

## TP n°1

### Prise en main d'une librairie de traitement d'images

L'objectif de ce TP est de manipuler et traiter une image à partir d'une librairie de traitement des images en langage C (écriture similaire au langage JAVA).

Les TP se dérouleront sous LINUX avec un terminal, un éditeur de texte et un logiciel de tracé de courbes GNUPLOT. Il est fortement conseillé de créer un répertoire TP\_image\_1 et de tout sauvegarder dans le même répertoire.

## 1) Seuillage d'une image

A partir des programmes `test_grey.cpp` et `image_pgm.h` téléchargés depuis :

[http://www.lirmm.fr/~wpuech/enseignement/DUT\\_SRC/1A/traitement\\_images/librairie](http://www.lirmm.fr/~wpuech/enseignement/DUT_SRC/1A/traitement_images/librairie)

- Ouvrir ces fichiers avec un éditeur de texte et regarder leurs contenus.
- Ces programmes manipulent des images au format ppm. Rechercher des informations sur ce type de format. Télécharger des images au format ppm depuis :

[http://www.lirmm.fr/~wpuech/enseignement/DUT\\_SRC/1A/traitement\\_images/images/](http://www.lirmm.fr/~wpuech/enseignement/DUT_SRC/1A/traitement_images/images/)

- Compiler le programme `test_grey.cpp` de la manière suivante :

```
g++ -c test_grey.cpp (cette commande génère test_grey.o)
```

```
g++ -o test_grey test_grey.o (cette commande crée l'exécutable test_grey)
```

```
./test_grey image_entrée.pgm image_sortie.pgm  
valeur_du_seuil (exécute le programme)
```

En fait, le programme téléchargé permet de seuiller une image en 2 parties :

Si  $p(i,j) < \text{Seuil}$

alors  $p(i,j) = 0$ , (noir)

Sinon  $p(i,j) = 255$  (blanc)

- Tester plusieurs valeurs de seuil.

## 2) Seuillage d'images avec plusieurs niveaux S1, S2 (,S3)

A partir du programme utilisé dans la première partie, écrire un nouveau programme afin de seuiller une image :

- en 3 parties
- en 4 parties.

Pour trois parties :

Si  $p(i,j) < S1$

alors  $p(i,j) = 0$

Sinon Si  $p(i,j) < S2$  alors  $p(i,j) = 128$  (gris)

Sinon  $p(i,j) = 255$

Dans le cas où  $S1 < p(i,j) < S2$ , proposer une autre valeur pour  $p(i,j)$ .

Tester plusieurs valeurs de  $S1$  et  $S2$ , puis  $S1$ ,  $S2$  et  $S3$ .

### 3) Profil d'une ligne ou d'une colonne d'une image

Ecrire un programme profil permettant d'afficher à l'écran sur 2 colonnes les indices et les niveaux de gris d'une ligne ou d'une colonne d'une image. Cette programme aura comme arguments, le nom l'image, une information précisant s'il s'agit d'une ligne ou d'une colonne, et un indice indiquant le numéro de la ligne ou de la colonne.

Au lieu d'afficher les valeurs des pixels d'une ligne ou d'une colonne à l'écran, nous souhaitons maintenant les visualiser sous forme de courbes à l'aide du logiciel GNUPLOT.

Pour cela, il faut rediriger l'affichage de l'écran vers un fichier que nous appellerons `profil.dat` qui sera ensuite visualisé avec le logiciel Gnuplot.

Visualiser les profils d'une ligne et d'une colonne d'une image au format pgm indiquée par l'enseignant.

A l'aide du logiciel GNUPLOT, pour visualiser une courbe :

```
plot 'fich.dat' with lines
```