TP - Sous-échantillonnage

1 - Sous-échantillonnage naïf

1. Charger l'image contenue dans le fichier *image.png*, disponible à l'adresse `http://www.di.ens.fr/~waldspur/tds/13_14_s2/image.png`. La stocker dans une variable `im`, sous la forme d'un tableau de 0 et de 1.

   La commande `im = imread('image.png')` permet de lire l'image et de la stocker dans la variable `im`.

Normalement, le tableau `im` est de taille 178×336.

2. Définir une image sous-échantillonnée `im_sub`, de taille 89×168 par :

$$\texttt{im\_sub}[m,n] = \texttt{im}[2m,2n] \qquad (\forall m = 0,...,88 \text{ et } n = 0,...,167)$$

   Attention, en Octave/Matlab, les tableaux sont indexés à partir de 1. Le réel `im[0,0]` est donc stocké dans la case `[1,1]` du tableau correspondant.

3. Enregistrer `im_sub` dans un fichier.

   La commande `imwrite(im_sub,'im_sub.png')` enregistre l'image représentée par `im_sub` dans le fichier `im_sub.png`.

Normalement, le résultat obtenu est assez inesthétique. Dans la suite du TP, on va améliorer la méthode en appliquant un filtre à l'image `im` avant le sous-échantillonnage.

2 - Flou local avant sous-échantillonnage

4. Calculer les signaux `im_blur_x` et `im_blur` définis par :

$$\texttt{im\_blur\_x}[m,n] = \frac{1}{4}\texttt{im}[m-1,n] + \frac{1}{2}\texttt{im}[m,n] + \frac{1}{4}\texttt{im}[m+1,n]$$
$$\texttt{im\_blur}[m,n] = \frac{1}{4}\texttt{im\_blur\_x}[m,n-1] + \frac{1}{2}\texttt{im\_blur\_x}[m,n] + \frac{1}{4}\texttt{im\_blur\_x}[m,n+1]$$

Ici, les indices sont considérés modulo 178 selon les lignes et modulo 336 selon les colonnes.

5. Définir `im_blur_sub`, la version sous-échantillonnée de `im_blur`, et l'enregistrer dans un fichier.

3 - Troncature en fréquence

6. Soit h le filtre de taille 178×336 dont la transformée de Fourier bidimensionnelle vaut :

$$\begin{aligned}\hat{h}[k,l] &= 1 \text{ si } k \in \{-44,...,44\} \text{ et } l \in \{-83,...,83\} \\ &= 0 \text{ sinon}\end{aligned}$$

À nouveau, les indices sont considérés modulo 178 et modulo 336.
Calculer la convolution circulaire $\texttt{im} \star h$:

$$\texttt{im} \star h[m,n] = \sum_{k,l} \texttt{im}[k,l]h[m-k,n-l]$$

La transformée de Fourier rapide et son inverse sont implémentées en Octave/Matlab : il s'agit des fonctions `fft` et `ifft` pour des signaux à une dimension et des fonctions `fft2` et `ifft2` pour des signaux à deux dimensions.

7. Normalement, $\mathtt{im} \star h$ est un signal réel. En raison des erreurs numériques, il se peut néanmoins que la partie imaginaire du signal calculé par Octave/Matlab ne soit pas strictement nulle.
Vérifier que la partie imaginaire est bien négligeable par rapport à la partie réelle. Projeter le signal sur l'ensemble des signaux réels.

8. Définir `im_trunc_sub`, la version sous-échantillonnée de $\mathtt{im} \star h$ et l'enregistrer dans un fichier.

9. Normalement, l'image que vous venez d'obtenir est facilement lisible mais présente des bavures. À quoi est-ce dû ?
Pouvez-vous refaire les questions 6 à 8 en utilisant à la place de h un filtre h_2 de même support mais donnant un résultat plus agréable visuellement ? Enregistrer à nouveau le résultat.

10. Envoyer le code et les quatre images à `waldspur@di.ens.fr`, le 11 mars au plus tard.