

Traitement et Synthèse d'Image – S7

TP1 – Filtrage

Marion Foare

Objectifs.

- Filtrage dans l'espace direct et dans l'espace de Fourier
- Implémentation de l'algorithme des K-means

Déroulement. Ce TP est à effectuer par binôme sous Matlab/Octave. Vous trouverez sur CPe-campus une archive contenant l'ensemble des fichiers nécessaires à la réalisation de ce TP.

Configuration. Ce TP nécessite les librairies suivantes :

- Matlab : *Image Processing Toolbox*

- Octave : *Image toolbox*
puis, en début de script :

```
pkg install -forge image  
pkg load image
```

Évaluation en séance. Vous expliquerez à l'intervenant :

- votre démarche (algorithme, équations sur lesquelles vous vous êtes appuyés, etc.),
- le choix et l'influence de vos paramètres,
- les résultats obtenus avec votre interprétation.

Nous rappelons que toute tentative de copie entraînera une sanction de l'ensemble des binômes concernés.

1 Préparation

- 1 – Rappeler les fonctions Matlab permettant :
 - de lire une image
 - de filtrer une image
 - d'afficher une image (plusieurs fonctions existent, préciser pour chacune ses spécificités, notamment en terme de gestion de la dynamique)
- 2 – A quoi sert la fonction `im2double` ? Pourquoi est-elle indispensable lorsqu'on manipule des images ?
- 3 – Quelles lignes de code permettent de calculer :
 - le gradient d'une image I
 - la norme du gradient de I
 - le gradient normalisé de I
- 4 – Que fait la fonction `meshgrid` ?

2 Filtrage coupe-bande dans l'espace de Fourier

On souhaite supprimer la trame d'une image. Pour cela, on implémente un filtre de Butterworth coupe-bande, dont on rappelle ci-dessous le gain :

$$H[u, v] = \frac{1}{1 + \left(\frac{B\sqrt{u^2 + v^2}}{u^2 + v^2 - n_c^2} \right)^{2p}}$$

On pourra par exemple prendre pour paramètres du filtre $p = 2$, $n_c = 100$ et $B = 120$ la largeur de bande.

- 1 – Lire et afficher l'image 'journal.png'. Obtenir la taille $[h, w]$ de l'image.
- 2 – Génération du filtre de Butterworth :
 - En utilisant la fonction `meshgrid`, obtenir deux matrices donnant respectivement les coordonnées U et V de chaque pixel. On fera en sorte que le pixel au centre de l'image ait pour coordonnées $(0,0)$:
`[U V] = meshgrid(-w/2+1/2:w/2-1/2, -h/2+1/2:h/2-1/2)`
 - À partir de U et V , obtenir la matrice D , donnant la distance au centre pour chaque pixel de l'image.
 - À partir de D , en déduire le gain H de Butterworth (sans utiliser de boucle). On pourra afficher le filtre en 2D sous forme d'image.
- 3 – Calculer la transformée de Fourier 2D de l'image en utilisant la fonction `fft2`. On n'oubliera pas, après avoir effectué la FFT de l'image, de réarranger les données afin que la fréquence nulle se trouve au centre de l'image (`fftshift`). Afficher le module de la transformée de Fourier.
- 4 – Filtrer l'image par H dans le domaine fréquentiel.
- 5 – Revenir à l'espace direct par transformée de Fourier inverse 2D et afficher l'image filtrée.
- 6 – Appliquer ce filtre à l'image qui vous aura été désignée, en ajustant les paramètres au besoin.

3 Filtrage passe-haut dans l'espace direct : détection de contours

On cherche à détecter les contours de l'image 'flower.png' à l'aide du filtre de Sobel.

- 1 – Décomposer le filtre de Sobel en produit (matriciel) de deux filtres 1D, dont un filtre de lissage et un filtre dérivatif. Pourquoi a-t-on, dans les filtres de détection de contours, une partie 'lissage' ?
- 2 – Calculer les composantes horizontale et verticale du gradient en chaque pixel de l'image, à l'aide du filtre de Sobel et de la fonction de filtrage de Matlab. On enregistrera la composante verticale du gradient de chaque pixel dans la matrice G_v et la composante horizontale dans le tableau G_h (sans utiliser de boucle).
- 3 – En déduire la norme G du gradient en chaque pixel. Afficher sur une même figure, G , G_v et G_h .
- 4 – Reprendre les questions précédentes en ayant ajouté à l'image de départ un bruit blanc Gaussien centré d'écart-type 0.1 en utilisant la fonction `randn`. Qu'observez vous ?

4 Filtrage linéaire vs non linéaire : débruitage

On souhaite comparer les performances de débruitage de plusieurs filtres en présence de différents types de bruit. Pour cela, on peut par exemple utiliser l'Erreur Quadratique Moyenne (MSE), qui permet de quantifier objectivement la qualité d'une estimation en sortie de filtre. Étant donnée l'image bruitée I de taille $h \times w$, et sa version débruitée \hat{I} en sortie de filtre, la MSE est définie par :

$$MSE = \frac{1}{hw} \sum_{i=1}^h \sum_{j=1}^w [I(i, j) - \hat{I}(i, j)]^2$$

Dans cet exercice, on travaillera sur l'image 'flower.png', ou tout autre image de votre choix.

- 1 – Générer les deux images bruitées suivantes :
 - I_g correspondant à l'image originale dégradée par un bruit blanc Gaussien d'écart-type $\sigma = 0.1$, en utilisant la fonction `randn`;
 - I_{sp} correspondant à l'image originale dégradée par un bruit poivre et sel de densité 0.5, en utilisant la commande Matlab `imnoise`.
- 2 – Comparer qualitativement (ie : en observant les images résultantes) les performances de débruitage de chacun des filtres suivants :
 - filtre moyenneur de taille n ,
 - filtre de Butterworth passe-bas d'ordre $p = 5$ et d'indice de coupure n_c (cf exercice 2),
 - filtre médian de taille n .

Pour cela on testera, pour chaque filtre, différentes valeurs du paramètre de ce filtre (n ou n_c).
- 3 – Comparer quantitativement les performances de chacun des filtres, en traçant la MSE en fonction de différentes valeurs du paramètre du filtre (n ou n_c).

Préparation pour le TP2

5 Filtrage à partir d'histogramme : seuillage par K -means

- 1 – Lire et afficher l'image 'flower.png'.
- 2 – Écrire une fonction implémentant l'algorithme de K -means à K régions :
 - Choisir aléatoirement K intensités m_i .
 - Assigner à chaque pixel le label 1 si son intensité est plus proche de la valeur m_1 , le label 2 si son intensité est plus proche de la valeur m_2 , ..., le label K si son intensité est plus proche de la valeur m_K . On stockera ces labels dans la matrice `labels`.
 - Mettre à jour les valeurs de m_i en prenant la moyenne des intensités des pixels de label k .
 - Itérer jusqu'à ce que les moyennes m_i ne changent plus.
- 3 – Afficher l'image segmentée en 2 régions avec l'algorithme des K -means. Comment pourrait-on remonter à un seuil à partir de cet algorithme?
- 4 – Discuter du choix de K pour la segmentation de l'image 'flower.png'.