

## TEL411 – Digital Image Processing

Doutsis Effrosyni, PhD

### Assignment 5

Due date: Sunday, November 8, 2020

In this exercise, we are interested in a problem commonly encountered in image processing; the **segmentation**. The goal of segmentation is to extract an object(s) (here a cell) from an image in order to be able to study i.e. the shape, volume, number, orientation, etc. To do this, we will use some default MATLAB functions. We propose to detect the cell which is illustrated in the following Figure.

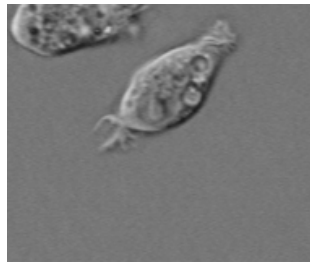


Fig. 1 – MATLAB Image *cell.tif*.

To read and display your image you should use the following commands:

```
I = imread('cell.tif');  
imshow(I, 'displayrange', []);
```

In order to detect the cell, we can, for example, follow the following "pipeline" by using operators from mathematical morphology (MATLAB functions `imopen()`, `imclose()`, `imdilate()`, `imerode()`, `imtophat()`):

1. Contour detection: calculation of the morphological gradient followed by thresholding from a threshold determined by the Otsu method (`graythresh()` and `im2bw()` functions).
2. Contour connection: the first contour extraction step usually contains holes in the detected contour which must therefore be filled. A dilation of the previous binarization with a well-chosen structuring element (s) allows you to do this.
3. Filling of holes: this step consists of filling the holes which are present inside the objects (`imfill()` function). This can also be done from a sequential filter alternating openings and closings.
4. Removal of small connected components and smoothing of the contour: typically an erosion with a well-chosen structuring element.

Figure 2 shows the steps described above. Note that in the last image (on the right), only the cell that is contained entirely in the image is segmented. To achieve this, you can use the `imclearborder()` function to delete the cell that does not interest us.



Fig. 2 – Example of the main steps you are asked to implement.

### **Hint**

Don't hesitate to use MATLAB's help function in order to better understand how these functions work.

**What to turn in**

You should turn in both your code and a report. Justify your choices and taking care to briefly detail how the functions you will use work. Make sure you illustrate the results as in Fig. 2 and explain each of them.