Traitement d'images : TP2

Si vous utilisez Octave, les commandes restent les mêmes. Il faut juste mettre la commande `pkg load image` au début de votre code pour charger le package image.

Certaines fonctions de traitement d'images matlab ne peuvent être exécutées que sur des images dont les valeurs sont des réels entre 0 et 1. Avant d'appliquer une telle fonction, il faut donc vérifier que la matrice représentant l'image contient bien de telles valeurs et, si besoin, modifier le codage.

Certaines fonctions ne peuvent être exécutées que sur des images en niveaux de gris. Avant d'appliquer une telle fonction, il faut donc vérifier que la matrice représente bien une image en niveaux de gris et, au besoin, effectuer une conversion.

Terminer les exercices du TP précedent.

Exercice 1 Etirement et égalisation d'histogramme

1. Ecrivez une fonction effectuant un étirement d'histogramme qui fonctionne sur une image en niveaux de gris, codée avec des réels entre 0 et 1.
2. La fonction permettant d'effectuer une égalisation d'histogramme est la fonction histeq. Elle fonctionne sur des réels codés entre 0 et 1.
3. Application à l'amélioration d'images
   - Chargez et affichez les images image2.bmp, image3.jpg et image4.jpg
   - Convertissez-les en images en niveaux de gris. Vérifiez que l'image est codée avec des réels entre 0 et 1. Si ce n'est pas le cas, transformez l'image pour qu'elle soit codée avec des réels entre 0 et 1.
   - Affichez les histogrammes des images converties.
   - Effectuez sur chacune d'elles un étirement et une égalisation d'histogramme.
   - Affichez les images obtenues par ces opérations ainsi que leurs histogrammes.

Exercice 2 Fonction FIND

L'instruction `[X,Y] = find(condition)` crée deux matrices X et Y contenant respectivement les abscisses et les ordonnées de tous les pixels de I dont la valeur satisfait la condition.
Par exemple, `[X,Y]=find(I==5)` trouve les coordonnées de tous les pixels dont la valeur dans I est 5.

1. Trouver, avec la fonction `find`, les coordonnées de tous les pixels blancs de l'image binaire image_bin.bmp. En déduire leur nombre.

Attention pour les exercices suivants, les fonctions `filter2`, `medfilt2` et `im2bw` ne fonctionnent que sur des images en niveaux de gris dont les valeurs des pixels sont des réels entre 0 et 1. Avant d'appliquer l'une de ces fonctions sur une matrice image, vérifiez que celle-ci ne représente pas une image couleur ou une image indexée et que les valeurs sont bien des réels entre 0 et 1. Si ce n'est pas le cas, effectuez les conversions nécessaires.

Exercice 3 Filtrage

Dans une image, les basses fréquences correspondent à des variations lentes de l'intensité (zones uniformes), et les hautes fréquences à des variations rapides (contours, bruits). Le filtrage d'une image par un filtre passe-bas donne une image floue, c'est à dire une image possédant peu de hautes fréquences (les contours sont adoucis par le filtre). D'une manière générale, un passe-haut accentue les contours et le bruit, tandis qu'un passe-bas réduit le bruit et adoucit les contours.

1. Chargez l'image `house.jpg` et testez les filtres ci-dessous. Notez à chaque fois l'effet produit par le filtrage sur les zones uniformes et sur les zones non uniformes. Expliquez les résultats obtenus.

$$f1 = \frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad f2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad f3 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

2. Comparez les résultats obtenus par le filtre f1 et les filtres f12, f13 et f14 ci-dessous :

$$f12 = \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad f13 = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$f14 = \frac{1}{65} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Exercice 4 Ajout de bruit sur une image

La fonction `imnoise` permet d'ajouter du bruit sur une image. Regardez l'aide matlab pour voir les différents paramètres.

— chargez l'image `house.jpg` et faites les conversions pour avoir des valeurs réelles entre 0 et 1.

— ajoutez du bruit poivre et sel sur l'image initiale. Affichez l'image obtenue.

— ajoutez du bruit gaussien sur l'image initiale et faites varier les paramètres de moyenne et variance du bruit. Affichez les images obtenues.

Exercice 5 Atténuation du bruit par filtrage

L'image `house_gaus.bmp` est bruitée par un bruit dit gaussien. L'image `house_sap.bmp` a été bruitée par un filtre dit poivre et sel. Appliquez sur les deux images un filtre passe-bas (f1) et un filtre médian (fontion `medfilt2`). Sur quel type de bruit sont plus efficaces ces filtres ? Quels sont les inconvénients du filtrage sur le reste de l'image ?