

**Niveau:** Master1 Télécommunication

**Responsable du cours:** Zoubeida MESSALI

**Enseignante de TP:** Soumia SID AHMED

## TP1 : Toolbox de Matlab de traitement d'images et de la vidéo

Après ce TP, vous serez capable de:

- I. Maîtriser les commandes de base pour de Manipulation des images digitales sous Matlab
- II. Traitements sur la couleur et palette
- III. Les séquences d'images et vidéo (multi frame array)

### I. Représentation des images digitales sous Matlab

En Matlab ©, une image  $I(m,n)$  est la réponse d'un pixel situé au  $m^{\text{ème}}$  ligne et  $n^{\text{ème}}$  colonne commençant du haut gauche de l'origine 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, pour la figure 1, on peut récupérer les valeurs ou les modifier aisément. Le point  $(m,n)$  est appelé pixel (picture cell), sa valeur  $I(m,n)$  réelle traduit le degré de luminosité de l'image au point  $(m,n)$  : On l'appelle niveau de gris de l'image au point  $(m,n)$  ( $m$  : numéro des lignes et  $n$  : numéro des colonnes).

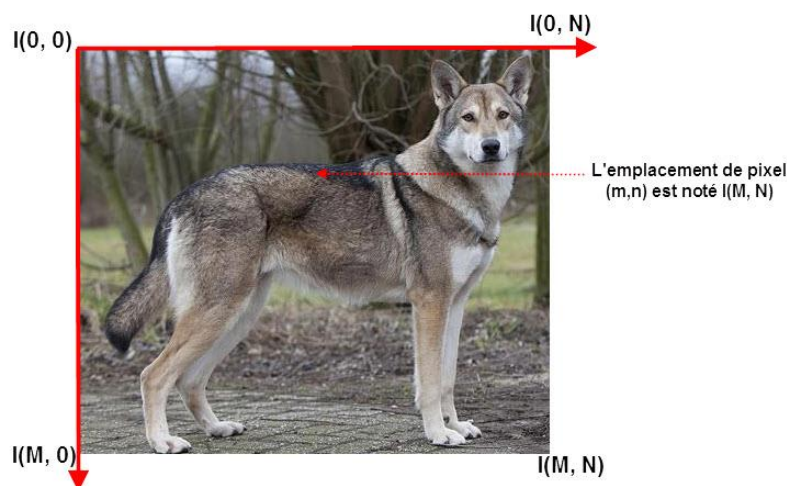


Figure 1 : La représentation de L'images sous Matlab

Une image monochrome peut être définie comme une fonction 2D de l'intensité lumineuse,  $I(x,y)$ , en fonction de la localisation spatiale ( $x$  et  $y$  sont les coordonnées spatiales).

Une image couleur est définie, non plus par une fonction 2D mais 3D. La dimension supplémentaire est égale à 3 car l'image est codée suivant 3 composantes, par exemple le rouge, le vert et le bleu pour un système RVB.

On peut représenter une image numérique par une matrice 2D pour une image monochrome (Figure 2) et par une matrice 3D pour une image couleur (Figure 3).

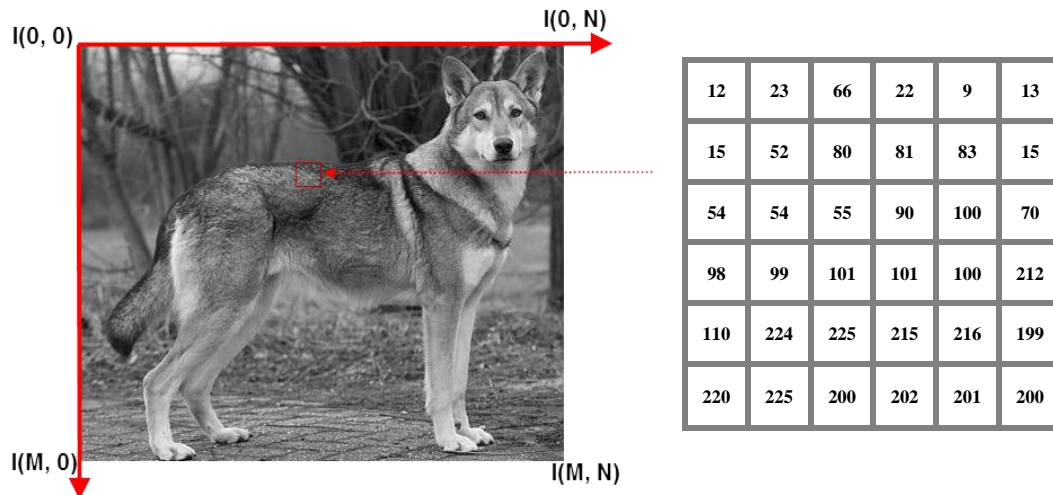


Figure 2 : La représentation d'une image en niveau de gris sous Matlab

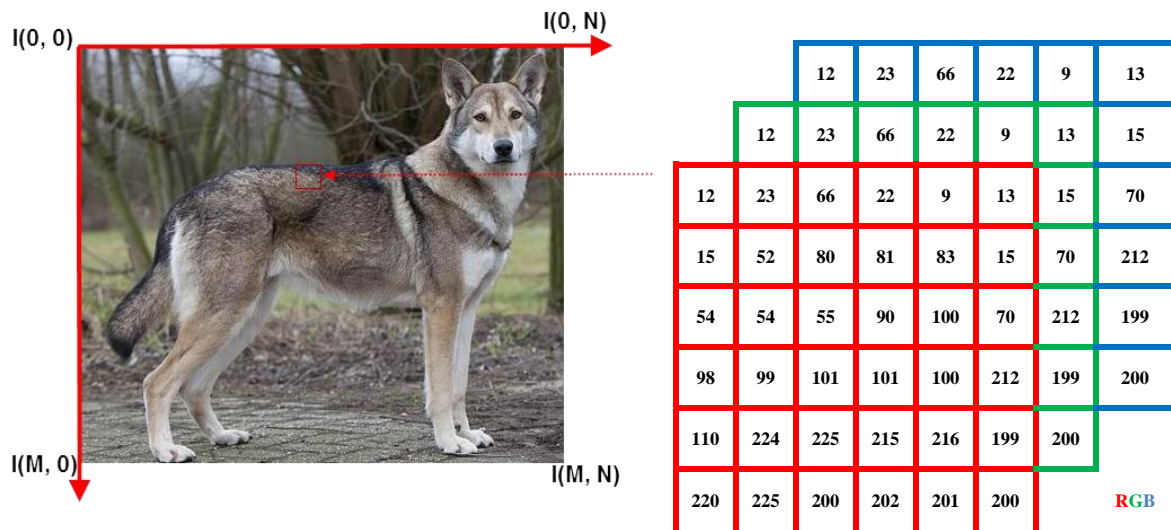


Figure 3 : La représentation d'une image RGB sous Matlab

### Exemple 1 du code Matlab :

```
I1=imread('cameraman.tif'); %Lire l'image et retourner une matrice I1
imfinfo('cameraman.tif'); % Retourne les caractéristiques de l'image
imwrite(I1,'cameraman.jpg','jpg'); % Enregistrer la matrice I1 sous le format JPEG
```

Les principales fonctions de traitement d'images sous Matlab se trouvent dans la boîte à outils (toolbox) image processing (traitement d'images). L'aide sur cette boîte à outils est obtenue en tapant *help images* en ligne de commande de Matlab. Ensuite, l'aide sur une commande particulière est obtenue en tapant *help* suivi du nom de la commande, (ex : *help imshow*).

### Manipulation 1 avec Matlab :

Matlab est capable de lire et de décoder les fichiers images JPEG, TIFF, BMP, PNG, HDF, PCX ou XWD. Une image sous Matlab est représentée sous forme d'une matrice.

Avant de traiter une image dans Matlab, il faut la lire et décoder son format afin de la transformer en une matrice de valeurs.

Ecrire un script qui permet de :

1. Lisez une image avec Matlab © et l'afficher
2. Afficher l'image sur une figure

## II. Traitements sur la couleur et palette

En Matlab ©, l'affichage de l'image se fait à travers un Colormap. Ce dernier associe à chaque nuance de couleur une valeur numérique pour donner une représentation visuelle de l'image.

### Exemple 2 du code Matlab :

```
A=imread('cameraman.tif'); % Lire l'image cameraman
imshow(A);                 % Afficher une image
imagesc(A);                % Une autre méthode d'affichage
axis image;                % Repère de l'image
axis off;                  % Désactiver les labels sur les axes
colormap(gray);            % Afficher l'image en niveaux de gris
```

Le Colormap le plus courant est le « grayscale »- niveaux de gris, qui associe toutes les nuances du gris : du noir (valeur 0 : absence de lumière) jusqu'au blanc (valeur maximale qui peut être 2 exposant n, avec n est un entier naturel).

### Manipulation 2 avec Matlab :

1. Lisez l'aide de la fonction *colormap*.  
Ecrire un script qui permet de :
2. Appliquez un Colormap autre que grayscale sur une image
3. Définissez votre Colormap dans une fonction à part et revenir au script pour l'appliquer

### III. Les séquences d'images et vidéo (multiframe image arrays)

Pour certaines applications, vous pouvez avoir besoin de travailler avec des séquences d'images liées par le temps ou la vue, comme l'imagerie par résonance magnétique (IRM) ou les images vidéo.

Le Toolbox de Matlab de traitement d'images comprend deux fonctions, *immovie* et *montage*, qui fonctionnent avec un type spécifique de matrice multi-dimensionnelle appelée multi frame array, plusieurs images sont stockés dans le même tableau. Chaque image est appelée frame (en anglais).

#### Exemple 3 du code Matlab :

---

```
%Créer un montage à partir d'une liste de fichiers
srcFiles= dir('AT3_1m4_*.tif'); % Liste le contenu du répertoire
fileNames={srcFichiers.name}'; %Liste les noms des images
montage(fileNames,'Size',[2 5]) ; %Affiche un montage des images spécifiées dans
fileNames
```

---

#### Exemple 4 du code Matlab

---

```
%Créer un montage à partir d'une liste de fichiers
load mri ; %Ajoute deux variables à l'espace de travail: D et map
mov = immovie(D,map) ; %retourne la vidéo
imshow(mov) ; % ouvre l'application Video Viewer.
```

---

#### Manipulation 3 avec Matlab :

- 
1. Lisez l'aide de la fonction *cat*.
  2. Créez une matrice multi-dimensionnelle (5 images).
  3. Faire le montage de ces images.
  4. Faire une vidéo en utilisant ces images
- 

#### Manipulation avancées :

- 
1. Récupérer le niveau de gris du pixel (50,100) de l'image tire dans *indemos* ;
  2. Récupérer le quart haut droit de l'image *tire.tif* et le stocker dans la variable *imgQuart*.
-

## Solutions

### Manipulation 1 :

Lisez une image avec Matlab © et l'afficher

```
I=imread('cameraman.tif');
```

Afficher l'image sur une figure

```
figure(1) ; imshow(I) ;
```

### Manipulation 2 :

Appliquez un Colormap autre que grayscale sur une image

```
Io=imread('cameraman.tif');
```

```
imagesc(Io);axis off;axis square;colormap(jet);
```

```
imagesc(Io);axis off;axis square;colormap(summer);
```

Définissez votre Colormap dans une fonction à part et revenir au script pour l'appliquer

Main :

```
Io=imread('cameraman.tif');
```

```
figure(1) ; imagesc(Io);axis off;axis square;colormap(gray);
```

```
figure(2) ; palette=tpmap(50);
```

```
imagesc(Io);axis off;axis square;colormap(palette);
```

Function

```
function palette=tpmap(rows)
```

```
palette = rand(rows,3);
```

```
end
```

### Manipulation 3:

Créez une matrice multi-dimensionnelle (5 images)

```
I1=imread('AT3_1m4_01.tif');
```

```
I2=imread('AT3_1m4_02.tif');
```

```
I3=imread('AT3_1m4_03.tif');
```

```
I4=imread('AT3_1m4_02.tif');
```

```
I5=imread('AT3_1m4_03.tif');
```

```
A = cat(4,I1,I2,I3,I4,I5) ;
```

Faire le montage de ces images.

```
montage(A,'Size',[1 5]) ;
```

Faire une vidéo en utilisant ces images

```
ColorMaps=colormap('gray');
```

```
mov=immovie(A,ColorMaps);
```

```
implay(mov);
```

### Manipulation avancées :

Récupérer le niveau de gris du pixel (50,100) de l'image tire dans imdemos ;

```
I=imread('tire.tif');
```

```
NG=I(50,100) ;
```

Récupérer le quart haut droit de l'image tire.tif et le stocker dans la variable imgQuart.

```
[m n]=size(I) ; quarterMatrix = I (1 : 1 : 102, 1 : 1 : 116);
```

