Université Echahid Hamma Lakhder El Oued.

Module : Image, son, vidéo codage et transmission

TP1

Manipulations d'images sous Matlab

Objectifs du TP1 :

L'objectif de ce TP est de prendre en main les outils de traitement d'images les plus classiques à l'aide du logiciel Matlab. En effet, nous visons à se familiariser avec les toolbox **Image Processing** et **Image Acquisition**.

1- Indications sur Matlab

Matlab est un logiciel de calcul scientifique permettant de développer des solutions à des problèmes techniques. Il permet de réaliser du calcul numérique et de tracer des graphiques pour visualiser et analyser les données. Il dispose d'un langage et d'un environnement de programmation interactifs ainsi que d'outils pour concevoir des interfaces utilisateur graphiques. Matlab est associé à des boites à outils appelés **TOOLBOX** permettant d'accéder à des fonctions spécifiques à un domaine d'application comme le **traitement d'images** par exemple.
Les TPs de traitement d'images réalisés avec Matlab nécessitent ainsi la toolbox **Image Acquisition** et la toolbox **Image Processing**.

2- Le help de MatLab

Les fonctions de la toolbox **Image Processing** peuvent être listées en tapant : **help nom d'une fonction** donnant une explication sur la fonction et souvent un exemple qui peut vraiment être essayé.
Les images peuvent être obtenues à partir de :
C:\ProgramFiles\MATLAB\R2012b\toolbox\images\imdemos.

3- Rappels sur la notion d'image

Une image réelle est obtenue à partir d'un signal continu bidimensionnel comme par exemple un appareil photo ou une caméra. Sur un ordinateur, on ne peut pas représenter de signaux continus, on travaille donc sur des valeurs discrètes.
A cet effet, une image numérique est définie comme un signal fini bidimensionnel échantillonné à valeurs quantifiées dans un certain espace de couleurs. Elle est constituée de points (pixels).

- **Signal fini :** une image possède des dimensions finies, exemple : 640x480, 800x600 points.
- **Signal bidimensionnel :** une image possède deux dimensions : largeur, hauteur.
- **Signal échantillonné :** les pixels d'une image sont régulièrement espacés sur une grille carrée.
- **Valeurs quantifiées :** les valeurs des pixels appartiennent à un intervalle borné connu.
- **Espace de couleur :** il existe de nombreuses façons de percevoir les couleurs d'une image, l'espace de représentation le plus connu est l'espace rgb (rouge-vert-bleu).

Autrement dit, une image est une matrice MxN de valeurs entières prises sur un intervalle borné [0,Ng] où Ng est la valeur maximale du niveau de gris.

p(i,j) est le niveau de gris du pixel de coordonnées ligne i et colonne j dans l'image. p(i,j) ∈ [0, Ng]. Les valeurs des niveaux de gris sont des entiers.

Stocker de grandes **images** sur le disque dur d'un ordinateur prend beaucoup de place. Les nombres entiers sont stockés en écriture binaire, c'est-à-dire sous la forme d'une succession de **0** et de **1**. Chaque 0 et chaque 1 se stocke sur une unité élémentaire de stockage, appelée bit. Pour stocker l'image complète, on a donc besoin de MxNx8bits.

La disposition des pixels (**le maillage**) est généralement en ligne et colonne (maillage rectangulaire); quelques applications peuvent présenter un maillage différent (utilisation de pavés hexagonaux par exemple).

Pour stocker le niveau de gris, on utilise généralement un nombre entier (**type uint8 sur 8 bits ou uint16 sur 16 bits**) plutôt qu'un scalaire Matlab (**type double qui utilise 64 bits**). Matlab gère ces formats de données directement.

4- Prise en main

- Les **fonctions** sont éditées dans la fenêtre de commandes et **exécutées** en appuyant sur la touche **Entré**.
- Le point virgule à la fin d'une fonction permet d'éviter d'afficher les données résultats de la fonction exécutée ou de séparer plusieurs fonctions sur une même ligne de commande.
- Plusieurs fonctions et commandes peuvent être saisies dans un fichier qui sera enregistré avec l'extension **.m**.
- Chaque variable déclarée dans Matlab est stockée dans l'espace des variables à partir duquel il est possible de consulter la taille et le type de la variable ainsi que d'éditer son contenu par un double-click sur le nom de la variable. A cet effet, un scalaire est un tableau de taille 1×1 ; un vecteur est un tableau à 1 dimension de taille $1\times n$; une matrice est un tableau à 2 dimensions de taille $m\times n$.

5- Acquisition d'images sous Matlab

Les fonctions MatLab pour **lire** et **enregistrer** les images sont **imread** et **imwrite**. Les fonctions disponibles pour afficher les images sont **image**, **imagesc** et **imshow**.

Exemple :

```
IMG=imread('cameraman.tif');
imshow(IMG) ;
Chemin = 'E:\';
imwrite (IMG,fullfile(Chemin,'cameraman.tif'),'tif');
```

1) L'image binaire

Une image binaire est une matrice rectangulaire dont les éléments valent 0 ou1.Bien qu'il n'y ait que deux valeurs possibles, Matlab représente les éléments de cette matrice par des réels en double précision (8 octets par élément). Lorsque l'on visualise une telle image, les 0 sont représentés par du noir et les 1 par du blanc.

Exemple :

```
A = [0 0 1 0 0 ;0 1 1 1 0 ;1 1 1 1 1 ;0 1 1 1 0 ;0 0 1 0 0] ;
B = [A A A A A ; A A A A A ; A A A A A ; A A A A A; A A A A A] ;
```

```
imshow(A , 'notruesize');
imshow(B, 'notruesize');
```

Nous remarquons que **imshow** prend en compte deux paramètres : le nom de l'image, et un paramètre optionnel,'notruesize', qui, s'il est présent, autorise Matlab à choisir la taille de l'image affichée (cela est utile en particulier pour les petites images).

2) L'image en niveaux de gris

Les images en niveaux de gris sont mémorisées sous la forme d'une matrice formée chacune d'entiers de 0 à 255 correspondant à un octet (format MatLab : uint8).
Nous pouvons l'appeler également une image d'intensités.
Notons qu'une image en niveaux de gris peut consister aussi en des réels compris entre 0 (noir) et 1 (blanc). En effet, les valeurs comprises entre 0 et 1 représentent les différents niveaux de gris. Par exemple : 0.2 représente un gris foncé et 0.8 un gris clair.

Exemple 1 :
Nous allons créer une image d'intensités de 4*7 colonnes) constituée de lignes verticales dont les 7 pixels (quatre lignes et niveaux de gris passent progressivement du noir au blanc (de la gauche vers la droite).

```
I = ones(4,1)*[(0:6)/6];     % ones permet de créer des tableaux contenants des valeurs de 1
imshow(I, 'notruesize');   % ones(4,1) va créer 4 lignes et une colonne pour le tableau
```

Pour préciser l'échelle de gris, il suffit de donner les bornes d'affichage (qui peuvent être différentes des bornes de l'image) :
imshow(image, [niv_min niv_max]).

Exemple 2 :

```
IMG=imread('cameraman.tif');
imshow(IMG, [12 24]);
```

Pour un ajustement automatique sur les niveaux extrêmes :
imshow(image, [ ]) ;

Exemple 3 :
Nous allons manipuler les valeurs de niveau de gris, en les convertissant de : **uint8** en **double** (les niveaux de gris seront composés de réels entre 0 et 1).

```
IMGBis=double(IMG)/255; % Convertir en double
figure(1); imshow(IMGBis);
```

Si nous vérifions la variable IMGBis dans le workspace, nous constaterons que les niveaux de gris sont bien en double.

Les commandes suivantes permettent d'afficher une courbe dont les valeurs prises sont les valeurs des pixels de l'image lorsqu'on parcourt cette image de haut en bas puis de gauche vers la droite.

```
val=IMGBis(:);
figure(2); plot(val);
```