

Traitement d'images

Projet – Coder la ligne de partage des eaux

Ce projet consiste à coder, en Matlab, la ligne de partage des eaux en utilisant vos propres fonctions (**pas le droit d'utiliser les fonctions du package image de matlab**). Dans le fichier zip contenant cet énoncé, vous trouverez les fonctions **mydilate** et **mystrel** : la première réalise une dilatation d'une image en niveau de gris, et la seconde permet de générer des éléments structurants.

Vous devrez remettre votre code ainsi qu'un rapport (correctement présenté, expliquant chacune des étapes de votre travail et les difficultés rencontrées) avant le 12 Janvier à minuit (fin de journée). Vous devrez envoyer tout cela par mail à l'adresse mail indiquée sur le polycopié de cours. Une pénalité de -5 points par jour de retard sera appliquée. Le rapport vaudra pour environ la moitié de la note du projet.

Partie 1 : Coder une érosion

Dans cette première partie, vous devez écrire la fonction **myerode**, qui réalise l'érosion d'une image en niveau de gris (attention, ce n'est pas du tout le même code que la fonction réalisée en TP qui fonctionne sur des images binaires seulement).

Indice : n'oubliez pas que l'on peut facilement, en passant par le dual, réaliser une érosion à partir d'une dilatation.

Test des résultats : l'image *test_erode.png* est le résultat de l'érosion de *chien.png* par un élément structurant de type 'diamond' et de taille 6.

Partie 2 : Coder un gradient morphologique

Maintenant, vous devez réaliser la fonction **mygrad**, qui calcule le gradient morphologique d'une image à partir d'un élément structurant donné en paramètre.

Test des résultats : l'image *test_gradient.png* est le résultat du gradient morphologique de *chien.png* par un élément structurant de type 'square' et de taille 3.

Partie 3 : Etiquetage des composantes connexes

Ecrivez une fonction **myreconstructbin** qui prend en paramètre une image binaire *Im*, une image de marqueurs *M* et un élément structurant *EI*, et réalise la reconstruction inférieure binaire de *M* dans *Im* à l'aide de l'élément structurant *EI* (en utilisant la dilatation binaire **mydilatebin**). Puis écrivez la fonction **mylabel** qui étiquette chaque composante connexe d'une image binaire (et qui utilisera **myreconstructbin**).

Test des résultats : testez votre fonction d'étiquetage sur l'image *tools.png*.

Partie 4 : Coder une reconstruction inférieure

Maintenant, écrivez une fonction **myreconstruct**, qui prend en paramètre une image *Im*, une image de marqueurs *M* et un élément structurant *El*, et réalise la reconstruction inférieure de *M* dans *Im* à l'aide de l'élément structurant *El*.

Indice : En matlab, la fonction `numel(find(A~=B))` renvoie combien d'éléments sont différents entre les images A et B... Pratique pour tester si deux images sont égales ou non.

Test des résultats : l'image *test_myreconstruct.png* est le résultat de la reconstruction inférieure de *chien.png* à partir des marqueurs de l'image *chien_marqueurs.png*, par un élément structurant de type 'square' et de taille 3.

Trop lent ? Vous vous rendrez compte, après vos tests, que la reconstruction que vous avez codé est certainement très lente... Pour la rendre plus rapide, voici une possibilité : utilisez la fonction **mydilate_quick** pour réaliser la dilatation, et ne dilatez, à chaque tour, que les pixels du résultat qui ont changé par rapport au tour précédent (**mydilate_quick** est rapide s'il y a peu de pixels à dilater).

Partie 5 : La ligne de partage des eaux

Il est temps d'écrire la fonction **mywatershed**, qui prend en paramètre une image *Im*, une image de marqueurs *M* et un élément structurant *El*. Voici l'algorithme que vous devrez implémenter :

1. Calculez le gradient *Grad* de l'image *Im*.
2. On veut forcer les minimums globaux de l'image *Grad* à être les pixels de *M*. Pour ce faire, on doit modifier l'image *Grad* : on met les pixels de *M* à 0 dans l'image *Grad*. Ensuite, on inverse l'image *Grad* et on effectue une reconstruction inférieure de *M* dans l'image inversée. On inverse le résultat et on obtient *Grad2*.
3. On étiquette, dans *L*, chaque composante connexe de *M*. On aura donc, dans *L*, tous les pixels d'une même composante connexe de *M* à une même valeur.
4. Pour chaque seuil *s* de l'image *Grad2* (en partant de 0)
 5. On pose *G* comme l'ensemble des pixels de *Grad2* inférieurs ou égaux à *s*.
 6. Pour chaque valeur *v*>0 dans l'image *L*
 7. On pose *K* comme l'ensemble des pixels de *L* égaux à *v*
 8. On réalise la reconstruction inférieure binaire de *K* dans l'image *G*, et on ajoute le résultat aux pixels de *L* (en leur donnant comme valeur *v*)
9. Le résultat est *L*.

Test des résultats : l'image *test_watershed.png* est le résultat de la ligne de partage des eaux de *flower.png* à partir des marqueurs de l'image *flower_marqueurs.png*, par un élément structurant de type 'square' et de taille 3.

Testez votre algorithme aussi sur l'image *cells.png* à partir de l'image *cells_marqueurs.png*.