

TP Télévision Numérique



Université de Tlemcen

Dr. BELGACEM Nassima

Université de Tlemcen

Faculté de technologie

Département de
Télécommunications

E m a i l :

belgacem_nassima@yahoo.fr

1.0

23/02/2024

Table des matières

Objectifs	4
Introduction	5
I - Pré-requis	6
II - Exercice	7
III - Les ressources	10
IV - Chapitre I : Initiation au traitement d'images	11
1. Objectifs du TP	11
2. Qu'est ce qu'une numérisation ?	12
3. L'image numérique	12
4. Différents types d'images numériques	12
4.1. Images binaires (Images Noir et blanc)	12
4.2. Images en niveaux de gris	13
4.3. Images couleur RVB	15
5. La résolution d'une image numérique	16
6. Formats d'images numériques	17
6.1. Les standards universels	17
7. Histogramme	18
8. Formats d'image sur disque	19
9. Fonctions matlab utiles pour la réalisation du TP	19
10. Exemples du code Matlab	21
10.1. Lecture d'une Image	21
10.2. Affichage d'une Image	21
10.3. Détermination de la taille d'une Image	21
10.4. Palette des couleurs d'une Image	21
11. Manipulation	23
11.1. Première manipulation	23
11.2. Deuxième manipulation	24
11.3. Troisième manipulation	26
11.4. Quatrième manipulation	26

12. Solution de TP N°01	27
13. Exercice	30
14. Conclusion	32
Conclusion	33
Solutions des exercices	34
Glossaire	39
Abréviations	42
Bibliographie	43
Webographie	44

Objectifs

1- Comprendre, manipuler et maîtriser les commandes de base pour la manipulation des images numériques sous Matlab.

2-Lecture et affichage d'une image.

3-Extraire les matrices de couleur Rouge (R), Vert (G), et Bleu (B) d'une image couleur.

4-Identifier et extraire la composante de luminance d'une image.

5- Initiation à la compression des images fixes et vidéo.

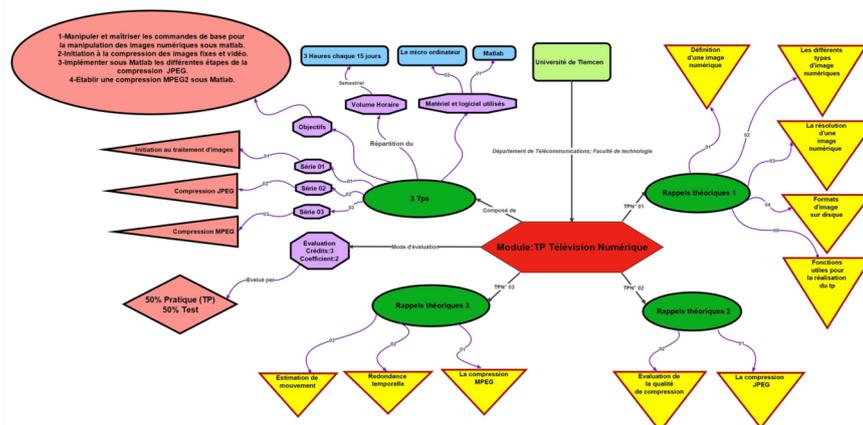
6- Implémenter sous Matlab les différentes étapes de la compression JPEG.

7-Identifier les algorithmes et les techniques utilisés dans la compression JPEG.

Introduction

La télévision numérique offre un aperçu essentiel sur l'évolution des technologies de diffusion et de réception des programmes télévisuels. Avec l'avènement de la télévision numérique, les signaux audiovisuels sont transformés en données binaires, permettant une transmission plus efficace et offrant de multiples avantages par rapport à la télévision analogique. Cette transition vers la télévision numérique représente une avancée majeure dans l'histoire de la diffusion télévisuelle. Contrairement à la télévision analogique qui transmet des signaux sous forme d'ondes continues, la télévision numérique encode les signaux en données numériques, ce qui permet une utilisation plus efficace de la bande passante et une meilleure qualité d'image et de son. L'une des caractéristiques les plus remarquables de la télévision numérique est sa capacité à offrir une gamme plus étendue de canaux et de services, tels que la haute définition (HD), la télévision à la demande (VOD*), et la télévision interactive. De plus, la télévision numérique permet l'intégration de services supplémentaires, tels que l'accès à internet et la transmission de données complémentaires liées au programme diffusé. La télévision numérique présente également des avantages significatifs en termes de qualité d'image et de son. Grâce à la compression des données, les images sont généralement plus nettes et les sons plus clairs. De plus, la transmission numérique réduit les interférences et les distorsions, offrant ainsi une expérience de visionnage plus stable et cohérente. En résumé, l'avènement de la télévision numérique marque une évolution majeure dans le domaine de la diffusion télévisuelle, offrant une meilleure qualité, une plus grande variété de contenus et une expérience utilisateur améliorée.

Enseignant: Dr.BELGACEM Nassima
Département Télécommunications



I Pré-requis

Il est utile de prendre en compte quelques prérequis qui peuvent faciliter l'apprentissage et la compréhension des concepts fondamentaux

1-Il est essentiel de comprendre les concepts fondamentaux du traitement d'image, tels que la résolution, la couleur et les formats de fichiers.

2-Acquérir au moins une connaissance de base en programmation est crucial, car la manipulation d'images implique souvent l'utilisation de langages de programmation pour répondre aux besoins spécifiques dans ce domaine.

3-Connaître les différentes espaces de couleurs.

4-Comprendre les principales méthodes de synthèse des couleurs pour les images numériques.

5-Maîtrise d'au moins un langage de programmation, car la mise en œuvre de la compression JPEG implique souvent la manipulation de données et d'algorithmes.

6-Distinguer entre les formats d'images afin de les compresser.

Pour tester ces six pré-requis, un test est mis à votre disposition

II Exercice

[solution n°1 p.34]

Exercice

1- Quel est le processus qui crée une image numérique à partir d'une image analogique ?

- La numérisation de données
- La conversion numérique-analogique
- La digitalisation d'images
- La compression d'images

Exercice

2- Quels sont les concepts fondamentaux du traitement d'image ?

- Résolution, Compression, Conversion
- Résolution, Couleur, Formats de fichiers
- Transparence, Opacité, Contraste
- Interpolation, Éclairage, Texture

Exercice

3- Quelles fonctions MATLAB sont couramment utilisées dans le traitement d'images ?

- imread : Pour lire une image à partir d'un fichier.
- imshow : Pour afficher une image à l'écran.
- randn : Pour générer des valeurs aléatoires selon une distribution normale

Exercice

4- Comment les couleurs sont-elles souvent représentées dans les images numériques ?

- En utilisant des valeurs de luminosité uniquement
- En utilisant des valeurs de teinte et de saturation uniquement
- En utilisant des matrices de pixels RVB (Rouge, Vert, Bleu)

- En utilisant des valeurs de transparence uniquement

Exercice

5- Comment pouvez-vous convertir une image RGB en une image en niveaux de gris en utilisant MATLAB ?

- rgb2gray
- gray2rgb
- convert2gray
- grayscale

Exercice

6-Qu'est-ce que le traitement d'images numériques implique généralement ?

- La conversion d'images analogiques en images numériques
- La manipulation et la modification d'images à l'aide d'algorithmes informatiques
- La création d'images artistiques à l'aide de logiciels spécifiques

Exercice

7-Quelle est l'importance de la maîtrise d'un langage de programmation dans le contexte de la compression JPEG ?

- Permettre de créer des images numériques à partir de zéro
- Faciliter la compréhension des concepts de base de la compression d'images
- Aider à mettre en œuvre les algorithmes de compression et à manipuler les données nécessaires

Exercice

8-Quelles compétences sont essentielles pour comprendre les fondamentaux du traitement d'images numériques ?

- Connaissance des méthodes de cuisson d'images
- Capacité à reconnaître les noms de différents logiciels de retouche d'images
- Compréhension des concepts de résolution, de couleur et de format d'images

Exercice

9-Pourquoi la maîtrise des bases du traitement d'images numériques est-elle importante pour la compression JPEG ?

- Pour pouvoir identifier les types de compression les plus efficaces

- Pour comprendre l'impact des différents algorithmes de compression sur la qualité de l'image
- Pour connaître les différents formats de fichiers d'images et leurs avantages

Exercice

10-Quel est l'avantage de la maîtrise d'un langage de programmation dans la mise en œuvre de la compression JPEG ?

- Permettre de créer des animations vidéo à partir d'images fixes
- Faciliter la collaboration avec d'autres concepteurs graphiques
-

Aider à personnaliser et à optimiser les algorithmes de compression pour répondre à des besoins spécifiques

III Les ressources

Pour aider les étudiants rencontrant des difficultés, nous recommandons vivement de consulter les ressources suivante

http://elearning.univ-biskra.dz/moodle2019/pluginfile.php/20103/mod_resource/content/4/Chapitre2_papier.pdf

http://igm.univ-mlv.fr/~dr/XPOSE2013/La_compression_de_donnees/jpeg.html

IV Chapitre I : Initiation au traitement d'images

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années. On désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations. De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

Par conséquent, le traitement d'images ne se limite pas à une seule approche ou une seule application, mais englobe une multitude de méthodes et de techniques qui peuvent être adaptées à différents besoins et domaines d'application. Que ce soit dans le domaine médical, l'industrie, la surveillance, la vision par ordinateur, ou d'autres domaines encore, le traitement d'images joue un rôle crucial dans l'exploitation et l'analyse des données visuelles dans le monde moderne.

<https://www.youtube.com/watch?v=lhbAwAgv1dU>

1. Objectifs du TP

Dans le cadre de ce TP sur le traitement d'images numériques sous Matlab, nous aborderons les objectifs suivants

1. Compréhension des fondements du traitement d'images numériques sous Matlab.
2. Définir une connaissance adéquate des caractéristiques propres aux images
3. Comprendre, manipuler et maîtriser les commandes de base pour la manipulation des images numériques sous Matlab.
4. Maîtrise des commandes de base pour la lecture et l'affichage d'une image dans Matlab.
5. Comprendre les concepts de représentation des couleurs dans une image numérique.
6. Application des commandes appropriées pour convertir une image en niveaux de gris (RVB vers niveaux de gris).
7. Utiliser les outils Matlab pour analyser les valeurs des pixels dans les matrices de couleur.

2. Qu'est ce qu'une numérisation ?

La numérisation* fait référence au processus de conversion d'informations analogiques* en données numériques*. Cela implique de prendre des données sous forme analogique, telles que des images, du texte ou du son, et de les transformer en une série de chiffres binaires* (0 et 1) compréhensibles par un ordinateur ou d'autres dispositifs électroniques.

Par exemple, la numérisation d'une image implique de convertir les variations de lumière et de couleur en pixels*** numériques. De même, la numérisation d'un document implique de convertir le texte imprimé en un format numérique lisible par une machine.

La numérisation joue un rôle crucial dans de nombreux domaines, y compris la photographie, l'archivage de documents, la production de films, la musique, etc. Elle permet le stockage, la manipulation et le partage efficaces des informations, ainsi que leur traitement automatisé* par des logiciels* et des algorithmes**.

3. L'image numérique

Une image numérique* est une représentation visuelle stockée sous forme de données numériques, généralement dans un format informatique spécifique*. Elle est composée de pixels, chaque pixel étant un point élémentaire de l'image contenant des informations de couleur et de luminosité. Les images numériques peuvent être créées à l'aide d'appareils photo numériques***, de scanners*** ou de logiciels de création graphique*, et elles sont largement utilisées dans divers domaines tels que la photographie*, la conception graphique*, l'illustration*, le cinéma, la médecine*, etc. [1]*.

4. Différents types d'images numériques

Il existe plusieurs types d'images numériques, chacun ayant ses propres caractéristiques et utilisations spécifiques. Voici quelques-uns des types d'images numériques les plus courants, avec leurs définitions :

4.1. Images binaires (Images Noir et blanc)

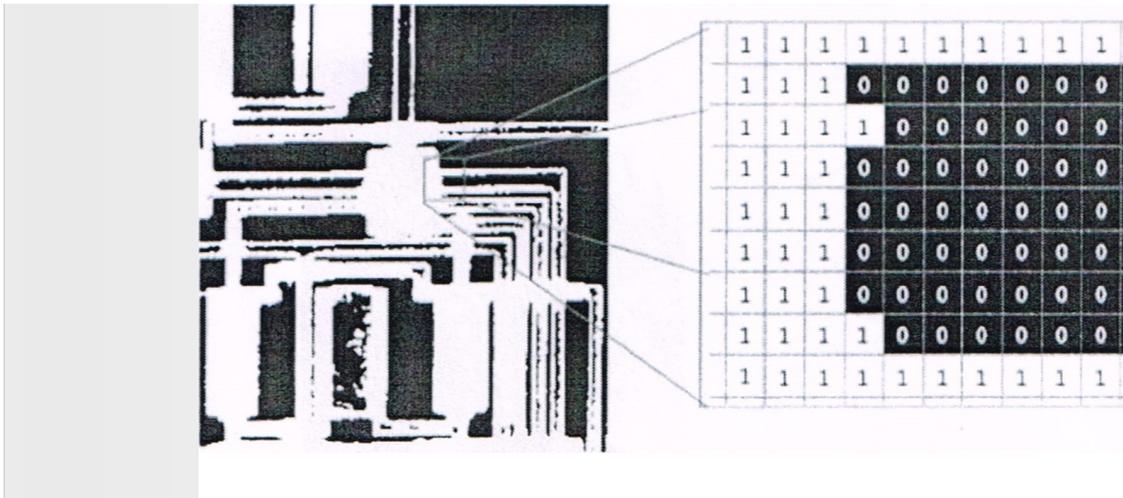
Une image binaire* est un type d'image numérique dans lequel chaque pixel ne peut prendre que deux valeurs distinctes*, généralement représentées par le noir* et le blanc*. Ces valeurs sont souvent associées à des états logiques*, où le noir peut représenter la présence d'un objet ou une caractéristique spécifique* dans une région donnée de l'image*, tandis que le blanc peut représenter son absence [2].***

Fondamental

Chaque pixel est représenté par un bit (0/1) (I(x,y) et (0,1).

0 : noir, 1 : blanc

L'image binaire est un tableau de 0 et de 1.



Les valeurs des pixels dans une image binaire

4.2. Images en niveaux de gris

Une image en niveaux* de gris est une image où chaque pixel a une valeur représentant son niveau de luminosité**, allant du noir au blanc.

Elles sont utilisées dans les applications où la couleur n'est pas nécessaire, comme dans les diagrammes scientifiques ou les images médicales*.

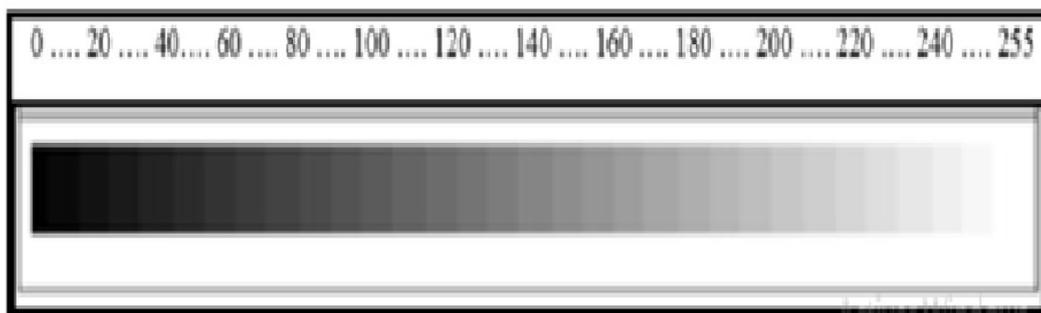
• Dans ce cas on dispose d'une échelle* de NG, chaque pixel est représenté par x bits : 256 NG 8bits $[(x,y) \in \mathbb{Z} \times \mathbb{Z}]$.

• 0 : noir ,127 : gris moyen , 255 : blanc

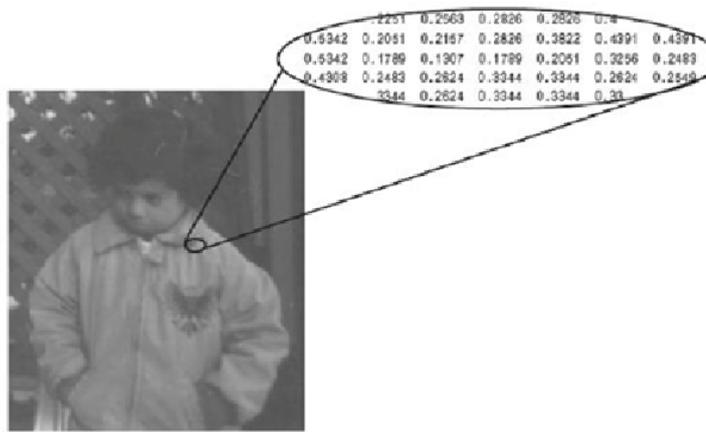
Exemple

Image en niveaux de gris (8 bits) de taille 128 × 128 :

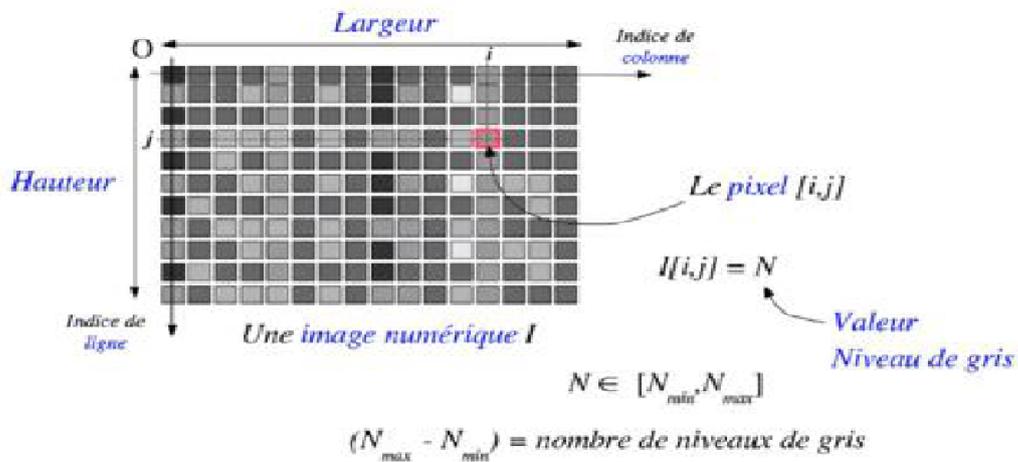
$128 \times 128 \times 8 = 131072$ octets = 16 Koctets



Valeurs des niveaux de gris et teintes de gris correspondantes



Les valeurs des pixels dans une image d'intensité définissent les niveaux de gris



Le point (i,j) est appelé pixel (picture cell). Pour une image monochrome, la valeur du pixel (i,j) réelle traduit le degré de luminosité de l'image au point (i,j) : On l'appelle niveau de gris de l'image au point (i,j) (i : numéro des lignes et j : numéro des colonnes).

L'accès à un élément particulier d'une image est indexé par le nom et la position de cet élément.

Une image monochrome « Figure 2 » peut être définie comme une fonction 21) de l'intensité lumineuse, (x,y), en fonction de la localisation spatiale (x et y sont les coordonnées spatiales).

On peut représenter une image en niveaux de gris (monochrome) par une matrice 21) d'entiers dont les valeurs représentent des nuances de gris « Figure 2 ». Les pixels d'une image en niveaux de gris sont généralement de type uint 8 (ou uint 16), ils ont des valeurs entières dans la plage [0, 255] [vers , 65535]).

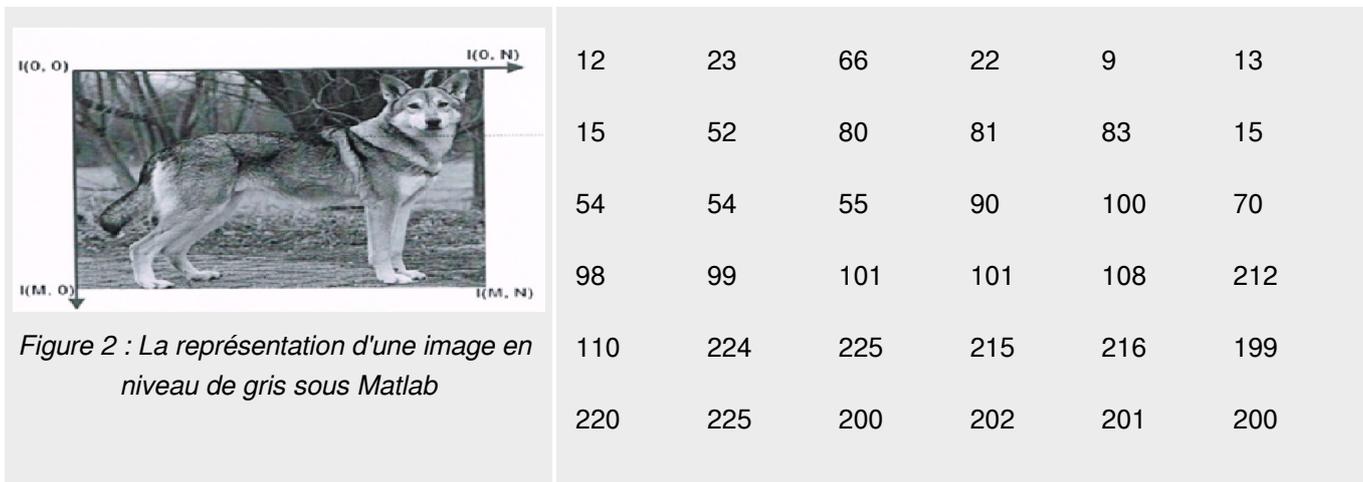


Figure 2 : La représentation d'une image en niveau de gris sous Matlab

4.3. Images couleur RVB

Une image couleur est composée de pixels ayant chacun une valeur pour les composantes rouge, verte et bleue (RVB^{*}), ou d'autres systèmes de couleur comme CMJN^{*}. La combinaison de ces valeurs crée une large gamme de couleurs. Les images couleur sont utilisées dans la plupart des applications graphiques, y compris la photographie, les graphiques informatiques et la conception web[3]^{*}.

Ces images sont codées en utilisant les trios couleurs fondamentales **RVB**. (l'image est constituée de 3 plans ou 3 matrices: 1 matrice pour chaque couleur).

Une image couleur « figure 3 » 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 couleur par une matrice 3D « Figure 3 ». Chaque pixel est un triplet correspondant aux composantes rouge, vert et bleu.

- Chaque couleur est codée sous forme d'un octet (8 bits: intensité de 0 à 255).
- on code ainsi **16777216** couleurs différentes.
- **Image couleur RVB**: $I_R(x,y) \ I_V(x,y) \ I_B(x,y)$.

Exemple

Image en couleurs (32 bits) de taille 256 × 256 :

$$256 \times 256 \times 32 = 256 \text{ Koctets}$$

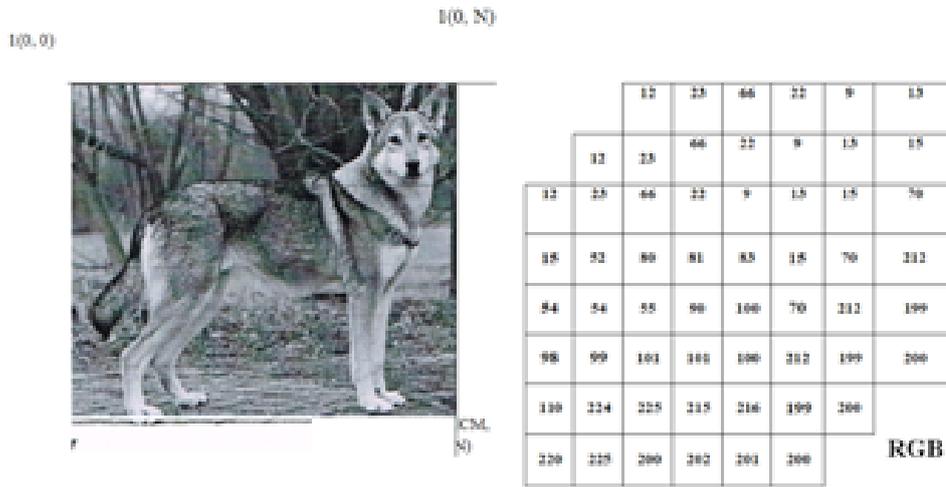
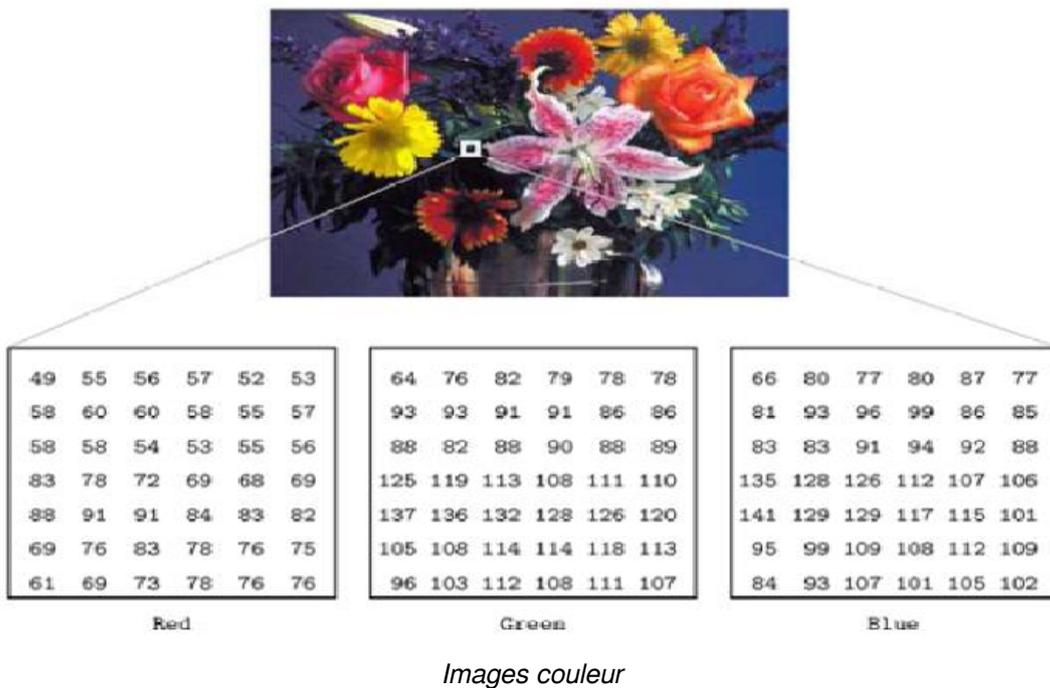


Figure 3 : La représentation d'une image RGB sous Matlab



5. La résolution d'une image numérique

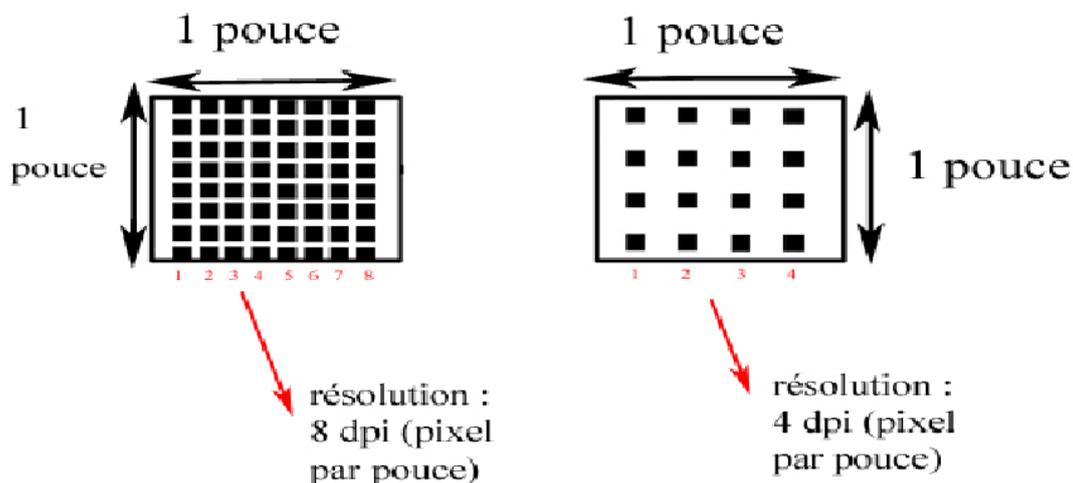
La résolution d'une image est le nombre de pixels contenus dans l'image par unité de longueur. Elle s'exprime le plus souvent en ppp (point par pouces) ou en dpi*, parfois en point par cm.

Exemple

Une résolution de **300 dpi** signifie que l'image comporte **300** pixels dans sa largeur et **300** pixels dans sa hauteur, elle est donc composée de **90000** pixels (**300 x 300 ppp**).

Rappel

1 pouce (ou inch) 2,54cm.



6. Formats d'images numériques

Les formats d'images numériques sont des normes utilisées pour encoder et stocker des images numériques sur des ordinateurs et d'autres dispositifs électroniques. Chaque format d'image numérique a ses propres caractéristiques, avantages et inconvénients. Voici quelques-uns des formats d'images numériques les plus couramment utilisés :

- **JPEG*** : C'est l'un des formats les plus répandus pour les photographies numériques. Il utilise une compression avec perte, ce qui signifie que des informations sont perdues lors de la compression, mais il permet de réduire considérablement la taille du fichier.
- **PNG*** : Le format **PNG** est couramment utilisé pour les images avec des zones de transparence. Contrairement au **JPEG**, il utilise une compression sans perte, préservant ainsi la qualité de l'image, mais généralement avec des tailles de fichiers plus importantes.
- **GIF*** : Les **GIF** sont souvent utilisés pour les images animées ou pour les images simples avec une palette de couleurs limitée. Ils utilisent une compression sans perte, mais sont généralement limités à 256 couleurs.
- **BMP*** : Les fichiers **BMP** stockent des images sous forme de matrices de pixels. Ils peuvent être non compressés ou compressés sans perte, ce qui signifie que les fichiers **BMP** peuvent être très volumineux.

6.1. Les standards universels

Les standards universels tels que **GIF**, **Bitmap** et **TIFF** sont largement utilisés dans divers domaines pour stocker et échanger des images numériques, offrant une compatibilité étendue avec de nombreux logiciels et plates-formes. En revanche, les standards médicaux comme **DICOM** sont essentiels pour l'imagerie médicale, garantissant l'interopérabilité et la cohérence des données entre les différents équipements et systèmes de santé. Enfin, les standards propriétaires, tels que ceux développés par des entreprises telles que **Philips** et **Siemens**, sont souvent spécifiques à leurs produits et nécessitent des logiciels propriétaires pour leur accès et leur traitement.

on peut cité ces différent standards :

- standards universels : **GIF**, **bitmap**, **tiff***.
- Standards médicaux : **DICOM***.
- Standards propriétaires : **Philips**, **Siemens**.

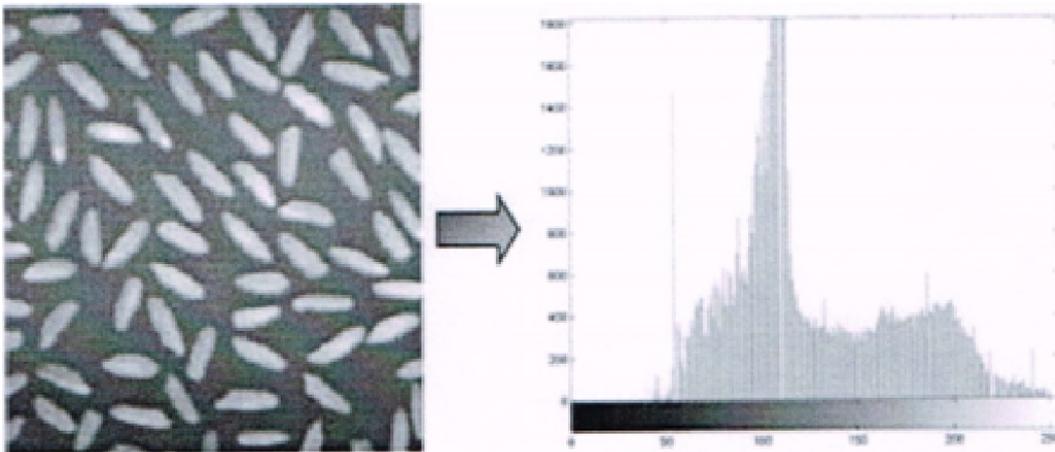
🔗 Exemple

- **BMP (Bitmap)** : matrice de bits codés en couleur (jusqu' à 24 bits/pixel).
- **GIF** : format compressé avec codage 8 bits/pixel.
- **JPG* (jpeg)** : format de compression d'images photographiques.

7. Histogramme

Il s'agit de répartir les niveaux de gris en un ensemble d'intervalle de même taille et ensuite de représenter le nombre de pixels de chaque intervalle en fonction de la moyenne de ces intervalles. L'histogramme ne dépend pas de la façon dont les pixels sont organisés dans l'image. Cela peut aussi se voir comme la probabilité d'observer un pixel de tel niveau de gris. La figure suivante montre l'histogramme.

Une image de 256 * 256 pixels, codée sur 8 bits par pixel, aurait une taille de 256 pixels de large par 256 pixels de haut, avec chaque pixel pouvant prendre 256 valeurs différentes (de 0 à 255). Cette image aurait une taille totale de $256 * 256 = 65\,536$ pixels.



🔗 Exemple : Exemple simple de calcul d'histogramme pour une image

- L'image «A» comporte 3 niveaux de gris différents: 0, 1 et 2.
- Compter le nombre de pixels pour chaque niveau de gris, à l'aide de la matrice des valeurs de luminance.
- Les niveaux 0, 1 et 2 sont respectivement représentés par 24, 12 et 28 pixels => représentation de cette population de pixels sur l'histogramme.

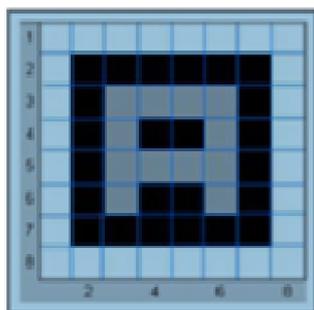
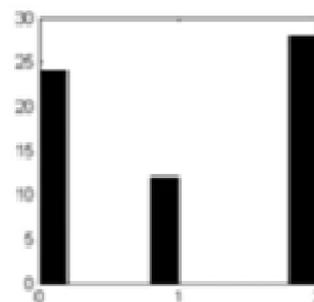


Image « A » en
niveaux de gris

2	2	2	2	2	2	2	2
2	0	0	0	0	0	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	0	0	0	0	0	2
2	2	2	2	2	2	2	2

Matrice des valeurs de
luminance des pixels de
l'image « A »



Histogramme de
l'image « A »

8. Formats d'image sur disque

Sur un disque, les images peuvent être stockées dans différents formats, chacun ayant ses propres caractéristiques en termes de compression, de qualité et de compatibilité. Voici quelques-uns des formats d'image les plus courants :

- Le format **Matlab** (fichiers *.mat)
- Les formats graphiques standard : les formats jpeg et tiff (fichiers *.jpg et *.tif)

Pour le choix d'un format :

- Format tiff : images binaires (compression parfaite)
- Format jpeg : images de scènes réelles
- Format matlab : tout type d'image (sans compression), lorsque l'on souhaite éviter toute dégradation

9. Fonctions matlab utiles pour la réalisation du TP

Voici une liste de fonctions MATLAB utiles pour la réalisation d'un TP (Travaux Pratiques) impliquant le traitement d'images :

`imread`: Charge une image à partir d'un fichier.

Exemple : `img = imread('image.jpg');`

`imshow`: Affiche une image.

Exemple : `imshow(img);`

`imhist`: Affiche l'histogramme d'une image.

Exemple : `imhist(img);`

`size`: Retourne la taille d'une matrice (ou d'une image).

Exemple : `image size = size(img);`

`imfinfo`: Retourne des informations sur un fichier image.

Exemple : `info = imfinfo('image.jpg');`

`rgb2gray`: Convertit une image RGB en une image en niveaux de gris.

Exemple : `grayImg = rgb2gray(rgbImg);`

`colormap`: Définit ou récupère la carte de couleurs actuelle.

Exemple : `colormap(gray);`

`rgb2hsv`: Convertit une image RGB en une image HSV.

Exemple : `hsvImg = rgb2hsv(rgbImg);`

Ces fonctions peuvent être utilisées ensemble pour effectuer diverses opérations de traitement d'images, telles que la conversion de couleurs, l'affichage d'histogrammes, etc.

10. Exemples du code Matlab

10.1. Lecture d'une Image

Sous Matlab, on peut lire des images aux formats : jpg,tif, png, bmp

Pour lire une image, on utilise la commande suivante .

```
im =imread(filename,ext)
```

filename : est le nom de l'image.

ext : est l'extension du format de stockage de l'image.

Exemple :

```
im = imread('cameraman', 'tif) ou im = imread('cameraman.tif)
```

Si l'image n'est pas située dans .

- le dossier MATLAB (1)
- ou dans le dossier où sont sauvegardées les images de démonstration de MATLAB(2) Il faudra préciser, dans filename, le chemin (la racine) vers le dossier qui héberge l'image

Exemple :

```
im = imread('C:\Users \Public\Pictures \Sample Pictures\koala', 'jpg')
```

10.2. Affichage d'une Image

Pour afficher une image, on utilise la commande suivante imshow(im) : Permet d'afficher l'image «im».

Exemple :

```
('cameraman.tif');
```

```
imshow(im) ou imshow(cameraman.tif);
```

10.3. Détermination de la taille d'une Image

[m,n,d]=size(im) : Permet de connaître la taille d'une image, où :

- m est le nombre de ligne et n et le nombre des colonnes de l'image
- Si l'image est .
 - 1-à niveau de gris, d=1
 - 2-en couleur, d=3

Exemple :

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

```
[m,n,d]=size(im);
```

10.4. Palette des couleurs d'une Image

```
cm = colormap • colormap(cm)
```

Permet :

- d'extraire la palette des couleurs en cours et l'affecte à la variable `cm`
- Ou de modifier la palette des couleurs selon le contenu de `cm`
- Il existe des palettes de couleurs préprogrammées sous MATLAB, telles que : 'jet', 'hsv'. (voir `help colormap`)

11. Manipulation

11.1. Première manipulation

Toutes les images utilisées sont des images de démonstration de MATLAB, donc il n'y a pas lieu de préciser le dossier qui héberge l'image.

1. Lire et afficher l'image à niveau de gris 'coins.png', affecter-la à la variable im1.
2. Afficher-la.
3. Aller au gestionnaire des variables (Workspace) et déduire : la taille et le type de la variable im1.
4. Lire et afficher l'image en couleur 'onion.png', affecter-la à la variable im2.

Dans le gestionnaire des variables, déduire : la taille et le type de la variable de im2, Que présentent : im1(: , : , 1) , im2(: , : , 2) et im3(: , : , 3).

5. Définir les tailles des deux images im1 et im2 au moyen de la commande "size", et valider les observations faites dans les questions 3 et 4.
6. im2 est une image couleur de type RGB. Pour extraire les matrices de couleur rouge, R, de couleur verte, G, et de couleur bleu, B, on procèdera comme suit :

```
% extraction des matrices RGB
im2_R=im2(: , : , 1);
im2_G=im2(: , : , 2) ;
im2_B=im2(: , : , 3)
% affichage figure(3)
subplot(131),imshow(im2_R); title('R');
subplot(132), imshow(im2_G); title('G');
subplot(133), imshow(im2_B); title('B');
```

7. Les images R, G et B sont affichées en niveaux de gris (intensité de la lumière), pour les afficher en couleur, il faudra changer la palette des couleurs (colormap), et les afficher dans des figures indépendantes :

```
% affichage
cm=colormap; % contient les 3 palettes RGB
% garder que la couleur R
cmR=cm ; cmR( :,2)=0; cmR( :,3)=0 ;
figure, imshow(im2_R); title('R'); colormap(cmR);
% garder que la couleur G
cmG=cm; cmG(:,1)=0; cmG(:,3)=0;
figure, imshow(im2_G); title('G'); colormap(cmG);
% garder que la couleur B
```

```
cmB=cm; cmB(:,1)=0; cmB(:,2)=0 ;
```

```
figure, imshow(im2_B); title('B'); colormap(cmB);
```

Aller au gestionnaire des variables pour visualiser le contenu de cmR, cmG et cmB.

Commenter.

Remarque

On ne peut pas afficher les images dans la même figure, car une figure prend en compte une seule palette de couleur

11.2. Deuxième manipulation

La luminance d'une image correspond à l'intensité de lumière entrant dans la formation de chaque pixel.

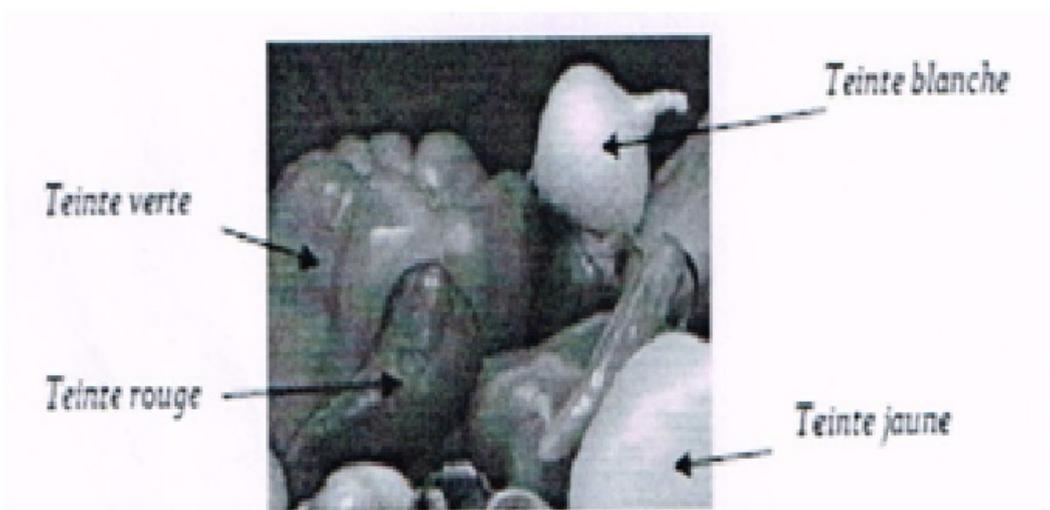
- Pour une image en niveaux de gris I, est elle-même la luminance.
- Pour une image en couleur RGB, la luminance est :

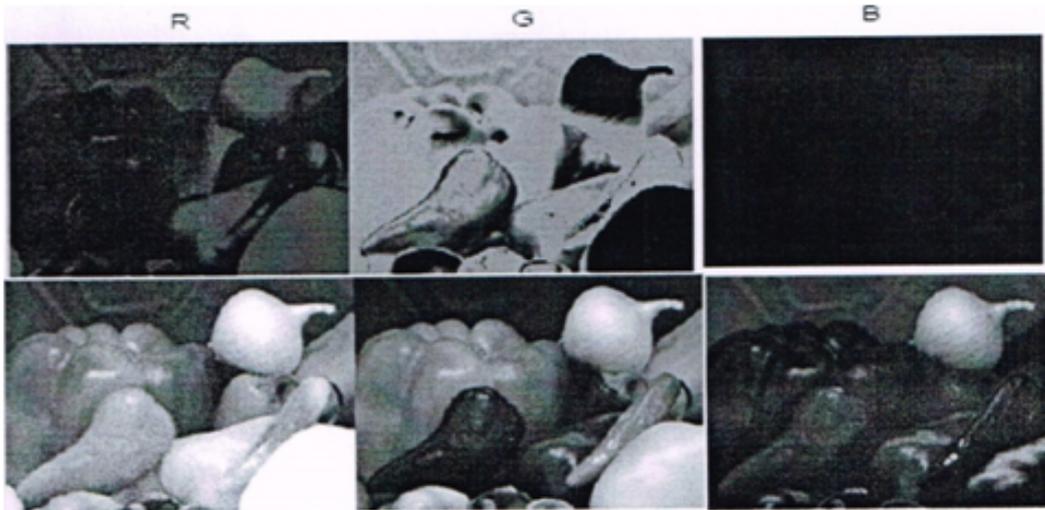
$$I = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (1)$$

Sous MATLAB pour créer la luminance d'une image couleur, il existe la commande `im_gray = rgb2gray(im_RGB)`

Convertit une image couleur im RGB en une image en niveau de gris (luminance) im gray Dans la figure suivante .

- Analyser les images couleurs R, G et B avec leurs luminances (intensité de la lumière) et comparer avec l'image originale.
- Pourquoi l'intensité du B est faible et celle du vert est élevée.





1. En utilisant l'expression de la luminance, créer l'image luminance associée à l'image couleur 'onion.png' qui est lue et stockée dans im2 et dont les matrices couleurs R, G et B sont extraites respectivement dans im2_R, im2_G et im2_B. Afficher-la.

2. Au lieu de ce calcul, utiliser à présent la commande rgb2gray pour créer cette même image luminance. Afficher-la dans la même figure.

3. Si on vous dit que les images obtenues dans la question 7 de la manipulation 1, représentent respectivement les luminances du Rouge, du Vert et du Bleu (intensités des lumières Rouge, Verte et Bleue). On propose une autre démarche pour afficher les matrices couleurs sans manier le Colormap.

- Création des luminances des couleurs R, G et B

```
[m, n, d] = size (im2) ;
imm=zeros (m, n, d) ;
immR=imm; immG=imm, immB=imm,.
immR ( : , : , 1 ) =mat2gray (im2_R) ;
immG ( : , : , 2 ) =mat2gray (im2_G) ;
immB ( : , : , 3 ) =mat2gray (im2_B)
```

- Affichage des matrices couleurs

```
subplot (231) , imshow (im2_R) ; title ('R') ;
subplot (232) , imshow (im2_G) ; title ('G') ;
subplot (233) , imshow (im2_B) ; title ('B') ;
```

- Affichage des luminances couleurs

```
subplot (234) , imshow (imm R) ;
subplot (235) , imshow (imm G) ;
subplot (236) , imshow (immB) ;
```

Comparer les images couleurs R, G et B avec leurs luminances.

11.3. Troisième manipulation

L'espace de colorimétrie HSV représente l'image par trois grandeurs : la teinte H (Hue), la saturation S et la luminance V (Value).

Sous MATLAB pour transformer une image RGB en en image HSV, ou vis-versa, on utilise la commande .

```
[H S V] = rgb2hsv(im_RGB)
```

```
Im_HSV = rgb2hsv(im_RGB)
```

Convertit une image couleur im_RGB de type RGB en une image HSV

```
[R G B] = hsv2rgb(im_HSV)
```

 Convertit une image couleur im_HSV de type HSV en une image RGB.

1.Créer les composantes H, S et V associée à l'image couleur 'onion.png', qui a été lue et stockée dans im2.

2.Afficher la composante H en couleur (colormap hsv) et les composantes S et V en niveaux de gris. Commenter.

```
[H S V] = rgb2hsv(im2) ;
```

```
figure, subplot(121), imshow(im2);
```

```
subplot(122), imshow(H); title('H'); colormap('hsv')
```

```
figure, subplot(121), imshow(S); title('S');
```

```
subplot(122), imshow(V); title('V'); colormap('gray')
```

3.A présent créer l'image HSV associé à l'image associée à l'image couleur 'onion.png' et afficher-la.

```
imHSV=rgb2hsv(im2); figure, imshow(imHSV);
```

11.4. Quatrième manipulation

L'espace de colorimétrie YUV représente l'image par trois grandeurs . la luminance Y, les chrominances U et V, selon les expressions :

$$Y=0.2989*R+0.5870*G+0.1140*B$$
$$U=0.5960*R+0.2740*G+0.3220*B$$
$$V=0.2110*R+0.5230*G+0.3120*B$$

Ce système est utilisé en télévision.

Sous MATLAB pour transformer une image RGB en en image YUV, ou vis-versa, on utilise la commande .

```
[H S V] = rgb2ntsc(im_RGB)
```

Ou

```
Im_YUV = rgb2ntsc(im_RGB)
```

Convertit une image couleur im_RGB de type RGB en une image NTSC de type YUV.

Convertit une image NTSC de type im_YUV en une image couleur de type RGB.

1.Créer puis afficher l'image im_YUV, de type YUV associée à l'image 'onion.png', qui a été lue et stockée dans im2.

```
imYUV=rgb2ntsc (im2); figure, imshow(imYUV);
```

2.A présent, extraire les composantes Y, U et V à partir de l'image imYUV, puis afficher-les.

```
Y=imYUV(:,:,1); subplot(131), imshow(Y); title('Y') ;
```

```
U=imYUV(:,:,2); subplot(132), imshow(U); title('U') ;
```

```
V=imYUV(133) ;subplot(133) ; imshow(V), title('V') ;
```

3. Tester d'autres palettes de couleurs pour afficher les composantes U et V.

12. Solution de TP N°01

Le programme effectue plusieurs manipulations sur des images. Dans la première manipulation, il affiche les composantes RGB* d'une image et permet la sélection de canaux de couleur spécifiques. Ensuite, il convertit l'image en niveaux de gris et crée des matrices de luminance pour chaque composante de couleur. La troisième manipulation consiste à convertir l'image en espace de couleur HSV* (Teinte, Saturation, Valeur) et à afficher ses composantes. Enfin, il convertit l'image en espace de couleur YUV (Luminance, Chrominance) et affiche ses composantes.

Voici le programme sous MATLAB qui effectue les quatre manipulations

```
1 Première manipulation : Affichage des composantes RGB d'une image et sélection de canaux de
  couleur spécifiques.
2 clc
3 clear all
4 close all
5 %Io=imread('cameraman.tip');coins.pm
6 im1=imread('coins.um');
7 [M,N,D]=size(im1);
8 figure(1)
9 imshow(im1);
10 im2=imread('onion.onc');
11 [m,n,d]=size(im2);
12 figure(2)
13 imshow(im2);
14 im2_R=im2(:,:,1);
15 im2_G=im2(:,:,2);
16 im2_B=im2(:,:,3);
17 %affichage
18 figure(3)
19 subplot(131),
20 subplot(132),
21 subplot(133),
22 imshow(im2_R); title('R');
23 imshow(im2_G);title('G');
24 imshow(im2_B);title('B');
25 cm=colormap; contient les 2 palettes RGB garder que la couleur H
26 cmR=cm;cmR(:,2)=0;cmR(:,3)=0;
27 figure(4), imshow(im2_R);title('R');colormap(cmR);
28 %garder que la couleur G
29 cmG=cm;cmG(:,1)=0;cmG(:,3)=0;
30 figure(5),
31 imshow(im2_G);title('G');colormap(cmG);
32 cmB=cm;cmB(:,1)=0;cmB(:,2)=0;
33 figure(6),
34 imshow(im2_B);title('B');colormap(cmB);
35 deuxième manipulation: Conversion de l'image en niveaux de gris et création de matrices de
  luminance pour chaque composante de couleur.
36 question1
```

```

37 im_gray1= 0.2989 *im2_R +0.5870*im2G+0.1140*im2_B;
38 figure(7)
39 imshow(im_gray1);
40 question2
41 im_gray=rgb2gray(im2);
42 figure(8)
43 imshow(im_gray);
44 question3
45 Création des luminances des couleurs R,GetB
46 [m,n,d]=size(im2);
47 imm=zeros(m,n,d);
48 immR=imm;
49 immG=imm;
50 immB=imm;
51 immR(:,:,1)=mat2gray(im2_R);
52 immG(:,:,2)=mat2gray(im2_G);
53 immB(:,:,3)=mat2gray(im2_B);
54 % affichage
55 figure(9)
56 subplot(231),imshow(im2_R);title('R');
57 subplot(232),imshow(im2_G);title('G');
58 subplot(233),imshow(im2_B);title('B');
59 % affichage des luminanes couleurs
60 subplot(234),imshow(immR);
61 subplot(235),imshow(immG);
62 subplot(236),imshow(immB);
63 Troisième manipulation: Conversion de l'image en espace de couleur HSV (Teinte, Saturation,
    Valeur) et affichage des composantes HSV
64 [HSV]=rgb2hsv(im2); figure(10)
65 subplot(121),
66 imshow(im2);
67 subplot(122),
68 imshow(H);title('H');colormap('hsv')
69 figure(11)
70 subplot(121),imshow(S);title('S');
71 subplot(122),imshow(V);title('V');colormap('gray');
72 Quatrième manipulation: Conversion de l'image en espace de couleur YUV (Luminance, Chrominance
    ) et affichage des composantes YUV
73 % imYUV=rgb2hsv(im2);
74 imYUV=rgb2ntsc(im2);
75 figure(12)
76 imshow(imYUV);
77 figure(13)
78 Y=imYUV(:,:,1);
79 U=imYUV(:,:,2);
80 V=imYUV(:,:,3);
81 subplot(131),
82 subplot(132),
83 subplot(133),
84 imshow(Y);title('Y');
85 imshow(U);title('U');
86 imshow(V);title('V');

```

Dans la première manipulation, nous allons explorer différentes opérations sur des images en utilisant MATLAB. Nous commencerons par lire et afficher une image en niveaux de gris, puis une image en couleur. Ensuite, nous effectuerons diverses manipulations telles que l'extraction des canaux R, G et B et le changement de palette de couleurs pour les afficher dans leurs couleurs d'origine

Dans la deuxième manipulation, nous allons examiner et comparer les images en couleurs R, G et B en tenant compte de leurs luminances respectives.

1. Comparaison des images couleur :

o Regardez les images couleur R, G et B individuellement. Observez les différentes teintes et couleurs présentes dans chaque canal.

o Notez les différences de couleur entre les canaux. Par exemple, le canal R mettra en évidence les zones où le rouge est dominant, tandis que le canal B mettra en évidence les zones où le bleu est dominant, et ainsi de suite.

2. Comparaison des luminances :

o Comparez les images de luminance R, G et B. Ces images montrent la quantité de lumière présente dans chaque pixel, sans tenir compte de la couleur.

o Observez les variations de luminance à travers les images. Les zones plus lumineuses indiqueront des parties de l'image qui sont plus brillantes, tandis que les zones plus sombres indiqueront des parties de l'image qui sont plus sombres.

3. Comparaison entre les images couleur et leurs luminances :

Cela vous permettra de voir comment les variations de couleur dans les images couleur se traduisent en variations de luminance dans les images de luminance.

o Notez que les images couleur peuvent présenter des détails supplémentaires en termes de couleur et de teinte, tandis que les images de luminance mettent davantage l'accent sur les variations de luminosité.

Dans la troisième manipulation, nous allons explorer la transformation des images entre l'espace de couleur RVB (Rouge-Vert-Bleu) et l'espace de couleur HSV (Teinte-Saturation-Luminance) à l'aide de MATLAB. Nous commencerons par extraire les composantes H (Teinte), S (Saturation) et V (Luminance) d'une image couleur 'onion.png'. Ensuite, nous afficherons la composante H en couleur, tandis que les composantes S et V seront affichées en niveaux de gris pour une meilleure visualisation

Dans la quatrième manipulation, nous allons explorer la conversion des images entre l'espace de couleur RVB (Rouge-Vert-Bleu) et l'espace de couleur YUV (Luminance-Chrominance). L'espace de colorimétrie YUV est souvent utilisé dans les applications de télévision. Nous débuterons par la conversion de l'image couleur 'onion.png' en une image de type YUV à l'aide de la fonction `rgb2ntsc` de MATLAB. Ensuite, nous extraierons les composantes Y (Luminance), U et V (Chrominances) de l'image YUV et les afficherons individuellement pour une analyse plus détaillée. Enfin, nous explorerons l'utilisation de différentes palettes de couleurs pour visualiser les composantes U et V, afin d'observer comment elles affectent la perception des informations de chrominance

13. Exercice

[solution n°2 p.36]

Exercice

1- Qu'est-ce que le traitement numérique ?

- L'ensemble des techniques permettant de modifier une image numérique
- L'ensemble des techniques permettant de modifier la répartition des niveaux de luminosité de l'image
- L'ensemble des techniques permettant de modifier l'aspect de pixelisé de la photo

Exercice

2- Quel est l'objectif principal du traitement d'image ?

- Créer des illusions d'optique
- Améliorer la qualité des photos
- Modifier la réalité physique

Exercice

3. La taille en octet d'une image numérique :

- Dépend de la profondeur de la couleur de l'image
- Dépend du nombre de pixels qui forment l'image
- Diffère entre une image bitmap et une image vectorielle

Exercice

4. La résolution d'une image :

- C'est le nombre de pixels qui forment l'image
- C'est le nombre de pixels par pouce
- C'est mesurée en pixel
- Plus elle est élevée plus l'image est de meilleure qualité

Exercice

5- Une image vectorielle :

- Est formée par des pixels
- Occupe un espace mémoire important
- Ne perd pas de qualité suite à un agrandissement

- Formée par des formes géométriques

Exercice

6- Les périphériques suivants sont des périphériques d'acquisition d'image numérique :

- Scanner
- Souris
- Imprimante
- Appareil photo numérique

Exercice

7- En quelle unité se définit une image :

- En centimètre
- En Pixels
- En pouce

Exercice

8- Qu'est ce que le contraste d'une image ?

- La différence entre les couleurs froides et chaudes
- La répartition lumineuse d'une image
- La différence entre les couleurs primaires et secondaires

Exercice

9- Quelle est la fonction principale d'un histogramme dans le traitement d'image ?

- Visualiser la distribution des couleurs dans l'image
- Réduire la taille de l'image pour économiser de l'espace de stockage
- Appliquer des effets artistiques à l'image

Exercice

10- Quelle est la signification de l'acronyme RGB ?

- Red, Green, Blue
- Read, Gain, Brightness
- Realistic Graphic Bitmap

14. Conclusion

L'initiation au traitement d'image offre une porte d'entrée captivante dans le monde de la manipulation visuelle numérique. En explorant les concepts fondamentaux de la modification d'images, les apprenants acquièrent une compréhension précieuse de la manière dont les images peuvent être transformées et améliorées pour répondre à divers besoins créatifs et professionnels. Cette initiation permet également de découvrir les outils logiciels et les techniques utilisés dans le domaine, offrant ainsi une base solide pour des explorations plus avancées.

Au-delà des simples ajustements de luminosité et de contraste, l'initiation au traitement d'image introduit les étudiants aux concepts avancés tels que la retouche photo, la composition, le montage et la manipulation d'images en trois dimensions. Cela ouvre un monde de possibilités artistiques et techniques, permettant aux apprenants d'exprimer leur créativité tout en développant des compétences pratiques essentielles.

Conclusion

En combinant les connaissances acquises dans ce TP sur l'initiation au traitement d'image avec la compréhension des techniques de compression **JPEG**, on peut pleinement apprécier l'importance de la manipulation d'images dans différents domaines, tels que la photographie numérique, la vidéo, la vision par ordinateur, etc. Le traitement d'image fournit les outils nécessaires pour analyser, modifier et améliorer les images, tandis que la compression **JPEG** offre des moyens efficaces de stocker et de transmettre ces images de manière économique. Cette synergie entre le traitement d'image et la compression **JPEG** offre aux professionnels de l'informatique la possibilité de créer, manipuler et partager des contenus visuels de manière innovante et efficace

Solutions des exercices

> **Solution** n° 1

Exercice p. 7

Exercice

1- Quel est le processus qui crée une image numérique à partir d'une image analogique ?

- La numérisation de données
- La conversion numérique-analogique
- La digitalisation d'images
- La compression d'images

Exercice

2- Quels sont les concepts fondamentaux du traitement d'image ?

- Résolution, Compression, Conversion
- Résolution, Couleur, Formats de fichiers
- Transparence, Opacité, Contraste
- Interpolation, Éclairage, Texture

Exercice

3- Quelles fonctions MATLAB sont couramment utilisées dans le traitement d'images ?

- imread : Pour lire une image à partir d'un fichier.
- imshow : Pour afficher une image à l'écran.
- randn : Pour générer des valeurs aléatoires selon une distribution normale

Exercice

4- Comment les couleurs sont-elles souvent représentées dans les images numériques ?

- En utilisant des valeurs de luminosité uniquement
- En utilisant des valeurs de teinte et de saturation uniquement

En utilisant des matrices de pixels RVB (Rouge, Vert, Bleu)

En utilisant des valeurs de transparence uniquement

Exercice

5- Comment pouvez-vous convertir une image RGB en une image en niveaux de gris en utilisant MATLAB ?

`rgb2gray`

`gray2rgb`

`convert2gray`

`grayscale`

Exercice

6-Qu'est-ce que le traitement d'images numériques implique généralement ?

La conversion d'images analogiques en images numériques

La manipulation et la modification d'images à l'aide d'algorithmes informatiques

La création d'images artistiques à l'aide de logiciels spécifiques

Exercice

7-Quelle est l'importance de la maîtrise d'un langage de programmation dans le contexte de la compression JPEG ?

Permettre de créer des images numériques à partir de zéro

Faciliter la compréhension des concepts de base de la compression d'images

Aider à mettre en œuvre les algorithmes de compression et à manipuler les données nécessaires

Exercice

8-Quelles compétences sont essentielles pour comprendre les fondamentaux du traitement d'images numériques ?

Connaissance des méthodes de cuisson d'images

Capacité à reconnaître les noms de différents logiciels de retouche d'images

Compréhension des concepts de résolution, de couleur et de format d'images

Exercice

9-Pourquoi la maîtrise des bases du traitement d'images numériques est-elle importante pour la compression JPEG ?

Pour pouvoir identifier les types de compression les plus efficaces

Pour comprendre l'impact des différents algorithmes de compression sur la qualité de l'image

Pour connaître les différents formats de fichiers d'images et leurs avantages

Exercice

10- Quel est l'avantage de la maîtrise d'un langage de programmation dans la mise en œuvre de la compression JPEG ?

- Permettre de créer des animations vidéo à partir d'images fixes
- Faciliter la collaboration avec d'autres concepteurs graphiques
- Aider à personnaliser et à optimiser les algorithmes de compression pour répondre à des besoins spécifiques

> **Solution n°2**

Exercice p. 30

Exercice

1- Qu'est-ce que le traitement numérique ?

- L'ensemble des techniques permettant de modifier une image numérique
- L'ensemble des techniques permettant de modifier la répartition des niveaux de luminosité de l'image
- L'ensemble des techniques permettant de modifier l'aspect de pixelisé de la photo

Exercice

2- Quel est l'objectif principal du traitement d'image ?

- Créer des illusions d'optique
- Améliorer la qualité des photos
- Modifier la réalité physique

Exercice

3. La taille en octet d'une image numérique :

- Dépend de la profondeur de la couleur de l'image
- Dépend du nombre de pixels qui forment l'image
- Diffère entre une image bitmap et une image vectorielle

Exercice

4. La résolution d'une image :

- C'est le nombre de pixels qui forment l'image
- C'est le nombre de pixels par pouce
- C'est mesurée en pixel

Plus elle est élevée plus l'image est de meilleure qualité

Exercice

5- Une image vectorielle :

- Est formée par des pixels
- Occupe un espace mémoire important
- Ne perd pas de qualité suite à un agrandissement
- Formée par des formes géométriques

Exercice

6- Les périphériques suivants sont des périphériques d'acquisition d'image numérique :

- Scanner
- Souris
- Imprimante
- Appareil photo numérique

Exercice

7- En quelle unité se définit une image :

- En centimètre
- En Pixels
- En pouce

Exercice

8- Qu'est ce que le contraste d'une image ?

- La différence entre les couleurs froides et chaudes
- La répartition lumineuse d'une image
- La différence entre les couleurs primaires et secondaires

Exercice

9- Quelle est la fonction principale d'un histogramme dans le traitement d'image ?

- Visualiser la distribution des couleurs dans l'image
- Réduire la taille de l'image pour économiser de l'espace de stockage
- Appliquer des effets artistiques à l'image

Exercice

10- Quelle est la signification de l'acronyme RGB ?

- Red, Green, Blue
- Read, Gain, Brightness
- Realistic Graphic Bitmap

Glossaire

Algorithmes

Des séquences d'instructions ou de règles bien définies pour résoudre un problème ou exécuter une tâche donnée dans un nombre fini d'étapes.

Analogique

Se réfère à une forme de représentation où les grandeurs physiques sont représentées par des valeurs continues.

Appareil photo numérique

Un dispositif électronique qui capture des images et les enregistre sous forme numérique.

Blanc

L'autre des deux valeurs possibles pour un pixel dans une image binaire, représentant généralement l'absence d'un objet ou d'une caractéristique spécifique.

Caractéristique spécifique

Une propriété ou un élément d'intérêt dans une image, souvent représenté par la présence ou l'absence de pixels noirs.

Chiffres binaires

Système de numération utilisant deux chiffres, 0 et 1, qui forme la base du langage informatique.

Conception graphique

Le processus de communication visuelle et de résolution de problèmes à l'aide d'éléments graphiques et de texte.

Données numériques

Informations représentées sous forme de nombres ou de chiffres, généralement binaires (0 et 1), qui peuvent être traitées par des ordinateurs ou d'autres dispositifs électroniques.

Échelle de niveaux de gris

Une gamme de valeurs utilisées pour représenter les niveaux de luminosité dans une image en niveaux de gris, souvent de 0 (noir) à 255 (blanc) dans un système de 8 bits.

État logique

Dans le contexte des images binaires, l'association des valeurs noir et blanc à des concepts logiques, où noir peut représenter "vrai" ou la présence, tandis que blanc peut représenter "faux" ou l'absence.

Format informatique spécifique

La structure dans laquelle les données d'une image numérique sont encodées, comme JPEG, PNG, GIF, TIFF, etc.

Illustration

Une représentation visuelle ou graphique, souvent utilisée pour accompagner du texte ou transmettre une idée.

Image binaire

Un type d'image numérique où chaque pixel ne peut prendre que deux valeurs distinctes, généralement noir et blanc.

Image en niveaux de gris

Une image où chaque pixel a une valeur représentant son niveau de luminosité, allant du noir au blanc.

Image numérique

Une représentation visuelle stockée sous forme de données numériques, généralement dans un format informatique spécifique

Images médicales

Des images utilisées dans le domaine médical pour diagnostiquer, surveiller et traiter les patients, comprenant des radiographies, des IRM, des scanners CT, etc.

Logiciel de création graphique

Un programme informatique utilisé pour créer, éditer ou manipuler des images graphiques.

Logiciels

Des programmes informatiques conçus pour effectuer des tâches spécifiques sur un ordinateur ou un dispositif électronique

Médecine

La science et la pratique de diagnostiquer, de traiter et de prévenir les maladies et les blessures chez les humains, où les images numériques sont souvent utilisées pour les diagnostics médicaux et les recherches.

Noir

Une des deux valeurs possibles pour un pixel dans une image binaire, souvent associée à la présence d'un objet ou d'une caractéristique spécifique.

Numérisation

Le processus de conversion d'informations analogiques en données numériques.

Partage d'informations

L'action de rendre des données ou des informations accessibles à d'autres personnes ou entités.

Photographie

L'art, la pratique ou le métier de prendre et de traiter des photographies.

Pixels

Les plus petites unités d'une image numérique, composées de points individuels représentant différentes couleurs et intensités.

Région de l'image

Une zone spécifique de l'image définie par un ensemble de pixels, où la présence ou l'absence de caractéristiques peut être déterminée.

Scanner

Un périphérique électronique qui convertit des documents ou des images en données numériques.

Traitement automatisé

L'utilisation de logiciels et d'algorithmes pour exécuter des tâches ou des processus sans intervention humaine directe.

Valeur de luminosité

La valeur attribuée à chaque pixel d'une image en niveaux de gris, indiquant son intensité lumineuse relative. Cette valeur peut varier de 0 (noir) à 255 (blanc) dans un système de 8 bits.

Valeurs distinctes

Dans le contexte d'une image binaire, il s'agit des deux seules valeurs possibles pour chaque pixel : noir et blanc.

Abréviations

BMP : Bitmap

CMJN : Il s'agit de quadrichromie, donc 4 couleurs : cyan, magenta, jaune et noir.

DICOM : Digital Imaging and Communications in Medicine

DPI : dots per inch

GIF : Graphics Interchange Format

HSV : Hue Saturation Value

JPG : Joint Photographic Experts Group

PJEG : Joint Photographic Experts Group

PNG : Portable Network Graphics

RGB : pour Red-Green-Blue

RVB : désigne trois couleurs : rouge, vert et bleu

Tiff : Tag Image File Format

VOD : video on demand

Bibliographie

Cours "Compression par transformée de codage JPEG"

Webographie

<https://iast.univ-setif.dz/documents/Cours/CoursInformatiqueL1GAT21.pdf>. Représentation d'une image numérique.

<https://www.univ-constantine2.dz/files/Theses/Informatique/Magistere/Mohamed-Sandeli.pdf>. Traitement d'images par des approches bio-inspirées (Application à la segmentation d'images)

<https://efreidoc.fr/M1/Traitement%20d%27images/Cours/2011-12.cours.complet.img.pdf>. La COULEUR dans les IMAGES Et Indexation par le Contenu

<http://rdoc.univ-sba.dz/bitstream/123456789/2546/1/THESE.pdf>. Sécurité des images Numériques compressées JPEG