

INF8725 - Traitement de signaux et d'images

TP2 - Filtrage d'image dans le domaine spatial et fréquentiel

Automne 2016

Chargé de laboratoire : Hamza Bendaoudi

Objectifs :

Ce laboratoire a pour objectif de manipuler et analyser les images dans le domaine spatial et fréquentiel.

Documents à remettre :

Les exercices doivent être codés dans un fichier TP.m. Les réponses aux questions doivent être incluses dans le code. Les exercices doivent être séparés par des cellules (*Insert cell divider* ou `%%`). Vous devez bien identifier chaque exercice et sous-question, et bien commenter le code.

Créer un fichier html à l'aide de *Publish to html* de Matlab pour avoir un fichier html de votre code et de vos graphiques. Veuillez remettre tous vos fichiers (.m et dossier html) dans un seul fichier zip et nommez ce fichier selon vos matricules (Mat1_Mat2.zip).

Pour inclure les fonctions dans le html, ajouter 'type fonction.m' dans votre .m principal. Vérifier également que les graphiques et les figures sont lisibles dans le html.

Une pénalité de 3 points sera appliquée si ces consignes ne sont pas respectées.

Notice

Lors de la conception de vos algorithmes, portez attention aux types de vos données (uint8, double, etc.). Lors de la manipulation des images, il sera fort probablement nécessaire de passer en double pour faire vos calculs puis de revenir en uint8 pour afficher vos images.

1 Première partie : Filtrage spatial

Exercice 1 (5 points) : Rehaussement d'image

Vous trouvez une vieille bande vidéo d'un film classique. Malheureusement, celle-ci est détériorée. Vous voulez la restaurer, mais vous n'avez pas beaucoup de temps. Alors, vous songez à faire un rehaussement simple avec quelques filtres de bases.

1. (1.5 points) Chargez l'image *img1.jpg* avec la fonction *imread*. Cette image est trop sombre. Implémentez la fonction *Egalisation_Histogramme* qui applique une égalisation d'histogramme à une image. Cette fonction prend en entrée une image et retourne l'image égalisée. Ne pas utiliser la fonction *histeq* de Matlab. Appliquez-la à votre image et affichez le résultat.
2. (1 point) L'image égalisée semble très bruitée. Implémentez la fonction *Filtre_Gaussien* qui applique un filtre gaussien sur une image. Cette fonction prend en entrée l'image à filtrer et un masque gaussien, puis retourne l'image filtrée. Ne pas utiliser la fonction *fspecial* pour générer un masque gaussien. Utilisez la fonction *conv2* pour effectuer la convolution entre l'image et le masque (utilisez le paramètre 'same' pour obtenir un résultat de même taille que l'image d'entrée). Appliquez cette fonction à l'image égalisée avec le masque gaussien 5x5 suivant et affichez le résultat :

$$Masque\ Gaussien = \frac{1}{90} * \begin{bmatrix} 1 & 2 & 1 & 2 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 8 & 18 & 8 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 1 & 2 & 1 \end{bmatrix} \quad (1)$$

3. (1 point) Bien que vous ayez enlevé l'ensemble du bruit, les contours de l'image vous semblent dégradés. Vous pensez donc à l'algorithme de rehaussement de contours que vous avez vu en cours.

L'algorithme de rehaussement de contour est le suivant :

$$\begin{aligned} I_g &= G * I \\ I_r &= I_g + K \cdot \Delta^2 I_g \end{aligned} \quad (2)$$

Où G est un masque gaussien.

Vous aurez besoin du filtre Laplacien pour cet algorithme. Implémentez tout d'abord la fonction *Filtre_Laplacien*. Celle-ci prend en entrée une image puis retourne l'image filtrée. Le masque laplacien est donnée ci-dessous. Ne pas utiliser la fonction *fspecial* pour générer le masque Laplacien. Attention, ici il ne faut pas reconvertir l'image en uint8 car vous allez perdre toutes les valeurs négatives. Pour afficher le laplacien, vous devez faire une mise à l'échelle de votre image.

$$Masque\ Laplacien = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3)$$

4. (1 point) Implémentez la fonction *Rehaussement_Contour* qui prend en entrée une image, un masque gaussien et le paramètre K , puis retourne l'image rehaussée. Appliquez cette fonction à l'image dont les contours ont été dégradés en utilisant le masque gaussien 3x3 ci-dessous et un paramètre $K = 1.2$. Affichez le Laplacien et le résultat du rehaussement.

$$\text{Masque Gaussien} = \frac{1}{16} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

5. (0.5 point) Que se passe-t-il avec l'image rehaussée ? Pourquoi ?

Exercice 2 (5 points) : Compteur de monnaie

On vous a confié la tâche de réaliser un compteur de monnaie. Puisque la monnaie canadienne est essentiellement ronde, vous vous dites que ce serait l'occasion parfaite pour utiliser vos notions de filtrage morphologique.

1. (0.5 point) Chargez l'image *monnaie.png*. Affichez cette image.
2. (1 point) Implémentez une fonction *Binariser* qui met à 0 tous les pixels se trouvant en dessous du seuil, et à 255 tous les pixels se trouvant sur le seuil ou au dessus. Cette fonction prend en entrée une image et un seuil, puis retourne l'image binaire. Ne pas utiliser la fonction *im2bw*. Binarisez votre image avec un seuil égale à 250. Bien sûr, puisque nous voulons que ce soit les pièces qui soient blanches, vous devez inverser la couleur. Affichez l'image binaire.
3. (1.5 points) Vos pièces ne sont pas totalement pleines. Effectuez une fermeture sur l'image binaire afin de fermer les trous, tout en ne changeant pas trop la taille des pièces. Vous devez choisir un bon élément structurant. Affichez le résultat.
4. (2 points) Implémentez la fonction *Compter_Monnaie* qui compte le total de la monnaie de votre image binaire à l'aide des opérateurs morphologiques *imerode* ou *imdilate*. Voici des rayons de disques qui pourront vous être utiles : 67, 64, 60, 55, 45, 40. Aussi, vous pouvez utiliser la fonction *bwlabel* pour compter le nombre de composantes connexes dans l'image.

2 Deuxième partie : Filtrage spectral

Exercice 3 (3.5 points) : Transformée de Fourier 2D

1. (0.5 point) Chargez les images *Barres_Verticales.png*, *Barres_Horizontales.png* et *Barres_Obliques.png*, puis affichez les.
2. (1.5 points) Calculez les TFD des images avec la fonction *fft2*. N'oubliez pas de récupérer la composante réelle de la TFD avec *abs* et de normaliser vos valeurs. Utilisez ensuite la fonction *fftshift* pour centrer votre TFD. Pour des fins de visualisation, utiliser $\log(1 + \text{TFD}(\text{IM}))$ pour que les valeurs extrêmes n'empêchent pas la visualisation des autres valeurs. Aussi, il est fortement conseillé de forcer *imshow* à prendre les valeurs min et max de votre image comme noir et blanc, au lieu de 0 et 1 dans le cas des images de types double, ce qui ne correspond pas forcément aux valeurs de votre TFD. Pour forcer *imshow*, il faut lui passer une matrice vide en deuxième paramètre (ou une matrice contenant [min max]) pour lui dire de prendre le min et le max de votre image comme noir et blanc.
3. (0.5 points) Tournez l'image *Barres_Verticales.png* de 70 degrés avec l'aide de *imrotate*. Utilisez l'option *bilinear* et *crop* de la fonction *imrotate*. Calculez et affichez la TFD de l'image tournée.
4. (1 point) Quelles propriétés de la transformée de Fourier 2D pouvez-vous inférer de ces résultats ?

Exercice 4 (6.5 points) : Filtrage spectral

1. (0.5 point) Chargez l'image *NewHouse.png*, convertir l'image en grayscale, calculez sa TFD puis affichez les. Attention, ici il faut conserver la phase de la TFD, donc ne pas conserver uniquement *abs(TFD)*. Appliquez *abs* seulement pour l'affichage.
2. (1 point) Sur le spectre de l'image originale, identifiez sur un schéma à quelle texture (porte, toit, fenetre, cheminée) correspondent les différents pics (vous pouvez faire des tests avec l'interface *signal2D* pour vous faire une idée).

3. (1 point) Créez un filtre passe-bas gaussien dans le domaine **spectral** avec la fonction *fspecial* pour récupérer uniquement la texture du toit. La taille de votre filtre doit être de la même taille que l'image. Vous devez jouer avec le paramètre sigma pour obtenir l'effet désiré.
4. (1.25 point) Créez un filtre qui vous permettra de conserver uniquement la texture des deux fenêtres. Ici, vous pouvez utiliser le filtre qui vous convient . Il est possible de créer un filtre idéal dans Paint puis d'effectuer un seuillage sur l'image résultante.
5. (1 point) Créez un filtre qui vous permettra de conserver uniquement la texture de la porte. Ici, vous pouvez utiliser le filtre qui vous convient
6. (0.5 point) Que se passe-t-il si on ne conserve pas la composante moyenne (fréquence 0) dans les filtres ?
7. (0.5 point) Quelle est la différence entre utiliser un filtre idéal ou un filtre *Butterworth* lors du filtrage ?
8. (0.75 point) Un filtre est utilisé. La fréquence de coupure de ce filtre change progressivement. À chaque itération, vous obtenez un élément de la maison en plus. Vous obtenez les éléments dans l'ordre suivant : la porte, la cheminée, les fenêtres, le mur, puis le toit et finalement la couleur moyenne de l'image. Quel est ce filtre ?