

TP n°3

Erosion et dilatation d'une image

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. Il est fortement conseillé de créer un répertoire TP_image_3 et de tout sauvegarder dans le même répertoire.

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

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

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

- a) Prendre l'image au format pgm indiquée par l'enseignant, 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.
- b) 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) A partir de l'image seuillée la plus intéressante obtenue à la question 1.a, 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) 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.a, 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.
 - a. Dans une première approche, reprendre les programmes `erosion.cpp` et `dilatation.cpp` et les appliquer 6 fois dans l'ordre proposé.
 - b. Ecrire un nouveau programme unique qui enchainera les 3 dilations, les 6 érosions et enfin les 3 dilations.
 - c. Comparer les 2 images obtenues. 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.a 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).

Si le pixel de l'image dilatée appartient à l'objet et que le pixel de l'image seuillée appartient au fond, alors le pixel correspondant de l'image de sortie appartiendra au contour (0).

- a) écrire 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.