I. Introduction

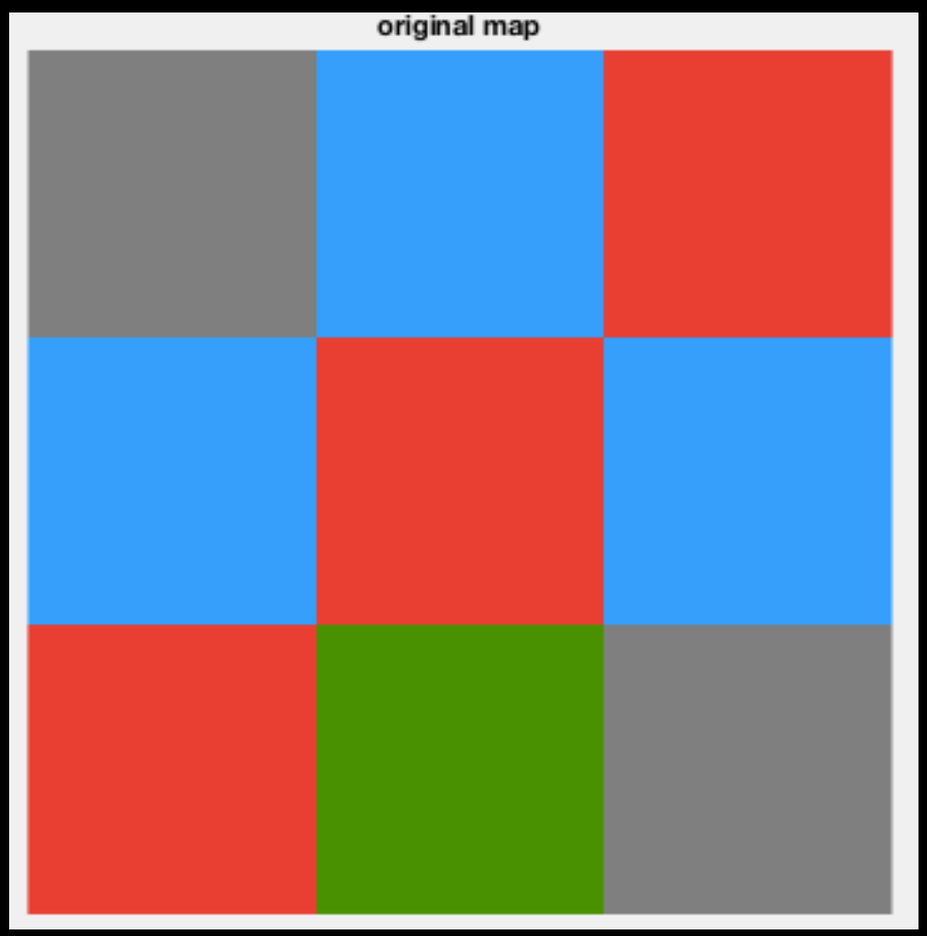
- **EXERCICE 1 : Table de couleurs.**

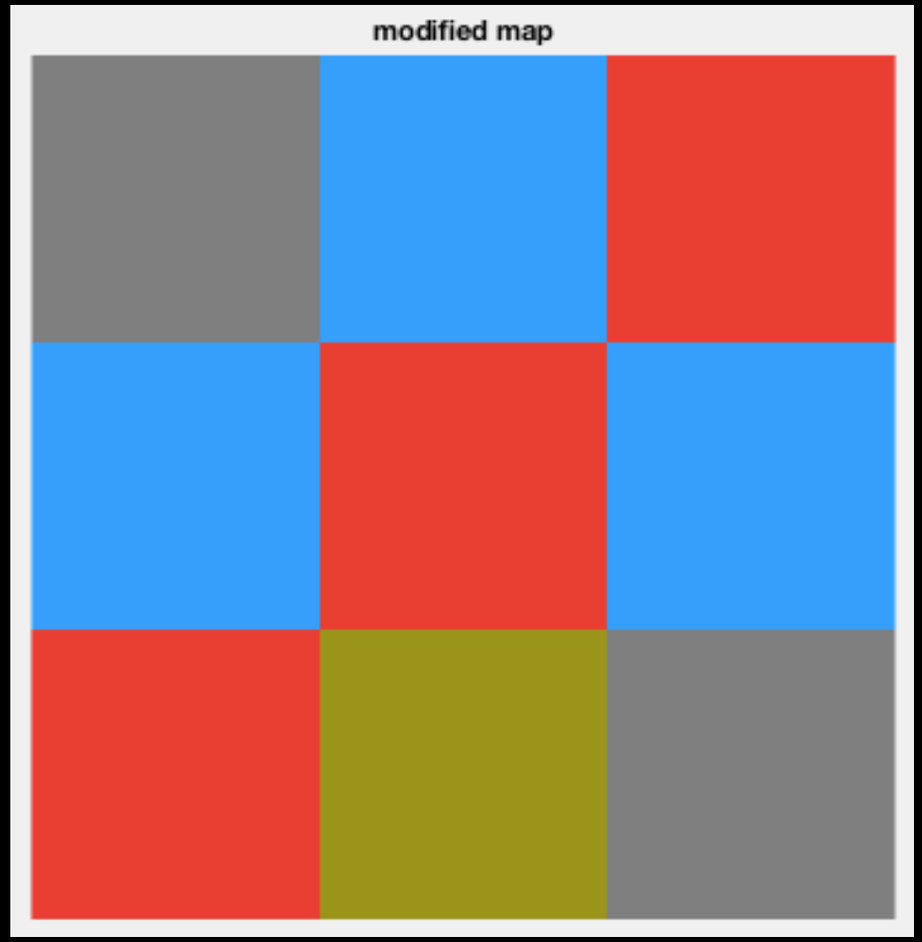
  Reprendre l'image en couleurs indexées de tout à l'heure (définie par *//* et *map*), puis modifier [*] la table de couleurs et expliquer le résultat.

  ([*] : par exemple, changer le vert pur à 60% de luminosité en ocre jaune (R et G), toujours à 60% de luminosité…)

I. Introduction

```
clear
close all

II=[1 2 3; 2 3 2; 3 4 1];
map=[.5 .5 .5; 0 .6 1; 1 0 0; 0 .6 0];
figure(1), imshow(II, map, 'InitialMagnification', 'fit')
title('original map')

map(4,1)=.6
figure(2), imshow(II, map, 'InitialMagnification', 'fit')
title('modified map')
```

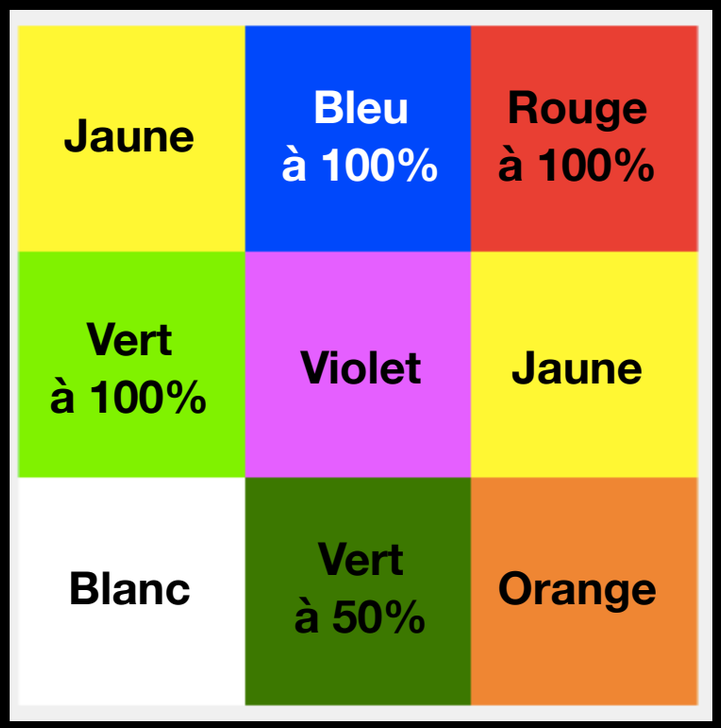




I. Introduction

- **EXERCICE 2 : Détermination d'une représentation indexée.**

  Déterminer les matrices permettant d'obtenir l'image suivante :




```
clear
close all

map=[1 1 0; 0 0 1; 1 0 0; 0 1 0; 1 0 1; 1 1 1; 0 .5 0; 1 .5 0];
II=[1 2 3; 4 5 1; 6 7 8];
figure, imshow(II, map, 'InitialMagnification', 'fit')
```

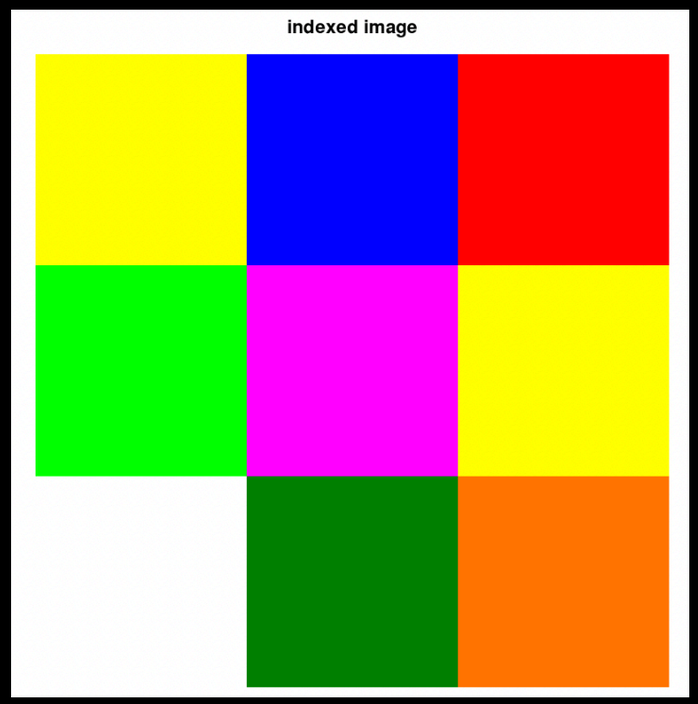
I. Introduction

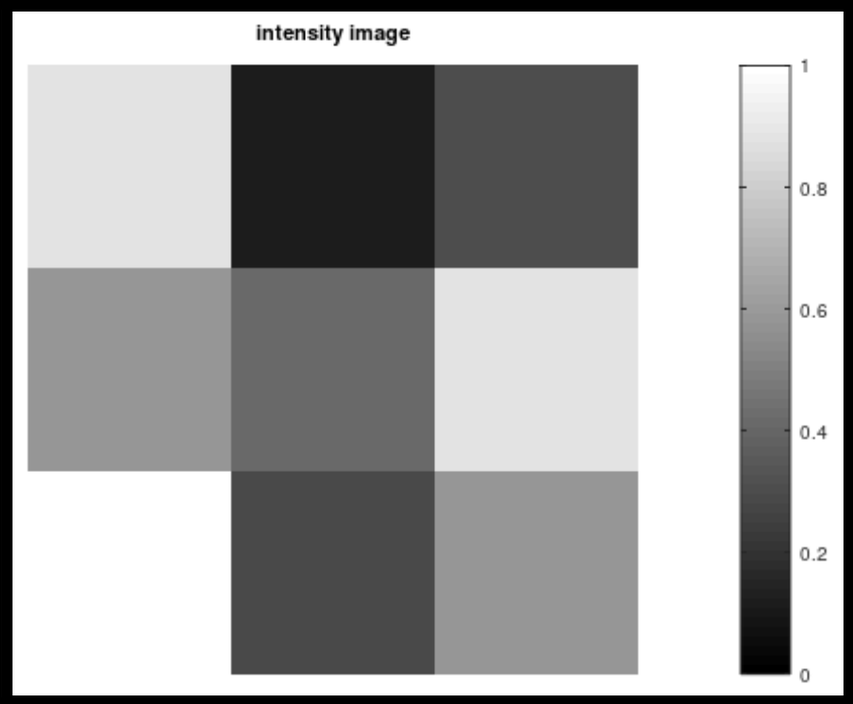
- **EXERCICE 3 : Changement de format.**

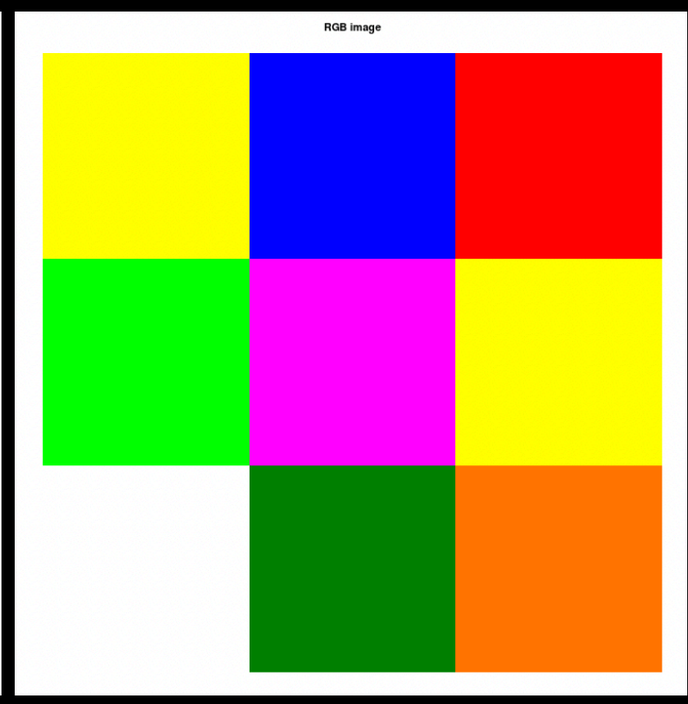
  Transformer l’image indexée de l’exercice précédent en image d’intensité et en image RGB. Les visualiser.

I. Introduction

```
clear
close all

map=[1 1 0; 0 0 1; 1 0 0; 0 1 0; 1 0 1; 1 1 1; 0 .5 0; 1 .5 0];
II=[1 2 3; 4 5 1; 6 7 8];

figure(1), imshow(II, map, 'InitialMagnification', 'fit')
%OCTAVE: figure(1), imshow(II, map), title('indexed image')

Iint=ind2gray(II, map);
figure(2), imshow(Iint, 'InitialMagnification', 'fit')
%OCTAVE: figure(2), imshow(Iint), title('intensity image'), colorbar

IRGB=ind2rgb(II, map);
figure(3), imshow(IRGB, 'InitialMagnification', 'fit')
%OCTAVE: figure(3), imshow(IRGB), title('RGB image')
```





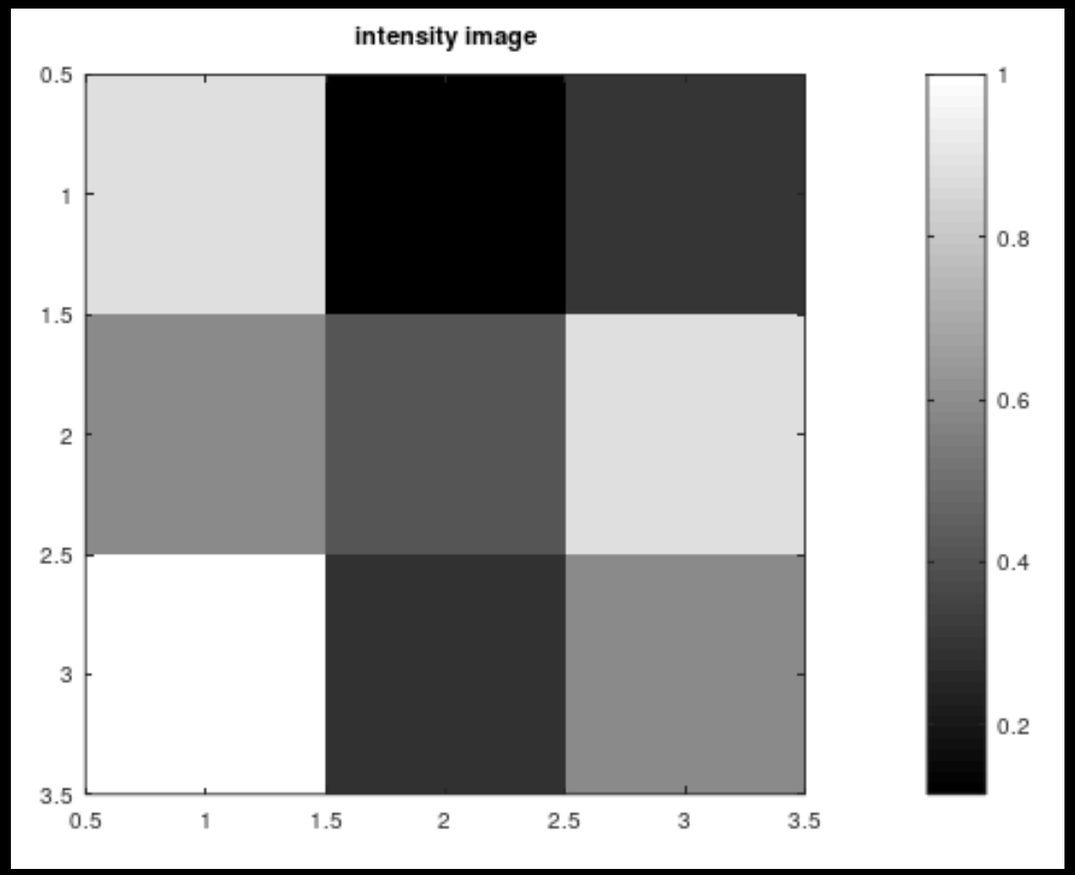


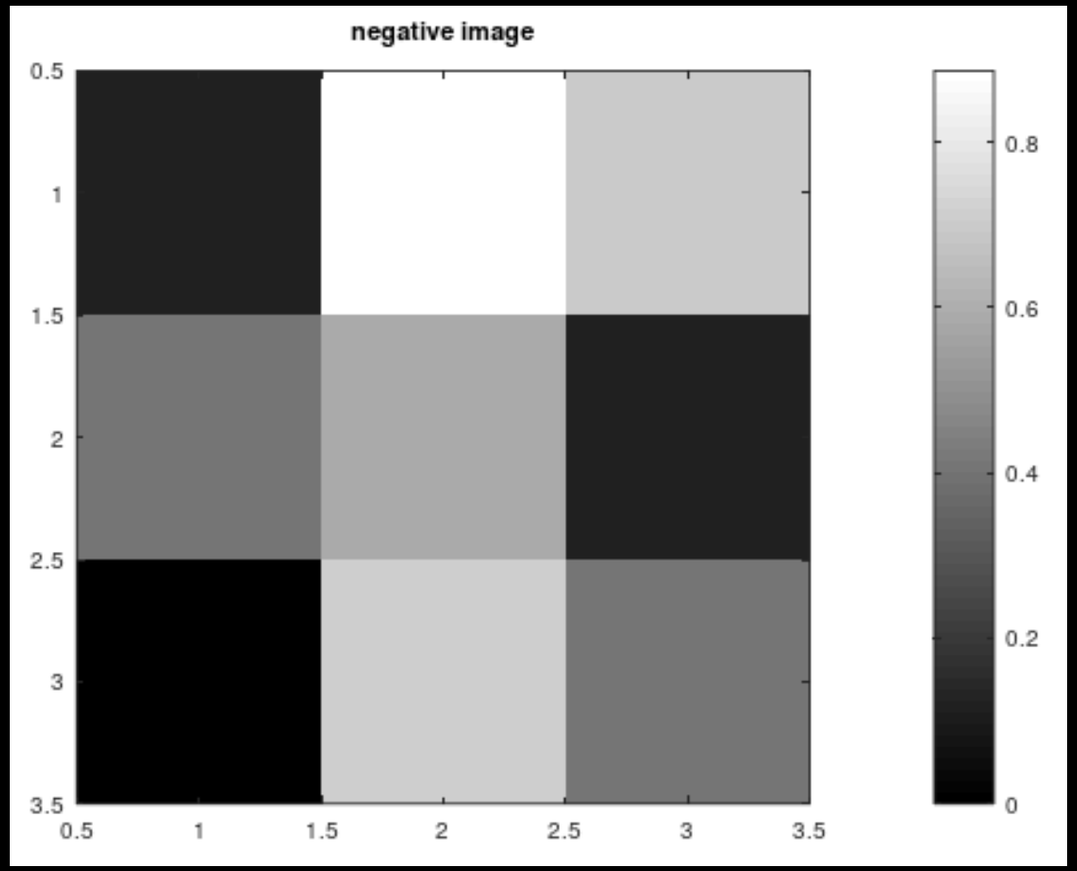
I. Introduction

- **EXERCICE 4 : Représenter le *négatif* de l'image en niveaux de gris de l'exemple précédent (*Iint*, le résultat de *ind2gray(II,map)*).**

I. Introduction

```
clear
close all

map=[1 1 0; 0 0 1; 1 0 0; 0 1 0; 1 0 1; 1 1 1; 0 .5 0; 1 .5 0];
II=[1 2 3; 4 5 1; 6 7 8];

Iint=ind2gray(II, map);
figure(1), imagesc(Iint), axis('square'), title('int. img'), colorbar
colormap(gray)

Ineg=1-Iint;
figure(2), imagesc(Ineg), axis('square'), title('neg. img'), colorbar
colormap(gray)
```





I. Introduction

4- FORMATS D'IMAGE SUR LE DISQUE

- **Avant toute chose :**

  >> pwd **(pour voir où l'on se trouve)**
  >> ls **(pour savoir ce qu'il y a là)**
  >> mkdir MATLAB **(ou '*OCTAVE*' ou autre)**
  >> cd MATLAB **(pour y aller)**
  >> mkdir 0-images **(pour créer le répertoire *0-images*)**

  puis y mettre dedans les images que vous trouverez sur

  https://lagrange.oca.eu/carbillet/enseignement/M2-GBM/images

  >> cd 0-images
  >> ls
  >> cd .. **(pour remonter d'un cran)**
  >> mkdir 1-intro
  >> mv exo.m 1-intro/. **(pour y mettre *exo.m*)**

  si vous avez appelé l'exercice précédent « exo.m »…

  >> cd 1-intro
  >> pwd
  >> ls

I. Introduction

- **Format Matlab/Octave : *.mat**

  Exemple sur l'image en niveau de gris définie précédemment *lint* (aller préalablement dans *1-intro*)

  >> save Image.mat lint → **« *Image.mat* » apparaît**
  >> clear lint
  >> whos → ***lint* n'existe plus !**
  >> load Image.mat
  >> whos → ***lint* est réapparue…**
  >> imshow(lint, 'InitialMagnification', 'fit') **(ou *imagesc(lint), colormap(gray)*)**

  Fonctionne aussi pour les images indexées (en sauvant les 2 tableaux II et map dans le même fichier : *save Image_ind.mat II map*) ou les images RGB.

I. Introduction

- **Formats graphiques usuels : jpeg, tiff, png, gif, etc.**

  Exemple sur *lint* en .tif

>> imwrite(lint, ‘Image.tif’, ‘tif’) → **« *Image.tif* » apparaît**

>> clear lint

>> whos → ***lint* n’existe plus !**

>> lint = imread(‘Image.tif’, ‘tif’);

>> whos → ***lint* est réapparue…**

>> imshow(lint, ’InitialMagnification’, ‘fit’)

`'jpg'` or `'jpeg'`	JPEG — Joint Photographic Experts Group	8-bit, 12-bit, and 16-bit Baseline JPEG images **Note** `imwrite` converts indexed images to RGB before writing data to JPEG files, because the JPEG format does not support indexed images.

I. Introduction

- **Remarques finales :**

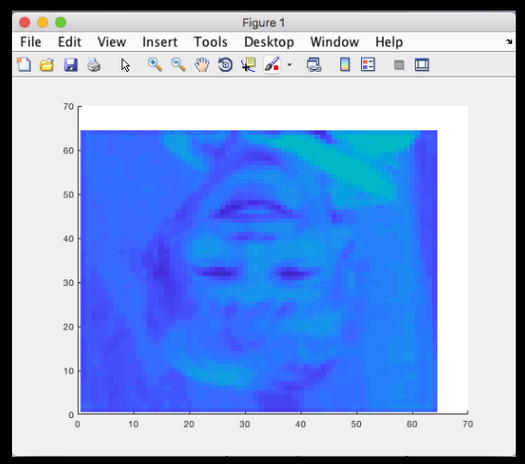
  - Du *help* de *imwrite* :

```
If the input array is of class double, and the image is a grayscale
or RGB color image, imwrite assumes the dynamic range is [0,1] and
automatically scales the data by 255 before writing it to the file as
8-bit values.
```

=> quand on charge l'image avec IMREAD, elle est donc de type *uint8* (à 256 niveaux de gris de 0 à 255)…

- Plein de TIFF (et autres formats) dans :
/Applications/MATLAB_R.app/toolbox/images/imdemos/***

- Image « cachée » :
>> image **(qui est en fait une erreur !)**



II. Analyse élémentaire d'images

1- HISTOGRAMME

- **Densité de probabilité des niveaux de gris d'une image = histogramme NORMALISÉ**

- **En abscisses : le nombre de niveaux de quantification de l'image**

- **En ordonnées : le nombre de pixels de l'image correspondant à chaque niveau de quantification**

- **Sous Matlab/Octave : instruction IMHIST**
  (faire « *>> help imhist* » pour lire les caractéristiques de la fonction (ou aussi « *>> doc imhist* » sous Matlab))
  (sous Octave : il faut au préalable avoir chargé le package 'image' à l'aide de « *>> pkg load image* »)

II. Analyse élémentaire d'images

```
-- Function File: imhist (I)
-- Function File: imhist (I, N)
-- Function File: imhist (X, CMAP)
-- Function File: [COUNTS, X] = imhist (...)
    Produce histogram counts of image I.

    The second argument can either be N, a scalar that specifies the
    number of bins; or CMAP, a colormap in which case X is expected to
    be an indexed image.  If not specified, N defaults to 2 for binary
    images, and 256 for grayscale images.

    If output is requested, COUNTS is the number of counts for each bin
    and X is a range for the bins so that 'stem (X, COUNTS)' will show
    the histogram.

    _Note:_ specially high peaks that may prevent an overview of the
    histogram may not be displayed.  To avoid this, use 'axis "auto y"'
    after the call to 'imhist'.

    See also: hist, histc, histeq.

Additional help for built-in functions and operators is
available in the online version of the manual.  Use the command
'doc <topic>' to search the manual index.

Help and information about Octave is also available on the WWW
at https://www.octave.org and via the help@octave.org
mailing list.
```

II. Analyse élémentaire d'images

- **Exemple :**
  >> I = [0.5 0.4 0.3 ; 0.4 0.3 0.4 ; 0.3 0.4 0.5];
  >> [n, x] = imhist(I, 11)
  rend :
  n = [0 0 0 3 4 2 0 0 0 0 0]
  x = [0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0]

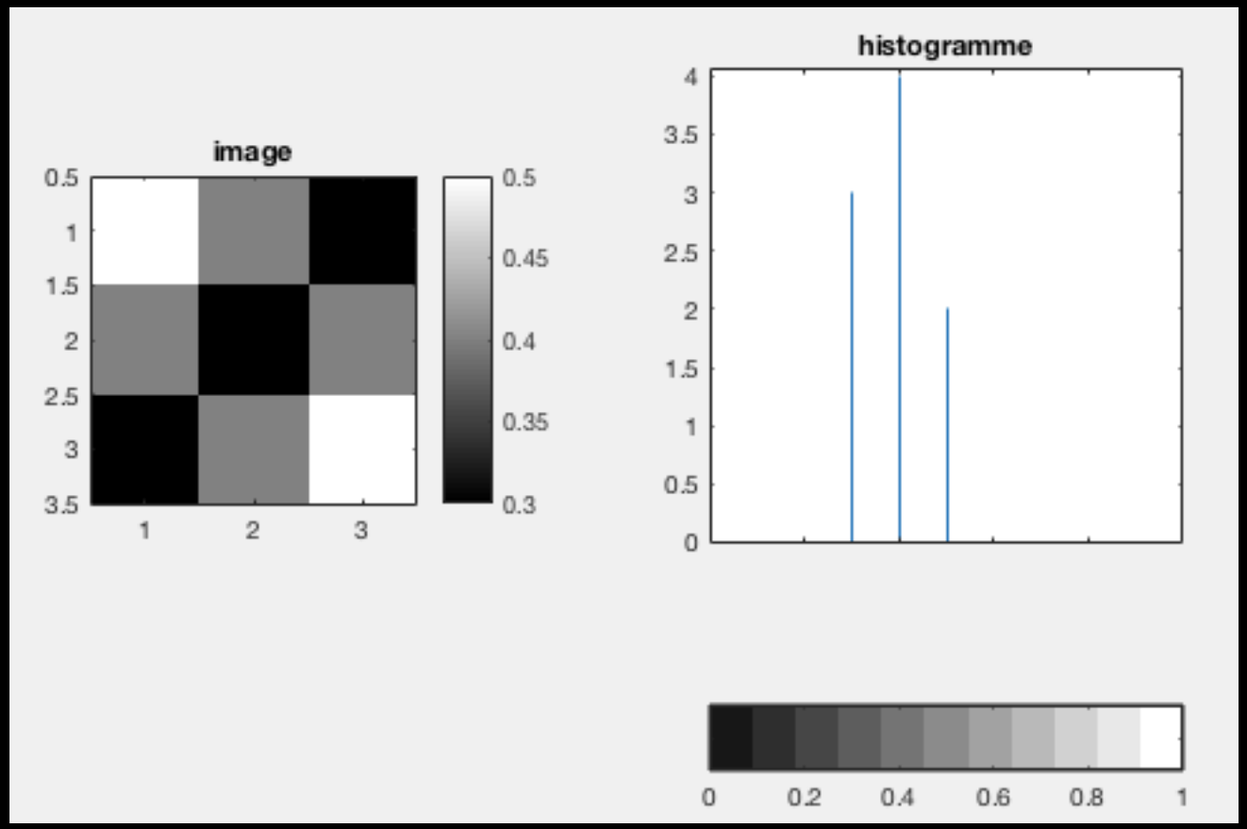
  >> figure, imhist(I, 11)
  rend : la représentation de cet histogramme

- **Représenter à la fois l'image et son histogramme :**

  >> figure, subplot(1, 2, 1), colormap(gray)
  >> imagesc(I), colorbar, axis('square')
  (ou : ***>> imshow(I, 'InitialMagnification','fit'), colorbar*****)**
  >> subplot(1, 2, 2), imhist(I, 11)

II. Analyse élémentaire d'images

```
clear
close all

I=[.5 .4 .3; .4 .3 .4; .3 .4 .5]
[n, x] = imhist(I, 11);
'niveaux de quantification dans l"image = ', x
'nb de px dans chq niveaux = ', n

figure
subplot(1,2,1)
imagesc(I), colormap(gray), colorbar, axis('square'), title('image')
subplot(1,2,2)
imhist(I, 11), axis('square'), title('histogramme')
```



II. Analyse élémentaire d’images

- **EXERCICE 1 : Lire une image plus compliquée (p.ex. frog.jpg) en couleur. La convertir en image d’intensité puis tracer son histogramme ainsi que celui de son négatif, sur 256 niveaux. Comparer.**