

## TP 1 : Traitements et outils de base

### 1. Visualisation d'une image

- (a) Chargez (`imread`) puis affichez (`imshow`) l'image 'mandrill.jpg'.
- (b) Convertissez l'image couleur en niveaux de gris (`rgb2gray`), puis affichez le résultat.
- (c) Affichez chacune les composantes rouge, verte et bleue de l'image couleur.
- (d) Affichez les composantes de teinte, saturation et valeur (`rgb2hsv`) de l'image couleur (pour afficher la teinte en image couleur, on pourra mettre à 1 la saturation et la valeur de l'image HSV, puis la convertir en image RGB avec la fonction `hsv2rgb`)
- (e) La représentation naturelle de l'image avec `imshow` code l'intensité des pixels par une couleur. Une image en niveau de gris peut également être représentée comme une courbe en trois dimensions, l'intensité correspondant alors à l'altitude de chaque point. Utilisez `mesh` pour représenter par une courbe en trois dimensions l'image convertie en niveaux de gris. Attention, il est indispensable de convertir l'image en `double`.
- (f) Affichez l'image convertie en niveaux de gris avec une représentation en fausse couleur : `imshow(monImage, 'Colormap', jet)`. Afficher la barre de couleur correspondante (`colorbar`).
- (g) Visualisez l'image convertie en niveaux de gris en précisant la plage d'intensité dans `imshow`. Testez successivement : `[0,1]`, `[0,100]`, `[100,255]` et `[]`.

### 2. Histogramme et amélioration du contraste

- (a) Chargez l'image contenue dans 'ct.mat' (`load`) puis affichez là.
- (b) Lors de l'interprétation des images, le radiologue choisit une dynamique de visualisation adaptée aux organes qu'il souhaite analyser. Testez successivement : `[]`, `[-150, 250]` et `[-1400, 200]` et analysez les différences.
- (c) Calculez le min et le max de l'image. Convertissez ensuite cette image en une image ayant des niveaux de gris entre 0 et 255 (une fois la mise à l'échelle faite, on prendra soin de convertir l'image en `uint8`). Afficher l'histogramme de cette image convertie (`imhist`). On notera par la suite cette image  $I_{norm}$ .
- (d) On souhaite maintenant afficher l'histogramme de l'image originale. Dans ce cas, `imhist` ne fonctionne pas car l'image n'est pas codée sur 256 niveaux de

gris. Utilisez pour cela la fonction `hist`. Cette fonction requiert un vecteur en entrée. On pourra convertir l'image en un vecteur grâce à la syntaxe `I(:)`. Testez l'influence du choix du nombre de bins (valeur par défaut, règle de Sturges :  $1 + \log_2 n$  avec  $n$  le nombre de pixels dans l'image, max-min, ...)

- (e) Sur l'image  $I_{norm}$ , appliquez les opérations algébriques suivantes :  $I_{norm}^2$ ,  $I_{norm}^4$ ,  $\log I_{norm}$ . Visualisez et commentez leurs effets. Que se passe-t-il si on ne prend pas soin de convertir au préalable l'image  $I_{norm}$  en double ?
- (f) Effectuez une égalisation de l'histogramme (`histeq`) de l'image  $I_{norm}$ . Est-ce que l'objectif de la méthode est atteint ?

### 3. Transformée de Fourier

- (a) Chargez l'image `sin32.png`.
- (b) Calculez sa transformée de Fourier discrète (TFD) avec la fonction `fft2` (après avoir converti l'image en *double*).
- (c) Affichez le module de la TFD de l'image. La TFD est généralement représentée avec les basses fréquences au centre de l'image, contrairement au résultat calculé par `fft2`. Pour ramener les basses fréquences au centre, utilisez `fftshift`.
- (d) Comment s'interprète le module de la TFD ?
- (e) Effectuez les mêmes opérations avec les images `sin16.png` puis `sin8.png`. Que concluez-vous ?
- (f) Terminez en analysant les transformées de Fourier des images `holes.tiff`, `wall.tiff`, `weave.tiff`, `straw.tiff` et `lena.tiff`.

### 4. Hautes et basses fréquences d'une image

- (a) Chargez l'image '`lena.tiff`'.
- (b) Calculez sa transformée de Fourier discrète (TFD) avec la fonction `fft2` et `fftshift`.
- (c) Multipliez sa transformée de Fourier à l'aide d'une fonction rectangle de largeur  $\Delta f$  afin de récupérer uniquement les basses fréquences (on construira la fonction rectangle en utilisant la fonction `zeros`, puis en mettant à 1 les valeurs au centre de l'image).
- (d) Calculez la transformée de Fourier inverse du résultat (`fftshift` puis `ifft2`) afin de visualiser l'image basse fréquence correspondante.
- (e) Refaites les mêmes opérations en récupérant cette fois-ci les hautes fréquences.
- (f) Faites varier  $\Delta f$  et regarder l'influence sur les images hautes et basses fréquences correspondantes.

## 5. Phénomène de crénelage (*aliasing*)

Le crénelage est un phénomène qui apparaît lorsque l'image est mal échantillonnée. Ce phénomène peut être évité en filtrant l'image convenablement avant l'échantillonnage. Cet exercice a pour but de comparer le sous-échantillonnage sans filtre, avec un filtre gaussien puis un filtre idéal.

- (a) Chargez et affichez l'image `toit.jpg`.
- (b) Sous-échantillonnez l'image d'un facteur trois; il suffit pour cela de prendre un pixel sur trois dans chaque dimension, ce qui peut être fait avec l'instruction `I(1:3:end,1:3:end)`. Comment se traduit visuellement le phénomène de crénelage?
- (c) En sous-échantillonnant l'image, la fréquence d'échantillonnage est-elle modifiée? Le théorème de Shannon est-il respecté? Comment s'appelle l'équivalent monodimensionnel du crénelage?
- (d) Appliquez un filtre gaussien (fonction `fspecial`) sur l'image avant le sous-échantillonnage. Réglez les paramètres du filtre afin de diminuer au maximum le crénelage.
- (e) Appliquez un filtre idéal sur l'image avant le sous-échantillonnage. Choisissez convenablement les fréquences de coupure du filtre pour diminuer le crénelage sans altérer l'image. Un filtre passe-bas idéal élimine complètement les fréquences supérieures à la fréquence de coupure sans modifier les fréquences en deça. Pour implémenter le filtre idéal, il est plus simple de travailler dans le domaine de Fourier pour définir le filtre idéal.