

TD I : Prise en main de Matlab, codage d'images numériques

I.1/ Prise en main de Matlab

a/ Copier le répertoire fourni « TP_image_1 » sur votre disque M.

b/ Ouvrir l'éditeur de code de Matlab et remplacer le répertoire de travail par défaut par le répertoire fourni « M:\...\TP_1 ».

c/ Lancer l'éditeur de scripts et créer un script « PremierTraitement.m ». Ecrire le programme ci-dessous permettant la lecture du fichier image « `image1.bmp` », le chargement de son contenu dans une matrice `I_0` puis son affichage dans une fenêtre graphique :

```
I_0 = imread('image1.bmp');  
figure(1);  
imshow(I_0);  
title('Image originale');
```

Exécuter ce script grâce au menu « Debug->Save and Run » ou en cliquant sur l'icône correspondant. Afficher dimensions de la matrice `I_0` grâce à l'instruction :

```
size(I_0)
```

d/ Une image en niveau de gris est un signal à deux dimensions que l'on peut représenter par une fonction $z=f(x,y)$ où x et y représentent la position du pixel et z la valeur du niveau de gris associée. Pour illustrer ce codage de l'image, représenter les données de l'image « `image1.bmp` » dans un graphique en 3D grâce à l'instruction :

```
figure(2);  
mesh(double(I_0));
```

Noter la conversion de type des données image en double (elles sont codées en `uint8`, c-à-d sur un octet, dans le fichier).

A partir du graphique, retrouver la dynamique du signal (étendue de variation des niveaux de gris).

A quels niveaux de gris correspondent les valeurs minimale et maximale ?

Sur combien de bits est codé chaque pixel ?

Refaire le même travail sur l'image « `image2.bmp` ». Notez la différence de dimensions. Conclusion.

f/ Pour commencer on se propose de parcourir l'image en modifiant certains pixels par un simple traitement (sans grand intérêt pratique). Charger et afficher les deux images « `image3.bmp` » et « `image4.bmp` ». Remplacer les pixels de « `image3.bmp` » dont le niveau de gris est supérieur à un seuil $s_2=250$ par les pixels de mêmes indices de l'image « `image4.bmp` ». Afficher le résultat.

Même question avec les images « `image3c.bmp` » et « `image4c.bmp` ».