

ENSTA - 2e année / ESE42
Morphologie mathématique
Année 2017-2018
TP N°1

Préliminaires

Le TP sera réalisé à partir d'un logiciel dédié à l'apprentissage de la morphologie mathématique, qui permet d'expérimenter des opérateurs simples au niveau de l'interface graphique, écrite en Tcl/Tk, ainsi que de programmer des nouveaux opérateurs, en C++, en s'appuyant sur des méthodes déjà écrites, en particulier pour ce qui concerne les lectures et écritures d'images, et la gestion des bords (voir commentaires dans le code).

La première phase consiste donc à récupérer et installer l'archive suivante, qui contient également des images d'exemple :

http://perso.ensta-paristech.fr/~manzaner/Cours/Ese42/tps_ese42.tgz

Installez l'archive, recompilez l'interpréteur et lancez l'exécution du script en suivant les instructions fournies dans le fichier *ReadMe.txt*.

1 Opérateurs de base et composés

EXPÉRIMENTEZ : A partir des opérations disponibles sur l'interface *Inti*, calculez les opérations morphologiques suivantes :

- Erosion
- Dilatation
- Gradient interne
- Gradient externe
- Gradient morphologique (symétrisé)
- Laplacien morphologique
- Ouverture
- Fermeture
- Top-hat
- Top-hat conjugué

Quel est l'effet du changement de la condition de bord dans l'interface sur les opérations morphologiques ?

Comment vérifier expérimentalement la relation de dualité des opérateurs (e.g. Erosion/Dilatation, Ouverture/Fermeture) ?

2 Un détecteur de contours rudimentaire

Selon le modèle classique de Marr et Hildreth, les *contours* d'une image correspondent aux lieux de passage par zéro du *laplacien* Λ . On calcule cet ensemble à partir des 2 ensembles suivants :

$$\Lambda^+ = \{(x, y); \Lambda(x, y) > 0\}, \text{ et} \\ \Lambda^- = \{(x, y); \Lambda(x, y) < 0\}.$$

Ces 2 ensembles sont évidemment disjoints, mais on peut calculer grossièrement les lieux de passages par zéro par exemple de la façon suivante :

$$\Lambda^0 = \delta_B(\Lambda^+) \cap \Lambda^-,$$

où δ_B désigne la dilatation morphologique par une boule élémentaire (i.e. définissant la topologie).

Comme en général, ce détecteur fournit une réponse trop dense pour être exploitable, on le combine avec un seuillage *du module du gradient* G , soit :

$$G_s = \{(x, y); G(x, y) \geq s\}.$$

L'ensemble des *contours* C est finalement défini par :

$$C = G_s \cap \Lambda^0$$

APPLIQUEZ ce détecteur de contours directement sur l'interface *Inti* en prenant pour Λ et pour G respectivement le laplacien et le gradient morphologiques, et en testant sur 2 ou 3 images en niveaux de gris.

Quels sont les paramètres de votre détecteur ? Quel est l'impact de ces paramètres sur le résultat ?

Quels sont selon vous les limites à la pertinence et à l'exploitation de ce détecteur ?

3 Transformée en Tout-ou-rien.

Etant donné une image binaire $I \subset \mathbb{Z}^2$, un couple (H, M) d'éléments structurants $H, M \subset \mathbb{Z}^2$ tels que $H \cap M = \emptyset$, on appelle *transformée en tout-ou-rien* (TTR) de I par le couple (H, M) , l'image :

$$I \otimes (H, M) = \varepsilon_H(I) \cap \varepsilon_M(I^c) = (I \ominus \check{H}) \cap (I^c \ominus \check{M}).$$

EXPÉRIMENTEZ la recherche de configurations sur images binaires par TTR sur l'interface *Inti* (Sur le tableau de définition de l'élément structurant, le bouton de gauche définit H (en blanc), le bouton de droite définit M (en noir)).

Appliquez les TTR au débruitage de l'image binaire *texte_bruit.gif*, commentez les résultats.

4 Opérateurs connexes

EXPÉRIMENTEZ la reconstruction géodésique, pour une image binaire, puis pour une image en niveau de gris. Comment peut-on "calculer" une reconstruction sur l'interface *Inti* sans utiliser le bouton "reconstruction" ? Comment cet algorithme est-il programmé efficacement ?

APPLIQUEZ la reconstruction géodésique à l'image binaire *particules.gif* pour éliminer les particules de "diamètre" inférieur à 20 pixels sans modifier les autres. Quelle est l'opération que vous avez utilisée ?

CALCULEZ l'ouverture et la fermeture par reconstruction de l'image *tree_noise.gif*. Présentez et commentez les résultats.

EXPÉRIMENTEZ l'opération d'érosion ultime sur une image binaire. Quel rapport existe-il entre cette opération et la reconstruction géodésique ? Appliquez l'érosion ultime à la singularisation de particules dans l'image binaire *coffee.gif*.

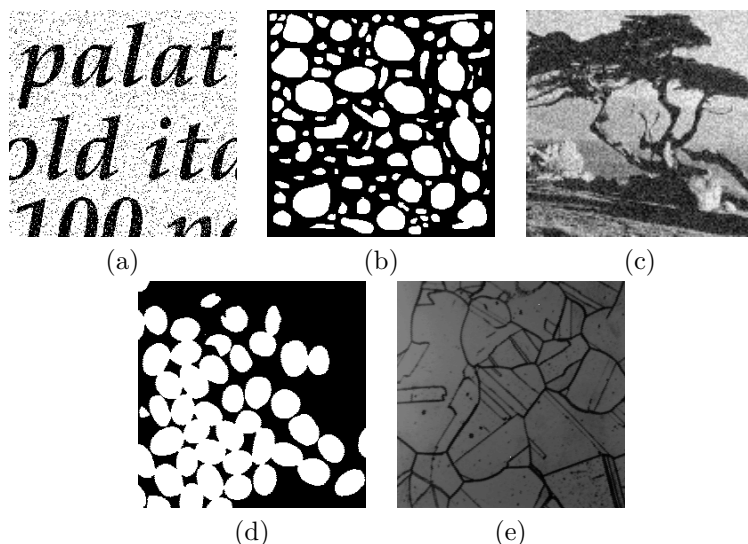


FIGURE 1 – Les images d'exemple : (a) *texte_bruit.gif*, (b) *particules.gif*, (c) *tree_noise.gif*, (d) *coffee.gif*, (e) *laiton.gif*

5 Lignes de Partage des Eaux

Expérimentez l'opérateur de lignes de partage des eaux (LPE - Menu "niveau de gris"), sur l'image *laiton.gif*. Comment interpréter cet opérateur ? Sur quels types d'images l'applique-t-on en général ?

Observez et interpréter le phénomène de sur-segmentation sur une image naturelle.

Appliquez une fermeture par reconstruction sur l'image de gradient avant la LPE. Quel est l'effet de cet opérateur sur le résultat de la LPE ?

Appliquez un filtrage de dynamique sur l'image originale avant le calcul du gradient. Appliquez le gradient, puis la LPE. Quel est l'effet de l'opérateur de filtrage de dynamique ?

Combinez enfin les 2 opérateurs avant le calcul de la LPE et interprétez le résultat.

Attention : la ligne de partage des eaux est affichée en jaune sur l'interface, mais sera interprétée en niveau de gris si on applique un opérateur *après la LPE*, ce qui n'aura donc pas de sens en général.

6 Programmation : Opérateur de Contraste

Soit ξ et ψ deux opérateurs morphologiques, le premier étant extensif, et le deuxième anti-extensif. L'opérateur de réhaussement de contraste χ est défini

comme suit :

$$\chi(x) = \arg \min_{\omega \in \{\xi, \psi\}} \{|\omega(x) - x|\}(x)$$

PROGRAMMEZ l'opérateur de contraste générique :

```
void Contraste(Image<int>& p, int type);
```

où *type* définit la nature des opérateurs ξ et ψ (par exemple *type* = 0 correspond au couple (érosion, dilatation), *type* = 1 à (ouverture, fermeture) et *type* = 2 à (ouverture, fermeture) par reconstruction. L'élément structurant est défini par l'utilisateur sur l'interface.

7 Programmation : Nivellement

Le Nivellement peut être vu comme une généralisation de la reconstruction, utilisée dans le cas où il n'y a pas de relation d'ordre entre le marqueur et l'image de référence.

PROGRAMMEZ en utilisant la procédure de reconstruction géodésique disponible dans le fichier *morfo.cc* de l'interface *Inti*, la procédure :

```
void Nivellement(Image<int>& p, Image<int>& p_ref, int cx);
```

qui calcule le nivellement de l'image marqueur *p* dans l'image de référence *p_ref*, en utilisant la connexité *cx* (*cx* = 4 ou 8).

EXPÉRIMENTEZ les nivellements pour calculer les filtres alternés séquentiels par reconstruction d'une image en niveaux de gris sur plusieurs niveaux. Observez l'évolution de profils 1d (bouton de droite sur les images) pour les f.a.s et les f.a.s par reconstruction. Commentez les résultats.

Les filtres alternés séquentiels peuvent être calculés "à la main" depuis l'interface *Inti*, ou, mieux, en programmant votre propre procédure :

```
void FAS(Image<int>& p, int premier, int rayon);
```

qui calcule le filtrage alterné séquentiel de l'image *p* par le filtre :

$$\Theta_{rayon} = \phi_{rayon} \gamma_{rayon} \dots \phi_1 \gamma_1$$

(ouverture d'abord) si *premier* = 0, et

$$\Xi_{rayon} = \gamma_{rayon} \phi_{rayon} \dots \gamma_1 \phi_1$$

(fermeture d'abord) si *premier* = 1.