TD III : Prétraitements globaux

III.1/ Traitements par histogramme

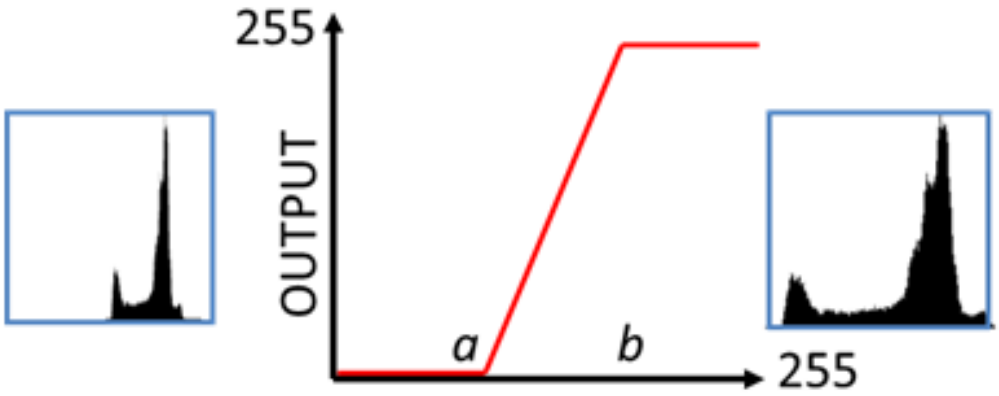
a/ Calcul d'histogramme :

Charger une l'image « image6.bmp ». Calculer son histogramme puis tracer son graphique. Utiliser pour cela l'instruction suivante :

```
[h,j]=imhist(I);
figure(1), stem(j,h);
```

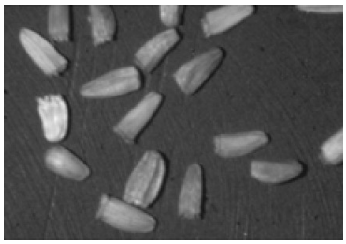
b/ Effectuer un étalement d'histogramme sur l'image avec différente valeurs des facteurs *a* et *b* (voir cours). Constater l'effet sur l'image.

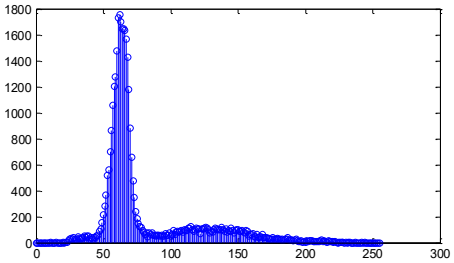
c/ Même questions avec « image7.bmp ».

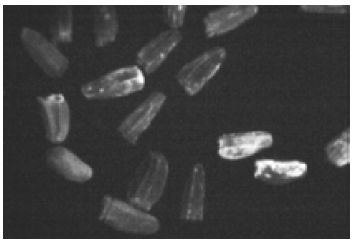


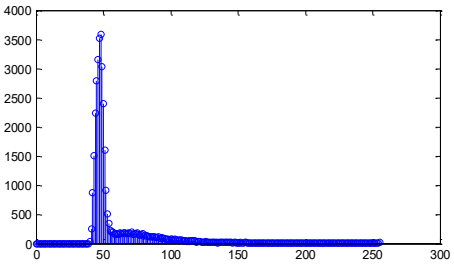
III.2/ Exemple d'application : Séparation objet/fond par seuillage automatique

Dans des applications telles que le tri ou le comptage automatiques, il peut s'avérer nécessaire de séparer des objets du fond de l'image. On se propose d'utiliser l'histogramme des niveaux de gris pour définir de façon automatique le seuil de séparation entre un objet et le fond d'une image. Cette opération peut s'avérer délicate lorsque les changements de conditions d'acquisition d'image surviennent (éclairage, sensibilité du capteur, angle de vue,…). Le seuil de séparation doit alors être adaptatif comme nous le montre la figure ci-dessous.



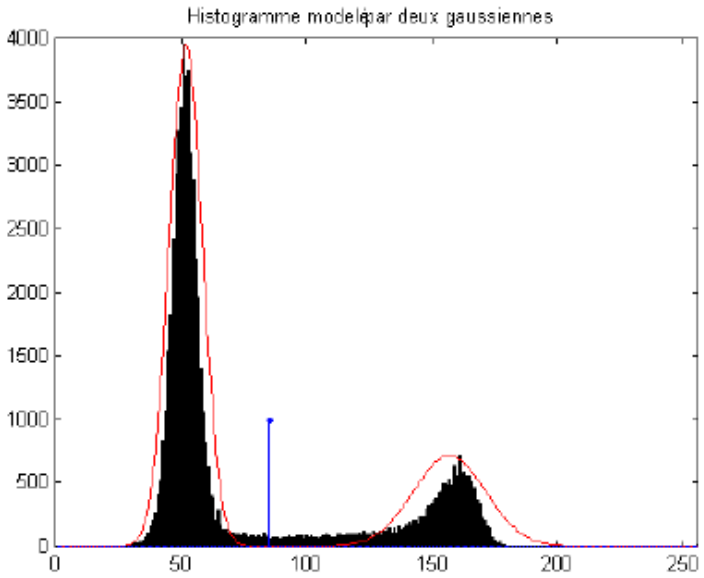






Le but de la modélisation de l'histogramme par une somme de deux gaussiennes est de le lisser mais aussi de le rendre bimodal. De cette façon, on n'aura qu'un minimum, qui correspondra au seuil choisi.

$$G(i) = A_0 . e^{\frac{-(i-m_0)^2}{2\sigma_0^2}} + A_1 . e^{\frac{-(i-m_1)^2}{2\sigma_1^2}}$$



Il faut donc commencer par effectuer une approximation de l'histogramme par une fonction G(i) en identifiant ses paramètres $X = [A_0, m_0, \sigma_0, A_1, m_1, \sigma_1]$ qui minimisent l'erreur quadratique :

$$EQ = \sum_{i=0}^{255} [G(i) - h(i)]^2$$

On utilise pour cela l'algorithme d'optimisation au sens des moindres carrés non linéaires qui converge itérativement vers le vecteur optimal X à partir d'une estimation grossière X_0.

1/ Pour visualiser le problème, charger l'image « image8.bmp » et visualiser son histogramme. Faire de même pour l'image « image9.bmp ». Constater le déplacement du seuil optimal.

2/ Ecrire la fonction d'erreur `E = ErrFun(X,h)` qui pour un vecteur de paramètres X et un histogramme h retourne le vecteur `E` des erreurs de taille 256 contenant les `E(i)= G(i)-h(i)`.

3/ Choisir une estimation grossière de X_0 pour commencer l'optimisation itérative :

```
sigma0 = 10; m0 = 50; A0 = 2000;
sigma1 = 10; m1 = 100; A1 = 100;
X0 = [sigma0, m0, A0, sigma1, m1, A1];
```

4/ Utiliser la fonction `lsqnonlin()` pour optimiser cette fonction, dite de coût, avec les données de l'image utilisée.

```
X0 = [sigma0, m0, A0, sigma1, m1, A1];
X = lsqnonlin(@(X) ErrFun(X,h), X0);
```

5/ Compléter le code pour calculer le seuil optimal. Il s'agit en fait de trouver l'index `i` entre `m0` et `m1` tel que G(i) soit minimale.

6/ Sélectionner les pixels de l'image dont le niveau de gris est supérieur au seuil trouvé grâce à l'instruction `find` puis les colorer en rouge pour mettre en évidence les objet par rapport au fond.