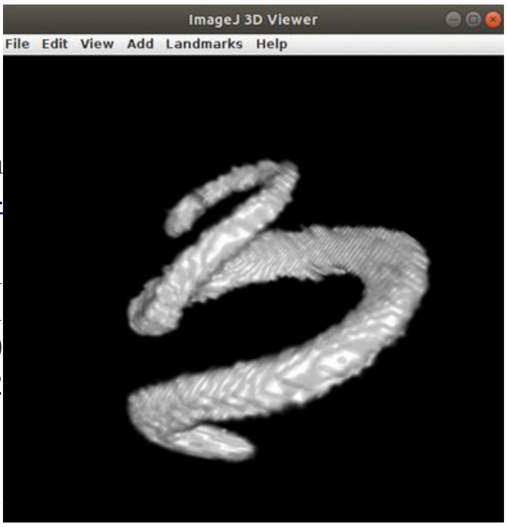
Methods for 3D Analysis in Bio-medical Images (an ImageJ/Fiji based tutorial)

A few links : http://gibbs.engr.ccny.cuny.edu/technical/Tracking/RoachTrack.php for MATLAB fans *or* http://fiji.sc/Fiji or http://icy.bioimageanalysis.org/ or http://www.ipol.im/ . **I chose Fiji.**

Resources: http://www.math-info.univ-paris5.fr/~lomn/Cours/CV/BME including the lecture (file ***TPVideo.pdf*** and software fiji-linux64.Core.tar.gz if needed otherwise install from http://fiji.sc/Fiji and ***Data***)

Start by visualizing a 3D biological sample with the 3D viewer. Use the sample *Bat Cochlea Volume* image. ***Menu Plugins/3D Viewer.***



Then we will work with the **Fluo-C3DL-MDA231** dataset (from the ISBI 2015 challenge http://celltrackingchallenge.net/3d-datasets/).
They are MDA231 human breast carcinoma cells infected with a pMSCV vector including the GFP sequence, embedded in a collagen matrix (3D) - Microscope: Olympus FluoView F1000 Objective lens: Plan 20x/0.7 Voxel size (microns): 1.242 x 1.242 x 6 Time step (min): 80).

They are organized as 30 z stacks over 11 time steps sequence. The ***.tif*** are multi-image formats (the z stacks). Take your time to explore and understand the dataset files.

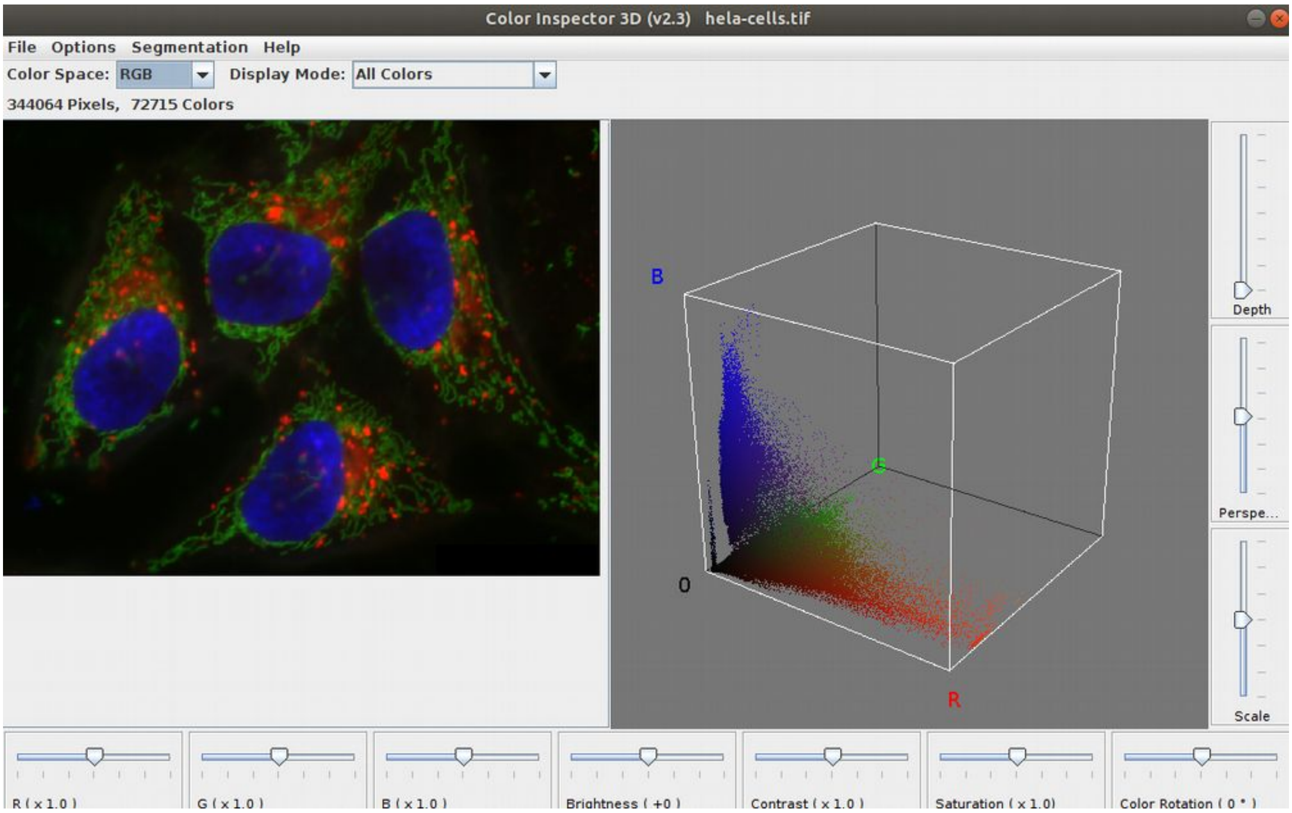
Start by visualizing the 5th frame in 3D of the *01_GT/TRA* set (GT for Ground Truth and TRA for Tracked, in the ***01_GT_TRA.tar.gz*** which is an excerpt of the whole data set or out of the **Fluo-C3DL-MDA231.zip** file see again http://celltrackingchallenge.net/annotations/ for more information about annotations and in particular https://public.celltrackingchallenge.net/documents/Naming%20and%20file%20content%20conventions.pdf).

The z scale is less resolved that the focal plane (xy) scale.
Use the ***Volume Viewer plugin*** and change the z scale by a factor of 5. Keep on exploring that plugin.

From the ***man_track005.tif stack z sequence,*** use the ***Analyze/3D Object Counter*** tool. *(settings: find a good threshold manually on the slice number 24). Then play with the min size in voxels parameter. Take your time to explore and analyze the results.*

Use the ***Image/Scale*** tool to set the resolution for the volume calculation. Set x=y scale to 1 and z scale to 5. What happens ? Can you save the resulting stack? Can you estimate the true volume with the metric resolution above of the first cell tracked over the z axis.

On the previous original image ***t005.tif*** (30 z stacks), threshold the stack with ***Image/Adjust/Threshold*** and find an optimal threshold ? Remove shape noise with morphological filters.



Color Inspector 3D (v2.3) hela-cells.tif

Color Space: RGB

Display Mode: All Colors

344064 Pixels, 72715 Colors

Open the **HeLa Cells** sample. Where do the HeLa cells come from ? Play with the ***Plugins/Color Inspector 3D*** tool. This sample uses the ***Image/Color/Colour Deconvolution*** tool (when you ask to show information). For instance in the RGB fluorescent space, put the blue channel at full power. Display the 3D histogram of color. Do the same operation in the deconvolved space (Press 'I'). (see http://www.mecourse.com/landinig/software/cdeconv/cdeconv.html)-

Just for the Engineers audience (optional)

How to create 3D document for reporting. As before using ***Menu Plugins/3D Viewer*** and the ***man_track005.tif*** z stack, visualize the 3D scene as a red surface. Export it as *test.u3d* file. Save the sample text as *test.tex* file (see latex editor for scientific communications). Then use the unix command *$pdflatex* to generate a *test.pdf* file with a 3D embedded visualization of the scene (command unix *$acroread test.pdf*).

But tricky depending of the OS and the pdf viewer version as well. http://vaastavanand.com/blog/2017/Generating-3D-PDFs-with-U3D/

But if you succeed, can be fun for your project or internship report. Windows OS easier : https://gitlab.com/agrahn/media9

If you get time, explore the tutorials at http://3dviewer.neurofly.de/ with the landmark (registration) example using the *Screencasts* documentation.

We switch to Python (no more Fiji/ImageJ) in case you want to start programming something in an easy way. Explore the Python example 3D.py in the Repertory **Python** just by running it (check if cv2 is installed ou opencv library for python). This code create a 3D numpy array and feed another one with real data from a z-stack the one corresponding to the cochlea volume. The data are in the folder ***VolumeData.zip*** to unzip in your current working directory.

```
$python 3D.py
```

Spend time understanding the way it works. Then try to answer the questions within the code.

For the curious ones, https://bic-berkeley.github.io/psych-214-fall-2016/preparation.html you can explore another module in python nibabel (to read/write neuroimaging file formats in Python).

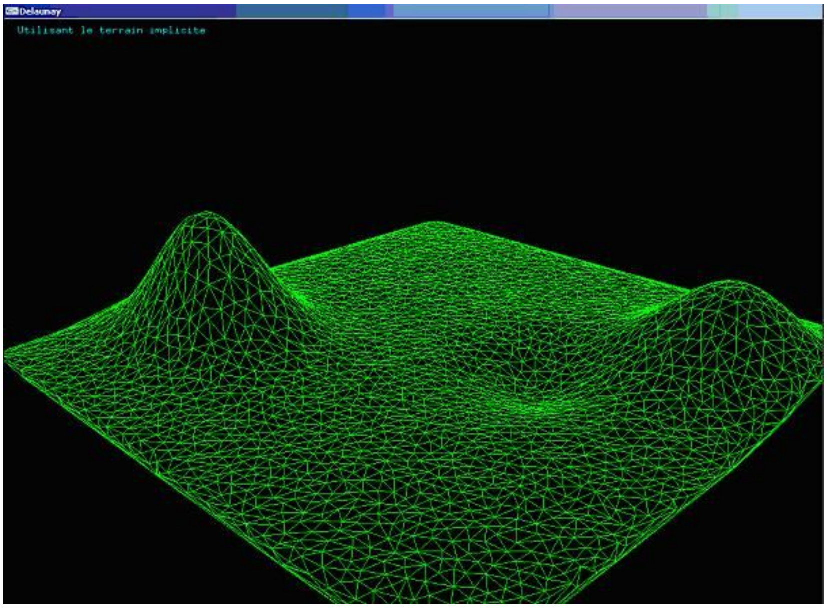
Matlab variant if you prefer:

Some 2D to 3D implementations are quite straightforward (except for the file format). Try to implement in Matlab the erosion process on a 3D matrix. You will use the ***bat-cochlea-volume.tif*** 3D stacks of 2D images (either from my website or by saving it in tiff format from ImageJ/Fiji). Then how to import it in matlab : a few hints with mmreader https://fr.mathworks.com/matlabcentral/answers/295849-how-do-i-stack-several-tiff-files-into-a-3d-stack-in-mat-format or with videoreader (as 3D is alike 2D+t) for the algorithmic data structure :
https://fr.mathworks.com/matlabcentral/answers/380269-applying-threshold-to-video-pixels

---

Some 3D implementations are harder to transpose from 2D to 3D due to topological issues for instance. If you are interested apply the Delaunay triangulation in 2D to the pattern provided in annex by using the algorithm described here : https://fr.mathworks.com/matlabcentral/answers/380269-applying-threshold-to-video-pixels

It is the core algorihtm to build 3D mesh of surface like here :
https://nanolive.ch/products/3d-microscopes/cx/ or
https://github.com/domlysz/BlenderGIS/wiki/Make-terrain-mesh-with-Delaunay-triangulation
But the 3D version is a bit more tricky :
http://openalea.gforge.inria.fr/doc/vplants/tissue/doc/_build/html/user/reconstruction/delaunay3D/index.html



```
Utilisant le terrain implicite
```