

TP n°1

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

L'objectif de ce premier TP est de manipuler et traiter des images à partir d'une librairie de traitement des images

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 au format pgm

A partir des programmes téléchargés depuis :

https://www.lirmm.fr/~wpuech/enseignement/donnees_multimedia/librairie/

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

https://www.lirmm.fr/~wpuech/enseignement/donnees_multimedia/images/

- c) Compiler le programme `test_grey.cpp` et exécuter le.

En fait, le programme téléchargé permet de seuiller une image pgm (niveau de gris) en 2 parties :

Si pixel d'entrée $p_{in}(i,j) < \text{Seuil}$
alors pixel de sortie $p_{out}(i,j) = 0$, (noir)
Sinon $p_{out}(i,j) = 255$ (blanc)

- d) Tester plusieurs valeurs de seuil.

2) Seuillage d'une image pgm avec plusieurs niveaux S1, S2, S3

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

- a) en 3 parties
- b) en 4 parties.

Pour trois parties :

Si $p_{in}(i,j) < S1$
alors $p_{out}(i,j) = 0$
Sinon Si $p_{in}(i,j) < S2$ alors $p_{out}(i,j) = 128$ (gris)
Sinon $p_{out}(i,j) = 255$

Dans le cas où $S1 < p_{in}(i,j) < S2$, proposer une autre valeur pour $p_{out}(i,j)$.

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

3) Histogramme d'une image pgm

- Ecrire un programme `histo.cpp` permettant d'afficher à l'écran, puis ensuite de sauvegarder dans un fichier, les données de l'histogramme d'une image. Le fichier contiendra 2 colonnes : indice et occurrence des niveaux de gris.
- Compiler et exécuter votre programme avec une image en niveau de gris au format pgm.
- A l'aide du logiciel GNUPLOT, visualiser l'histogramme :

```
> plot "histo.dat" with lines
```

4) Profil d'une ligne ou d'une colonne d'une image pgm

Ecrire un programme `profil.cpp` 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. Ce 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
```

5) Seuillage d'une image couleur (ppm)

A partir d'une image en couleur au format ppm, écrire un programme permettant de seuiller **séparément** chacune des trois composantes couleurs (Rouge, Vert, Bleu) de cette image. 3 paramètres `S_R`, `S_G`, `S_B` sont alors nécessaires.

Visualiser l'image de sortie pour différentes valeurs de seuils.

6) Histogrammes des 3 composantes d'une image couleur (ppm)

A partir d'une image en couleur au format ppm, écrire un programme permettant de visualiser à l'écran, puis de sauver dans un fichier, les données des histogrammes des trois composantes couleur (Rouge, Vert, Bleu) de cette image.

Le fichier contiendra 4 colonnes : indice (de 0 à 255) et occurrence du rouge, puis du vert et enfin du bleu. A l'aide du logiciel GNUPLOT, visualiser sur un même graphique les trois histogrammes :