

TP3 : Morphologie par reconstruction

1 Coder une reconstruction inférieure et supérieure

1. Vous devez écrire la fonction **mycondilata**, qui prend en paramètre une image M , un élément structurant E et une image I , et renvoie la dilatation conditionnelle de M par l'élément E restreinte à I .
2. Vous devez écrire la fonction **myconderode**, qui prend en paramètre une image M , un élément structurant E et une image I , et renvoie l'érosion conditionnelle de M par l'élément E restreinte à I .
3. Vous devez écrire la fonction **myreconinf**, qui prend en paramètre une image M , un élément structurant E et une image I , et renvoie la reconstruction inférieure, par l'élément structurant E , de l'image I à partir du masque M .

L'image *result_reconinf.png* est le résultat de la reconstruction inférieure de *chien.png* par un élément structurant de type *cercle* de rayon 5, par le masque *mask_inf.png*.

4. Vous devez écrire la fonction **myreconsup**, qui prend en paramètre une image M , un élément structurant E et une image I , et renvoie la reconstruction supérieure, par l'élément structurant E , de l'image I à partir du masque M .

L'image *result_reconsup.png* est le résultat de la reconstruction supérieure de *chien.png* par un élément structurant de type *cercle* de rayon 5, par le masque *mask_sup.png*.

2 Coder une ouverture et une fermeture par reconstruction

Vous devez maintenant coder une ouverture par reconstruction, puis une fermeture par reconstruction. Les images *result_opencond.png* et *result_closecond.png* sont respectivement les résultats de l'ouverture par reconstruction et de la fermeture par reconstruction de *chien.png* par un élément structurant de type *cercle* et de taille 20, reconstruites par Γ_8 .

3 Plus loin avec les grains de riz

Nous continuons de travailler sur l'image *rice.png*, et plus particulièrement sur l'image seuillée et nettoyée obtenue lors de la dernière séance.

1. Choisissez un pixel p de l'image appartenant à un grain de riz, et construisez une image vide M où tous les pixels sont à 0, sauf p qui vaudra 255.
2. Effectuez une reconstruction inférieure de l'image I , par l'élément structurant Γ_8 , à partir de M .
3. En interprétant le résultat obtenu, écrivez un programme permettant de compter le nombre de grains de riz sur l'image.

Nous allons maintenant effectuer des calculs plus avancés pour caractériser les grains de riz.

1. Commencez par supprimer tous les grains de riz touchant le bord de l'image.
2. Pour chaque grain de riz restant, affichez la largeur du grain de riz ainsi que sa longueur (approximative).

3. Générez une image couleur où apparaissent en bleu les grains de riz les plus courts, et en rouge ceux les plus longs.

Indice : `cv2.COLORMAP_JET` est une carte de couleur (colormap) qui devrait correspondre à ce que l'on souhaite obtenir. Voir <http://docs.opencv.org/2.4/modules/contrib/doc/facerec/colormaps.html>

4 La tête dans les étoiles

Dans l'image *etoile.png*, on souhaite compter le nombre d'étoiles présentes. On définit une étoile comme un objet d'au moins un pixel dont la valeur de luminosité est de 50 niveaux de gris supérieur à celle de son entourage.

Quelle transformation appliquer ici à l'image afin d'en extraire les étoiles, étant donné que l'on se base ici sur la hauteur des éléments, et non sur leur forme? Comptez le nombre d'étoiles de l'image, et représentez chaque étoile par une croix rouge (un petit signe plus).

Votre approche fonctionne-t-elle avec l'image *comete.jpg*?