

Image Segmentation and image analysis - Practical work 1

ImageJ

ImageJ is a free software written in Java for processing and analysis of 2D and 3D images. It provides the user with many functionalities that can be extended through macros and plugins (see the website <http://rsbweb.nih.gov/ij/>).

To run it, use the script `run`. It simply contains the following line where `<PLUGINDIR>` is replaced by the path to the plugins directory and `<IMAGEDIR>` by the path to the java archive of ImageJ :

```
java -Dplugins.dir=<PLUGINDIR> -cp <IMAGEJDIR>/ij.jar ij.ImageJ
```

How to use ImageJ for 2D images processing and analysis

Some test images are supplied via the menu `[File/Open Samples]`. The documentation can be found from the item *Menu Commands* of the *Documentation* part of the web site.

1. Test the histogram computation `[Analyze/Histogram]`, the different thresholding algorithms `[Image/Adjust/Threshold]`, the edge detection `[Process/Find edges]`, the contrast setting.
2. Open the sample **Diatoms** (supplied on the web via `[File/Import/URL]` at URL <http://rsbweb.nih.gov/ij/images/Diatoms.jpg>). How to obtain the main contours of the objects in black on a white background?
3. Open the sample **Embryos**. We aim to measure and count the big circular cells. Convert the image to gray scale, apply a thresholding algorithm and reduce the noise. Select a well circular cell with the Magic Wand tool. Measure it with `[Analyze/Measure]` to obtain area, perimeter and circularity values.
Select other cells, for example globally round cells with irregular contours, to obtain their area and circularity. Deduce the values of `[Analyze/Analyze particles]` which allow to capture round cells of similar size whether their contour is circular or not.
4. Count the number of white particles in `TEM_Filter`.

How to use ImageJ for the visualization and processing of 3D images

In ImageJ a 3D image is represented as a stack of 2D images.

1. Open the sample **MRI Stack**. Browse the image stack with the scroll bar. Open the volume viewer `[Plugins/3D/Volume Viewer]`. Change the z-aspect to 4. Visualize the data volume. Look for the eyes, the white matter, the gray matter.
2. Open the image `flybrain.zip` (available via `[File/Import/URL]`). Visualize the data volume with the plugin `[Plugins/3D/ 3D Viewer]`. The plugin documentation is available at <http://3dviewer.neurofly.de/>. Test the geometric transforms (rotation, translation, zoom).
3. The file `sphere.tif` contains a spherical gray level gradation centered in the middle of the data volume. Open the image with `3D Viewer` and visualize an isosurface of the data. Change the threshold defining the surface.
4. Use in the same way `3D Viewer` to visualize the surface of the segmented region stored in the image `bat-cochlea-volume.zip`. Look for existing plugins of ImageJ that can be used to obtain the geometric information of a 3D region.
5. Is it possible to apply the different image processing filters on image stacks? Are they computed slice by slice or on the full 3D volume?

How to write an ImageJ plugin

The API documentation is available at :

<http://rsbweb.nih.gov/ij/developer/api/>.

1. Look at a first example of ImageJ plugin : `Binarize_2D_Image.java`. It simply binarizes a gray level image according to a threshold which is given by the user. Try it and understand its source code. Remark that the name of the plugin contains an underscore to make it appear in the menu.

A comprehensive tutorial is available at :

<http://imagingbook.files.wordpress.com/2013/06/tutorial171.pdf>

2. Replace the manual setting of the threshold by an automatic computation with the Otsu's method.
3. Same question in 3D (the initial plugin is `Binarize_3D_Image.java`).

An object `ImagePlus` contains meta-data (title,...) and an instance of the class `ImageProcessor` (2D image) or an instance of the class `ImageStack` (image 3D). In the case of a 3D image, the method `getProcessor` returns an `ImageProcessor` to access the pixels of the current slice. The current slice is set via the method `setSlice` of the class `ImagePlus`.