

TD 4 UE Vision : Morphologie et filtre médian

1) L'élément structurant

La transformation morphologique modifie la valeur d'un pixel de l'image en fonction de la valeur de ses voisins dans l'élément structurant. Charger l'image «[bureau256.png](#)». L'élément structurant B est un masque binaire ; il est défini par la fonction suivante : `>> B = strel('square',3);` Tapez B pour le visualiser.

On peut choisir différentes formes : '[diamond](#)', '[square](#)', '[line](#)', et modifier sa taille. Ici, l'élément structurant est un carré de taille 3×3 pixels. On peut également fabriquer son masque soi-même, par exemple le disque 5×5 : `>> B = [0 0 1 0 0; 0 1 1 1 0; 1 1 1 1 1; 0 1 1 1 0; 0 0 1 0 0];`

2) Dilatation / Erosion

Les commandes suivantes permettent d'appliquer à une image I une **dilatation** notée δ ou une **érosion** ε .

`>> IM2 = imdilate(I, B); figure, imagesc(IM2); colormap(gray), title('Image dilatee');`

`>> IM3 = imerode(I, B); figure, imagesc(IM3); colormap(gray), title('Image erodee');`

Tester différents éléments structurants pour vous familiariser et observer les résultats qui en découlent.

Que constatez-vous si vous utilisez l'élément structurant : `>> B = strel('diamond',1);` ?

Que faut-il appliquer afin de faire disparaître les barreaux horizontaux des fenêtres sur l'image du bureau ?

3) Gradient Morphologique

Le gradient morphologique correspond généralement à la différence entre l'image dilatée/érodée et I . Utiliser la valeur absolue et un seuil puis afficher les contours en vert sur l'image originale I .

4) Ouverture/Fermeture

L'ouverture est composée d'une érosion puis d'une dilatation. La fermeture est composée d'une dilatation puis d'une érosion. Ces commandes existent sous Octave/Matlab :

`>> imo = imopen(imb, B);` et `>> imf = imclose(imb, B);`

Testez différents éléments structurants pour faire disparaître les barreaux de l'image du bureau.

5) White Top Hat

Pour extraire les barreaux horizontaux, calculer le *white top hat* par : « [originale](#) – [ouverture](#) », seuiller et afficher en vert sur l'image originale (utiliser l'élément structurant en 1)).

6) Rehaussement de contraste

Charger l'image du bureau et appliquer la méthode rehaussement de contraste suivante :

où B représente l'élément structurant (cf. 1)).

Comparer le résultat avec la méthode proposée ci-dessous (en choisissant correctement les valeurs (α, β)) :

$$\begin{cases} \chi_B(x,y) = \varepsilon_B(x,y) & \text{si} & \varepsilon_B(x,y) \leq I(x,y) \leq \varepsilon_B(x,y) + \alpha (\delta_B(x,y) - \varepsilon_B(x,y)) \\ \chi_B(x,y) = I(x,y) & \text{si} & \varepsilon_B(x,y) + \alpha (\delta_B(x,y) - \varepsilon_B(x,y)) < I(x,y) < \delta_B(x,y) - \beta (\delta_B(x,y) - \varepsilon_B(x,y)) \\ \chi_B(x,y) = \delta_B(x,y) & \text{si} & \delta_B(x,y) - \beta (\delta_B(x,y) - \varepsilon_B(x,y)) \leq I(x,y) \leq \varepsilon_B(x,y) \end{cases}$$

Avec $0 \leq \alpha, 0 \leq \beta, (\alpha + \beta) < 1$.

Vous pouvez comparer avec le filtre non linéaire du TD1, question 10).

7) Squelettisation

Charger l'image « [pieces.png](#) » et regarder sur quel plan couleur les pièces sont les plus visibles. Choisir ainsi ce plan ; après avoir converti en *double*, seuiller le plan choisi pour ne faire apparaître quasiment que les pièces appelée I_s . On peut appliquer une *dilatation* pour boucher quelques trous ou utiliser « *imfill* ».

Utiliser la fonction « *bwmorph(mon_plan_choisi, 'skel', Inf)* » pour squelettiser l'image. Que se passe-t-il si l'on change le paramètre « Inf » par une valeur (entière pas très grande) ?

8) Erosion ultime

Sur I_s , appliquer la méthode de l'érosion ultime afin de déterminer la taille des objets. A la suite de chaque érosion, on applique une ouverture que l'on somme au résultat précédent, (cf. fig. ci-dessous).

Les objets les plus gros apparaîtront en plus clairs avec un pic au milieu de l'objet. On peut ainsi déterminer leur taille/rayon. Seuiller le résultat pour ne faire apparaître qu'un point en vert sur les pièces de deux euros.

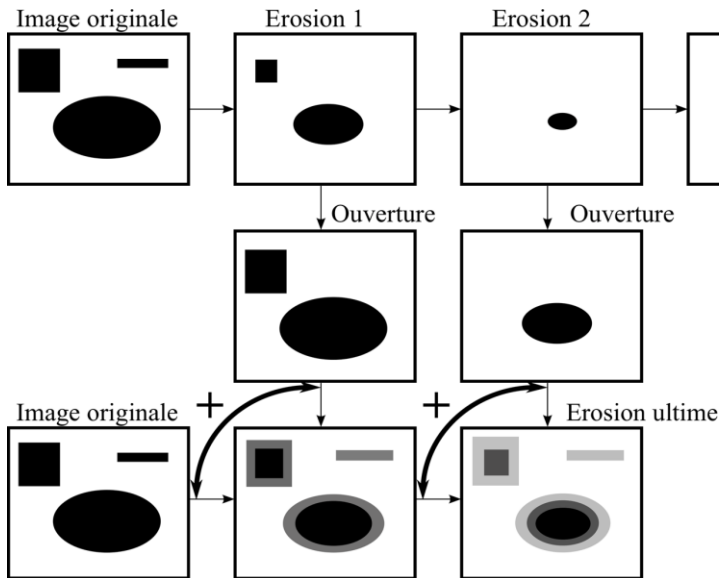


Figure : Schéma de l'érosion ultime (illustration avec image inverse)

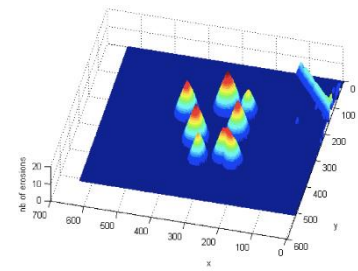


Figure : Erosion ultime des pièces



Figure : Centre des pièces détectés

9) Image bruitée

Chargez l'image « [bureau256.png](#) » que l'on nommera J puis la visualiser. Ajouter du bruit « poivre et sel » :
`>> imb=double(imnoise(J, 'salt & pepper', 0.1)) ; %bruit poivre et sel`

10) Filtres alternés séquentiels

Pour réaliser un filtre alterné séquentiel, il suffit de créer une ouverture puis une fermeture avec un élément structurant B donné puis répéter le processus avec un élément structurant plus grand. Cela fait disparaître les petits éléments dans l'image, donc il est possible de restaurer une image de cette façon. Créer une boucle qui applique 10 filtres alternés séquentiels avec des éléments structurants de taille croissante : 1 à 10 sur l'image imb . Utiliser l'option 'square' puis créez un disque pour l'élément structurant, que remarquez-vous ?

11) Filtre median

Regarder l'aide de la fonction « [medfilt2](#) » en tapant « `>> help medfilt2` » puis appliquez cette fonction à « imb ». Créer une boucle en appliquant le filtre plusieurs fois avec une taille croissante sur le résultat précédent (avec une taille augmentant de 1 pour chaque nouveau filtre). Comparez le résultat final avec celui utilisant seulement la taille finale du filtre (i.e., sans itération).

12) Rapport signal sur bruit

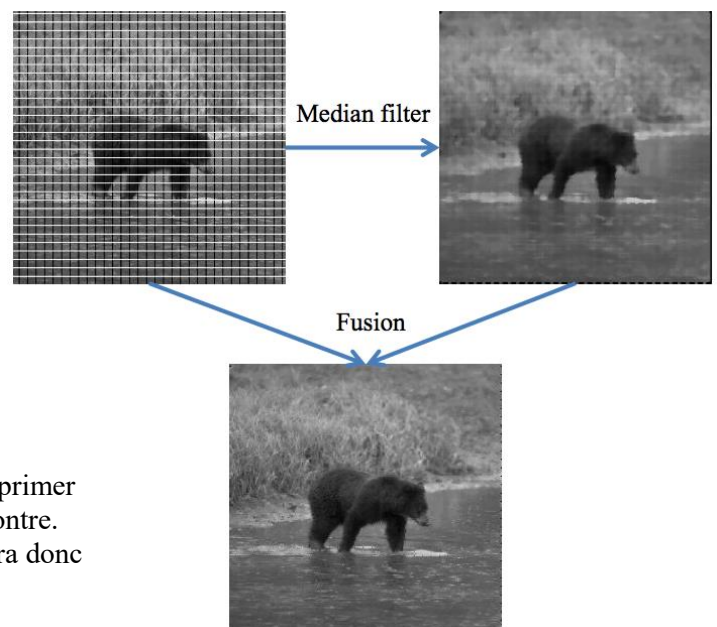
Coder la fonction « [f_SNR.m](#) » qui calculer le SNR entre deux images I et J de même taille :

$$SNR(I, J) = 10 \cdot \log_{10} \frac{\sum_{i=1}^M \sum_{j=1}^N I^2(i, j)}{\sum_{i=1}^M \sum_{j=1}^N (J(i, j) - I(i, j))^2}$$

où M représente le nombre de ligne de I ,
 N représente le nombre de colonnes de I ,
la fonction logarithme s'écrit « [log10](#) ».

13) Comparaison de débruitage

Bruiter une image appeler lb , calculer le SNR entre I et lb . Débruiter avec le filtre médian. Tester différents bruits et niveaux. Enfin, comparez les valeurs de SNR avec différents paramètres du filtre médian ?



14) Inpainting d'image

Charger l'image « [pentagon_bar.png](#) ».

Trouvez une technique à l'aide du filtre médian pour supprimer les barres blanches et noires comme dans l'exemple ci-contre. Les autres pixels ne doivent pas être changés (on utilisera donc des images intermédiaires).