

# Analyse d'images

Le traitement d'images permet de transformer des images. L'analyse d'images permet d'extraire des informations contenues dans une image. Il est aussi possible d'effectuer des tâches plus complexes de reconnaissance et d'analyse de scènes.

## Utilisation de la librairie PIL

La librairie Pillow devrait prendre la succession, avec des appels très semblables.

Voici un exemple de lecture d'une image, de conversion en monochrome et ensuite en tableau de données numériques de NumPy. Une transformée de Fourier bidimensionnel est effectuée, ainsi que quelques autres traitements, visualisations, ainsi qu'une sauvegarde :

```
<sxh python; title : FFT-structures_2D_01.py> #! /usr/bin/env python # -*- coding: utf-8 -*- """ Digital Image processing with PIL
```

```
""" # sources & refs :
```

```
http://stackoverflow.com/questions/2652415/fft-and-array-to-image-image-to-array-conversion #  
http://stackoverflow.com/questions/14577007/grayscale-image-to-numpy-array-for-fourier-transform  
# http://optipng.sourceforge.net/pngtech/img/lena.png #  
http://stackoverflow.com/questions/14452824/saving-image-in-python-pil/14472089#14472089
```

```
import PIL import Image import numpy as np import matplotlib.pyplot as plt
```

```
dirfile="" #filename="x10001-512.tif" filename="ech14x50001-512.jpg" im =  
Image.open(str(dirfile+filename)) im = im.convert('L') #convert to grayscale im.show() data =  
np.asarray(im) # a is readonly print type(data), data.ndim, data.shape, data.dtype datafft =  
abs(np.fft.rfft2(data)) datafft[0, 0] = 0 # remove DC component for visualization print type(datafft),  
datafft.ndim, datafft.shape, datafft.dtype plt.imshow(np.abs(np.fft.fftshift(datafft)),  
interpolation='nearest') plt.show() plt.imshow(np.abs(np.fft.fftshift(datafft))[192:320, 192:320],  
interpolation='nearest') plt.show() im2 = Image.fromarray(datafft) im2.show() im2.save('img2.gif')  
</sxh>
```

L'[image de départ](#) correspond à un film mince de polydiméthylsiloxane traité sous plasma 30 minutes, de manière à obtenir une contrainte interne de compression dans le film (15% dans une seule direction), ce qui provoque la déformation de l'interface et l'apparition de rides (UMONS, lab. Interfaces et fFluides Complexes).

## Références

- [PIL](#) : Python Imaging Library et la [documentation sur effbot](#)
  - [Tutoriel en français](#) sur des bases de traitement d'images avec PIL et numpy
- [Pillow](#), un fork pour le remplacement et l'évolution de PIL
- [scikits-image](#) : collection d'algorithmes en Python pour le traitement d'images

- [Galerie de Scikits-image](#)
- [imagescanner](#)
- [OpenCV python interface](#) (computer vision)
- [SimpleCV](#) : interface simplifié à OpenCV : “computer vision made easy”
- [pymaging](#) (pure python)

From:  
<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:  
[https://dvillers.umons.ac.be/wiki/teaching:progappchim:analyse\\_images?rev=1392644781](https://dvillers.umons.ac.be/wiki/teaching:progappchim:analyse_images?rev=1392644781)

Last update: **2014/02/17 14:46**

