

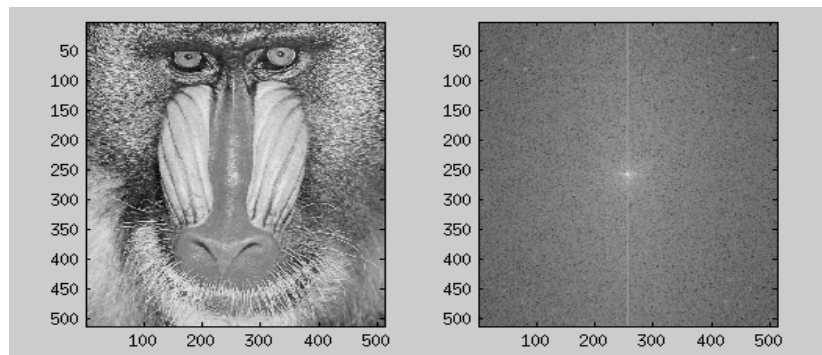
# Compte Rendu TME4 – BIMA

Binôme : Daniel Antunes Costa Gonçalves, Matthieu Wolfrom  
Groupe 1

## Exercice 1 :

Q1.

Pour afficher la transformée de Fourier de l'image mandrill.png on ouvre tout d'abord l'image avec `imread()`, on calcule ensuite sa transformée de Fourier avec `fft2()` et on centre les basses fréquences grâce à `fftshift()`. On ramène finalement le spectre centralisé en échelle logarithmique pour pouvoir visualiser les hautes fréquences qui ont une énergie négligeable autrement.

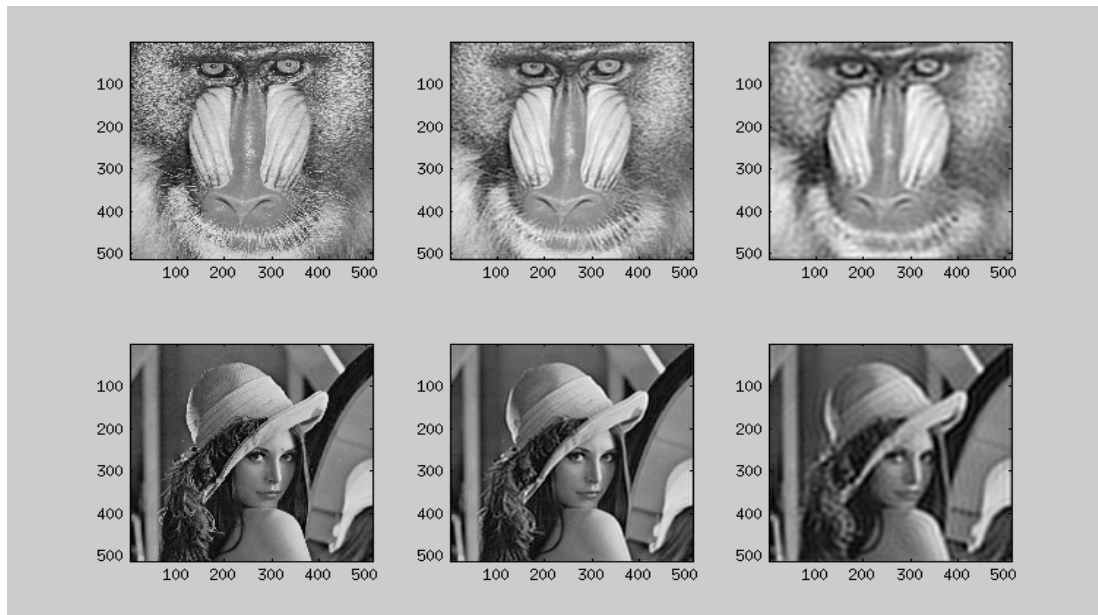


Q2.

Pour écrire la fonction `filtrePasseBasIdeal(n,m,fc)` on calcule tout d'abord les coordonnées du centre du filtre, à savoir  $(n/2, m/2)$ . Nous parcourons ensuite chaque case du filtre et si elle se situe à une distance supérieure à  $fc$  du centre du centre on la met à 0, sinon on la met à 1.

Q3.

Pour réaliser notre script on a créé une fonction `low_pass_filtering()` qui prend en paramètre une image `I` et une distance `fc` (fréquences à retenir). Elle calcule un filtre passe-bas idéal `h` grâce à notre fonction créée précédemment et la taille de l'image `I` ainsi que `fc`. On calcule ensuite sa transformée de Fourier `TF` et on fait une multiplication terme à terme entre `TF` et le filtre `h`. Il suffit ensuite de calculer la transformée de Fourier inverse avec les fonctions `ifft2()` et `ifftshift()`. Notre script doit donc juste ouvrir l'image à traiter et lui appliquer la fonction `low_pass_filtering()`.



*Sur l'image on voit d'abord à gauche les images originales, au centre les images après un filtrage avec  $f_c = \text{taille}/8$  et à droite  $f_c = \text{taille}/16$ .*

Q4.

- a) Lorsqu'on diminue  $f_c$  on remarque un lissage de l'image, les plus hautes fréquences sont filtrées. On remarque aussi des oscillations dans les résultats où  $f_c$  est plus petit, principalement dans l'image de Lena, ce qui conduit à un effet visuel de rebond.
- b) La première application est dans la compression d'images, c'est une méthode utilisée pour les formats jpeg par exemple. On utilise aussi ce type de filtrage passe-bas pour réduire le bruit d'une image.

## Exercice 2 :

Q1-2.

On commence par calculer le nombre de lignes et de colonnes de zéros à ajouter. Étant donné la taille  $d$  du filtre, on suppose  $d$  impair, on doit ajouter  $n = (d-1) / 2$  lignes et  $n$  colonnes de zéros sur chaque côté de l'image.

$\text{ImagePad}(I, h)$  est donc la fonction qui crée une image  $J$  de taille  $(\text{size}(I, 1) + 2*n, \text{size}(I, 2) + 2*n)$  où  $\text{size}(I, 1)$  est le nombre de lignes de l'image de départ et  $\text{size}(I, 2)$  est le nombre de colonnes de l'image de départ. Pour chaque position à une distance  $\leq n$  du bord contient un 0, le reste de l'image  $J$  est une copie de  $I$  décalée de  $n$  lignes et  $n$  colonnes. On obtient ainsi le zéro-padding de l'image  $I$  pour un filtre  $h$ .

	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0.7519	0.7152	0.3174	0.6357	0.6312	0	0
4	0	0	0.2287	0.6421	0.8145	0.9509	0.3551	0	0
5	0	0	0.0642	0.4190	0.7891	0.4440	0.9970	0	0
6	0	0	0.7673	0.3908	0.8523	0.0600	0.2242	0	0
7	0	0	0.6712	0.8161	0.5056	0.8667	0.6525	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

*Image représentant un zero-padding avec un filtre de taille 5x5*

Q3.

Pour réaliser la fonction  $\text{convolution}(I, h)$ , dans un premier temps, il est nécessaire d'appliquer une rotation de  $\pi$  sur le filtre  $h$ .

Comme on ne réalise pas de padding dans cette fonction, il sera impossible d'appliquer le filtre sur l'image entière, car les pixels sur les bords n'ont pas de pixels voisins tout autour.

Le décalage à effectuer s'exprime en fonction de la taille filtre, à savoir :

$$n = (\text{size}(h, 1) - 1) / 2.$$

On crée ensuite l'image résultat  $J$  de la même taille que l'image  $I$  et l'initialisant à 0.

On dispose maintenant de tous les éléments nécessaires pour appliquer le filtre  $h$  sur l'image  $I$ , à l'aide de 4 boucles imbriquées.

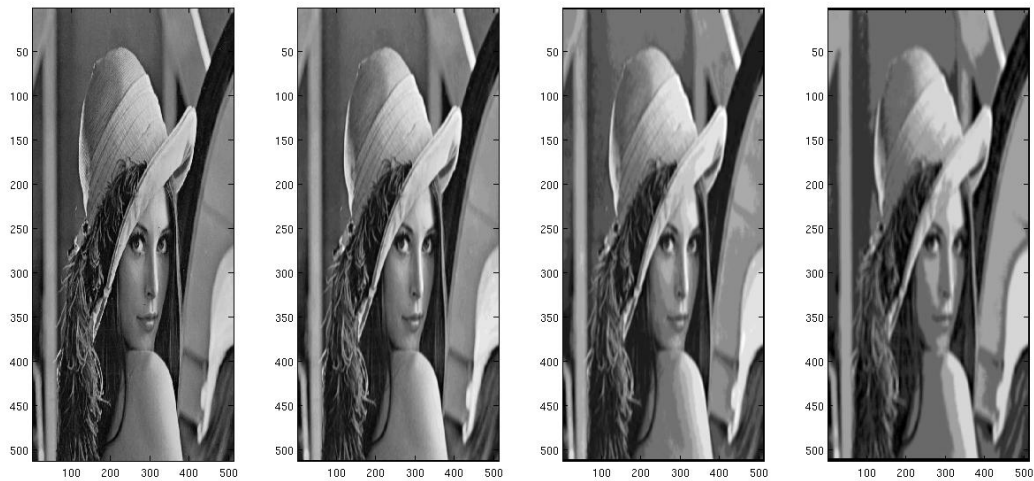
Les 2 premières boucles permettent de parcourir l'image  $I$  entre les coordonnées  $n+1$  à  $\text{size}(I) - n$ , cela garantissant que notre filtre ne dépasse pas des bords de l'image d'origine.

Ces boucles nous permettent de parcourir l'ensemble des pixels à traiter, on réalise ensuite 2 boucles supplémentaires qui parcourent le filtre et calcule la somme pondérée entre les pixels de  $I$  et les coefficients de  $h$ .

Cette fonction rend ainsi une image  $J$ , résultant de la convolution linéaire de  $I$  par  $h$ .

Q4.

En utilisant des filtres moyenneurs sur une image sans bruit, on constate un lissage des valeurs des pixels de l'image, sachant que plus le filtre est grand, plus le lissage est important.



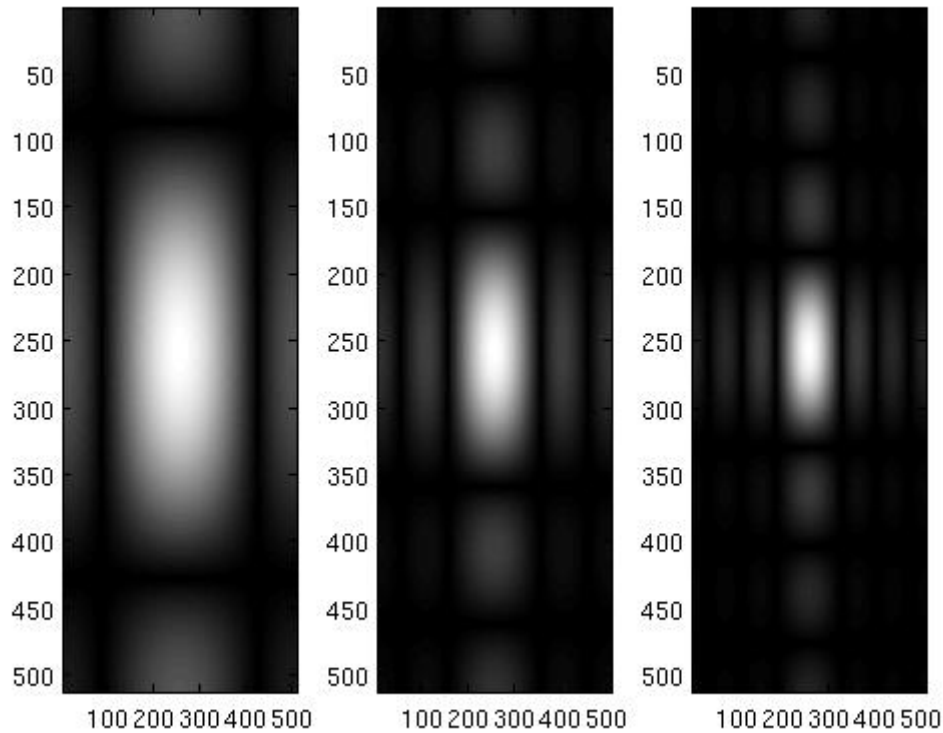
*Image d'origine à gauche, puis résultats de la convolution avec des filtres moyenneurs de taille 3x3, 5x5 et 7x7 dans l'ordre de la 2ème à la 4ème image*

Q5.

La fonction permettant de visualiser la fonction de transfert d'un filtre consiste à créer une image de la taille voulue (par exemple 512x512) initialisée à 0 partout. On recopie ensuite le filtre dans le coin en haut à gauche de l'image.

Q6.

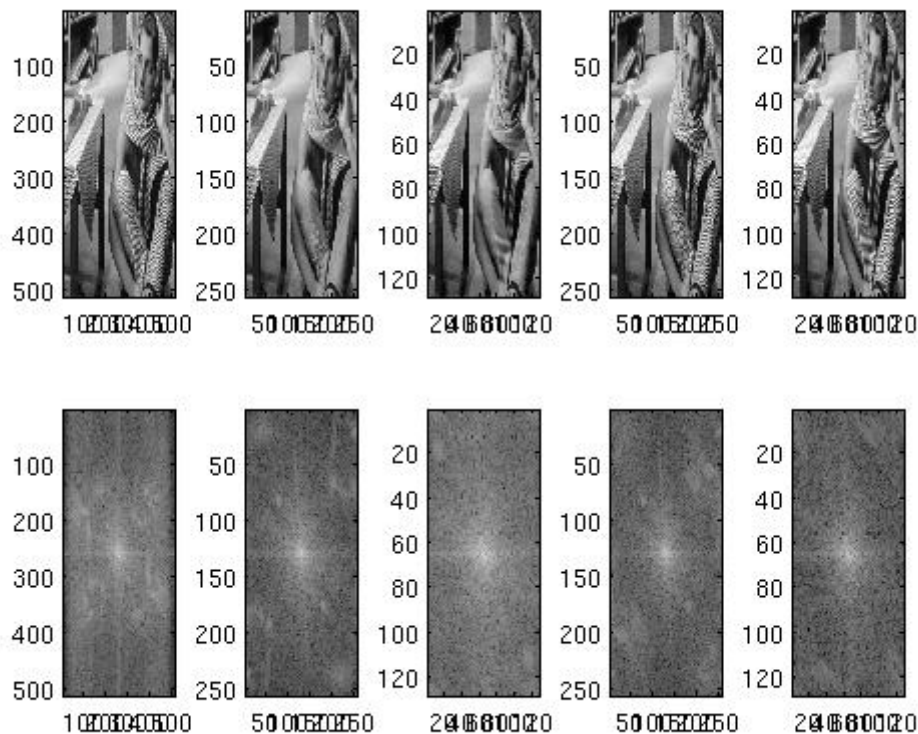
La visualisation des fonctions de transfert des 3 filtres moyenneurs permet de constater que ce sont bien des filtres passe-bas. Un filtre moyenneur n'est pas un filtre passe-bas idéal, car il ne conserve pas parfaitement les basses fréquences et il conserve également une partie des hautes fréquences, autrement dit ses coefficients ne sont pas 0 pour les hautes fréquences et 1 pour les basses fréquences.



*Fonctions de transfert des filtres moyenneurs 3x3, 5x5, 7x7, dans l'ordre*

## Exercice 3 :

Q2-3.



*Image originale à gauche, puis deux images sous-échantillonnées avec filtre anti-aliasing, et enfin deux images sous-échantillonnées sans filtre à droite. La deuxième ligne affiche le module centré des transformées de Fourier en échelle logarithmique.*

On crée dans un premier temps deux images à partir de l'originale en appliquant le filtre anti-aliasing fourni puis en effectuant deux sous-échantillonnage. Le filtrage anti-aliasing est un filtre passe-bas que l'on applique préalablement pour éviter les problèmes de recouvrement des fréquences lors du sous-échantillonnage.

La première image obtenue a subi un lissage par rapport à l'image de départ, cela se remarque principalement au niveau du pantalon, dont le motif a été majoritairement effacé.

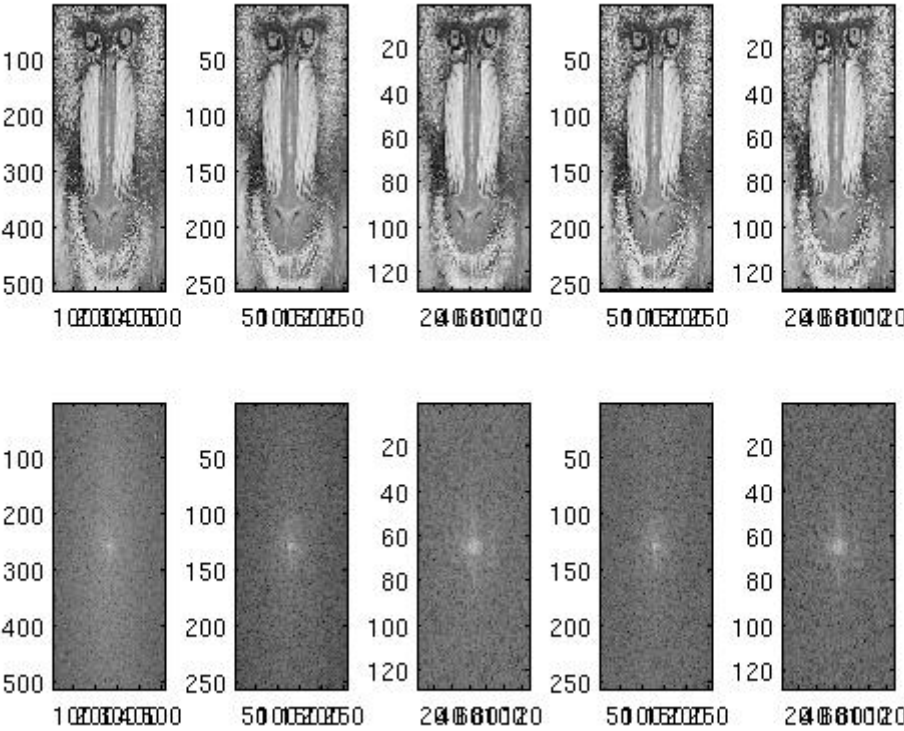
Dans la deuxième image, l'effet de lissage est encore plus marqué et on constate également un léger effet d'aliasing.

Les deux dernières images sont obtenues de la même manière mais sans appliquer le filtre.

Les images obtenues sans filtrage sont bien moins fidèles à l'image d'origine car les fortes fréquences sont totalement modifiées.

Le filtrage n'empêche pas la perte d'information lors du sous-échantillonnage mais permet d'obtenir des images moins déformées.

Q4.



## Exercice 4 :

Q1.

L'image I2 est une matrice 512x512 de nombres entiers uint8, au même titre que les images noir et blanc que l'on a traité auparavant.

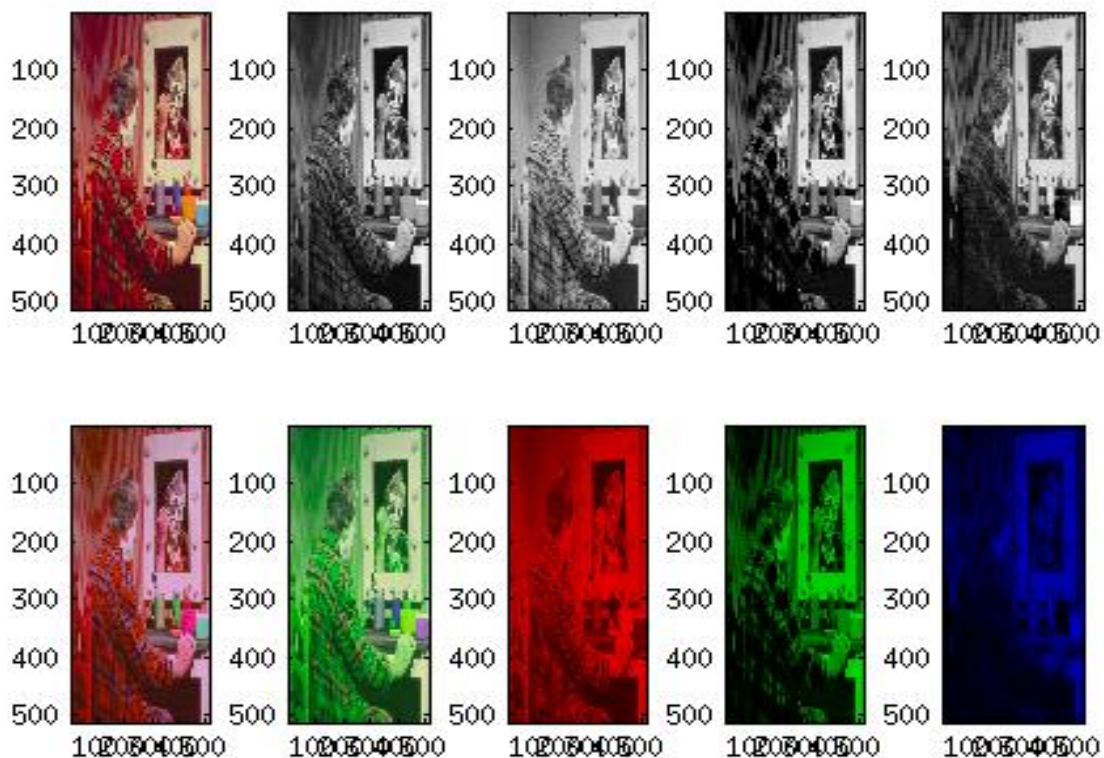
L'image I1 est un tableau à 3 dimensions 512x512x3 de nombres uint8, c'est une image couleur à 3 composantes, elle est donc représentée par 3 matrices 512x512 pour chacune de ses composantes.

Q2.

L'image I1 est une image couleur RGB, qui dispose donc des 3 composantes Red, Green et Blue pour définir ses couleurs.

On extrait les composantes dans trois images 512x512 séparées, en observant ces images en niveau de gris, on constate que la composante rouge donne une image plus claire que les deux autres, ce qui correspond au fait que le rouge est plus présent dans l'image que les autres couleurs.

Q3-4.



Dans l'ordre de gauche à droite, I1 ; I2 ; Ir et Iv et Ib qui sont les composantes isolées rouge, verte et bleue ; Ivb où les composantes vertes et bleues ont été échangées ; Irv où les composantes rouge et verte ont été échangées ; R et V et B qui sont respectivement le plan rouge en rouge, le plan vert en vert et le plan bleu en bleu, les plans étant ceux de I1.

Pour mélanger les plans de couleur, on copie l'image d'origine dans une nouvelle image puis on échange les plans voulus en remplaçant les matrices



correspondantes.

Pour visualiser un plan dans la couleur qui le correspond, il suffit de créer un tableau de la dimension correspondant à l'image d'origine (ici 512x512x3) rempli de 0, puis de recopier le plan voulu dans ce nouveau tableau. On obtient ici le plan dans la couleur correspondante et les autres plans valant 0, l'image ne sera que de la couleur voulue.