Biomedical imaging

# Image moments

*Max Langer, Loriane Weber, Marion Dohen*
*{mlanger,lweber}@esrf.fr, marion.dohen@gipsa-lab.grenoble-inp.fr*

Creatis, INSA-Lyon
European Synchrotron Radiation Facility
GIPSA-Lab, Grenoble-INP

December 2014

The aim of this lab is to study image moments by implementing analysis of centre of gravity (CoG), orientation and surface area of objects. You could imagine this as the image analysis step in an imaging chain, after image acquisition and pre-processing, and before a subsequent classification or decision making step. An example algorithm is given below (the parts marked *implement* have to be implemented by you even though implementations exist in matlab), but you are free to use another method or improve the one given below, as long as shape/area/orientation/CoG are calculated using moments.

A phase-contrast visible light micrograph of stained cells is given on the course page (cells_orig.jpg). Your algorithm should segment this image and extract surface area (in pixels), CoG, orientation and eccentricity of each cell. The result of your algorithm should be the input image marked up with the CoG (`scatter`), minor and major axis of each cell (`quiver`), and the area (text). You should also display the separate cells in different colours. Further, calculate the orientation and eccentricity histograms, and mean and standard deviation of area and eccentricity.

A short lab report should be submitted in electronic form as a PDF file. Hand in your work as a zip file containing your report, your code in one running .m file (script) and any supporting functions, along with the images you have used (reference your images with relative paths in your code).

## Algorithm

1. Convert the image to grayscale (`rgb2gray`)
2. Segment the image
3. Perform opening to separate touching objects and remove small objects (`imopen`)
4. Perform connected component analysis to identify each cell (`bwconncomp`)
5. Calculate the CoG of each cell using moments (*implement*)
6. Calculate the minor and major axes of each cell using the moment matrix (*implement*)
7. Calculate the eccentricity of each cell using the moment matrix (*implement*)