Traitement d'images
TP4 : filtres non-linéaires

Benoît Naegel, Julien Haristoy

1 Filtre médian

1.1 Définition

On considère une image en niveaux de gris I et une « fenêtre » M de taille $(N \times N)$, où N est un entier impair. Le résultat de l'application d'un filtre médian de taille N appliqué à l'image I est obtenu de la manière suivante :

Pour tous les points p de l'image I :

1. on considère la liste des valeurs des points contenus dans la fenêtre M centrée en p ;
2. on trie cette liste de valeur et on en détermine une *médiane* (valeur en positon centrale de la liste triée)
3. le point p de l'image résultante prend cette valeur médiane.

1.2 Travail à réaliser

Implanter l'opérateur de filtre médian et écrire un programme de test qui prend en paramètre la taille du filtre.

2 Morphologie mathématique

2.1 Rappel de la définition des opérateurs

On se place dans le cadre de la morphologie des images en niveaux de gris, et on considère un *élément structurant plat* B (B est une fonction qui vaut $-\infty$ en dehors d'un ensemble borné, où B vaut 0).
L'*érosion* d'une image I par B vaut alors, au point (x, y),

$$\varepsilon_B(I)(x, y) = \min\{I(x + i, y + j) \mid (i, j) \in B\},$$

tandis que la *dilatation* d'une image I par B vaut, au point (x, y),

$$\delta_B(I)(x, y) = \max\{I(x - i, y - j) \mid (i, j) \in B\}.$$

Comme d'habitude, on note ω_B et ϕ_B, respectivement, les opérateurs d'*ouverture* et de *fermeture* par l'élément structurant B.
Le *gradient morphologique* est défini au point (x, y) par

$$\gamma(I)(x, y) = \delta_D(I)(x, y) - \varepsilon_D(I)(x, y),$$

où D est le disque unité fermé (une définition explicite, dans le cas discret, en est donnéedans la section suivante.
La transformée en tout-ou-rien en niveaux de gris par l'élément structurant composite (A, B) (voir le support de cours) est définie par

$$HMT_{(A,B)}(I)(x, y) = \max\left(0, \varepsilon_A(I)(x, y) - \delta_B(I)(x, y)\right).$$

2.2 Travail à réaliser

1. Créer une classe `Structel` définissant la notion d'élément structurant. La classe permettra de stocker l'ensemble des points de l'élément structurant relativement à l'origine $(0, 0)$.
   Exemple : considérons l'élément structurant B_1 défini par le disque fermé de rayon 1 (4-voisinage de l'origine) :

   $$\begin{array}{ccc} & \bullet & \\ \bullet & \circ & \bullet \\ & \bullet & \end{array} \quad \text{ou encore, en terme de coordonnées :} \quad \begin{array}{ccc} & (0,-1) & \\ (-1,0) & (0,0) & (1,0) \\ & (0,1) & \end{array}$$

   (Les petits disques marquent les points appartenant à l'élément structurant, y compris, ici, l'origine, marqué par $\circ$.)
   L'ensemble B_1 sera codé par la liste des coordonnées de ses points : $\{(0, 0), (0, 1), (1, 0), (-1, 0), (0, -1)\}$.

2. Dans la classe `Structel`, implanter une méthode statique permettant, pour tout entier n, de retourner un élément structurant carré de côté $2n + 1$.
3. Dans la classe `Structel`, implanter une méthode statique permettant de créer un élément structurant défini par la boule euclidienne (disque) de rayon r : $B_r = \{(i, j) \mid i^2 + j^2 \leq r^2\}$
4. Implanter les fonctions d'érosion et de dilatation, permettant respectivement d'éroder et de dilater une image par un élément structurant.
5. De la même manière, implanter l'ouverture et la fermeture.
6. Implanter les gradients morphologique, interne et externe.
7. Implanter la transformée en tout-ou-rien par un élément structurant composite (A, B).

2.3 Programmes de test

Pour chaque opérateur, écrire un programme de test.
Pour la transformée en tout-ou-rien, tester l'élément structurant composite formé de

$$A = \begin{array}{ccc} \bullet & \bullet & \bullet \\ \bullet & \circ & \bullet \\ \bullet & \bullet & \bullet \end{array} \quad \text{et} \quad B = \begin{array}{ccccc} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & & & & \bullet \\ \bullet & & \times & & \bullet \\ \bullet & & & & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{array}$$

Notez bien que, pour B, l'origine (point central, marqué $\times$) n'est pas contenue dans la liste des points de B, au contraire de A.

3 Applications

Les images à traiter se trouvent dans le répertoire `TP4/images`.

3.1 Filtrage morphologique

Dans l'image `pcb_gray.pgm` on souhaite conserver au mieux les petits disques blancs tout en supprimant les autres structures de l'image.

3.2 Restauration

Dans l'image `barrat4.pgm` on souhaite éliminer les petites rayures noires en altérant le moins possible le reste de l'image.

3.3 Fissures

Dans l'image `soil.pgm` on souhaite mettre en valeur les petites fissures du sol. Dans l'image résultat, les fissures seront claires sur fond noir et auront les mêmes contours que dans l'image originale.

3.4 Granulométrie

Pour l'image `barrat2.pgm`, on souhaite obtenir la courbe granulométrique et le *pattern spectrum* pour des éléments structurants convexes du type « disques de rayons croissants ».

1. Écrivez un programme permettant d'effectuer une succession d'ouvertures avec un élément structurant de type disque de taille entière variant de 1 à 10, et calculant après chaque ouverture le *volume de l'image* (défini comme la somme des valeurs des points de l'image).
2. On souhaite afficher la courbe granulométrique (taille de l'ouverture en abscisse, volume de l'image en ordonnée) de l'image ainsi que son pattern-spectrum (dérivée discrète de la courbe granulométrique). En déduire la taille de la majorité des particules de l'image.
3. Vous effectuerez la granulométrie d'abord sur une image seuillée, puis sur l'image originale en niveaux de gris. Quelles différences constatez-vous ? Quelle méthode semble la plus robuste ?