MGM657 Outils Numériques pour l'Ingénieur Traitement d'Images

ludovic.charleux@univ-savoie.fr

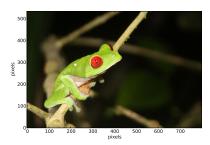
www.polytech.univ-savoie.fr

- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- 6 Érosion / Dilatation
- Comptage
- Contours



- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- Érosion / Dilatation
- Comptage
- Contours

Lire et afficher une image

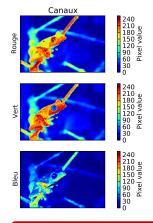


```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
im = Image.open(
'../data/grenouille.jpg')
fig = plt.figure(0)
plt.clf()
splt.imshow(im, origin = "lower")
plt.xlabel("pixels")
plt.ylabel("pixels")
plt.show()
```

Points clés

- PIL : utile pour lire et écrire divers formats d'images.
- Matplotlib : permet d'afficher les images

Canaux et couleurs



```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
im = Image.open('../data/grenouille.jpg')
rouge, vert, bleu = im.split()
rouge = np.array(rouge)
vert = np.array(vert)
bleu = np.array(bleu)
```

Types d'images

- Canal: 1 information (entier 8 bits)
- Image couleur : 3(+1) canaux
- Imagerie monochrome : 1 canal

Remarque

- Contexte scientifique : généralement un seul canal.
- On peut afficher une image monochrome avec une échelle de couleurs.



lmage = np.array

```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
im = Image.open('.../data/grenouille.jpg')
rouge, vert, bleu = im.split()
z = np.array(rouge)
```

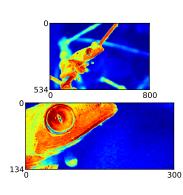
Sauvegarde



```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
im = Image.open('.../data/grenouille.jpg')
frouge, vert, bleu = im.split()
z = np.array(rouge)
z = np.uint8(cm.copper(z)*255)
jim2 = Image.fromarray(z)
im2.save("grenouille_saved.jpg")
```

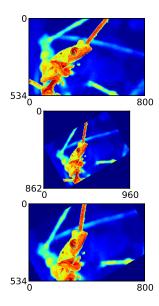
- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- Érosion / Dilatation
- Comptage
- Contours

Rognage (Crop)



```
from PIL import Image
  import numpy as np
 3 from matplotlib import pyplot as plt
 4 from matplotlib import cm
 5 im = Image.open('../data/grenouille.jpg')
 6 rouge, vert, bleu = im.split()
   z = np.array(rouge)
 8 \text{ ny}, \text{ nx} = \text{z.shape}
 9 | cx, cy = 200, 250
10 | zc = z[cx:-cx, cy:-cy]
11 nyc, n \times c = zc.shape
12
13 fig = plt.figure(0) # On cree une figure
14 plt.clf()
15 \mid a \times 1 = fig.add subplot(3,1,1)
16 plt.imshow(z, origin = "upper")
   plt.xticks([0, nx])
18 plt.yticks([0, ny])
19 \text{ ax2} = \text{ fig.add subplot}(3,1,2)
20 plt.imshow(zc, origin = "upper",
        interpolation = "nearest")
21 plt.xticks([0, nxc])
   plt.yticks([0, nyc])
23
   plt.show()
```

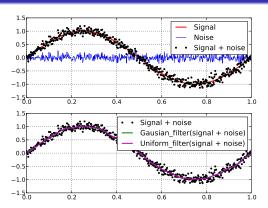
Rotation



```
1 from PIL import Image
2 import numpy as np
  from matplotlib import pyplot as plt
  from matplotlib import cm
  from scipy import ndimage
  im = Image.open('../data/grenouille.jpg')
7 rouge, vert, bleu = im.split()
  z = np.array(rouge)
   zrr = ndimage.rotate(z, 30.)
10 | zrn = ndimage.rotate(z, 30.,
11
     reshape = False)
12
13
  ny, nx = z.shape
  nyrr, nxrr = zrr.shape
15 nvrn. nxrn = zrn.shape
16
17
   fig = plt.figure(0) # On cree une figure
   plt.clf()
19 ax1 = fig . add _ subplot (3,1,1)
   plt.imshow(z, origin = "upper")
21 plt.xticks([0, nx])
22 plt.yticks([0, ny])
23 a \times 2 = fig.add subplot(3,1,2)
```

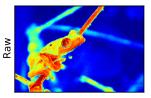
- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- Érosion / Dilatation
- Comptage
- Contours

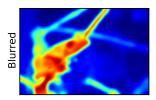
Lissage: exemple sur un signal 1D



```
import numpy as np
from matplotlib import pyplot as plt
from scipy import ndimage
x = np.linspace(0., 1., 500)
y_perf = np.sin(2. * np . pi * x)
noise = np.random.normal(loc = 0., scale = .1, size = len(x))
y = y_perf + noise
yg = ndimage.gaussian_filter(y, 10.)
ym = ndimage.uniform_filter(y, 20)
```

Lissage image

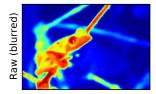


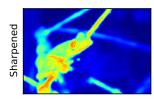


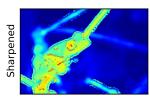
```
Blurred more !
```

```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from scipy import ndimage
im = Image.open('../data/grenouille.jpg')
rouge, vert, bleu = im.split()
z = np.array(rouge)
# Blur
gz10 = ndimage.gaussian_filter(z, 10.)
# Blur more!
11 zg30 = ndimage.gaussian filter(z, 30.)
```

Histogramme



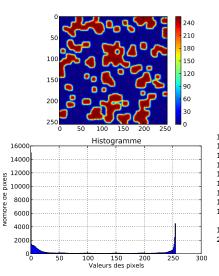




```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from scipy import ndimage
im = Image.open('../data/grenouille.jpg')
rouge, vert, bleu = im.split()
z = ndimage.gaussian_filter(np.array(rouge),
4)
# Sharpen
k = .5
zs1 = z + k * (z - ndimage.gaussian_filter(z,
1.))
zs2 = z + k * (z - ndimage.gaussian_filter(z,
2.))
```

- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- Érosion / Dilatation
- Comptage
- Contours

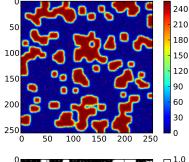
Histogramme



```
from PIL import Image
  import numpy as np
  from matplotlib import pyplot as plt
  im = Image.open('../Slides/figures/
        image.jpg')
   channels = im. split()
   z = np.array(channels[0])
   n classes = int(N**.5)
   fig = plt.figure()
   plt.clf()
   fig.add_subplot(2, 1, 1)
   plt.imshow(z, origin = "upper")
   plt.colorbar()
   fig.add subplot(2, 1, 2)
   plt.title('Histogramme')
16
   plt.ylabel('Nombre de pixels')
17
   plt.xlabel('Valeurs des pixels')
   plt.hist(z.flatten(), bins=n classes
        , histtype = "stepfilled")
19
   plt.grid()
   plt.show()
```

- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- A Histogramme
- Seuillage
- Érosion / Dilatation
- Comptage
- Contours

Seuillage



```
0.9
                                         0.8
                                         0.7
100
                                         0.6
                                         0.5
150
                                         0.4
                                         0.3
200
                                         0.2
                                         0.1
250
         50
               100
                     150
                           200
                                  250
```

```
1 from PIL import Image
2 import numpy as np
3 from matplotlib import pyplot as plt
  from matplotlib import cm
5 im = Image.open('../Slides/figures/
        image. ipg')
  channels = im.split()
   z = np.array(channels[0])
   seuil = 150
   zs = z > seuil
  fig = plt.figure()
11
   plt.clf()
12 fig.add subplot (2, 1, 1)
  plt.imshow(z, origin = "upper")
14
  plt.colorbar()
15 fig add subplot (2, 1, 2)
   plt.imshow(zs, origin = "upper", cmap =
16
        cm.gray)
17
   plt.colorbar()
18 plt.show()
```

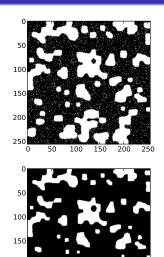
- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- 6 Érosion / Dilatation
- Comptage
- Contours

Érosion

200

250

50 100 150 200 250

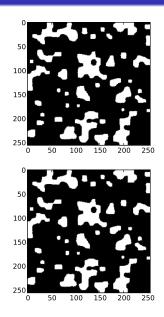


```
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1.0
0.9
8.0
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
```

1.0

```
0.9
8.0
    1 from PIL import Image
    2 import numpy as np
    3 from matplotlib import pyplot as plt
     from matplotlib import cm
    5 im = Image.open('../Slides/figures/
           image.jpg')
      channels = im.split()
      z = np.array(channels[0])
    8 seuil = 150.
      zs = z > seuil
      zss = ndimage.morphology.binary erosion
           (zs, structure=np.ones((3,3)))
      fig = plt.figure()
   11
   12
      plt.clf()
   13 fig.add subplot (2, 1, 1)
      plt.imshow(zs, origin = "upper", cmap =
            cm.gray)
   15
      plt.colorbar()
   16 fig.add subplot (2, 1, 2)
   17
      plt.imshow(zss. origin = "upper", cmap
           = cm.gray)
   18
      plt.colorbar()
      plt.show()
   19
```

Dilatation

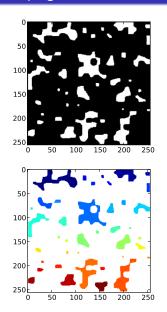


```
1.0
0.9
0.8
    1 from PIL import Image
    2 import numpy as np
0.7
    3 from matplotlib import pyplot as plt
0.6
      from matplotlib import cm
0.5
    5 im = Image.open('../Slides/figures/
            image.jpg')
0.4
       channels = im.split()
0.3
      z = np.array(channels[0])
0.2
    8 | seuil = 150.
0.1
      zs = z > seuil
    10 zse = ndimage.morphology.binary erosion
0.0
            (zs, structure=np.ones((3,3)))
       zsd = ndimage.morphology.binary erosion
1.0
            (zse, structure=np.ones((3,3)))
   12
      fig = plt.figure()
0.9
    13
       plt.clf()
0.8
      fig.add subplot(2.1.1)
0.7
       plt.imshow(zss, origin = "upper", cmap
            = cm.gray)
0.6
    16
       plt.colorbar()
0.5
   17
       fig.add subplot(2, 1, 2)
       plt.imshow(zse,
0.4
   18
                        origin = "upper", cmap
            = cm.gray)
0.3
      plt.colorbar()
   19
0.2
   20
       plt.show()
0.1
```

0.0

- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- Érosion / Dilatation
- Comptage
- Contours

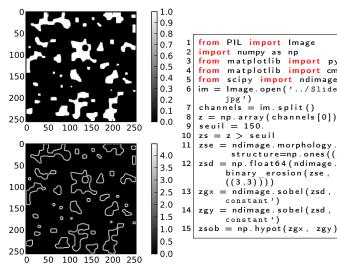
Comptage



```
1.0
0.9
    1 from PIL import Image
    2 import numpy as np
0.8
    3 from matplotlib import pyplot as plt
0.7
    4 from matplotlib import cm
    5 im = Image.open('../Slides/figures/
0.6
            image. ipg')
0.5
    6 channels = im.split()
0.4
       z = np.array(channels[0])
0.3
      | seuil = 150
      zs = z > seuil
0.2
    10 zse = ndimage.morphology.binary erosion
0.1
            (zs. structure=np.ones((3.3)))
0.0
       zsd = ndimage.morphology.binary erosion
            (zse, structure=np.ones((3,3)))
       zl, nombre = ndimage.measurements.label
40
            (zsd) # On compte les zones
36
      zl = np.where(zl == 0, np.nan, zl)
   14 fig = plt.figure()
32
   15 plt.clf()
28
    16 fig.add subplot (2, 1, 1)
24
    17
       plt.imshow(zsd, origin = "upper", cmap
           = cm.gray)
20
    18
       plt.colorbar()
16
    19
      fig.add subplot(2, 1, 2)
   20
       plt.imshow(zl, origin = "upper", cmap =
12
             cm.jet)
8
       plt.colorbar()
    21
    22
       plt.show()
```

- Ouverture/Fermeture
- Opérations basiques
- Filtrage
- 4 Histogramme
- Seuillage
- Érosion / Dilatation
- Comptage
- 8 Contours

Contours



```
3 from matplotlib import pyplot as plt
4 from matplotlib import cm
  from scipy import ndimage
  im = Image.open('../Slides/figures/image.
   z = np.array(channels[0])
  zse = ndimage.morphology.binary erosion(zs,
         structure=np.ones((3,3))
  zsd = np.float64 (ndimage.morphology.
       binary erosion(zse, structure=np.ones
  zgx = ndimage.sobel(zsd, axis=0, mode='
14 zgy = ndimage.sobel(zsd, axis=1, mode='
|zsob| = np.hypot(zgx, zgy)
```