

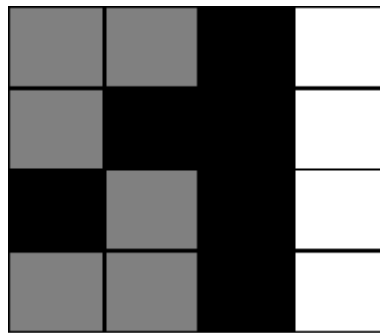
Exercice Chapitre 2 – Calcul d’histogrammes

Cet exercice va vous familiariser avec l’utilisation de la fonction Matlab *imhist* qui permet d’afficher l’histogramme d’une image. Un cas simple sur une image de faible taille est présenté afin de bien assimiler la notion d’histogramme. On vous propose ensuite de comparer les histogrammes de différentes images (monochromes et couleurs).

Histogramme

1 - Utilisez l’aide de Matlab (commande *help* ou *helpwin*) pour connaître le rôle de la fonction *imhist*.

2 – Créez la matrice *im1* qui représente l’image suivante :



avec les valeurs :

- 0 pour un pixel noir ;
- 127/255 pour un pixel gris ;
- 1 pour un pixel blanc.

Vérifiez que vos coefficients sont exacts en affichant *im1* avec la fonction *imagesc* (vous pouvez également créer une LUT pour obtenir un affichage en niveaux de gris. Pour appliquer cette LUT, utilisez la fonction *colormap*).

Affichez et commentez l’histogramme de *im1* obtenu avec la commande *imhist(im1,3)*.

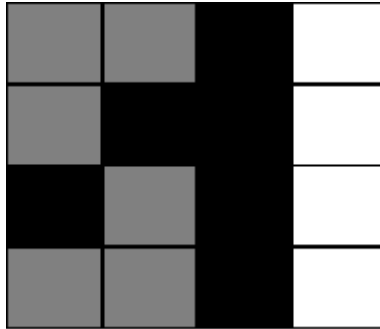
3 – Chargez les deux images monochromes « *FRUIT_LUMI* », et « *ISABE_LUMI* » dans votre répertoire de travail. Ouvrez ces deux images sous Matlab avec la fonction *imread*. Visualisez et comparez les histogrammes avec la fonction *imhist*.

4 – Chargez et ouvrez l’image couleur « *MANDRILL* ». Visualisez les histogrammes des différents plans R-V-B de cette image, en utilisant la fonction *imhist* plan par plan.

Correction de l'exercice : Calcul d'histogrammes

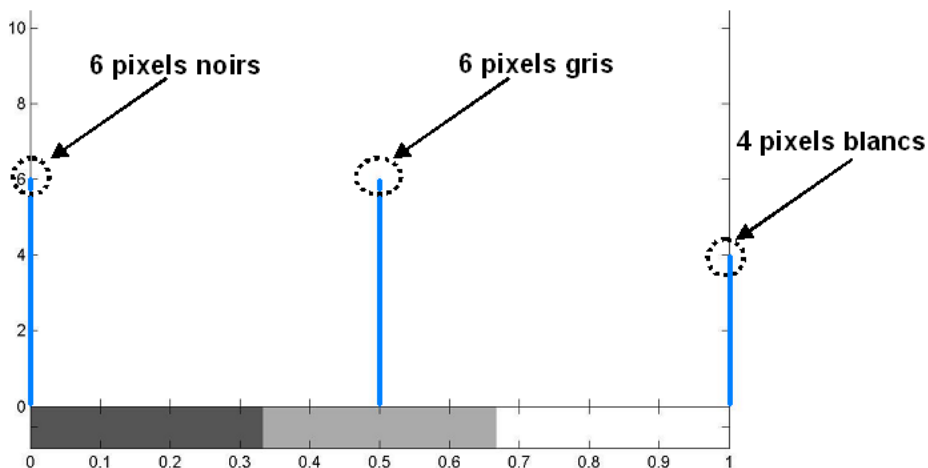
1 - La commande `imhist` permet de calculer et d'afficher l'histogramme des images en niveaux de gris. Pour les images couleurs la commande `imhist` est à utiliser plan par plan.

2 - Voici les commandes pour afficher l'histogramme de l'image 4x4 :



```
im1=[127/255 127/255 0 1;127/255 0 0 1;0 127/255 0 1; 127/255 127/255 0 1]
% Création de la LUT pour afficher en niveau de gris
r=[0 0.5 1];
v=[0 0.5 1];
b=[0 0.5 1];
map=[r' v' b'];
Création de la LUT pour afficher en niveau de gris
imagesc(im1);
colormap(map) ;
Affichage de l'histogramme
imhist(im1,3) ;
```

L'histogramme obtenu est le suivant :

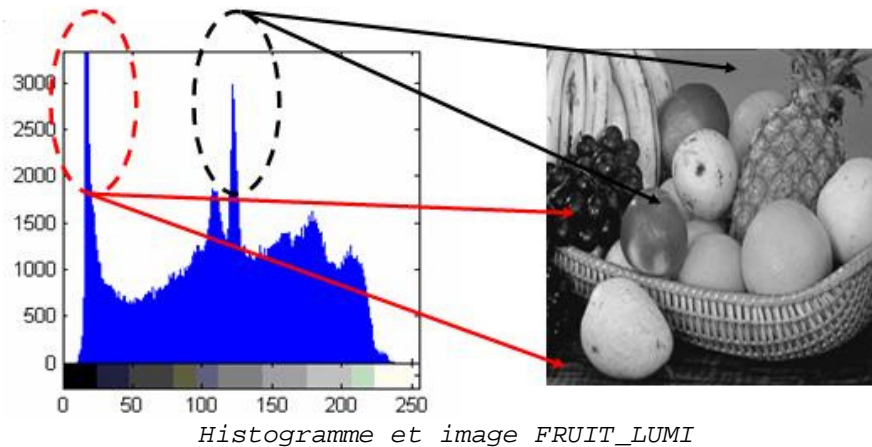


L'histogramme affiche le nombre de pixel pour chaque niveau de gris de l'image, il y'a donc 6 pixels noirs, 6 pixels gris et 4 pixels blancs (résultat facilement observable sur l'image 4x4 utilisée).

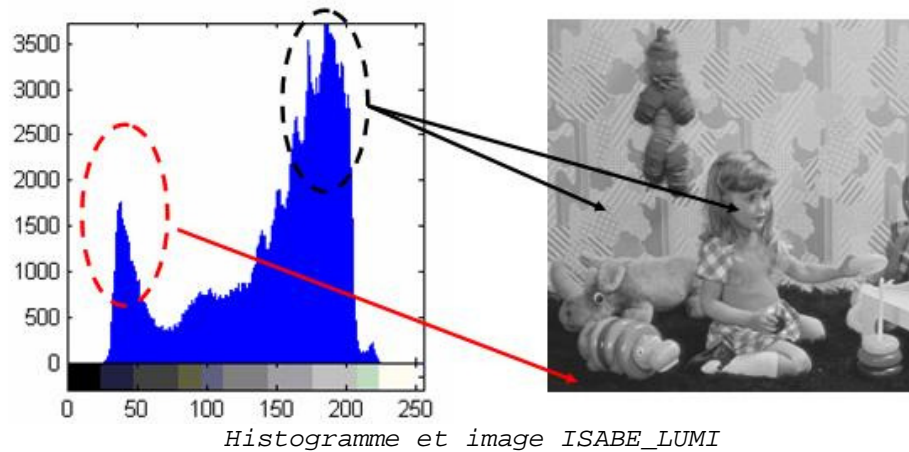
3 - Nous utilisons les images en niveaux de gris *FRUIT_LUMI.BMP* et *ISABE_LUMI.BMP*. Après avoir lu les fichiers images par la commande **imread**, nous calculons leurs histogrammes.

```
I = imread('FRUIT_LUMI.BMP') ;  
J = imread('ISABE_LUMI.BMP') ;  
imhist(I);  
imhist(J);
```

Les deux pics de populations de pixels dans l'histogramme de l'image *FRUIT_LUMI* correspondent aux régions les plus sombres de l'image (sol, raisin, ...) et aux régions de gris moyen (intensité proche de 127 pour les pommes, le fond, ...).



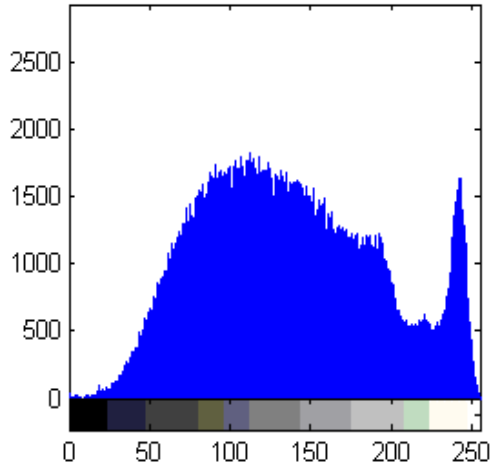
Voici l'histogramme obtenu pour l'image *ISABE_LUMI* :



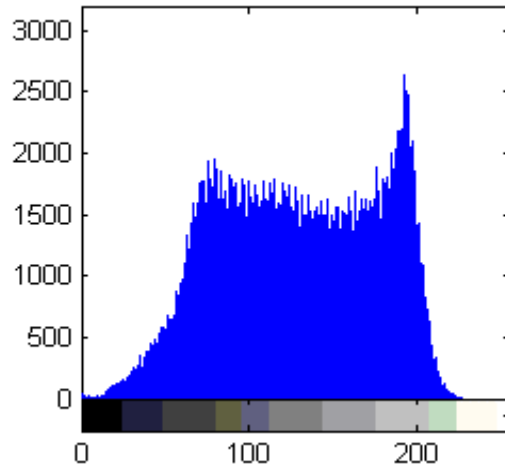
Dans ce cas les deux principaux pics de population de pixels correspondent au sol (partie la plus foncée) et à l'arrière plan de l'image (partie la plus claire).

4 -Pour les images couleurs, notamment l'image *MANDRILL.BMP*, on lit le fichier image avec la fonction **imread** puis on calcule son histogramme plan par plan.

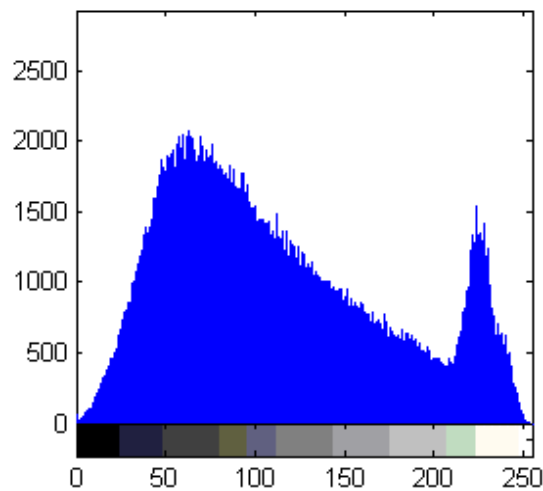
```
I = imread('MANDRILL.BMP') ;  
imhist(I(:,:,1)); % 1er plan : Rouge  
imhist(I(:,:,2)); % 2ème plan : Vert  
imhist(I(:,:,3)); % 3ème plan : Bleu
```



MANDRILL.BMP (Plan 1 : Rouge)



MANDRILL.BMP (Plan 2 : Vert)



MANDRILL.BMP (Plan 3 : Bleu)