Politechnika Łódzka
Instytut Elektroniki

Image Processing and Computer Graphics

Laboratory #3

M. Kociński, J. Blumenfeld

Medical Electronics Division
Institute of Electronics

Python Imaging Library (PIL)

The Python Imaging Library (PIL) provide us with additional 2D image processing algorithms. PIL consists of several modules. To become acquainted with its capabilities go throuout tutorial; additional exercises you can find here (subsections PIL 1, PIL 2, PIL 3).

Import functionality of PIL as Image.

In [2]:
```
import Image
```

```
Load image from file into variable im.
```

In [ ]:
```
im = Image.open('figures_for_student/FluorescentCells.jpg')
```

```
Print basic information about loaded image.
```

In [3]:
```
print im.format, im.size, im.mode
```

```
JPEG (512, 512) RGB
```

```
Image.show() method enables to display im variable in the default operating system viewer.
```

In [6]:
```
im.show()
```

Based on tutrial and PIL handbook do following operations:

- split image im and display its componentes (R,G,B). Pay attention to brightness level of each component
- create new image with changed component order (e.g. B,G,R)
- convert image form RGB to grayscale (L)
- convert Image objec to Numpy array
- rotate image

- blend image
- crop some Region Of Interest (ROI)

Perform various filtration types of the im variable. Apply filtration for various parameter values for each filtration. Note the results for values bigger than 1 and smaller than 1. Read additional information about image filtering and edge detection.

ImageEnhance Module

- Sharpness()
- Brightness()
- Contrast()
- Color()

ImageFilter Module

- MinFilter()
- MedianFilter()
- MaxFilter()
- BLUR
- CONTOUR
- DETAIL
- EDGE_ENHANCE
- FIND_EDGES
- SMOOTH
- SHARPEN
- Kernel() for laplace filter: kernel=(1,-2,1,-2,5,-2,1,-2,1)

Linear filtering

Compese your own averaging filter based on lecture. Use 3 x 3 mask. Propose two versions of filter:

- with the use of two for loops
- based on matrix operation

Multi-dimensional image processing (scipy.ndimage)

Load image "blood1.bmp". Perform image thresholding e.g. with thresh equal to 126. This is not optimal threshold value. Try to find better one. Theoretical background of thresholding you can find here.

Browse capabilities of morphological opearation of ndimage library. Theory on mathematical morphology is provided in lecture. On thresholded, binary image perform some operations:

- dilation
- erosion
- fill holes
- opening
- closing
- morphological gradient

Compose your own function that perform:

- binary dilation
- binary erosion
- binary opening
- binary closing
- morphological gradient