

TP de morphologie mathématique

Hugues Talbot

31 janvier 2014

1 Règles et principes

Ce TP est noté. Les règles usuelles (triche, plagiarisme, etc) s'appliquent. À la fin du TP vous devez me rendre un script python *commenté* représentant votre travail. Aucun autre rapport n'est demandé.

2 Prise en main du système PINK-PYTHON

Dans cette section nous allons nous familiariser avec le système PINK-PYTHON sous LINUX.

PINK-PYTHON est un logiciel d'analyse d'images en sources ouvertes (open-source) produit par une équipe de chercheurs du département informatique de l'ESIEE.

2.1 Architecture du système

Ce système est basé sur Python, et fait appel à un module dédié incluant la morphologie mathématique, la visualisation et d'autres opérateurs par exemple de géométrie et de topologie discrète.

2.1.1 Prérequis

Avant de pouvoir commencer, il faut suivre les étapes suivantes :

1. Se logger sous Linux sur une des lames de calcul : `blade-vm11.esiee.fr` ou `blade-vm12.esiee.fr`.
2. Choisir `blade-vm11` ou `blade-vm12`, les deux machines sont virtuelles et quasi-identiques.
3. Logez vous sur ces lames avec votre login / mot de passe ESIEE avec la commande suivante :

```
ssh -X votrellogin@blade-vm12.esiee.fr
```

4. Une fois logé, lancez un terminal, par exemple `konsole`.
5. Dans le terminal, lancer pink par la commande

```
pink.sh
```

6. Pointer le browser web vers le site suivant :
`http://www.pinkhq.com/`
7. Lancez un éditeur, par exemple `gedit` ou `kate`. Toutes les commandes que vous allez taper devront être sauvegardées dans un fichier dont vous vous servirez pour faire le rapport.
8. Au fur et à mesure que vous progresserez dans le TP, vos solutions devront apparaître comme des fonctions de Python.

2.1.2 Images

Les images que nous traiterons sont toujours au format standard PGM. Les images pour ce TP sont toutes disponibles dans le répertoire source suivant :

```
/user/talboth/Public/ISBS/TP/Images/tp4/
```

vous aurez besoin des images suivantes :

- numbers.pgm
- cell.pgm
- muscle.pgm

2.2 Commandes Unix/Linux et PINK

Pour continuer avec ce TP, il vous faut un minimum de connaissance des commandes UNIX. Par exemple, créez un répertoire pour ce TP dans votre répertoire principal `mkdir tp4morpho`. Copiez l'image `cells.pgm` du répertoire source vers le répertoire que vous venez de créer (commande `cp <source> <destination>`). Puis changez de répertoire vers celui que vous venez de créer (`cd tp4morpho`).

2.2.1 Documentation UNIX et PINK

Le système UNIX dispose d'un très grand nombre de commandes, qui vont bien au delà de ce TP. En général vous pouvez utiliser l'aide en ligne (cliquer sur le bouton avec une bouée comme icône). À partir de la console vous pouvez invoquer l'aide en ligne par la commande `man`, par exemple :

```
man mkdir
man cp
```

Vous pouvez aussi utiliser le navigateur de KDE ou de GNOME qui sont bien adaptés à une utilisation interactive, similaire à celle de Windows.

Notez que les commandes Unix/Linux restent disponible dans l'environnement PINK.

2.2.2 Commandes pink

Le système PINK utilise l'interpréteur python qui doit vous être familier, mais on ne peut pas se contenter de démarrer l'interpréteur python, c'est pourquoi nous lançons le script "pink.sh" pour bénéficier d'un environnement plus adapté.

Le module de PINK s'appelle simplement 'pink'. Pour démarrer une session pink, importez le module de la façon suivante (attention au majuscules !) :

Pink-python : importation des éléments du module

```
# la partie principale du module
from pink import cpp as pinklib
# importer certaines fonctions de conversion
import pink
# la partie affichage
from pink import Imview
```

Les commandes de PINK ont toutes la structure suivante (le % représente l'invite) :

```
resultat = pinklib.une_commande(donnees)
```

Par exemple lisons une image :

```
image= pinklib.readimage("numbers.pgm")
```

L'aide en ligne PINK est disponible sur le site <http://pinkhq.com>. Prenez un moment pour vous y familiariser.

Il existe aussi une aide interactive, par exemple essayez :

```
help(pink.readimage]
```

vous devriez voir :

Help on built-in function readimage:

```
readimage(...)
  readimage( (str)filename) -> object :
    reads an image from a 'pgm' file and returns an object with the appropriate type

  C++ signature :
    boost::python::api::object readimage(std::string)

(END)
```

2.2.3 Interactivité

Le programme `pink.sh` est interactif avec historique des commandes tapées (avec les flèches de direction hautes et basses), et raccourcis par la touche “TAB” (à gauche de la lettre A).

Par exemple, essayez :

```
I=pinklib.read<TAB>
```

où <TAB> correspond simplement à presser la touche TAB.

2.3 Visualisation

La visualisation des images s’opère par la commandes `Imview`, avec la syntaxe suivante :

Pink-python : lecture et affichage d’une image

```
image= pinklib.readimage("cells.pgm")
# ceci cr'ee un port de visualisation et visualise 'image'
viewer=Imview(image)
# ensuite on peut r'utiliser ce port, par exemple:
resultat=pink.monFiltre(image)
viewer.show(resultat , "resultat")
```

Dans le cas où on réutilise un même port plusieurs fois, il est possible de passer d’une image à l’autre par les touche d’espace et effacement. Le zoom avant et arrière s’effectue avec les touches “>” et “<” respectivement.

2.4 Exemple d’interaction aide-programmation

Dans le cours de ce TP vous aurez besoin des fonctions suivantes :

Morphologie

- `pinklib.genball`
- `pinklib.erosion`
- `pinklib.dilation`
- `pinklib.opening`
- `pinklib.closing`
- `pinklib.watershed` (plus exactement `pink.cpp.watershedMeyer2NM`)
- `pinklib.heightmaxima`

Arithmétique

- `pinklib.inverse`
- `pinklib.min`
- `pinklib.seuil`

Divers

- `pinklib.surimp`
- `pinklib.frame`
- `pinklib.point`

Exercice 2.1 (La fonction `erosion`)

1. Trouvez la documentation de la fonction `pinklib.erosion`
2. Trouver comment fonctionne la fonction `pinklib.erosion`. Expliquer les paramètres.
3. Trouvez la documentation de la fonction `pinklib.genball`. A quoi sert cette fonction ? A quoi sert l'argument `dimension` ? que veut dire le fait qu'il soit par défaut égal à 2 ?
4. Réaliser l'érosion de l'image `cell.pgm` par un disque de rayon 2.
5. Visualisez le résultat et comparez avec l'image de départ.
6. Trouvez la documentation de la fonction `pinklib.erosball`. Réalisez l'opération précédente avec cette fonction.

2.5 Fichier programme

Par la suite, il est nécessaire de créer un fichier exécutable contenant toutes les suites de commandes qui sont solution des exercices. Le format est le suivant :

- Vous devez l'appeler `nom1_nom2.py` où `nom1` et `nom2` sont les noms des composantes de votre binôme.
- Chaque ligne (sauf la première) commençant par un caractère `#` est un commentaire. N'hésitez pas à commenter vos fichiers.
- Chaque ligne non-commentée est une commande exécutable
- A chaque fois que vous faites un changement dans ce fichier, vous pouvez le recharger et l'exécuter par la commande
`%run nom1_nom2.py`

N'oubliez pas que vous devez rendre un tel fichier avec votre rapport.

3 Segmentation de nombres

L'image de la figure 1 est une image en niveaux de gris d'une suite de nombres :

Exercice 3.1 (Segmentation des nombres)

Proposez une procédure pour :

1. Corriger les variations du fond de l'image ;
2. Segmenter les nombres par seuillage.

Mettez de côté l'image finale et attachez-la au courrier électronique à la fin du TP.



FIGURE 1 – Image initiale `numbers.pgm`.

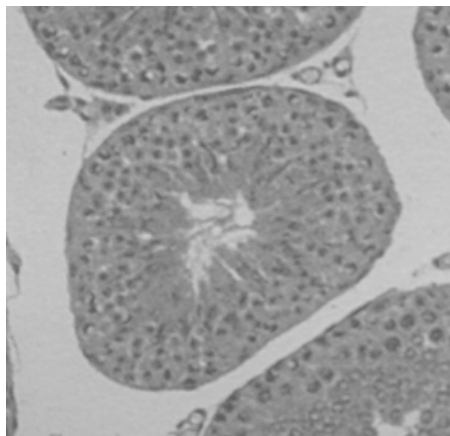


FIGURE 2 – Image initiale `cell.pgm`.

4 Segmentation d'une seule cellule

L'image de la figure 2 est une image en niveaux de gris d'un petit nombre de cellules :

Exercice 4.1 (Segmentation de la cellule centrale)

Proposez une procédure pour :

1. *Obtenir un gradient de l'image de cellules*
2. *Obtenir des marqueurs pour l'intérieur des cellules*
3. *Obtenir un marqueur de l'extérieur des cellules*
4. *Obtenir le contour des cellules en utilisant la ligne de partage des eaux watershed*
5. *Obtenir seulement la cellule centrale.*

Mettez de côté l'image finale et attachez la au courrier électronique à la fin du TP.

5 Segmentation de cellules de muscles

L'image `muscle.pgm` est une image en niveaux de gris de cellules musculaires.

Exercice 5.1 (Segmentation des cellules de muscle)

1. *Segmentez les cellule de muscles.*

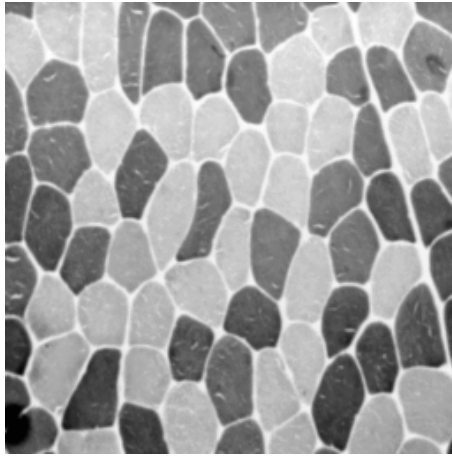


FIGURE 3 – Image initiale `muscle.pgm`.

2. *Eliminez ceux qui touchent le bord de l'image*
3. *Mesurez l'aire moyenne d'une cellule.*

Mettez de côté l'image finale et attachez la au courrier électronique à la fin du TP.

6 Vaisseaux de fond d'oeil

L'image `fondeoil.pgm` est une image de fundus en imagerie par fluorescence.

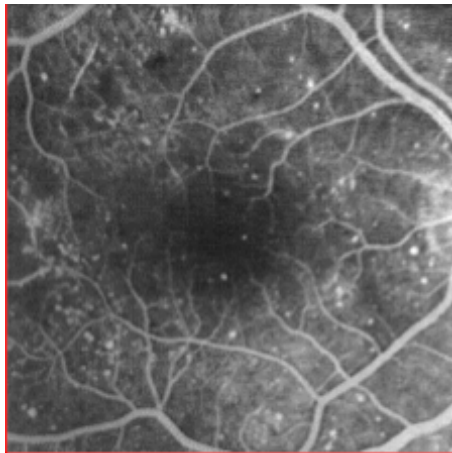


FIGURE 4 – Image initiale `numbers.pgm`.

Exercice 6.1 (Segmentation des anevrismes)

1. *Corrigez les variations du fond*
2. *Segmentez les vaisseaux*
3. *Segmentez les petits anevrismes*