Objectif du TP :

- Utilisation de la librairie « minilib.jar » - programmation Java de traitements simples

Rappels JAVA : Vous devez utiliser un éditeur de texte + le compilateur / interpréteur Java pour pouvoir exécuter ce code. Ce travail présuppose donc un minimum de pré requis en programmation.
Vous disposerez de 'run.bat' qui exécute le programme sous windows et de 'compile.bat' qui prend en argument le fichier java que vous souhaitez compiler.

1. * Augmenter la luminosité de l'image « lena.jpg »

2. * Seuillage sur l'image « lena.jpg » : si la lur courant est supérieure à un seuil alors la luminance vaut 255 et 0 sinon

3. * Contraste : modifier le contraste de l'image « len
   Cres = 5*(Cori-128)+128

4. * Filtres moyenneurs :
   - filtre moyenneur 3x3 sur « lena.jpg »
   - filtre gaussien 3x3 sur « lena.jpg »
   - afficher une image des contours (voir cour

5. * Filtre médian :
   - filtrer l'image « saltandpepper.png »
   - afficher une image du bruit

6. * Etirement d'histogramme

L'*étirement d'histogramme* (aussi appelé **d'histogramme**" ou "**expansion de la dynami** répartir les fréquences d'apparition des pixels s l'histogramme. Ainsi il s'agit d'une opération cons l'histogramme de telle manière à répartir au mieux l'échelle des valeurs disponibles. Ceci rev l'histogramme afin que la valeur d'intensité la plus fa que la plus haute soit à la valeur maximale.

De cette façon, si les valeurs de l'histogramme son unes des autres, l'étirement va permettre de four répartition afin de rendre les pixels clairs encore pixels foncés proches du noir.

que Eres (luminance resultante) soit comprise entre 0 et 255.

6.2 Coder l'algorithme qui permettra d'afficher l'image traitée par étirement d'histogramme.

votre code dans un document texte de votre choix question de référence). Vous rendrez ce docume séance (pas après) en indiquant vos noms. Vous pas le temps de tout faire : rendez ce que vous avez

7. * Histogramme couleur : changement de fond

De nombreux travaux utilisent la notion d'histogramme pour décrire la répartition de caractéristiques dans une image. Ainsi, il est possible de construire des histogrammes de couleurs, des histogrammes d'orientation des gradients, des histogrammes de réponse à des filtres, ... Dans le cas d'une image couleur, chaque pixel est décrit par un vecteur de dimension trois, codant l'information couleur. Il existe un grand nombre de codages possibles pour la couleur (RGB, HSV, YUV, YCrCb, LAB, ...). Parmi ces codages certains séparent la composante de chrominance de la composante de luminance. C'est le cas, par exemple du codage HSV (H : hue ou teinte, S : Saturation, et V : Value ou luminance).
Modification automatique de teinte : L'image billard large montre une table de billard. Comme cette dernière occupe la partie la plus importante de l'image, et que sa teinte est homogène, il en résulte un pic d'histogramme de teinte important autour de cette valeur. Vous devez rechercher ce pic, et effectuer une extraction de zone autour de cette valeur. Le masque ainsi crée pourra alors être utilisé pour modifier la teinte des pixels correspondants, en appliquant un offset. Le résultat obtenu sera alors une modification de la teinte des pixels du tapis du billard.
On utilisera les méthodes (java.awt.*) : Color.RGBtoHSB et Color.HSBtoRGB et la classe Color.