

TP2

September 21, 2018

1 TP 2 : Traitement d'image à l'aide de l'histogramme de niveau de gris

1.1 1 - Introduction

L'objectif de ce TP est de faire des rappels sur les librairies numpy et matplotlib. Nous réaliserons dans ce TP des fonctions basiques de traitement d'image.

Le TP sera à réaliser en python 3. Les librairies utilisées sont installées sur les machines de l'université, vous pouvez néanmoins les installer sur vos propres machines à l'aide de l'utilitaire pip présent par défaut avec python.

N'hésite pas à regarder régulièrement la documentation de ces librairies, des exemples d'utilisation accompagnent généralement l'explication de chaque fonction.

- Python 3: <https://docs.python.org/3/>
- Numpy: <https://docs.scipy.org/doc/numpy/reference/>
- Scipy: <https://docs.scipy.org/doc/scipy/reference/>
- Matplotlib: <https://matplotlib.org/contents.html>

À part si cela est précisé, vous ne devez pas utiliser directement de boucle (for,while) ou de branchement conditionnel (if) durant ce TP..

```
In [148]: import numpy as np
          import scipy as sc
          import scipy.misc
          import matplotlib.pyplot as plt
```

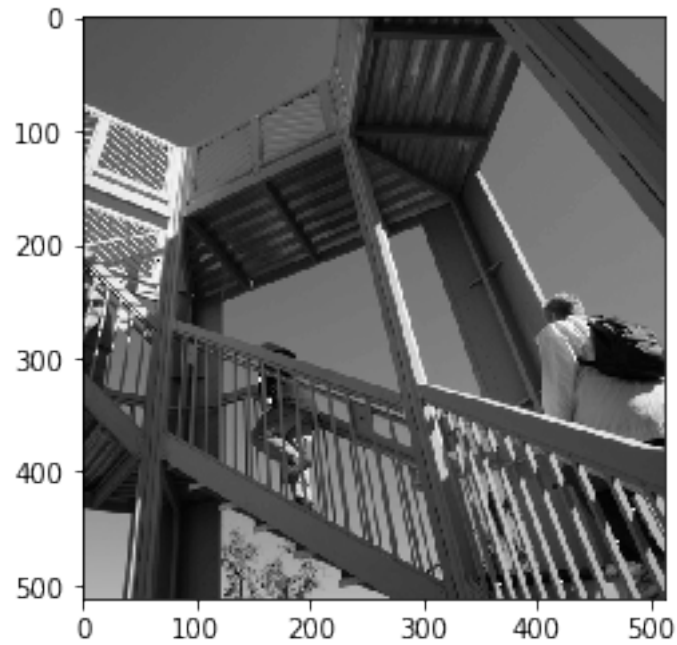
1.2 2- L'histogramme des niveaux de gris

En imagerie numérique, l'histogramme représente la distribution des intensités (ou des couleurs) de l'image. C'est une représentation graphique donnant pour chaque niveau de gris (ou couleur) le nombre de points de l'image ayant ce niveau de gris (couleur).

L'histogramme permet ainsi de visualiser la répartition des différents niveaux de gris de l'image. Pour chaque niveau de gris entre 0 (noir) jusqu'à 255 (blanc) en abscisse, vous avez le nombre de pixels de l'image ayant cette valeur en ordonnée.

Comme dans le TP 1, nous allons charger l'image *ascent* pour effectuer des traitements.

```
In [149]:
```



Nous allons maintenant afficher l'histogramme des niveaux de gris de cette image, grâce à la fonction suivante:

```
In [201]: def hist(im):  
            h,x = np.histogram(im,bins=255,range=(0,255))  
            plt.fill_between(x[:-1],0,h)  
            plt.axis((-1/10*(x[-1]-x[0]),x[-1]+1/10*(x[-1]-x[0]),0,np.max(h)))  
  
            hist(im)  
            plt.show()
```