TP3 : Analyse d'image - détection de contour

TP à faire sous PC-Linux par binôme.
Compte-rendu à envoyer à l'adresse `nicolas.szafran@univ-grenoble-alpes.fr` avant le 10 mars 2017 en indiquant comme sujet de l'e-mail :
[L3-Info] - Image - TP3 - *noms du binôme* .

Pour ce compte rendu, faire une archive avec :
- les trois fichiers `lib_image.cpp`, `lib_image.hpp`, `CImg.h`
- votre fichier `Makefile`,
- vos programmes sources des parties 1, 2 et 3,
- et un rapport au format PDF avec explications sur la structuration de vos programmes, instructions d'utilisation de ceux-ci, résultats, images et commentaires éventuels.

Récupérez le fichier archive `tp3.tar` à l'URL suivante :
http://www-ljk.imag.fr/membres/Nicolas.Szafran/ENSEIGNEMENT/L3
puis décompactez l'archive avec la commande `tar -xvf tp3.tar` .
Pensez à (re)créer la librairie ***lib_image*** : `make lib_image` .

---

1 - Filtres détecteurs de contour

1.1 - Traitement d'image en niveaux de gris

Le programme `filtre_contour_gris.cpp` charge une image `I` en niveaux de gris, applique le filtre *dérivée (méthode 1)* pour obtenir l'image `I_F` :

$$\texttt{I\_F} = \frac{1}{8}\sum_{k=1}^{8}|I_k| = \frac{1}{8}\big(|I_1|+|I_2|+|I_3|+|I_4|+|I_5|+|I_6|+|I_7|+|I_8|\big)$$

puis seuille automatiquement l'image `I_F` pour obtenir une image `I_S` en noir et blanc.

Exercice 1 :

Complétez ce programme pour :

1. écrire une fonction `filtre_derivee_methode2` qui calcule le filtre *dérivée* par la méthode 2 suivante :
$$\texttt{I\_F} = \max_k |I_k| = \max\big(|I_1|,|I_2|,|I_3|,|I_4|,|I_5|,|I_6|,|I_7|,|I_8|\big)$$
2. implémenter les filtres de *Sobel* et *Prewitt* en utilisant la méthode 2.

Comparez les filtres *dérivée*, *Sobel* et *Prewitt* (méthode 2) sur différentes images.

2 - Détection de droite par transformée de Hough

• Partie A

Dans cette partie, le but est de calculer la transformée de Hough d'une image en noir et blanc, d'afficher l'image et sa transformée de Hough.

Exercice 2 :

Complétez la fonction `transformee_hough` du programme `transformee_hough.cpp` afin de calculer la transformée de Hough d'une image en noir et blanc.
Testez ensuite votre programme complété avec différentes images tests.
Modifiez l'image `I` et observez la transformée de Hough correspondante :

1. dans la fonction `image_test_1`, modifiez la position du pixel blanc afin de tester différentes configurations,
2. dans la fonction `image_test_1`, mettez quelques pixels blancs alignés, et testez différentes configurations,
3. ensuite dans le programme principal, utilisez la fonction `image_test_2` à la place de `image_test_1`,
4. dans la fonction `image_test_2`, modifiez le segment afin de tester différentes configurations,
5. dans la fonction `image_test_2`, modifiez l'épaisseur du segment,
6. dans la fonction `image_test_2`, ajoutez d'autres segments, par exemple pour tracer un triangle.

• Partie B

Pour cette partie, modifier la fonction `main` du programme `transformee_hough.cpp` en fixant la variable `partieB` à `true` afin de pouvoir calculer les maxima locaux de la transformée de Hough `TH`, de les afficher, et de calculer et afficher les droites correspondantes dans l'image initiale.

Exercice 3 :

Testez ce programme sur différentes images-contour tests fournies (`contour-triangle.png`, `contour-cercle.png`, `contour-octogone.png`, `contour-rectangles.png`) : pour chaque image, adaptez éventuellement les paramètres `seuil` et `d` lors de l'appel de la fonction `sequence_maxima_locaux` afin d'obtenir un résultat satisfaisant.

3 - Traitement complet

Exercice 4 :

A partir des programmes `filtre_contour_gris.cpp` et `transformee_hough.cpp`, écrivez un programme qui effectue la traitement complet d'une image :

1. détection des contours par filtrage,
2. détection des droites contours par transformée de Hough.

Testez votre programme sur différentes images.