Traitement d'Images Numériques – TP2

L'Histogramme

Daniel Felipe González Obando
dgonzale@pasteur.fr

UNIVERSITÉ PARIS DESCARTES

Code pour le TP2

Exemples

https://github.com/imagej/tutorials

Code pour TP

https://github.com/danyfel80/descartes-image-L3

Exercice 1 – Calculer l'Histogramme

Soit l'image suivante en niveaux de gris (compris entre 1 et 100)

1. En considérant des classes d'amplitude 10, tracer l'histogramme de cette image.
2. Que pensez-vous de la qualité de la prise d'image ? Quelles modifications proposeriez-vous d'apporter pour améliorer l'image ?

12	10	8	13	25	50	46
8	7	8	10	20	23	31
11	9	10	14	28	30	37
14	11	10	26	31	28	32
50	9	11	31	35	33	41
7	12	33	35	41	50	70
13	10	35	38	75	73	72
9	14	41	45	71	76	75

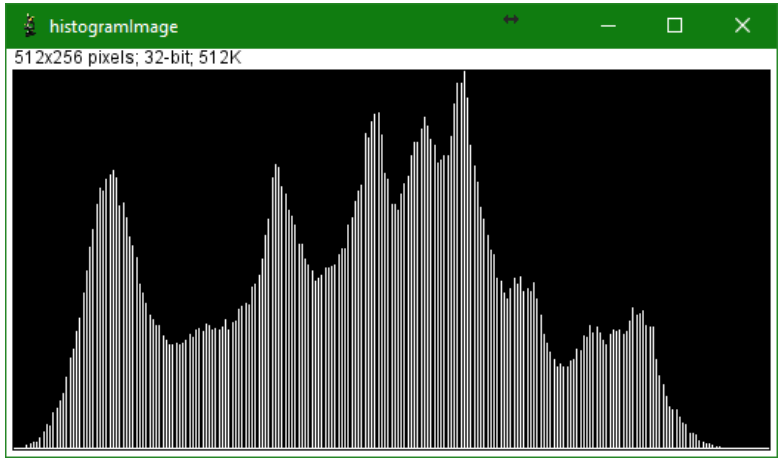
Exercice 2 – Affichage d'Histogramme

- Dans le logiciel Eclipse, éditez la classe `ComputeHistogram` dans le projet *Histogram*.

  Complétez le code de la méthode `run()` de manière à ce que dans le logiciel *ImageJ*, la commande :

  Plugins > TD 2 > Compute Histogram

  affiche l'histogramme de l'image sélectionnée.



TIP

Vous pourrez tester votre plug-in avec les images contenues dans le répertoire *testImages*, et en comparant le résultat obtenu avec le résultat de la commande *Analyze > Histogram*.

Exercice 3 - Étirement d'Histogramme

L'*étirement d'histogramme* (aussi appelé « linéarisation d'histogramme » ou « expansion de la dynamique ») consiste à répartir les fréquences d'apparition des pixels sur la largeur de l'histogramme. Ainsi il s'agit d'une opération consistant à modifier l'histogramme de telle manière à répartir au mieux les intensités sur l'échelle des valeurs disponibles. Ceci revient à étendre l'histogramme afin que la valeur d'intensité la plus faible soit à zéro et que la plus haute soit à la valeur maximale (255).

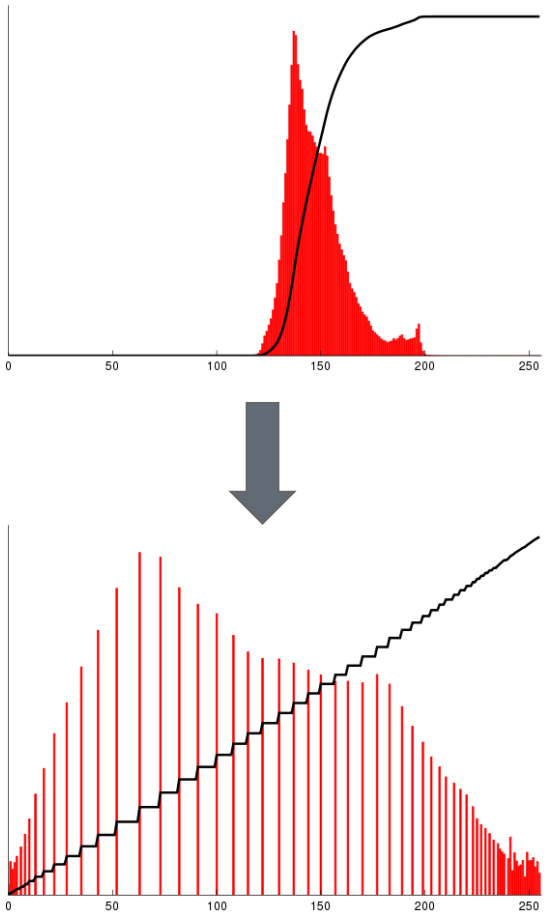
- Editez le plug-in `ExpandHistogram` du projet *Histogram* et complétez le code Java de la méthode `run()`. Testez le plug-in sur les images disponibles. Expliquez pourquoi le résultat n'est pas toujours satisfaisant.

Exercice 4 – Égalisation d'Histogramme

L'*égalisation d'histogramme* a pour but d'harmoniser la répartition des niveaux de luminosité de l'image, de telle manière à tendre vers un même nombre de pixel pour chacun des niveaux de l'histogramme. Cette opération vise à augmenter les nuances dans l'image.

En notant *h(i)* la valeur de l'histogramme pour le niveau de gris *i* et *C(i)* la valeur de l'histogramme cumulé pour le niveau de gris *i*, la LUT (*Look-Up Table*) permettant d'égaliser l'histogramme de l'image sera :

LUT(i) = G x C(i)/N ∀ i ∈[0..G[
(ici: G = 256 et N = largeur x hauteur)



Exercice 4 – Égalisation d’Histogramme

- Editez le plug-in `EqualizeHistogram` et complétez le code Java de la méthode `run()`. Testez le plug-in sur les images disponibles. Dans quels cas le résultat n'est-il pas satisfaisant ?
  A votre avis, est-ce une bonne idée d'effectuer une égalisation d'histogramme avant de faire une segmentation basée sur l'histogramme ? Pourquoi ?

Exercice 5 – Comparaison du Changement (Facultatif)

Complétez chacun des plug-ins précédents (sauf « ComputeHistogram ») de manière à afficher dans une fenêtre séparée la LUT correspondant à la transformation effectuée.
En abscisse : niveau de gris de l'image d'entrée.
En ordonnée : niveau de gris de l'image de sortie.