



## Artificial Intelligence based Image Processing (AIIP) (mai.nguyen-verger@cyu.fr)

### AIIP.1

Avant de programmer : Créer un répertoire de travail portant le nom « AIIP » et sous-répertoires suivants :

```
root/ AIIP /Document
      /Programmes
      /Rapport
```

Conseils pratiques : Tout programme commence par :

```
% Titre du programme
% Auteur
```

et les commandes de Matlab suivantes :

*clear all* ;% pour effacer toutes les variables issues de programmes précédemment exécutés.  
*close all* ;% pour fermer toutes les figures ouvertes précédemment.

### Partie 1. Création des image numériques

#### Partie 1(a). Création des image noires-blanches

Exercice 1 : image binaire

Créer un triangle blanc inférieur sur un fond noir de taille 10 x10 pixels (Fig. 1).

Exercice 2 : Crée un triangle blanc en haut dans un fond noir de taille NLxNC (avec NL=NC=100) et un triangle blanc en bas dans un fond noir de taille NL x NC, puis les visualiser (Fig. 2a, 2b)

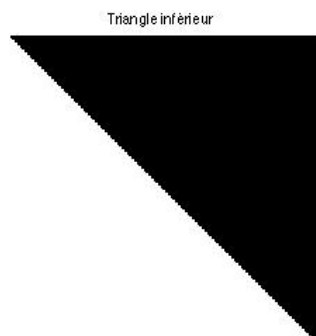


Fig. 1



Fig. 2a

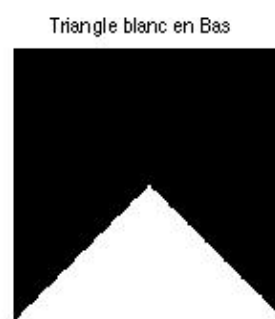


Fig. 2b

Exercice 3 : Écrire le programme pour créer une image binaire alternée de dimension NxN avec N=100 (les pixels dont le numéro est impair ont la valeur 0, pair la valeur 1). Fig.3

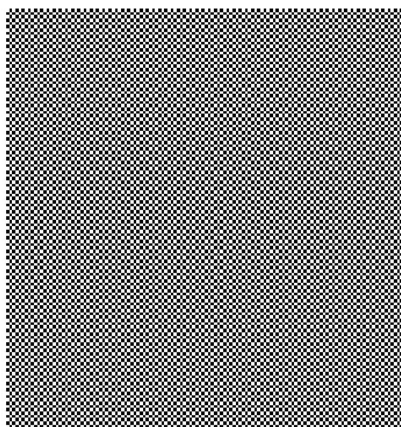


Fig. 3

Exercice 4 : Image d'intensité (en niveaux de gris) dégradée en ligne d'un rectangle de taille 10 x 8 (Fig.4)

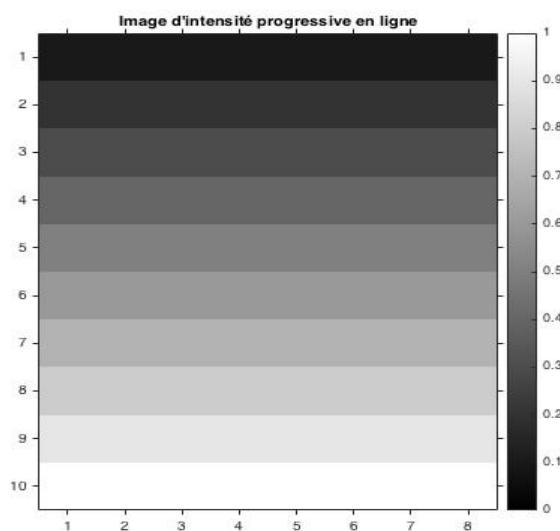


Fig. 4

Exercice 5 : Écrire le programme pour créer une image, de dimension  $N \times N$  (**avec  $N=50$ , puis  $N=100$** ), de niveaux de gris progressifs en fonction de position de pixel (la valeur de chaque pixel dépend de la position de ce pixel dans la matrice de  $N \times N$ ). Fig. 5  
Attention à la normalisation : les niveaux de gris doivent être entre  $[0,1]$  ou bien  $[0, 255]$ .



Fig. 5

## Partie 1(b). Image Couleur

Rappel du dosage des couleurs (mesures psychovisuelles) :

Rouge (r=1, g=0, b=0); Vert (r=0, g=1, b=0); Bleu (r=0, g=0, b=1);  
 Gris (r=g=b=0.5); Jaune (r=1, g=1, b=0); Violet (r=1, g=0, b=1);  
 Violet (r=1, g=0, b=1); Blanc (r=1, g=1, b=1); Orange (r=1, g=0.5, b=0);  
 Vert sombre (r=0, g=0.5, b=0)  
 Bleu-vert (r=0, g=0.6, b=1) ;

Exercice 6 : Représentation **RGB** :

(Rappel : créer trois matrices de couleur suivantes :

R= [...] ;  
 G=[...] ;  
 B=[...] ;

puis les concaténer dans une matrice à trois dimensions par :

I(:, :, 1)= R ;  
 I(:, :, 2)= G ;  
 I(:, :, 3)= B ;

**ou bien** utiliser la fonction de la concaténation **cat()** de Matlab :

I=**cat**(3,R,G,B) % commande pour la concaténation de trois matrices R, G, B  
 )

6.1) Création de l'image couleur suivante :

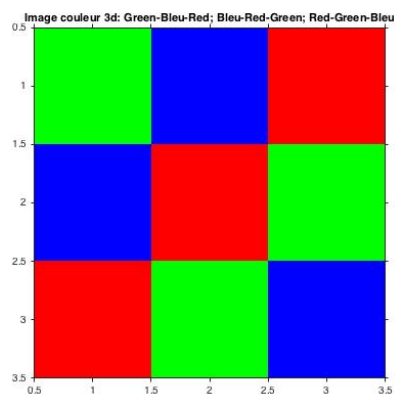


Fig. 6

6.2) Créer une image couleur dont l’affichage donne le résultat suivant :

jaune	violet	vert sombre
orange	blanc	jaune
rouge	bleu	Vert

6.3) Modifier les matrices R, G, B pour obtenir l’image suivante :

jaune	bleu	rouge
vert	blanc	jaune
violet	vert sombre	orange

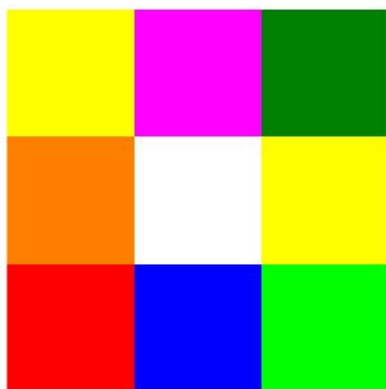


Fig. 7a

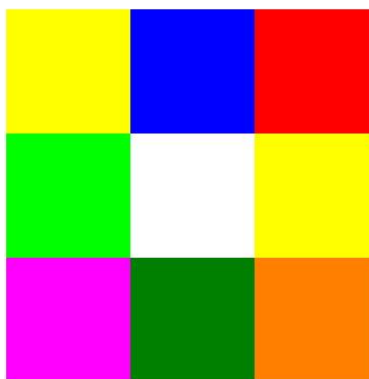


Fig. 7b

Exercice 7 : Représentation **indexée** des images couleurs :

(Rappel : créer :

- une table de couleurs appelée *map*=[...] qui est une matrice ***nx3*** (où ***n*** est le nombre de couleurs dans l'image) et
- une matrice des index de la même taille de l'image *I*=[...] qui contient les nombres entiers compris entre ***1*** et ***n***, chaque entier joue le rôle d'un index relatif à la table de couleurs, et
- afficher l'image par ***imshow(I,map)***.

)

```
map=[...];
I=[...];
imshow(I, map) ;
```

7.1) Créer la matrice *I* et la table de couleur *map* pour obtenir l'image dans Fig.7a

7.2) Modifier *I* et *map* pour obtenir l'image dans Fig.7b

7.3) Modifier la table de couleurs comme ci-dessous et afficher de nouveau l'image. Observer le résultat obtenu et expliquer le.

```
map=[0.4 0.2 0.4 ; 0 1 0 ; 1 0 0] ;
```

## AIIP. 2+3

(mai.nguyen-verger@cyu.fr)

### Histogramme des niveaux de gris d'images et transformations

- 1) Écrire un **algorithme** du CALCUL de l'histogramme des niveaux de gris d'une image de taille  $NL \times NC$  et de 256 niveaux de gris entre 0 (noir) et 255 (blanc).
- 2) Écrire un **programme** qui calcule et affiche les histogrammes des images dans les Fig. 1, Fig. 2a, Fig. 2b.  
(tracer une courbe en testant trois fonctions différentes: *plot(x,y)*, *stem()*, *bar()*).
- 3) Écrire un **algorithme** de l'ETIREMENT d'histogramme, des niveaux de gris d'une image entre [Imin et Imax], à 256 niveaux de gris entre [0, 255], avec 0 (noir) et 255 (blanc). La taille de l'image est  $NL \times NC$ .
- 4) Écrire un **programme** qui calcule, étire et affiche les histogrammes de l'image dans le fichier 'histo\_imageTest\_entree.png' se trouvant dans le répertoire «Images».
- 5) Écrire un **algorithme** de l'EGALISATION d'histogramme, des niveaux de gris d'une image entre [Imin et Imax], à 256 niveaux de gris entre [0, 255], avec 0 (noir) et 255 (blanc). L'image est de taille  $NL \times NC$ .
- 6) Écrire un **programme** qui calcule, égalise et affiche les histogrammes de l'image dans le fichier 'histo\_imageTest\_entree.png' se trouvant dans le répertoire «Images».

Les résultats sont présentés en deux colonnes (l'image à gauche et l'histogramme correspondant à droite) avec les intitulés des images et des axes. Exemples :

