

L'objectif de ce TP de traitement d'images sous Matlab est de présenter la notion d'image et d'effectuer des opérations simples d'analyse d'images.

1. **Aide** : L'aide sur les principales fonctions de traitement d'images sous Matlab est obtenue en tapant "**help images**" en ligne de commande de Matlab.
2. **Lire et écrire des images** :
  - Matlab est capable de lire et de décoder les fichiers images JPEG, TIFF, BMP, PNG, HDF, PCX ou XWD ...etc. Les exemples ci-dessous permettent de lire une image au format **jpg**, de la décoder dans la variable **img** et de l'afficher à l'écran dans une figure.

```
» img=imread('src1.jpg');  
» figure;imshow(img);  
» axis('img');  
» axis on
```

```
» img=imread('src1.jpg');  
» image(img);
```

La commande **axis('img')** rend l'image affichée carrée pour garder les proportions. L'appel à **axis on** permet l'affichage des graduations des axes.

- L'objet **img** est une série "**Ncanaux**" des matrices **Nlignes\*Ncolonnes**. Vous pouvez obtenir ses dimensions en utilisant la fonction **size**

```
» [Nlin, Ncol, Ncanaux] = size(img) ;
```

- Matlab offre une possibilité intéressante, en effet, il est possible d'afficher plusieurs images dans la même figure. Pour ce faire, il faut utiliser la commande **subplot** "**subplot(nombre\_ligne, nombre\_colonne, numéro\_zone)**". Dans l'exemple ci-dessous on souhaite afficher deux images sur la même ligne dans une seule figure.

```
» img2=imread('src2.jpg');  
» figure;subplot(1,2,1);imshow(img);  
» subplot(1,2,2);imshow(img2);
```

- Afin de fermer une figure sous Matlab, on tape **close** s'il s'agit de la dernière figure ouverte ou bien **close** avec le numéro de figure en paramètre pour fermer la figure sélectionnée. Pour fermer toutes les figures, on demande la fermeture avec le mot-clé **all**.

```
» figure;imshow(img);  
» figure;imshow(img2);  
» close(1)  
» close all
```

- Matlab autorise l'exportation d'images sous divers formats: BMP, TIFF, EPS, PS... par la commande "**print -dFORMAT fichier**" qui permet de sauvegarder une figure, comme illustré dans l'exemple ci-dessous.

```
» img=imread('donne.jpg');  
» figure;imshow(img);  
» print -djpeg result.jpg
```

Avec la fonction "*imwrite*" on peut aussi enregistrer une image dans un fichier, par le syntaxe suivant "*imwrite(img\_src, 'nom\_fichier\_sortie', 'format')*".

### 3. L'accès à un élément de l'image :

- L'accès à un élément particulier d'une image est indexé par le nom et la position de cet élément. Par exemple, si on conserve l'image **img** ci-dessus, on peut récupérer les valeurs ou les modifier aisément.

```
» img(3,2)  
» img(1:10,30:40)  
» img(1:3,31:39) = 0;  
» figure;imshow(img);
```

- Les valeurs des images lues sous Matlab sont entières, mais dans certaines circonstances, on a besoin de travailler sur des valeurs réelles. La transformation pour passer d'entier à réel utilise la fonction *im2double*.

```
» img=imread('src1.jpg');  
» figure;imshow(img);  
» imgdbl=im2double(img);  
» figure;imshow(imgdbl);  
» imgint=im2uint8(imgdbl);  
» figure;imshow(imgint);  
» imwrite(imgint,'test.jpg','jpeg');  
» whos
```

### 4. Sélection d'une sous image

Une sous-image pour matlab n'est autre qu'une sous-matrice. Pour créer une sous image dont les dimensions sont *delta\_lin* et *delta\_col* et dont le coin en haut à droite a pour coordonnées (*lin*, *col*) il suffit de saisir :

Resultat=image(*lin* : *lin*+ *delta\_lin*, *col* :*col*+ *delta\_col*)

```
» detail=img(30: 80,20:50);  
» figure;imshow(detail);
```