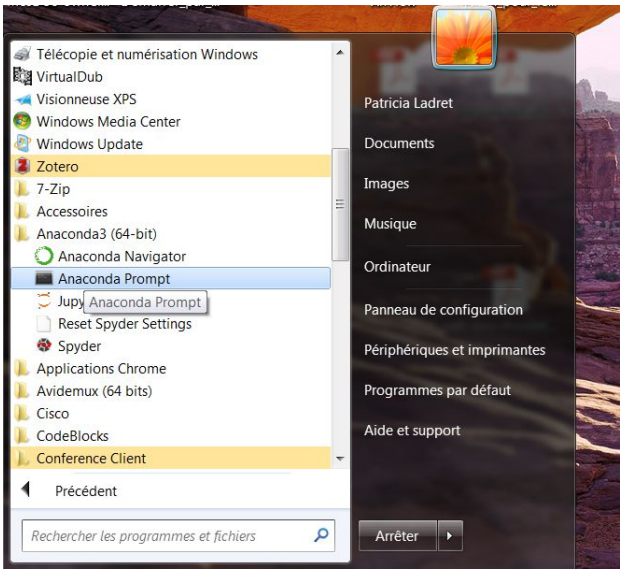
Image Processing: Sequence 1 exercises

0. Python package and Spyder editor installation

All the required softwares are available on any Phelma computer. If you want, you can install the required softwares on your own computer. All softwares are free.

0.1 Python install with Anaconda

- Go to the following site https://www.anaconda.com/distribution/ and choose the version compliant with your system.
- Anaconda for Python is installed with 150 packages. The main ones are used for Image and Signal processing: numpy, scipy, matplotlib, scikit-image, scikit-learn, etc...
- If you need to install other packages, launch an Anaconda command window,



The run the install by typing one of the following commands tat the Ananconda prompt:

- conda install [package_name]
- pip install [package_name] (or pip3 according to your system)

0.2 Mises à jours Anaconda et packages

For package updating:

- Global Anaconda package update
  - conda update anaconda (and before: conda update conda)
- Specific package update:
  - pip upgrade [package_name]
  - conda update [package_name]

0.3 Development Interface, IDE : Spyder

Once Anaconda is installed, access to development interface (IDE) Spyder is available.

Run the program by clicking on the icon: spyder

Before starting, look at the different available spyder windows

Rmks :

- ✓ Exercises marked as homework are supposed to be done at home before the corresponding follow-up session and exercises marked as classroom work are going to be done during the follow up session.

1. Image information and image display (homework)

Download the file Sequence1_part1_code.py. Each displayed image or curve must have a title.

- Select one **grey level** image among those available in the skimage.data module. Display the image and give its dimensions, its max and mean grey levels values.
- Select one **color** image among those available in the skimage.data module. Compute the image L defined as: L=(0.3*R+0.58*G+0.11*B). Display the color and the L image.
- Extract the luminance value of the line 200 of the L image and plot the luminance evolution along this line. Make the corresponding line appear on the image display figure so that it is possible to establish a correspondence between the chosen line and the displayed luminance profile.
- Download the *CH0SRC.tif* image. Display the image by using first skio.imshow() and then plt.imshow(). What do you notice when comparing both displays? Compute and display the histogram of the image in order to understand the difference between both displaying functions.

2. Image quantization (homework)

NOTA BENE: when an image is downloaded using *imread()* function, the luminance values related to the image are put in a uint8 matrix, that is a matrix with 8 bits unsigned integers. As a consequence, if this matrix has to be used in order to process the luminance values (for example by doing subtractions between pixel values) it is necessary to convert the matrix values into float values which can be done by using the from skimage.util import img_as_float function. To do the reverse conversion to obtain an image coded on 8 bits unsigned integers again, one might use the from skimage.util import img_as_ubyte function.

In order to evaluate the quantization's effect (that is the number of grey levels used in order to code the pixel luminance), the idea is to display an image using successively 256, 128, 64, 32, 16, 8, 4, 2 grey levels. To do that, download the file sequence1_pat2_code.py and use the proposed *quantize()* function.

- Apply it on the images *fruits.bmp*, *bacteria.tif*, *bonemarr.tif*, *blood1.tif*. For every image, at which grey level number do you notice a visual quality difference (please group all the results in a table)? According to the obtained results, do you think that visible information are lost when using 8 bits only to represent an image?

3. Image histogram (classroom work)

- In order to be familiar with the representation of an image with its histogram, compute, display and comment the histogram of the following images: *pout.tif*, *alumgrns.tif* and *moon.tif*.

- Build and display two different 8 bits coded images of size 256x256 pixels having the same histogram (do not forget to display the histogram of each image). What information is lost when an image is represented by its histogram?

4. Neighborhood impact (classroom work)

- Create 2 8 bits coded images of size 256x256 containing a 135 gray level square of size 56x56 at the center, the pixels of the background being 100 and 170 respectively. Display both images in the same figure. Comment the results.

5. Creating some specific images (classroom work)

- Create an image of size 256x256 with a decreasing of the 256 grey levels. Display the image and its histogram.
- Create an image of size 256x256 with black background and a white circle with radius 80 centered at pixel (128,128). Display the image
- By combining both images, create an image of size 256x256 with black background and a graded circle with radius 80 centered at pixel (128,128). Display the image