

## TP n°2

### Opérations morphologiques sur des images

L'objectif de ce TP est de continuer à manipuler et traiter une image à partir d'une librairie de traitement des images en langage C. Les TP se dérouleront sous LINUX avec un terminal, un éditeur de texte et un logiciel de tracé de courbes GNUPLOT.

#### 1) Seuillage d'une image et érosion de l'image binaire

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

- Choisir une image (validée par l'enseignant) issue d'un appareil photographique (pas une image de synthèse) et l'enregistrer au format pgm. Réduire la taille de l'image (soit 256x256 pixels, soit 128x128 pixels).
- Seuiller cette image en testant plusieurs valeurs de seuils. Modifier le programme `test_grey.cpp` afin que le **fond** de l'image (pixels < seuil) soit en **blanc** (255) et que les pixels des **objets** soient en **noir** (0). Comparer l'image seuillée avec au moins 3 valeurs différentes et en déduire le seuil le plus pertinent.
- A partir de l'image seuillée la plus intéressante, écrire le programme `erosion.cpp`, qui va permettre de supprimer les points objets isolés. Compiler ce programme et l'exécuter en utilisant l'image binaire obtenue précédemment avec le seuil le plus intéressant. Pour l'exécution, 3 arguments sont nécessaires, à savoir, le nom du programme, le nom de l'image d'entrée et le nom de l'image de sortie.

#### 2) Seuillage d'une image et dilatation de l'image binaire

- A partir de l'image seuillée la plus intéressante obtenue à la question 1.b, écrire le programme `dilatation.cpp`, qui va permettre de boucher des petits trous isolés dans les objets de l'image. Compiler ce programme et l'exécuter en utilisant l'image binaire obtenue précédemment avec le seuil le plus intéressant. Pour l'exécution, 3 arguments sont nécessaires, à savoir, le nom du programme, le nom de l'image d'entrée et le nom de l'image de sortie.

#### 3) Fermeture et ouverture d'une image de l'image binaire

La fermeture d'une image binaire consiste à enchaîner une dilatation et une érosion sur l'image binaire. Cela permet de boucher des trous dans les objets contenus dans l'image binaire.

L'ouverture d'une image binaire consiste à enchaîner une érosion et une dilatation sur l'image binaire. Cela permet de supprimer des points parasites du fond de l'image binaire.

- A partir de l'image seuillée la plus intéressante obtenue à la question 1.a, écrire le programme `fermeture.cpp`, qui va permettre de boucher des petits trous isolés dans les objets de l'image. Compiler ce programme et l'exécuter en utilisant l'image binaire obtenue précédemment avec le seuil le plus intéressant. Pour l'exécution, 3 arguments sont nécessaires, à savoir, le nom du programme, le nom de l'image d'entrée et le nom de l'image de sortie.

- b) A partir de l'image seuillée la plus intéressante obtenue à la question 1.b, écrire le programme `ouverture.cpp`, qui va permettre de supprimer des points parasites du fond de l'image binaire. Compiler ce programme et l'exécuter en utilisant l'image binaire obtenue précédemment avec le seuil le plus intéressant. Pour l'exécution, 3 arguments sont nécessaires, à savoir, le nom du programme, le nom de l'image d'entrée et le nom de l'image de sortie.
- c) Enchaîner la fermeture et l'ouverture sur la même image, c-à-d que l'image de sortie du premier programme sera utilisée comme image d'entrée du second programme. Que constatez-vous ?
- d) Afin d'avoir plus d'impact, nous vous proposons d'amplifier les effets fermeture et ouverture sur l'image binaire. L'idée est d'appliquer séquentiellement à l'image binaire, 3 dilations, 6 érosions et enfin 3 dilations. Que constatez-vous ?

#### 4) Segmentation d'une image

Le but de cette question est de développer une approche permettant de visualiser les contours d'une image. A partir de l'image seuillée la plus intéressante obtenue à la question 1.b et de l'image dilatée obtenue à la question 2, écrire un programme `difference.cpp` qui va nous permettre de visualiser les contours des objets contenus dans l'image :

Si les deux pixels (de l'image seuillée et de l'image dilatée) appartiennent au fond  
alors le pixel correspondant de l'image de sortie appartiendra au fond (255).

Si les deux pixels (de l'image seuillée et de l'image dilatée) appartiennent à l'objet  
alors le pixel correspondant de l'image de sortie appartiendra au fond (255).

Sinon le pixel correspondant de l'image de sortie appartiendra au contour (0).

- a) Ecrire le programme `difference.cpp`, compiler ce programme et l'exécuter en utilisant l'image binaire et l'image dilatées obtenues précédemment avec le seuil le plus intéressant. Pour l'exécution, 4 arguments sont nécessaires, à savoir, le nom du programme, les noms des deux images d'entrée et le nom de l'image de sortie.

#### 5) Question bonus : Extension aux images en niveaux de gris, puis en couleur

Le but de cette question est d'étendre les questions précédentes (érosion, dilatation, fermeture et ouverture) dans un premier temps aux images en niveau de gris (c-a-d sans seuiller l'image), puis aux images couleur.