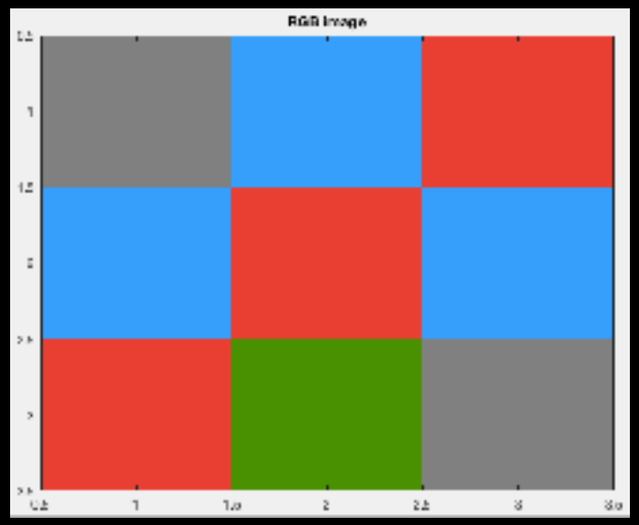
I. Introduction

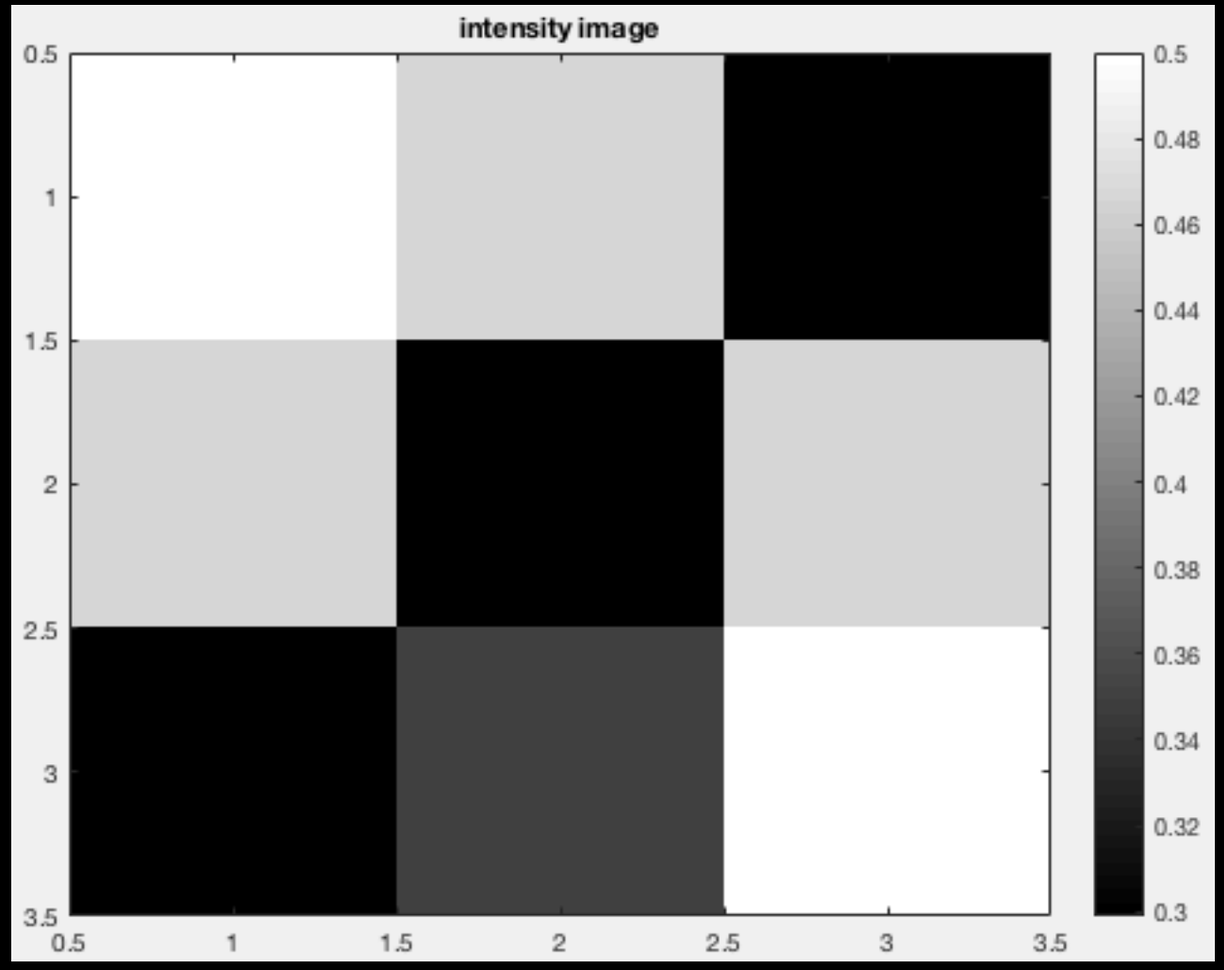
- **EXERCICE : Représenter le négatif de l'image lint de l'exemple précédent…**

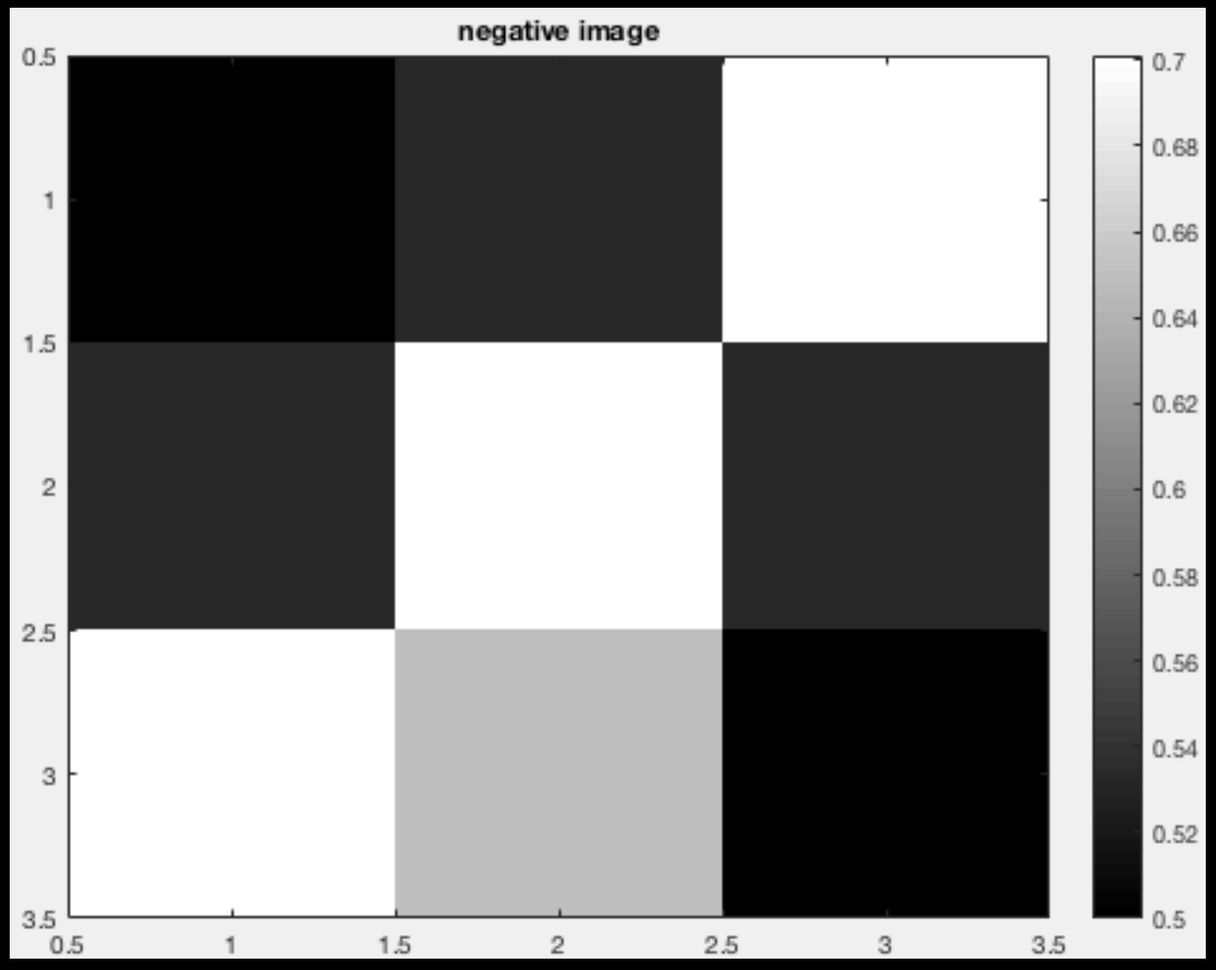
I. Introduction

```
1 -    clear
2 -    close all
3
4 -    R=[.5 0  1; 0  1  0; 1 0  .5];
5 -    G=[.5 .6 0; .6 0 .6; 0 .6 .5];
6 -    B=[.5 1  0; 1  0  1; 0 0  .5];
7
8 -    I3C=zeros([size(R) 3]);
9 -    I3C(:,:,1)=R;
10 -   I3C(:,:,2)=G;
11 -   I3C(:,:,3)=B;
12
13 -   figure(1), imagesc(I3C), title('RGB image')
14
15 -   Iint=rgb2gray(I3C);
16 -   figure(2)
17 -   imagesc(Iint), colormap(gray), title('intensity image'), colorbar
18
19 -   Ineg=1-Iint;
20 -   figure(3)
21 -   imagesc(Ineg), colormap(gray), title('negative image'), colorbar
```

I. Introduction







I. Introduction

4- FORMATS D’IMAGE SUR LE DISQUE

- **Avant toute chose :**
  >> pwd
  >> ls
  >> mkdir 1-introduction
  >> mkdir 0-images-exemples
  puis y mettre dedans les images que vous trouverez sur

https://lagrange.oca.eu/carbillet/enseignement/M2-MQM/0-images-exemples/

>> cd 0-images-exemples
>> ls
>> cd ..
>> ls
>> mv exo.m 1-introduction/.
si vous avez appelé l’exercice précédent « exo.m »…

>> cd 1-introduction
>> pwd
>> ls

I. Introduction

- **Format Matlab : *.mat**

>> save Image I3C
>> ls → **« Image.mat » sur le disque**
>> clear I3C
>> whos → **I3C n’existe plus sous Matlab**
>> load Image
>> whos → **I3C est réapparu…**

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

>> imwrite(I3C, ‘Image.tiff’, ‘tif’)
>> imwrite’I3C, ‘Image.jpg’, ‘jpg’)
>> ls → **« Image.tiff » et « Image.jpg »**
>> clear I3C
>> whos → **I3C n’existe plus sous Matlab**
>> Inew = imread(‘Image.tif’)
ou ‘Image.jpg’…
>> whos → **Inew est apparu…**
>> imagesc(Inew)

I. Introduction

- **Remarques finales :**

  - Du help de imwrite (à propos de ce que l'on vient de faire) :

```
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 de type *uint8*…

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

- Image « cachée » :
>> image (qui est en fait un 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 : instruction IMHIST**
  (faire « >> doc imhist » ou « >> help imhist »…)

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]

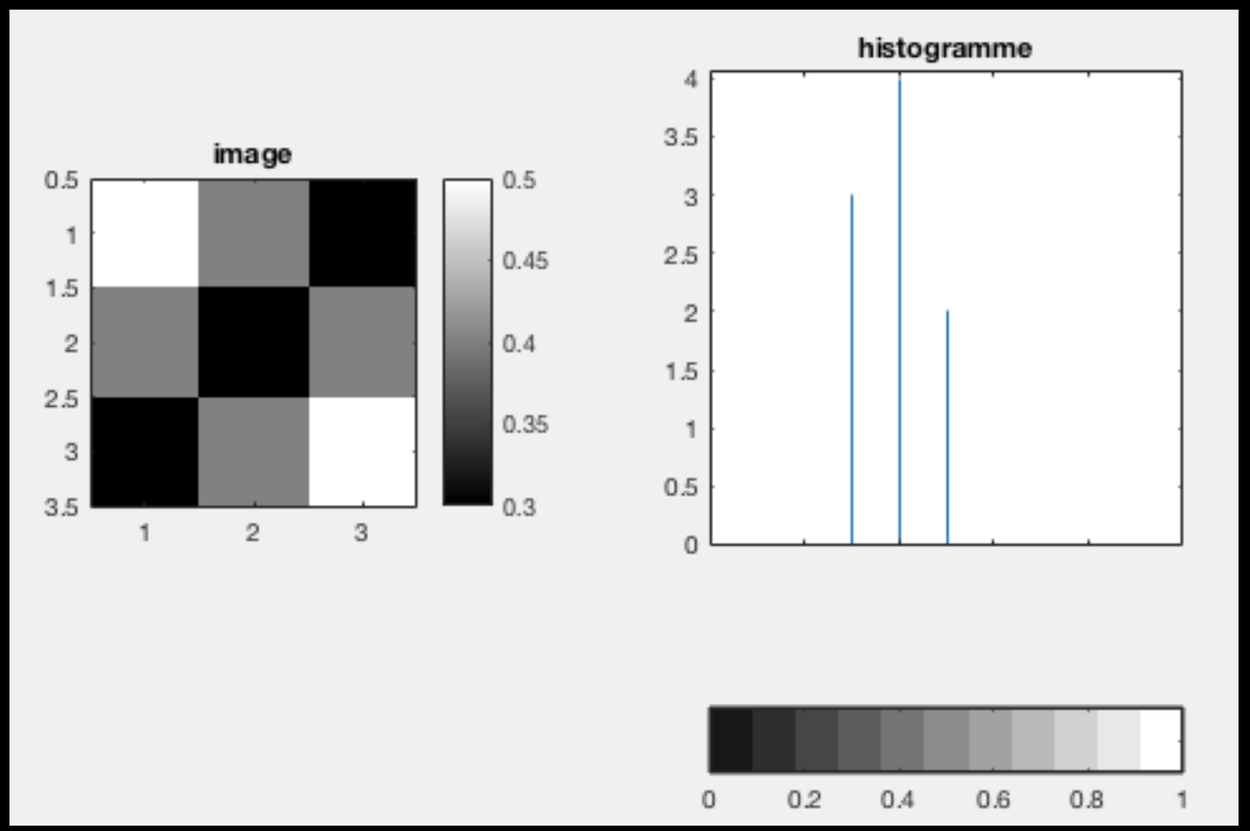
  >> figure, imhist(I, 11)
  rend : la représentation de cet histogramme

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

  >> figure
  >> subplot(1, 2, 1)
  >> colormap(gray)
  >> imagesc(I)
  >> subplot(1, 2, 2)
  >> imhist(I, 11)

II. Analyse élémentaire d'images

```
1 -    clear
2 -    close all
3
4 -    I=[.5 .4 .3; .4 .3 .4; .3 .4 .5]
5 -    [n, x] = imhist(I, 11);
6 -    'niveaux de quantification dans l''image = ', x
7 -    'nb de px dans chq niveaux = ', n
8
9 -    figure
10 -   subplot(1,2,1)
11 -   imagesc(I), colormap(gray), colorbar, axis('square'), title('image')
12 -   subplot(1,2,2)
13 -   imhist(I, 11), axis('square'), title('histogramme')
```

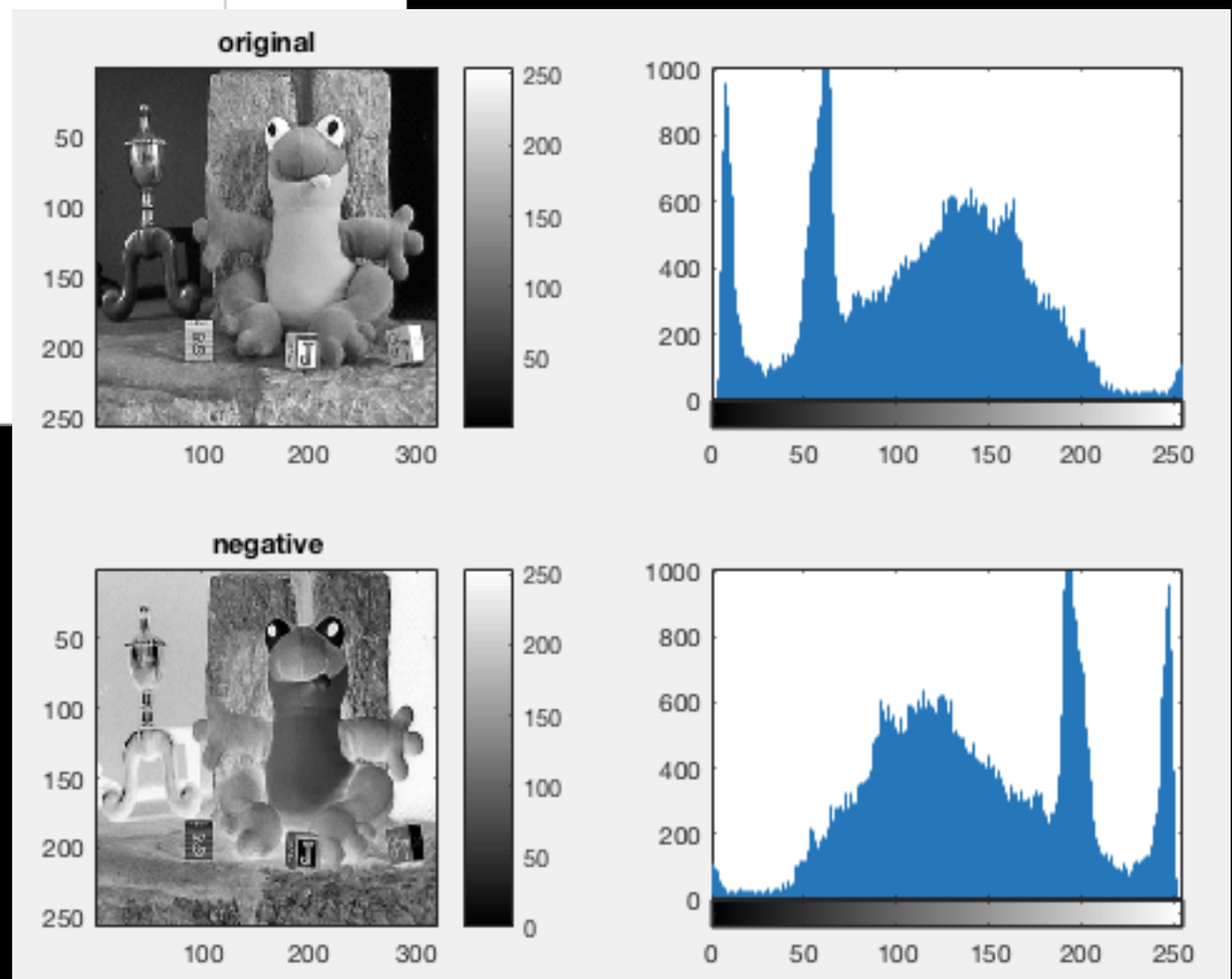


II. Analyse élémentaire d’images

- **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.**

II. Analyse élémentaire d'images

```
clear
close all

frog=imread('/Users/marcel/Documents/MATLAB/0-images-exemples/classiques/frog.jpg');
'type(frog) :', whos frog

frog_int=rgb2gray(frog);
'type(frog_int) :', whos frog_int
'min(frog_int) :', min(min(frog_int))
'max(frog_int) :', max(max(frog_int))

frog_neg=255-frog_int;
'min(frog_neg) :', min(min(frog_neg))
'max(frog_neg) :', max(max(frog_neg))

figure
subplot(2,2,1)
imagesc(frog_int), colormap(gray), title('original'), colorbar
subplot(2,2,2), imhist(frog_int, 256)
subplot(2,2,3)
imagesc(frog_neg), colormap(gray), title('negative'), colorbar
subplot(2,2,4), imhist(frog_neg, 256)
```



II. Analyse élémentaire d'images

2- ÉGALISATION D'HISTOGRAMME

- **Peut améliorer des images de mauvaise qualité (contraste : trop sombre/ trop claire, mauvaise répartition des niveaux d'intensité).**

- **But : rendre l'histogramme d'une image aussi plat que possible afin que chaque niveau d'intensité soit équitablement représenté (et ainsi profiter de toute la dynamique possible).**

- **Moyen : une transformation *f* des niveaux d'intensité qui pour l'intensité *i* d'un pixel donné dans l'image originale va rendre *f(i)* dans l'image que l'on va appeler « égalisée ».**

- **En général, on choisit une fonction *f* en escalier, en déterminant largeur et hauteur de chaque marche de manière à aplanir au mieux l'histogramme.**

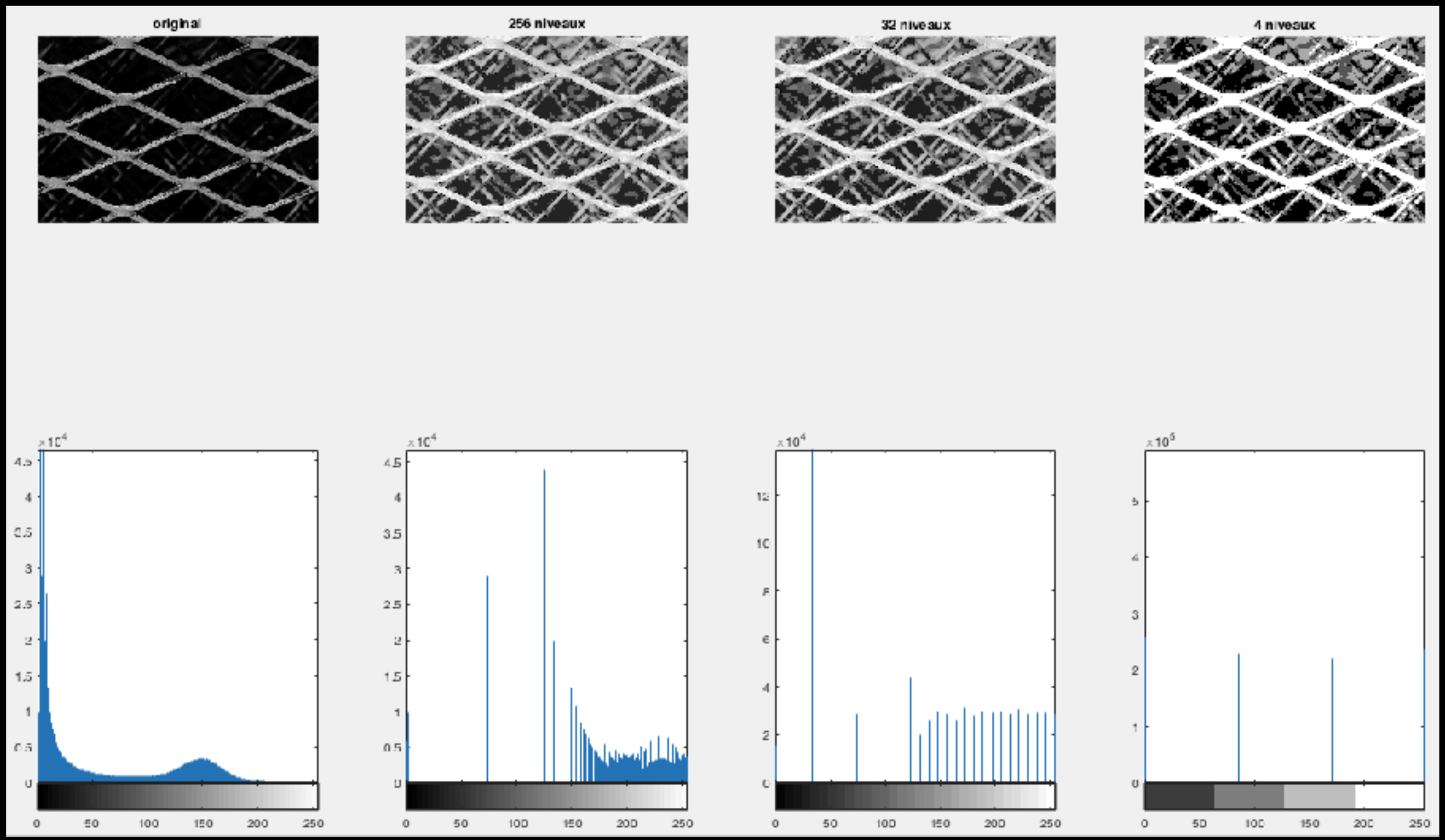
- **Sous Matlab : instruction HISTEQ**
  >> frog_eq = histeq(frog_int, n) ;
  avec : n = nombre de niveaux d'intensité dans l'image égalisée

II. Analyse élémentaire d'images

- **EXERCICE 2 : Égaliser l'histogramme d'une image de votre choix (p.ex. le grillage rouillé en niveaux de gris). Comparer avant et après (images et histogrammes). Égaliser avec moins de niveaux, que se passe-t-il ?…**

```
clear
close all

I=imread('/Users/marcel/Documents/MATLAB/0-images-exemples/materiaux/serveimage-8.jpeg'
Iint=rgb2gray(I);
figure, subplot(2,4,1), imshow(Iint), title('original')
subplot(2,4,5), imhist(Iint, 256)

n = 256;
Ieq = histeq(Iint, n);
subplot(2,4,2), imshow(Ieq), title('256 niveaux')
subplot(2,4,6), imhist(Ieq, n)

n = 32;
Ieq = histeq(Iint, n);
subplot(2,4,3), imshow(Ieq), title('32 niveaux')
subplot(2,4,7), imhist(Ieq, n)

n = 4;
Ieq = histeq(Iint, n);
subplot(2,4,4), imshow(Ieq), title('4 niveaux')
subplot(2,4,8), imhist(Ieq, n)
```

II. Analyse élémentaire d’images



II. Analyse élémentaire d'images

3-REMARQUE

- **L'histogramme représente une analyse en un seul point !**

  => négligence de toute corrélation entre les points (voisins ou pas)

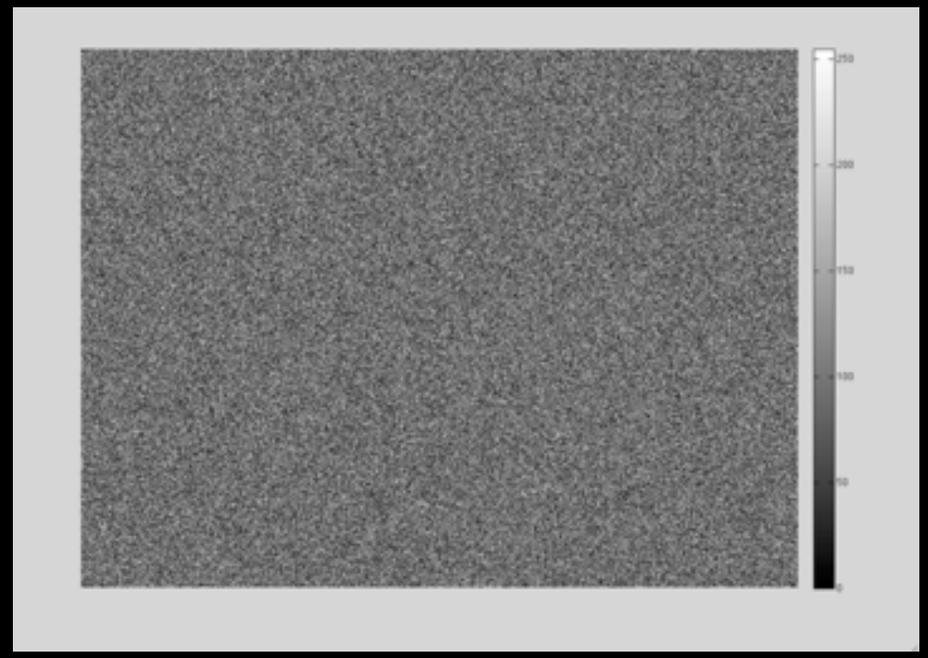
  => pas de notion de forme, de détails, de fréquence spatiale

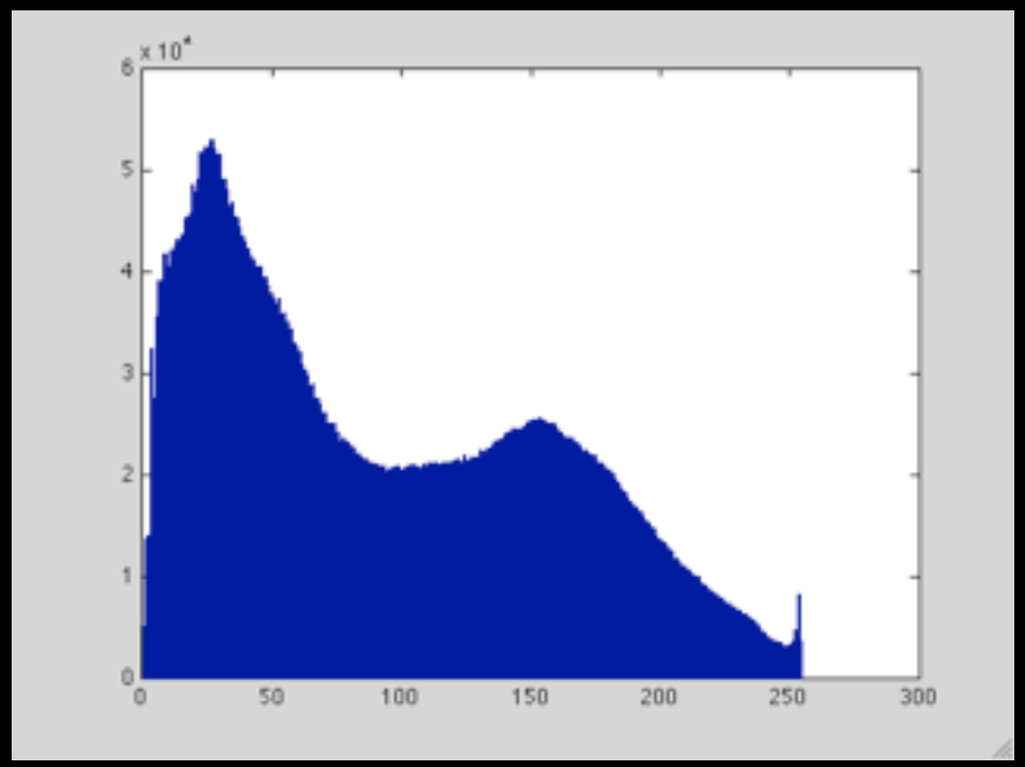
  => on peut redistribuer tous les pixels d'une image, la rendant méconnaissable, complètement différente ou même sans forme notable, sans texture aucune, elle aura toujours le même histogramme.

- **Pour avoir une idée de la texture : analyse en 2 points nécessaire**

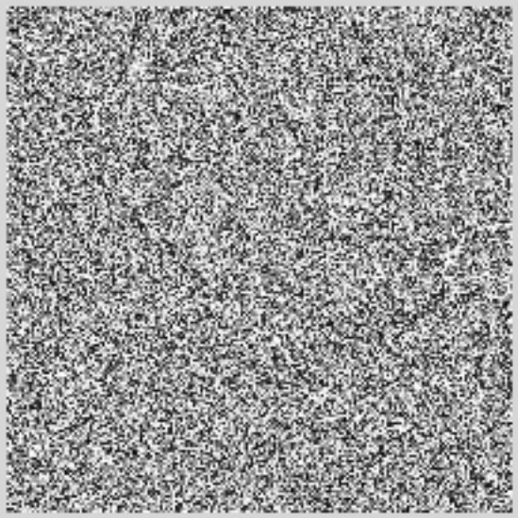
  => densité de probabilité bidimensionnelle, co-occurrence de valeurs en deux points d'une image…

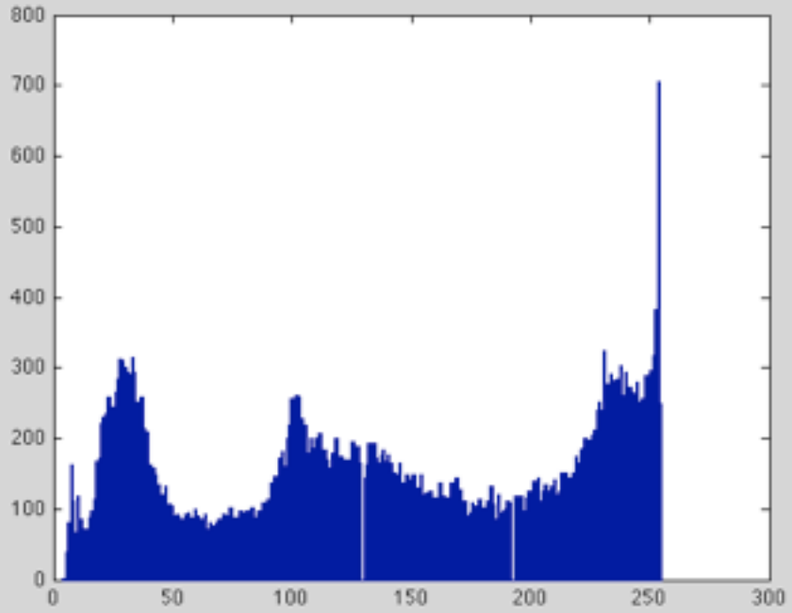
II. Analyse élémentaire d’images

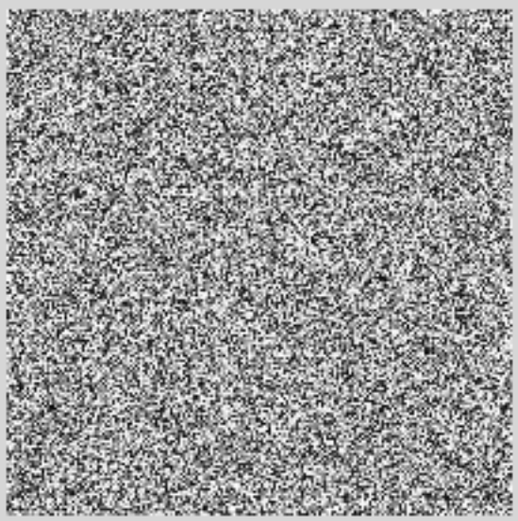




II. Analyse élémentaire d'images







II. Analyse élémentaire d'images

- **EXERCICE 3 : Reprendre l'image du début du chapitre, calculer son histogramme, puis redistribuer les valeurs des pixels dans une nouvelle image (ou même plusieurs). Calculer le nouvel histogramme. Comparer images et histogrammes.**

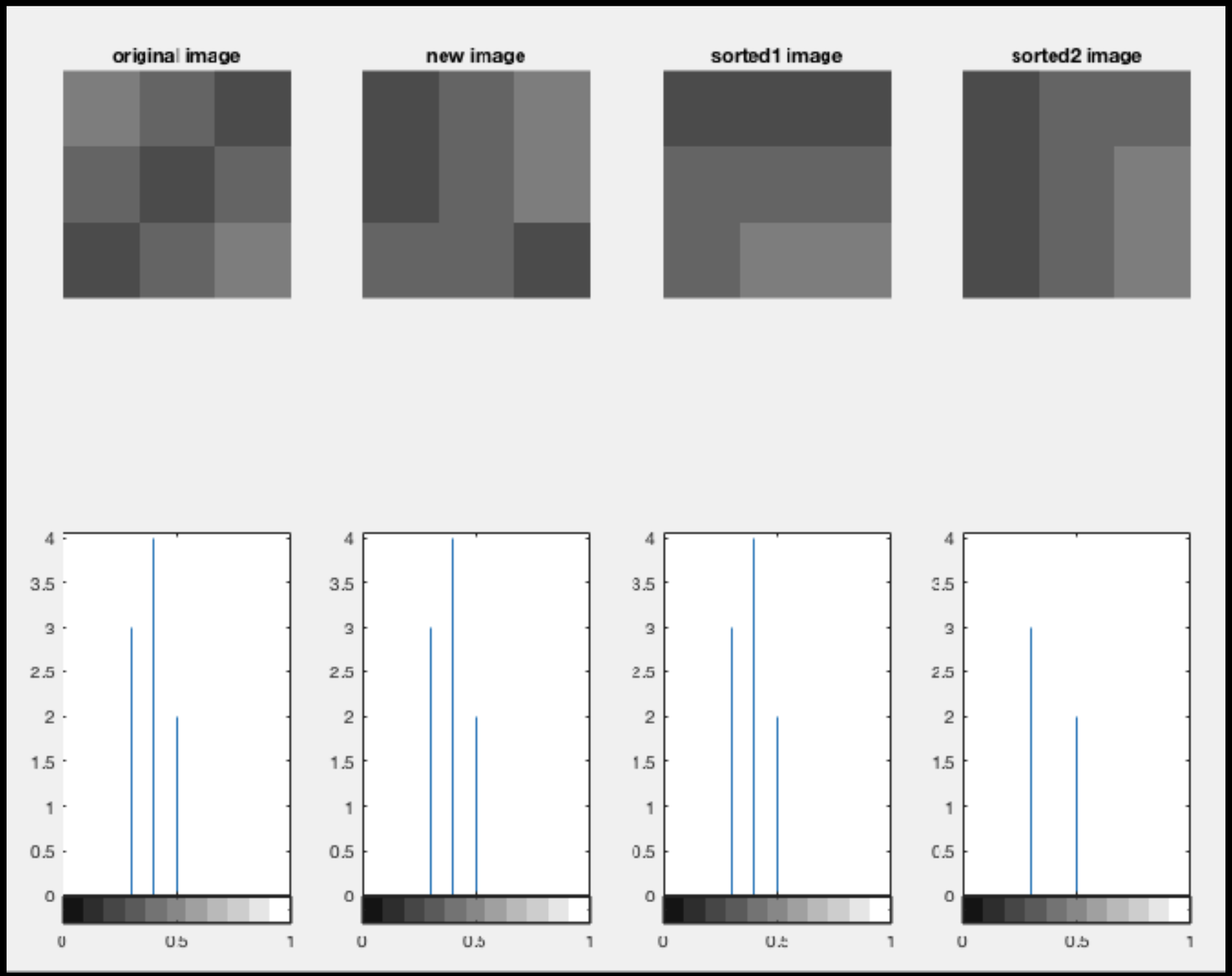
```
1 -    clear
2 -    close all
3
4 -    I=[.5 .4 .3 ; .4 .3 .4 ; .3 .4 .5];
5 -    figure
6 -    subplot(2,4,1), imshow(I), title('original image')
7 -    subplot(2,4,5), imhist(I,11)
8
9 -    I2=[.3 .4 .5 ; .3 .4 .5 ; .4 .4 .3];
10 -   subplot(2,4,2), imshow(I2), title('new image')
11 -   subplot(2,4,6), imhist(I2,11)
```

II. Analyse élémentaire d'images

- **EXERCICE 3 (suite) : Trouver une fonction pouvant transcoder l'image et vérifier que l'on a toujours le même histogramme quelque soit la distribution spatiale des pixels.**

```
clear
close all

I=[.5 .4 .3 ; .4 .3 .4 ; .3 .4 .5];
figure
subplot(2,4,1), imshow(I), title('original image')
subplot(2,4,5), imhist(I,11)

I2=[.3 .4 .5 ; .3 .4 .5 ; .4 .4 .3];
subplot(2,4,2), imshow(I2), title('new image')
subplot(2,4,6), imhist(I2,11)

I3=sort(sort(I),2)
subplot(2,4,3), imshow(I3), title('sorted1 image')
subplot(2,4,7), imhist(I3,11)

I4=sort(sort(I,2))
subplot(2,4,4), imshow(I4), title('sorted2 image')
subplot(2,4,8), imhist(I4,11)
```

II. Analyse élémentaire d’images



II. Analyse élémentaire d’images

- **EXERCICE 3bis : Appliquer à une image plus complexes (p.ex. la grenouille en niveaux de gris).**