TP Imagerie 3D - 5 (3 heures)

Emmanuel Faure

12 décembre 2017

Ce TP a pour objectif de vous faire programmer un algorithme de suivi cellulaire à partir d'images 4D. Il peut être effectué seul ou en binôme.

- ***Le TP est noté : le compte-rendu doit être envoyé*** *sous* ***forme électronique à :*** ***gerard.subsol@lirmm.fr*** *avant* ***le mercredi 20 décembre 2017 (minuit)***
- *Tout compte-rendu envoyé sous une mauvaise forme ou hors-délai sera sanctionné par un 0.*
- *Le compte-rendu doit inclure* ***la date, vos noms*** *et être composé d'au moins 1 à 2 pages de texte* ***décrivant la méthode*** *utilisée pour répondre aux questions ainsi* ***que la source de votre programme ou la liste des fonctions macros utilisées AVEC quelques captures d'écran*** *pour évaluer le résultat.*
- *Le tout doit être sous la forme d'un* ***unique fichier pdf****.*
- *Le TP peut se faire seul ou en binôme*
- *La participation active pendant le TP pourra aussi être prise en compte.*
- ***Tout plagiat sera lourdement sanctionné****.*

- **Master 2 Informatique**: obligatoirement C/C++ avec la bibliothèque CImg (http://cimg.sourceforge.net/)
- **Autres Masters** : au choix soit en C/C++ avec la bibliothèque CImg, soit en utilisant des macros du logiciel Fiji (http://fiji.sc/).

1. Ouvrir et visualiser la série temporelle d'images 3D (séries d'images 3D stack-i.hdr/img en 2 fichiers zip, le premier de faible taille avec 3 images 3D pour tester, le second avec les 17 instants suivants)

2. Pour chaque pas de temps, détecter les cellules présentes dans l'image 3D:
   - 2.1 - Utiliser un filtre (par exemple médian) pour éliminer le bruit dans les données.
   - 2.2 - Utiliser un seuil pour éliminer le fond et de ne garder que les noyaux cellulaires.
   - 2.3 - Eliminer les quelques pixels isolés par morphologie mathématique.
   - 2.4 - Identifier les cellules par composantes connexes
   - 2.5 - Calculer le barycentre de chacune des cellules et le sauvegarder dans un fichier.

3. Implémenter un algorithme de suivi de cellules. Pour cela, on pourra rechercher le plus proche voisin unique entre 2 pas de temps. Le principe est de trouver une correspondance unique entre deux listes de points 3D en calculant leur distance minimale. *Remarque* : si le nombre de cellules est différent entre les 2 pas de temps alors toutes les cellules ne pourront pas trouver de correspondance.

4. (optionnel) Eventuellement, visualiser le résultat sous forme de trajectoires 3D. On pourra créer un fichier .obj du type :

```
v 0.000000 2.000000 0.000000    /* Point 1
v 0.000000 0.000000 0.000000    /* Point 2
v 2.000000 0.000000 0.000000    /* Point 3
v 2.000000 2.000000 0.000000    /* Point 4
l 1 2 3 4                       /* Ligne reliant les points 1, 2, 3 et 4
```

qu'il est possible de visualiser avec Paraview (http://www.paraview.org/) par exemple.

Annexe CImg

Voir http://cimg.eu/reference/structcimg__library_1_1CImg.html

- get_blur_median
- threshold
- erode(3, 3, 3) pour une érosion de 1 voxel
- dilate(3, 3, 3) pour une dilatation de 1 voxel
- get_label
- load_analyze

Annexe Fiji

Voir http://rsb.info.nih.gov/ij/developer/macro/macros.html et http://rsb.info.nih.gov/ij/developer/macro/functions.html pour les macros classiques.

- File.open/File.close : pour ouvrir un fichier texte

Voir http://imagejdocu.tudor.lu/doku.php?id=plugin:analysis:3d_object_counter:start pour « 3D objects counter » (accessible par Analyze/3D objects Counter et 3D OC Options).

This plugin:

- counts the number of 3D objects in a stack.
- quantifies for each found object the following parameters:
  - Integrated density;
  - Mean of the gray values;
  - Standard deviation of the gray values;
  - Minimum gray value;
  - Maximum gray value;
  - Median of the gray values;
  - Mean distance from the geometrical centre of the object to surface;
  - Standard deviation of the distance to surface;
  - Median distance to surface;
  - **Centroid;**
  - Centre of mass;
  - Bounding box.
- generates results representations such as:
  - Objects' map;
  - Surface voxels' map;
  - **Centroids' map;**
  - Centres of masses' map.

On doit **d'abord paramétrer** l'algorithme par run("3D OC Options…"

On peut **alors appliquer la méthode** par run("3D Objects Counter", …" pour trouver les objets et afficher les coordonnées des centroïdes dans la fenêtre Results (par défaut, l'affichage se fait dans une fenêtre Results sauf si l'option Redirect to a été activée).

Pour lire un fichier texte contenant par exemple une liste de coordonnées de points 3D :

```
filestring=File.openAsString("….");        /* Ouverture et lecture d'un fichier texte
line=split(filestring, "\n");              /* Séparation des lignes
x=newArray(line.length);                   /* Création des tableaux des coordonnées x, y, z
y=newArray(line.length);
z=newArray(line.length);
for(i=0; i<line.length; i++)
        {
        columns=split(line[i]," ");        /* Séparation des colonnes par un espace
        x[i]=parseFloat(columns[1]);       /* Récupération d'un flottant dans la colonne 1 (la 2ème)
```

```
y[i]=parseFloat(columns[2]);
z[i]=parseFloat(columns[3]);
print(x[i]+" "+y[i]+" "+z[i]);
}
```

Pour afficher des lignes 3D, on peut éventuellement tester
http://imagejdocu.tudor.lu/doku.php?id=plugin:stacks:3d_tools:start