

TP2 - Opérations morphologiques sur des images

Stéphane Wouters - 20110928

Seuillage d'une image, érosion et dilatation

Seuillage

Résultat du seuillage au niveau 90, réalisé dans le TP1 :

```
./seuil lena.pgm seul.pgm 90
```



lena.pgm



seuil.pgm

Algorithme pour l'érosion et la dilatation

Au départ l'algorithme suivant était utilisé

- "Ne garder que les points qui ont 4 voisins noirs"

Finalement, l'algorithme suivant a été utilisé tout le long du TP :

- "Colorer tout les points en blanc sur l'image de sortie"
- "Si un point est noir, transformer tout ses voisins et lui même en noir sur l'image de sortie"

Le résultat semble le meme avec les deux algorithmes, bien que je n'ai pas analysé la différence avec précision.

Les points noirs isolés subissent ainsi cette transformation :



Résult érosion

```
./eroder seuil.pgm eroder.pgm
```



seuil.pgm



eroder.pgm

Résultat dilatation

```
./dilater seuil.pgm dilater.pgm
```



seuil.pgm



dilater.pgm

Fermeture et ouverture de l'image binaire

Fermeture (dilatation puis érosion)

```
./dilater seuil.pgm ouverture.pgm
```



seuil.pgm



ouverture.pgm

Fermeture (dilatation puis érosion)

```
./dilater seuil.pgm fermeture.pgm
```



seuil.pgm



fermeture.pgm

Fermeture puis ouverture

```
./fermeture seuil.pgm fermeture.pgm  
./ouverture fermeture.pgm fermeture-ouverture.pgm
```



seuil.pgm



fermeture-ouverture.pgm

On observe un effet de “lisage” sur l’image. Les points isolés ont été effacés, et les “trous” ont été bouchés.

Application

Suite d’instructions :

```
./dilater seuil.pgm s1.pgm  
./dilater s1.pgm s2.pgm  
./dilater s2.pgm s3.pgm
```

```
./eroder s3.pgm s4.pgm  
./eroder s4.pgm s5.pgm  
./eroder s5.pgm s6.pgm  
./eroder s6.pgm s7.pgm  
./eroder s7.pgm s8.pgm  
./eroder s8.pgm s9.pgm
```

```
./dilater s9.pgm s10.pgm  
./dilater s10.pgm s11.pgm  
./dilater s11.pgm resultat.pgm
```



resultat.pgm

Peu à peu, les contours s'amplifient et on perd en détail.

A moins que je n'ai pas compris la question, réaliser les transformation en appliquant les executables les uns après les autres ou executer les fonctions dans le même programme n'implique aucune différence sur les deux résultats, ce qui me semble parfaitement logique.

Segmentation d'une image

En parant de l'image seuillée

```
./difference seuil.pgm dilat.pgm diff1.pgm
```



seuil.pgm



diff1.pgm

En partant de l'image générée précédemment avec les 12 étapes

```
./dilater s12.pgm s12-dilat.pgm  
./difference s12-dilat.pgm diff2.pgm
```



s12.pgm



diff2.pgm

Adaptation aux images sans seuil

Pour cette partie, on utilise un niveau de seuil arbitraire. Tout les pixels au dessus de ce seuil sont copié aux voisins (ou en dessous pour l'érosion).



lena.pgm



fermeture-ouverture.pgm

Adaptation aux images couleurs

Même fonctionnemet, mais on applique l'algorithme sur chaque canal.



lena.ppm



fermeture-ouverture.ppm