INF8725 - Traitement de signaux et d'images

TP2 - Filtrage d'image dans le domaine spatial et fréquentiel
Automne 2019

Professeure : Lama Seoud
Chargés de laboratoire : Clément Playout, Gabriel Lepetit-Aimon

---

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

Remise du travail :
La date de remise est le 25 octobre à 23h30. Une pénalité de 3 points par jour sera appliquée lors d'un retard.

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.

Première Séance: Filtrage spatial

Note: 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 probablement nécessaire de passer en `double` pour faire vos calculs puis de revenir en `uint8` pour afficher vos images.

Exercice I (5 points) : Rehaussement d'image

Vous souhaiter restaurer une image donc la qualité a été détériorée, 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 *theArtist.png* 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-là à votre image et affichez le résultat.

2. (1 point) Pour la suite du rehaussement vous allez avoir besoin d'effectuer des convolutions sur l'image. Implémentez la fonction *Convolution* qui prend comme argument une image et un masque et renvoie la convolution des deux. Vous n'utiliserez que les opérations matricielles et au maximum 2 boucles for, en particulier l'utilisation de la fonction *conv2* est interdite. Pour que la taille de l'image convoluée soit la même que l'image de départ, vous effectuerez un *0 padding*.

3. (0,5 point) L'image égalisée semble très bruitée. Appliquez un flou gaussien à l'image égalisé en la convoluant avec le masque gaussien 5x5 suivant. Affichez le résultat.

$$Masque\ Gaussien = \frac{1}{90} \times \begin{bmatrix} 1 & 2 & 1 & 2 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 8 & 18 & 8 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 1 & 2 & 1 \end{bmatrix}$$

4. (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 :

$$I_g = G * I$$
$$I_r = I_g + k \times \nabla^2(I_g)$$

où G est le masque gaussien:

$$Masque\ Gaussien = \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

et où $\nabla^2(I_g)$ est le Laplacien de I_g (en notation nabla). On rappelle que le Laplacien peut être calculé par convolution de l'image avec le masque suivant:

$$Masque\ Laplacien = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Implémentez la fonction *Rehaussement_Contour* qui prend en entrée une image et le paramètre k, puis retourne l'image rehaussée. Appliquez cette fonction à l'image dont les contours ont été dégradés en faisant varier le paramètre k pour obtenir le meilleur rehaussement (il se peut que vous ayez à renormaliser l'image). Affichez le Laplacien et le résultat du rehaussement. Attention, le laplacien peut renvoyer des valeurs négatives, il ne faut donc pas simplement convertir son résultat en `uint8` (format non-signé) pour l'afficher.

5. (1 point) Que se passe-t-il avec l'image rehaussée? Pourquoi?
Un bruit persiste sur l'image, proposez un filtrage pour l'atténuer.

Exercice II (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 *pieces.jpg* et convertissez la en niveau de gris. 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 : 200, 140, 120, 110, 90 (on ignorera les pièces de 1 dollar, 50 et 1 cent qui sont absentes de l'image). Aussi, vous pouvez utiliser la fonction *bwlabel* pour compter le nombre de composantes connexes dans l'image.

Deuxième partie : Filtrage spectral

Exercice III (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 foncion *fft2*. N'oubliez pas de récupérer le module de la TFD avec *abs* et de normaliser vos valeurs. Utiliser ensuite la fonction *fftshift* pour centrer votre TFD. Pour des fins de visualisation, utiliser log(1 + TFD(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 IV (6.5 points) : Filtrage spectral

1. (0.5 point) Chargez l'image *maillot.png*, 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 (torse, manche gauche, manche droite, poche, col) 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 faire disparaitre les hachures de la poche. Les autres textures devront rester visible (bien que altérée). La taille de votre filtre doit être de la même taille que le spectre. Vous devez jouer avec le paramètre sigma pour obtenir l'effet désiré.

4. (0.5 point) En vous inspirant de la méthode précédente, créez le filtre passe-haut qui n'affichera que les contours du maillot et des rayures (vous pourrez afficher la valeur absolue de l'image résultant de la transformée de Fourier Inverse).

5. (1 point) Créez un filtre qui vous permettra de conserver uniquement la texture des deux manches. 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.

6. (1 point) Créez un filtre qui vous permettra de faire disparaitre la texture du torse. Ici, vous pouvez utiliser le filtre qui vous convient.

7. (0.5 point) Quelle est la différence entre utiliser un filtre idéal ou un filtre *Butterworth* lors du filtrage?

8. (0.5 point) Que se passe-t-il si on ne conserve pas la composante moyenne (fréquence 0) dans les filtres?

9. (0.5 point) Un filtre est utilisé. La fréquence de coupure de ce filtre change progressivement. À chaque itération, vous obtenez un élément du maillot en plus. Vous obtenez les éléments dans l'ordre suivant : la poche, les manches, le torse puis le col et finalement la couleur moyenne de l'image. Quel est ce filtre?