TP2 : Traitement d'image

TP à faire sous PC-Linux par binôme.
Compte-rendu à envoyer au format PDF (par exemple document *Libre Office Writer* exporté au format PDF) à l'adresse `nicolas.szafran@univ-grenoble-alpes.fr` avant le 22 février 2017 en indiquant comme sujet de l'e-mail :
[L3-Info] - Image - TP2 - *noms du binôme* .

Récupérez le fichier archive `tp2.tar` à l'URL suivante :
http://www-ljk.imag.fr/membres/Nicolas.Szafran/ENSEIGNEMENT/L3
puis décompactez l'archive avec la commande `tar -xvf tp2.tar` .
Pensez à (re)créer la librairie ***lib_image*** : `make lib_image` .

---

1 - Histogramme

1.1 - Image en niveaux de gris / image couleur

Le programme `histogramme1.cpp` charge une image en niveaux de gris à partir d'un fichier nommé *fichier_image*, calcule et affiche de différentes manières l'histogramme correspondant avec *nb_classes* classes.

Exercice 1 :

A partir de ce programme, écrivez un autre programme afin de charger une image couleur, puis calculer et afficher les trois histogrammes correspondant aux trois composantes *Rouge*, *Vert* et *Bleu* ainsi que l'histogramme correspondant à l'image couleur convertie en niveaux de gris.
Dans votre rapport, mettez le code source du programme, et les images d'exemples.

2 - Modification globale par fonction

Le programme `fonction1.cpp` montre un exemple de modification globale d'une image à l'aide d'une fonction.
Dans la fonction `traitement_image`, à partir d'une image `I` et d'une fonction `F` :

- l'histogramme de `I` sur 32 classes est calculé et affiché,
- sa moyenne et l'écart-type sont calculés et affichés,
- l'image modifiée `I_F` = `F(I)` est calculée et affichée,
- son histogramme sur 32 classes est calculé et affiché,
- sa moyenne et son écart-type sont calculés et affichés.

Dans un premier temps, testez les différentes fonctions-exemples vus en cours.

Exercice 2 :

Pour chacune des images suivantes, proposez une fonction afin de l'améliorer visuellement.

- `arbre-brume.png`
- `oiseau.png`
- `amphi-weil.png`
- `plante.png`

Dans votre rapport, pour chaque image, mettez l'expression de la fonction utilisée et l'image modifiée.

Exercice 3 :

L'image en niveaux de gris `belledonne.pgm` a un histogramme étendu sur $[\,0.15\,,\,1\,]$ et avec plusieurs pics.
Proposez une fonction affine par morceaux afin de :

- étendre l'histogramme sur $[\,0\,,\,1\,]$, c'est à dire déplacer la valeur 0.15 à la valeur 0.0.
- déplacer le pic le plus à gauche de la valeur 0.3 à la valeur 0.15,
- conserver le pic le plus à droite (dont la valeur est à peu près égale à 0.7).

Dans votre rapport, mettez l'expression de la fonction, et l'image modifiée .

3 - Transformations géométriques

3.1 - Agrandissement d'image

Le programme `agrandissement1.cpp` effectue différents agrandissements d'une même image en utilisant l'interpolation au plus proche.

Exercice 4 :

Complétez le programme pour ajouter une fonction d'agrandissement en utilisant l'interpolation bilinéaire.
Comparez ensuite les deux interpolations sur des images de votre choix.
Dans votre rapport, mettez le code source de votre fonction utilisant l'interpolation bilinéaire, et un comparatif des deux méthodes sur différentes images.

4 - Filtrage

4.1 - Filtre linéaire

Le programme `filtre_lineaire1.cpp` permet de définir des filtres linéaires particuliers (moyenne, chapeau, gaussien).
Dans ce programme, les indices pour accéder aux valeurs du noyau d'un filtre vont de $-n$ à n et non pas de 0 à $2n$:

$$R(x,y) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} K(i,j) \times I(x+i, y+j)$$

Exercice 5 :

Complétez ce programme (fonction `convolution`) afin d'implémenter l'opération de convolution.
Comparez ensuite les différents filtres sur des images de votre choix.
Dans votre rapport, mettez le code source de l'opération de convolution, indiquez quelle méthode vous utilisez pour les pixels de bord, et des exemples de filtres appliquées à des images.

4.2 - Filtre médian

Exercice 6 :

Complétez le programme `filtre_lineaire1.cpp` afin d'implémenter le filtre médian 3×3 (vous pouvez utiliser la fonction de tri fournie).
Comparez ensuite le filtre médian et le filtre moyenne 3×3 sur les images suivantes :

- `image_bruitee_1.png`
- `image_bruitee_2.png`
- `image_bruitee_3.png`

Dans votre rapport, mettez le code source du filtre médian, et le comparatif des deux filtres sur ces trois images.