

TP n°4

TP de traitement d'image

L'objectif de ce TP est de traiter une image et d'examiner les résultats au niveau des histogrammes. Il est fortement conseillé de créer un répertoire TP_image_4 et de tout sauvegarder dans le même répertoire. Prendre l'image au format pgm indiquée par l'enseignant.

http://www.lirmm.fr/~wpuech/enseignement/DUT_SRC/1A/traitement_images

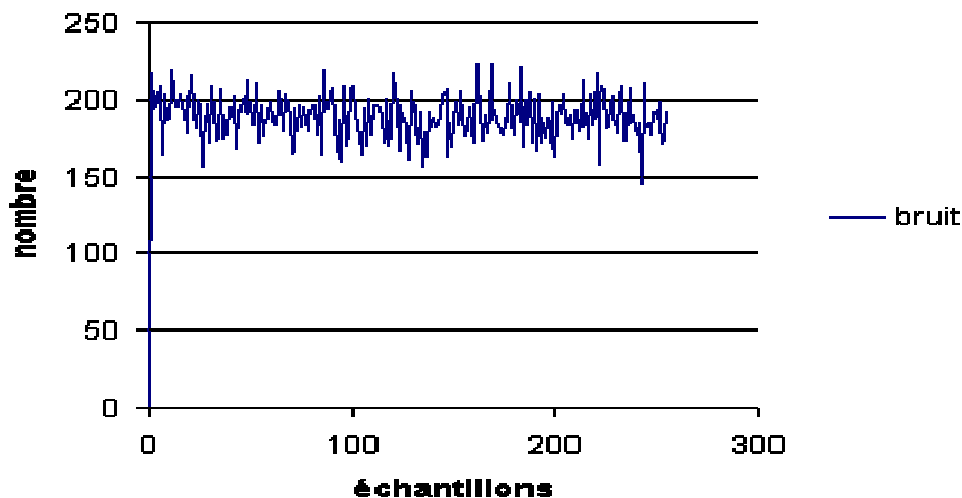
1) Histogramme d'une image

- a) A partir d'une image en niveau de gris au format pgm et des programmes `test_grey.cpp` et `image_ppm.h` téléchargés, écrire le programme `histogramme.cpp` qui permet d'afficher à l'écran puis de sauvegarder dans un fichier (`histo.dat`) les données de l'histogramme de cette image. Le fichier contiendra 2 colonnes : indice et occurrence des niveaux de gris.
- b) A l'aide du logiciel GNUPLOT, visualiser l'histogramme :
`plot 'histo.dat' with lines`

2) Bruit blanc additif

Un bruit blanc additif est un bruit qui se rajoute au signal avec une probabilité uniforme. Ce bruit peut modéliser, par exemple, l'impression d'une image.

- a) Ecrire un programme `bruit_blanc.cpp` qui rajoute un bruit blanc dans une image. Ce programme aura en entrée l'amplitude du bruit ajouté. Par exemple, si `ampli = 5`, alors le bruit ajouté sera un nombre aléatoire compris entre -5 et $+5$.
- b) Appliquer ce programme à l'image pgm avec 3 valeurs pour l'amplitude. Pour chacune de ces images visualiser l'histogramme et les profils.
- c) Prendre une image de synthèse et appliquer le même traitement.



3) Filtre flou

Créer un programme `filtre_flou.cpp` qui va remplacer, dans l'image de sortie, la valeur d'un pixel par la valeur moyenne de ce pixel avec ses 4 voisins :

$$p_out(i, j) = (p(i, j) + p(i-1, j) + p(i+1, j) + p(i, j-1) + p(i, j+1)) / 5.$$

Il est fortement recommandé de faire attention aux bords de l'image. En effet, les première et dernière colonnes, ainsi que les première et dernière lignes ne peuvent pas être traitées, il faudra conserver les valeurs initiales.

- Prendre une des images bruitées (obtenue à la question 2) et lui appliquer le filtre flou.
- Avec l'aide du logiciel GNUPLOT comparer les trois histogrammes (image originale, image bruitée, image floutée)

4) Filtre passe haut

Créer un programme `filtre_passe_haut.cpp` qui va remplacer, dans l'image de sortie, la valeur d'un pixel par la différence de ce pixel multipliée par 4 avec ses 4 voisins :

$$p_out(i, j) = 4 * p(i, j) - p(i-1, j) - p(i+1, j) - p(i, j-1) - p(i, j+1).$$

Il est également recommandé de faire attention aux bords de l'image. En effet, les première et dernière colonnes, ainsi que les première et dernière lignes ne peuvent pas être traitées, il faudra conserver les valeurs initiales.

- Prendre l'image originale et lui appliquer le filtre passe haut.
- Avec l'aide du logiciel GNUPLOT comparer les deux histogrammes (image originale et image filtrée)

Question bonus) Filtre médian

Créer un programme `filtre_median.cpp` qui va remplacer, dans l'image de sortie, la valeur d'un pixel par la valeur médiane entre ce pixel et ses 4 voisins.

La valeur médiane d'une série est par définition celle qui sépare l'échantillon en deux parties de population égale.

- Prendre l'image bruitée et lui appliquer le filtre médian.
- Avec l'aide du logiciel GNUPLOT comparer les deux histogrammes (image bruitée et image filtrée)