

Morphologie mathématique

Matériel

Récupérez l'archive `td-mm.tgz` elle contient

- l'énoncé du TD ;
- l'entête `mm.h` avec la définition des fonctions du module à réaliser ;
- le programme `mm_label.c` ;
- les images de test `lena.ppm` et `gear.ppm`.

Rappels

Soit B un élément structurant, f une image en niveaux de gris et p un pixel. Nous pouvons définir les opérations morphologiques suivantes :

- l'érosion : $\varepsilon_B(f)(p) = \min_{x \in B} f(p + x)$
- la dilatation : $\delta_B(f)(p) = \max_{x \in B} f(p + x)$
- l'ouverture : $\gamma(f)(p) = \delta(\varepsilon(f)(p))(p)$
- la fermeture : $\phi(f)(p) = \varepsilon(\delta(f)(p))(p)$
- le gradient morphologique : $|\nabla f| = \delta(f) - \varepsilon(f)$
- le laplacien morphologique : $\Delta f = \delta(f) + \varepsilon(f) - 2f$

1 Éléments structurants

La représentation des éléments structurants (`se`) se fera grâce à la structure de données `pnm` de la bibliothèque `bcl`. Les éléments structurants considérés dans ce TD sont :

- symétrique, l'origine se trouvant au centre de l'élément structurant,
- binaire c'est-à-dire que la forme de l'élément a pour valeur 1 ou 255 et son fond (s'il existe) à pour valeur 0,
- de taille impaire : ainsi si la demi-taille de l'élément structurant est n alors l'élément structurant sera de taille $(2n + 1) \times (2n + 1)$.

Implantation

Planter dans un module en C (`mm.c`) la fonction `pnm mm_se(int shape, int halfsize)` qui à partir d'un numéro de forme et d'une demi-taille retourne un objet `pnm` représentant l'élément structurant. Le paramètre `shape` $\in \{0, 1, 2\}$ permet de générer des éléments structurants de type carré, losange et disque. La figure 1 montre des exemples d'éléments structurant utilisant cette fonction. Vérifier que votre fonction fonctionne bien en affichant ce type d'image.

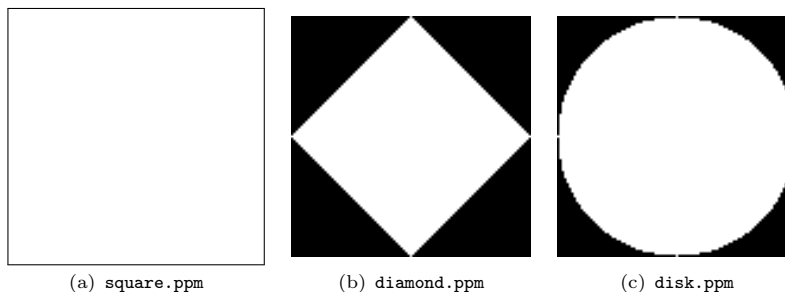


FIGURE 1 – Exemples d'images d'éléments structurants de demi-taille égale à 50 pixels

2 Opérateurs morphologiques

1. Planter les fonctions `mm_process`, `mm_greater` et `mm_lesser` dans votre module `mm`.

2. Implanter les opérateurs morphologiques suivants dont les formulations ont été rappelées. Chaque opérateur est à implanter dans un fichier différent.

- `mm_dil <se> <hs> <ims> <imd>` qui réalise une dilatation
- `mm_ero <se> <hs> <ims> <imd>` qui réalise une érosion
- `mm_open <se> <hs> <ims> <imd>` qui réalise une ouverture
- `mm_close <se> <hs> <ims> <imd>` qui réalise une fermeture
- `mm_subtract ima imb imd (imd=ima-imb)`
- `mm_add ima imb imd (imd=ima+imb)`
- Tester le gradient et le Laplacien morphologique.

`<se>`, `<hs>`, `<n>`, `<ims>` et `<imd>` sont respectivement un numéro d'élément structurant, une demi-taille, un nombre d'itération, une image source et une image destination.

Si vous utilisez l'image `lena.ppm`, vous devez obtenir des résultats semblables à ceux de la figure 2.

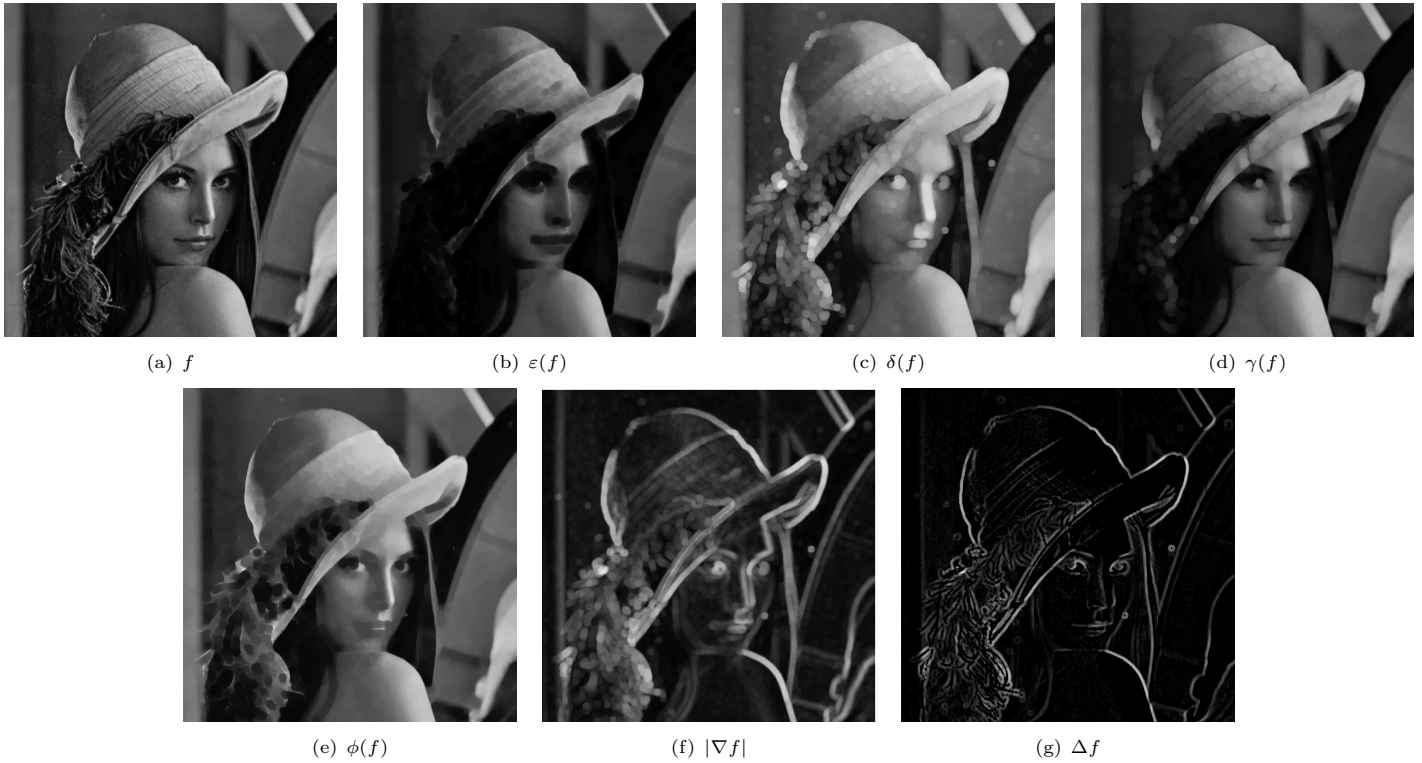


FIGURE 2 – Exemples d'opérations morphologiques avec un élément structurant de type disque et de demi-taille 5

3 Traitements morphologiques

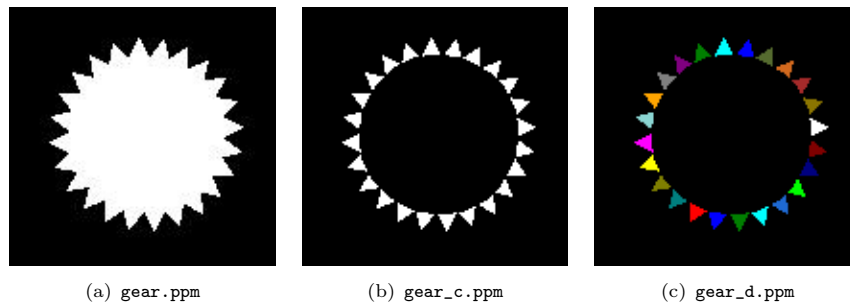


FIGURE 3 – Extractions d'objet par morphologie mathématique

La figure 3 montre l'extraction et l'isolement des dents pointues dans l'image. Cette extraction a été réalisée grâce à une succession d'opérations.

- à partir des opérateurs que vous avez implantés écrire le script shell **extract.sh** qui permet d'obtenir cette extraction et les différentes étapes du traitement.
- **aide 0** : afin de nettoyer l'image initiale des artefacts de compression, une binarisation de celle-ci a été faite. Une fonction de binarisation dans un intervalle $[\min, \max]$ sur une image f en un pixel x est :

$$f(x) = \begin{cases} 255 & \text{si } f(x) \in [\min, \max] \\ 0 & \text{sinon} \end{cases} \quad (1)$$

1. Planter le programme `mm_binarize <min> <max> <ims> <imd>`.
 2. Trouver le bon intervalle à appliquer pour obtenir l'image (b).
- **aide 1** : L'extraction des objets dans la dernière image a été réalisée par le programme `mm_label <ims>` fourni dans l'archive.
 1. Étudier le programme `mm_label` afin d'en comprendre le fonctionnement.
 2. À l'heure actuelle, le programme `mm_label` ne fait pas la distinction entre fond (pixels à 0) et forme. La conséquence est que le fond est considérée comme une composante connexe. Proposer un correctif du programme.
 3. Le programme `mm_label` ne fait que compter les composantes connexes. Proposer une modification du programme pour que celui-ci puisse afficher chaque composante avec une dans une couleur différente et dans une nouvelle image passé en paramètre (`mm_label <ims> <imd>`).

4 Traitements morphologiques couleurs

Proposer de nouvelles fonctions permettant l'extension de vos opérateurs morphologiques monochromes vers des opérateurs morphologiques couleurs en utilisant :

1. l'approche marginale : c'est-à-dire composante par composante
2. l'approche lexicographique. Avec la relation d'ordre $<$, nous avons :

$$(r, g, b) < (r', g', b') \begin{cases} \text{Vrai si} & r < r' \\ \text{Sinon vrai si} & r = r' \text{ et } g < g' \\ \text{Sinon vrai si} & r = r' \text{ et } g = g' \text{ et } b < b' \end{cases} \quad (2)$$

3. Comparer les différents résultats obtenus par ces trois approches différentes.