Département ATE
Master 1 en Systèmes des Télécommunications
Semestre 2 / Mars 2019
Module : Traitement d'Image

TP 1
Images et Colorimétrie

But

- *Lecture et affichage d'une image*
- *Extraire les matrices couleurs R, G et B d'une image couleur*
- *Luminance d'une image*
- *Conversion de l'espace de colorimétrie RGB vers l'espace HSV et ….*

1. Lecture et Affichage d'une Image

Commandes MATLAB

Pour lire ou afficher une image, on utilise les commandes suivantes :

`im = imread(filename,ext)`

Permet de lire une image, où
- `filename` : le nom de l'image.
- `ext` : l'extension du format de stockage de l'image.

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

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')
```

`imshow(im)`

Permet d(afficher l'image «im».

Exemple :
```
im=imread('cameraman.tif');
imshow(im)
```
ou
```
imshow('cameraman.tif');
```


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

Permet de connaitre 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 :
  - A niveau de gris, d=1
  - En couleur, d=3

Exemple :
```
im=imread('cameraman.tif');
[m,n,d]=size(im);
```

`cm = colormap ;`

`colormap(cm)`

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

Manipulation 1

(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**
   - Aller au gestionnaire des variables et déduire : la taille et le type de la variable de **im2**,
   - Que présentent : **im2(:,:,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 question 3. et 4.
6. **im2** est une image couleur de type RGB. Pour extraire les matrices de couler rouge, R, de couleur verte, G, et de couleur bleu, B, on procèdera comme suit :

```
%% extraction
im2_R=im2(:,:,1);
im2_G=im2(:,:,2);
im2_B=im2(:,:,3);
%% affichage
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

```
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

2. Luminance d'une image

- ***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*

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

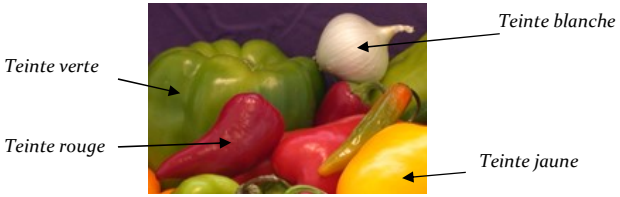
- *Sous MATLAB pour créer la luminance d'une image couleur, existe la commande :*

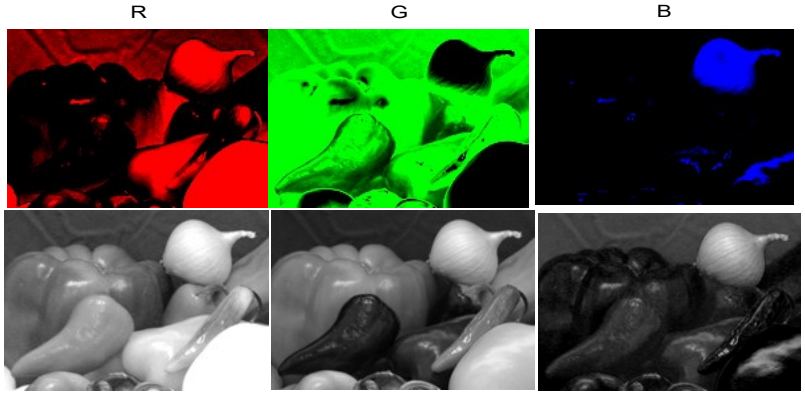
`im_gray = rgb2gray(im_RGB)`	`Convertit une image couleur im_RGBen une image en niveau de gris (luminance) im_gray`

Avant Propos

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?





Manipulation 2

1. En utilisant l'expression de la luminance, créer l'image luminance associée 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 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(immR);
subplot(235), imshow(immG);
subplot(236), imshow(immB);
```

- ***comparer les images couleurs R, G et B avec leurs luminances , et valider vos observations dans la section "Avant propos".***

3. De l'espace de colorimétrie RGB vers l'espace de colorimétrie HSV

- 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)`
`Ou`
`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 imageRGB`

Manipulation 3

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

1. *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,
figure,
subplot(121), imshow(S); title('S');
subplot(122), imshow(H); title('V'); colormap('gray')
```

2. *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);
```

4. De l'espace de colorimétrie RGB vers l'espace de colorimétrie YUV (ou YIQ)

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

Y=0.2989*R+0.5870*G+0.1140*B (2)

U=0.5960*R+0.2740*G+0.3220*B (3)

V=0.2110*R+0.5230*G+0.3120*B (4)

Ce système est utilisé en télévisions.

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

`[Y U V] = rgb2ntsc(im_RGB)`
`Ou`
`Im_YUV = rgb2ntsc(il_RGB)`
	`Convertit une image couleur im_RGB de type RGB en une image NTSC de type YUV.`

`[R G B] = ntsc2rgb(im_YUV)`	`Convertit une image NTSC de type YUV im_YUV en une image couleur de type RGB.`

Manipulation 4

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

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

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);
U=imYUV(:,:,2); subplot(132), imshow(U);
V=imYUV(:,:,3); subplot(133), imshow(V);
```

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