

IMA201 - 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

Copier les fichiers du TP sur votre compte. Le répertoire TP-morpho contient un code en python pour démarrer (tp_morpho.py), et un répertoire d'images. Vérifier que vous utilisez python 3. Sur les machines de l'Ecole, faire

```
export PATH=/cal/softs/anaconda/anaconda3/bin:$PATH
```

Cette commande 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.

Puis démarrer le code, par exemple avec spyder :

```
spyder tp_morpho.py &
```

Allez ensuite dans le répertoire Images. Vous pouvez aussi bien sûr travailler sur les images de votre choix.

Il faut d'emblée exécuter les SECTIONS 1 et 2 du code tp_morpho.py 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 11×11 pixels.

1 Morphologie mathématique binaire

Pour cette partie, utiliser l'image `cellbin.bmp` (ou `cafe.bmp`). L'image peut être lue et visualisée avec les commandes :

```
cell=skio.imread('cellbin.bmp')
```

```
plt.imshow(cell,cmap="gray"), ou viewimage(cell) si gimp est installé.
```

Tester les opérations de dilatation, érosion, ouverture et fermeture avec différentes formes et tailles d'éléments structurants (fonctions `morpho.dilation`, `erosion`, `opening` ou `closing`¹).

Commenter l'influence de la forme et de la taille de l'élément structurant sur les résultats.

2 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`

1. Tester les quatre opérations de base sur une des images à niveaux de gris. 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 3×3 et d'une dilatation par un carré de taille 5×5 ? 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. Pour aller plus loin, définir des segments dans plusieurs directions comme éléments structurants, et calculer le max des ouvertures obtenues pour chacun. Conclusion ?

3 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)...
```

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

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 ?

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. Essayer de seuiller le gradient, par exemple :

```
temp = grad  
temp[temp<50]=0
```

Pourquoi est-il difficile de trouver une bonne valeur de seuil ?

Dans la suite on utilisera le gradient original, avec éventuellement les petites valeurs mises à 0, les autres étant conservées.
3. Appliquer l'algorithme de ligne de partage des eaux sur l'image de gradient (voir la section correspondante dans le fichier d'exemples)
4. Commenter. Pour la visualisation, on peut sélectionner les lignes (points de valeur 0) et les superposer à l'image originale.
5. 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 Pour aller plus loin : 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.

La reconstruction par érosion s'obtient par dualité et peut être utilisée pour imposer des marqueurs à la ligne de partage des eaux. Soit m le marqueur, tel que m vaut 0 dans les zones à marquer (que l'on veut segmenter) 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. Cette idée pourra être testée si le temps le permet. Utiliser la propriété de dualité pour effectuer une reconstruction par érosion (la fonction `reconstruction` fait une reconstruction par dilatation, mais on peut la paramétrer pour utiliser une érosion).