation après chaque ouverture et reconstruction par érosion après chaque fermeture). Commenter.

### 4 Segmentation

1. Calculer le gradient morphologique (dilatation - érosion avec un élément structurant élémentaire, de taille 1), par exemple sur l'image `bat200.bmp`. Commenter.
2. Appliquer l'algorithme de ligne de partage des eaux sur l'image de gradient (voir la section correspondante dans le fichier d'exemples)
3. Commenter. Pour la visualisation, on peut sélectionner les lignes (points de valeur 0) et les superposer à l'image originale.
4. Essayer de filtrer l'image originale (par un filtre alterné séquentiel bien choisi par exemple) et/ou l'image de gradient (par une fermeture) avant d'appliquer la ligne de partage des eaux pour améliorer le résultat.
5. Eliminer les minima de dynamique inférieure à un certain seuil avant d'utiliser la ligne de partage des eaux.
6. Définir des marqueurs (à la main, ou par un pré-traitement), avec par exemple un marqueur dans l'objet à segmenter et un marqueur sur tout le bord de l'image (ou plusieurs marqueurs à l'extérieur de l'objet à segmenter). Soit  $m$  le marqueur, tel que  $m$  vaut 0 dans les zones marquées et 255 ailleurs. On calcule  $I' = I \wedge m$ , où  $I$  est l'image sur laquelle on veut appliquer la ligne de partage des eaux (image de gradient, ou image inversée dans le

cas de `laiton.bmp` par exemple). Puis on reconstruit  $I'$  par érosion à partir de  $m$ . L'image reconstruite n'a que des minima là où ils sont indiqués par  $m$ , et la ligne de partage des eaux peut alors être appliquée à cette image reconstruite. Commenter les résultats.

7. Comment pourrait on utiliser la ligne de partage des eaux (associée à d'autres étapes) pour segmenter les lignes noires dans l'image `bulles.bmp` ou `laiton.bmp`?