

1-Fonctions Matlab

Les fonctions *disp*, *display* ou *celldisp* permettent l'affichage de textes ou de données dans la fenêtre de commandes de Matlab.

```
>> disp('imagerie');  
imagerie
```

2-Type des images en mémoire

Matlab supporte 4 formats d'images :

- les images binaires,
- les images monochromes (dites aussi images d'intensités, images à niveaux de gris ou même images noir et blanc),
- les images couleur RGB,
- les images couleur indexées.

Il est possible de changer de format en utilisant les fonctions suivantes :

- *ind2gray* : indexé → intensité,
- *ind2rgb* : indexé → RGB,
- *rgb2ind* : RGB → indexé,
- *rgb2gray* : RGB → intensité,
- *im2bw* : intensité, indexé, RGB → binaire : c'est l'opérateur de binarisation. Une image binaire peut être également obtenue en utilisant des opérateurs de comparaison et des opérateurs logiques.

```
>> I=rgb2gray(i);  
>> figure(2);imshow(I);  
>> im2bw(i);
```

Par exemple, les instructions $(I==seuil)$ ou $((I>=seuil_bas) \& (I<seuil_haut))$ permettent d'obtenir des images binaires par comparaison des niveaux des pixels d'une image *I* à des valeurs de seuils.

Le passage d'une image couleur à une image d'intensité correspond à la transformation des composantes R, G, B en la composante Y des systèmes YUV qui séparent l'information de luminance et de chrominance.

Pour extraire une image composante à partir d'une image couleur, il suffit d'écrire par exemple :

```
I= imread('TP1.jpg') ;  
Ired=I(:,:,1); Igreen= I(:,:,2); Iblue= I(:,:,3);  
figure(3) ;Subplot(2, 2, 1); imshow(I); axis image;  
Subplot(2, 2, 2); imshow(Ired); title('red');  
Subplot(2, 2, 3); imshow(Igreen); title('green');  
Subplot(2, 2, 4); imshow(Iblue); title('blue');
```

Les valeurs des pixels de ces images peuvent être de différents types :

- logique (0 ou 1 pour les images binaires),
- entier non signé codé sur 8 bits (entre 0 et 255),
- entier non signé codé sur 16 bits (entre 0 et 65535),
- réel (entre 0 et 1).

```
>> 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
```

Les valeurs des images lues sous Matlab sont entières, mais dans certaines circonstances, on a besoin de travailler sur des valeurs réelles. Il est possible de changer le type des variables en utilisant les fonctions suivantes :

- *im2double* : codage d'images en type réel,
- *im2uint8* : codage d'images en type entier non signé sur 8 bits,
- *im2uint16* : codage d'images en type entier non signé sur 16 bits,
- *double* conversion de données en type réel,
- *uint8* conversion de données en type non signé sur 8 bits,
- *uint16* conversion de données en type non signé sur 16 bits.

Les images binaires sont codées en entier logique ou en réel logique et leurs pixels ont des valeurs uniquement égales à 0 ou à 1. L'affichage d'une image peut se faire, soit en réel et les valeurs des pixels sont alors comprises entre 0.0 et 1.0, soit en entier et les valeurs des pixels sont alors comprises, par exemple, entre 0 et 255 pour des entiers non signés codés sur 8 bits.

```

origine=imread('C:\exple.jpg');
Ired=origine(:,:,1);
[Nlin Ncol]=size(Ired);
figure(1);
subplot(1,3,1);
imshow(origine);colormap(gray(256));axis('image') ;
title('originale');
subplot(1,3,2);
imshow(Ired);colormap(gray(256));axis('image') ;
title('bande');
seuil = 150 % vous donnez une valeur au seuil
ImageBinaire = zeros( Nlin, Ncol ) ;

for lin=1:Nlin
    for col=1:Ncol
        if( Ired(lin,col) < seuil )
            ImageBinaire(lin,col) = 255 ;
        end
    end
end
subplot(1,3,3)
imshow(uint8(ImageBinaire));axis('image') ;
title('binairisé');
colormap(gray(2)) ;

```

3- Seuillage :

Le seuillage d'une image désigne la définition d'un seuil au-dessus ou en-dessous duquel on va garder certaines valeurs de niveaux de gris

```

» img=im2double(Ired);
» figure;subplot(1,2,1);imshow(img);
» result=(img>0.5).*img;
» subplot(1,2,2);imshow(result);

```

4- Opérations géométriques

Les opérations géométriques classiques sont permises avec la boîte à outils de traitement d'images: rotation, changement de taille, découpage...

L'exemple ci-dessous illustre la rotation d'une image avec Matlab. Dans le premier cas, "imgrot1" est plus grande que "img". Dans le second cas, le paramètre 'crop' impose un découpage de l'image et la taille de l'image "imgrot2" est la même que celle de l'image "img".

```

» figure;imshow(img);
» imgrot1=imrotate(img,3,'bilinear'); // nearest, bilinear or bicubic
» figure;imshow(imgrot1);
» imgrot2=imrotate(img,3,'bilinear','crop'); // crop or loose
» figure;imshow(imgrot2);

```

La syntaxe de la fonction imrotate est : **J = imrotate(I, angle, method, bbox)**

Le zoom permet d'agrandir une partie de l'image à l'aide de la souris, pour cela, il faut entrer la commande dans l'éditeur de Matlab puis sélectionner une zone de l'image à agrandir. Le bouton droit de la souris permet de revenir à la taille normale.

» zoom