

# TRAITEMENT D'IMAGE

## TP 1 : OUTIL DE TRAITEMENT D'IMAGE

### I) Manipulation d'image

Une image est une représentation Matricielle à deux ou trois dimensions. La manipulation d'une image sous Matlab respecte les mêmes règles de traitement d'une matrice. En outre, Matlab définit des fonctions spécifiques pour faciliter la gestion et l'amélioration des images définies dans son *Image Processing Toolbox*.

Pour afficher ces différentes fonctions, tapez `help images` ;

#### 1. Ouverture d'une image

**Pour lire l'image** « lena.jpg » puis l'afficher, tapez :

```
clear all ;  
clc ;  
lena_Img = imread('lena.jpg') ;  
imshow (lena_Img);
```

**Pour afficher la taille** de l'image, tapez :

```
Image_size = size (lena_Img) ;
```

Que présente chaque dimension de la matrice *lena\_Img* ?

**L'accès à un élément particulier d'une image** est indexé par le nom de la matrice et la position de cet élément. On peut récupérer ainsi les valeurs ou les modifier aisément. Par exemple, on peut modifier l'image « lena.jpg » comme suit :

```
Clear all;  
Clc;  
I = imread ('lena.jpg')  
I(1:3,31:50) = 0;  
I(2,3)  
Figure, imshow(I);close;
```

Travail à réaliser

Image acquise	Synthèse d'image																																																																
<p>Soit une image de dimension 8x8 présentée par la matrice de niveaux de gris suivante :</p> <p>Ecrire un script qui permet d'ouvrir une image, de la visualiser et d'afficher dans Matlab sa taille (nbLignes,nbColonnes).</p> <table><tr><td>5</td><td>150</td><td>238</td><td>35</td><td>113</td><td>213</td><td>77</td><td>5</td></tr><tr><td>58</td><td>113</td><td>233</td><td>51</td><td>5</td><td>5</td><td>150</td><td>219</td></tr><tr><td>5</td><td>5</td><td>0</td><td>127</td><td>128</td><td>240</td><td>150</td><td>217</td></tr><tr><td>123</td><td>5</td><td>60</td><td>150</td><td>90</td><td>250</td><td>173</td><td>151</td></tr><tr><td>5</td><td>150</td><td>200</td><td>250</td><td>16</td><td>5</td><td>5</td><td>126</td></tr><tr><td>194</td><td>5</td><td>80</td><td>192</td><td>191</td><td>150</td><td>138</td><td>5</td></tr><tr><td>150</td><td>44</td><td>207</td><td>3</td><td>51</td><td>180</td><td>5</td><td>209</td></tr><tr><td>4</td><td>103</td><td>2</td><td>190</td><td>5</td><td>109</td><td>177</td><td>164</td></tr></table> <p>Ecrire un script qui permet d'ouvrir une image, de la visualiser et d'afficher dans Matlab sa taille (nbLignes,nbColonnes).</p>	5	150	238	35	113	213	77	5	58	113	233	51	5	5	150	219	5	5	0	127	128	240	150	217	123	5	60	150	90	250	173	151	5	150	200	250	16	5	5	126	194	5	80	192	191	150	138	5	150	44	207	3	51	180	5	209	4	103	2	190	5	109	177	164	<p>Implémentez le code suivant pour générer et afficher l'image sinus :</p> <div><p><b>taille= 128</b></p><p><b>periode= 32</b></p><p><b>orientation = 0</b></p><p><b>amplitude= 128</b></p><p><b>val_moy = 0</b></p></div> <p>centre0=taille/2+1; %décalage pour fixer le point de rotation au centre de l'image i=1:1:taille; j=i; fl=(i-centre0); fc=(j-centre0); [Fl,Fc]= meshgrid(fl,fc); <b>img = val_moy +</b> <b>amplitude*sin(2*pi *(1/periode)</b> <b>* (Fl*cos(orientation)</b> <b>+ Fc*sin(orientation))));</b> img = floor(img); figure(1); imagesc(im1);colormap(gray); Avec</p>
5	150	238	35	113	213	77	5																																																										
58	113	233	51	5	5	150	219																																																										
5	5	0	127	128	240	150	217																																																										
123	5	60	150	90	250	173	151																																																										
5	150	200	250	16	5	5	126																																																										
194	5	80	192	191	150	138	5																																																										
150	44	207	3	51	180	5	209																																																										
4	103	2	190	5	109	177	164																																																										

Testez différents commandes qui permettent l'affichage des images et commentez.

## 2. Parcours d'une image

Ecrire un script ou une fonction qui prend en entrée L'image de la matrice ci-dessus et qui retourne le nombre de pixels de valeur 5 (ou autre valeur) dans l'image.

## 3. Affichage d'un bloc de l'image

Ecrire un script qui permet d'afficher le bloc supérieur gauche d'une image (la taille du bloc 32x32).

Changer dans l'image lena la partie des yeux par les niveaux de gris de ce bloc puis affichez l'image résultat.

## 4. Quantification d'une image

Pour visualiser l'image avec n niveaux de gris, il existe de nombreuses variantes.

```
clc;  
clear;  
close all;  
I = imread('eight.tif');
```

```
figure
imshow(I) % afficher l'image avec 256 niveaux de gris
n=8;% choisir le nombre de niveaux de gris
d=255/n-1; %déterminer le pas de quantification
J= floor(I/d);
figure, imagesc(J),colormap(gray)
figure, imhist(J)
```

Tester plusieurs valeurs de n et commentez.

### **5. Inversion des niveaux d'une image**

Ecrire un script ou une fonction Matlab qui permet de créer une nouvelle image  $N$  (le négatif de l'image) dont les niveaux de gris seront inversés par rapport à l'image d'origine  $I$ . C'est à dire :

$$N(i,j) = 255 - I(i,j)$$

### **6. Re-échantillonnage d'une image (résolution d'image)**

Tapez le code suivant qui permet de réduire le nombre de pixel de l'image par un facteur de 2 :

Sous-échantillonnage d'une image »X« par 2 dans chaque direction. Les colonnes et les lignes sont sous-échantillonnées par 2.

#### **a) variante 1**

```
s=size(X);
height=s(1); % nombre de lignes de l'image
width=s(2); % nombre de colonnes de l'image
% Créer une matrice de zeros de taille moitié dans chaque direction
Y=zeros(round(height/2),round(width/2));
% Remplir Y avec les données de X en prenant un pixel sur deux
for i=1:round(height/2)
    forj=1:round(width/2)
        Y(i,j)=X(2*i-1,2*j-1);
    end
end
```

#### **b) variante 2**

```
% Cette fonction utilise le sous-indexage Matlab
s=size(X);
Y=X(1:2:s(1),1:2:s(2));
```

#### **c) variante 3**

```
I = imread('rice.png');
J = imresize(I, 0.5);
Figure, imshow(I)
```

Comparer les 3 variantes des codes ci-dessus et commentez.

### **7. Représentation des images couleurs**

Espace de couleurs RGB

Une image couleur dans l'espace RGB (Red, Green, Blue) est composée de trois paramètres projetés sur trois plans de couleurs : Rouge, Vert et Bleu.

Le code ci-dessous vous permet de décomposer l'image « lena.jpg » en trois plans de couleur R, G, B. Affichez sur une même figure ces trois plans avec l'image d'origine.

```
Clear all;
Clc;
I = imread('lena.jpg');
r= I; g= I; b= I;
r(:, :, [2 3]) = 0;
g(:, :, [1 3]) = 0;
b(:, :, [1 2]) = 0;
```

### *Espaces de Couleurs*

L'espace de représentation standard est l'espace de couleurs RGB. Les couleurs finales sont obtenues par synthèse additive des trois couleurs primaires : Rouge, Vert et Bleu. Il existe cependant des problèmes qui peuvent nécessiter le changement d'espace de couleurs pour percevoir différemment l'image. Il y a des images où la couleur importe peu, par exemple des photographies de cellules vivantes (pseudo-transparentes), des images radar, des images satellites... Dans ce cas, l'espace RGB n'est plus utilisé. On lui préfère d'autres espaces comme HSV Hue/Saturation/Value ou YCbCr Luminance/Chrominance bleue/Chrominance rouge.

La boîte à outils images de Matlab gère les espaces de couleurs RGB, HSV, YCbCr, NTSC. Un exemple ci-dessous permet de vous rendre compte de l'utilisation d'un changement d'espace de couleurs. On lit une image colorée, on la passe dans l'espace HSV et on va changer la saturation avant de repasser dans l'espace RGB pour visualiser le résultat.

```
img = imread('fleur.jpg');
figure; imshow(img);
imgHSV = rgb2hsv(img);
figure; imshow(imgHSV);
imgHSV(100:200,300:400,1) = 0.7;
imgRGB = hsv2rgb(imgHSV);
figure; imshow(imgRGB);
imgHSV = rgb2hsv(img);
figure; imshow(imgHSV);
imgHSV(100:200,300:400,2) = 0;
imgRGB = hsv2rgb(imgHSV);
figure; imshow(imgRGB);
imgHSV = rgb2hsv(img);
figure; imshow(imgHSV);
imgHSV(100:200,300:400,3) = 0.3;
imgRGB = hsv2rgb(imgHSV);
figure; imshow(imgRGB);
```

Commenter le code et citer les modifications effectuées.