MGM657 Outils Numériques pour l'Ingénieur

Traitement d'Images

ludovic.charleux@univ-savoie.fr

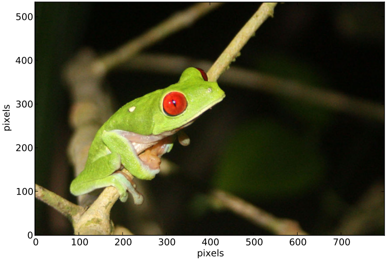
www.polytech.univ-savoie.fr

1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Plan

1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Lire et afficher une image

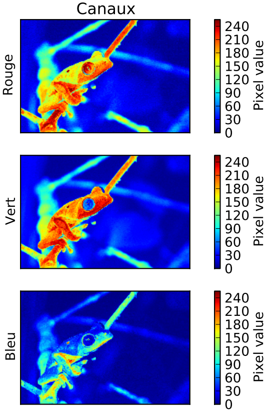


```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
im = Image.open(
  '../data/grenouille.jpg')
fig = plt.figure(0)
plt.clf()
plt.imshow(im, origin = "lower")
plt.xlabel("pixels")
plt.ylabel("pixels")
plt.show()
```

Points clés

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

Canaux et couleurs



```python
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

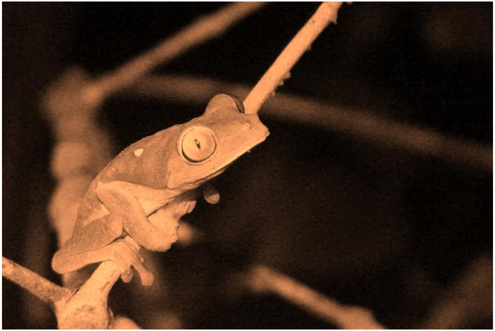
- Contexte scientifique : généralement un seul canal.
- On peut afficher une image monochrome avec une échelle de couleurs.

Image = 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)
```

```
>>> z
array([[16, 17, 19, ..., 10,  9,  8],
       [14, 15, 17, ..., 10,  9,  8],
       [15, 16, 18, ..., 11,  9,  8],
       ...,
       [25, 24, 24, ..., 19, 20, 20],
       [24, 23, 23, ..., 17, 19, 19],
       [23, 23, 22, ..., 18, 18, 18]], dtype=uint8)
>>> z.shape
(534, 800)
>>> nx, ny = z.shape
>>> z.dtype
dtype(’uint8’)
```

Sauvegarde

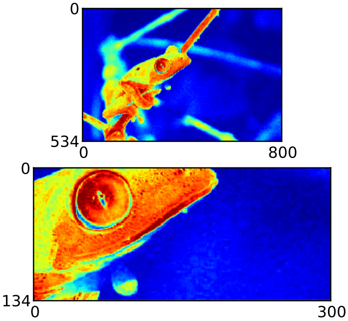


```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
im = Image.open(’../data/grenouille.jpg’)
rouge, vert, bleu = im.split()
z = np.array(rouge)
z = np.uint8(cm.copper(z)*255)
im2 = Image.fromarray(z)
im2.save("grenouille_saved.jpg")
```

Plan

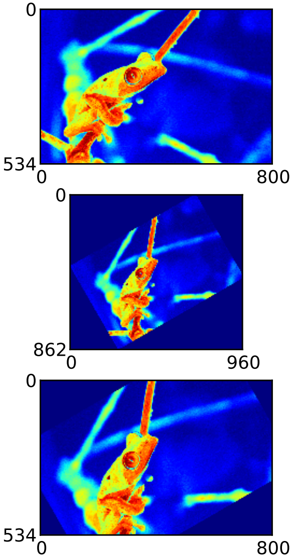
1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Rognage (*Crop*)



```
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)
ny, nx = z.shape
cx, cy = 200, 250
zc = z[cx:-cx, cy:-cy]
nyc, nxc = zc.shape

fig = plt.figure(0) # On cree une figure
plt.clf()
ax1 =fig.add_subplot(3,1,1)
plt.imshow(z, origin = "upper")
plt.xticks([0, nx])
plt.yticks([0, ny])
ax2 =  fig.add_subplot(3,1,2)
plt.imshow(zc, origin = "upper",
    interpolation = "nearest")
plt.xticks([0, nxc])
plt.yticks([0, nyc])
plt.show()
```

Rotation

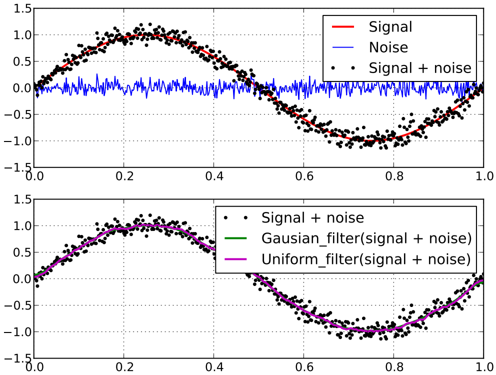


```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
from scipy import ndimage
im = Image.open('../data/grenouille.jpg')
rouge, vert, bleu = im.split()
z = np.array(rouge)
zrr = ndimage.rotate(z, 30.)
zrn = ndimage.rotate(z, 30.,
  reshape = False)

ny, nx = z.shape
nyrr, nxrr = zrr.shape
nyrn, nxrn = zrn.shape

fig = plt.figure(0) # On cree une figure
plt.clf()
ax1 =fig.add_subplot(3,1,1)
plt.imshow(z, origin = "upper")
plt.xticks([0, nx])
plt.yticks([0, ny])
ax2 =  fig.add_subplot(3,1,2)
```

Plan

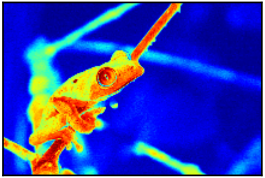
1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Lissage : exemple sur un signal 1D

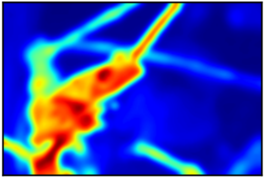


```python
import numpy as np
from matplotlib import pyplot as plt
from scipy import ndimage
x = np.linspace(0., 1., 500)
y_perf = np.sin(2. * np . pi * x)
noise = np.random.normal(loc = 0.,scale = .1,  size = len(x))
y = y_perf + noise
yg = ndimage.gaussian_filter(y, 10.)
ym =  ndimage.uniform_filter(y,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

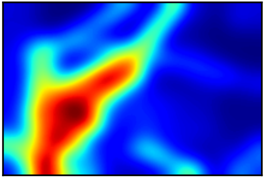
Lissage image



Raw



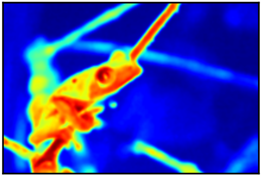
Blurred



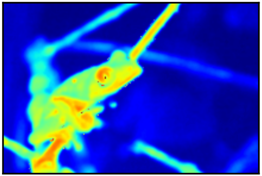
Blurred more !

```python
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
zg10 = ndimage.gaussian_filter(z, 10.)
# Blur more !
zg30 = ndimage.gaussian_filter(z, 30.)
```

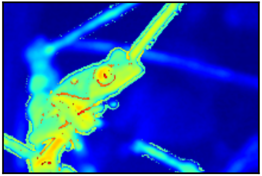
Histogramme



Raw (blurred)



Sharpened



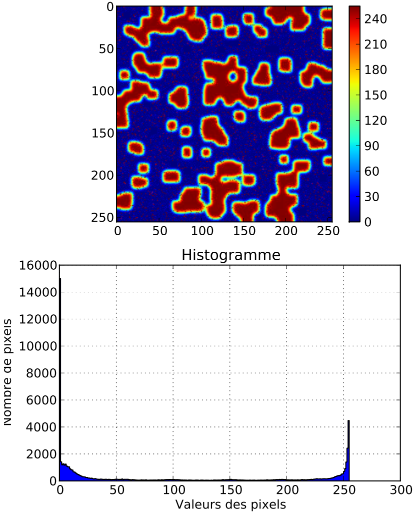
Sharpened

```
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.))
```

Plan

1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Histogramme

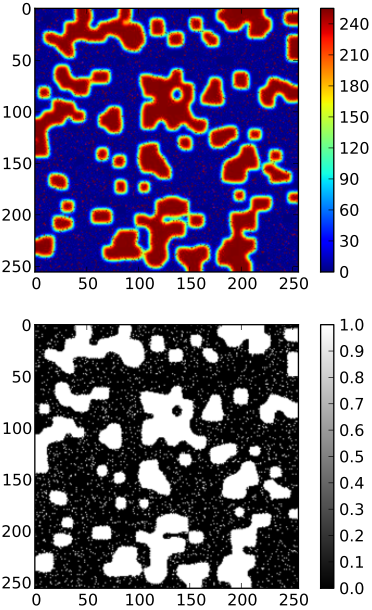


```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
im = Image.open(’../Slides/figures/image.jpg’)
channels = im.split()
z = np.array(channels[0])
N  = z.size
n_classes = int(N**.5)
fig = plt.figure()
plt.clf()
fig.add_subplot(2, 1, 1)
plt.imshow(z, origin = "upper")
plt.colorbar()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

Plan

1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Seuillage

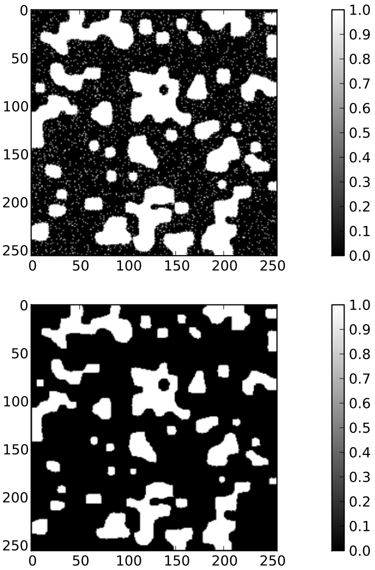


```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
im = Image.open('../Slides/figures/
    image.jpg')
channels = im.split()
z = np.array(channels[0])
seuil = 150.
zs = z > seuil
fig = plt.figure()
plt.clf()
fig.add_subplot(2, 1, 1)
plt.imshow(z, origin = "upper")
plt.colorbar()
fig.add_subplot(2, 1, 2)
plt.imshow(zs, origin = "upper", cmap =
    cm.gray)
plt.colorbar()
plt.show()
```

Plan

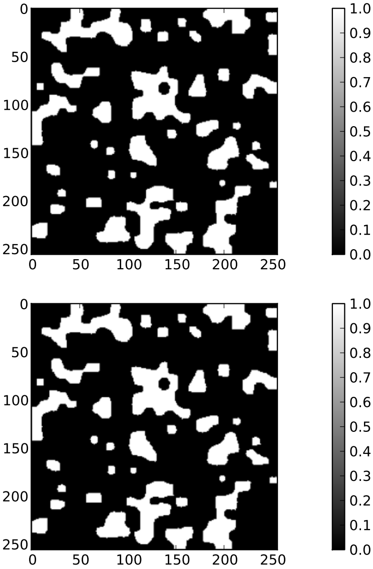
1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. **Érosion / Dilatation**
7. Comptage
8. Contours

Érosion



```python
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
im = Image.open('../Slides/figures/image.jpg')
channels = im.split()
z = np.array(channels[0])
seuil = 150.
zs = z > seuil
zss = ndimage.morphology.binary_erosion(zs, structure=np.ones((3,3)))
fig = plt.figure()
plt.clf()
fig.add_subplot(2, 1, 1)
plt.imshow(zs, origin = "upper", cmap = cm.gray)
plt.colorbar()
fig.add_subplot(2, 1, 2)
plt.imshow(zss, origin = "upper", cmap = cm.gray)
plt.colorbar()
plt.show()
```

Dilatation

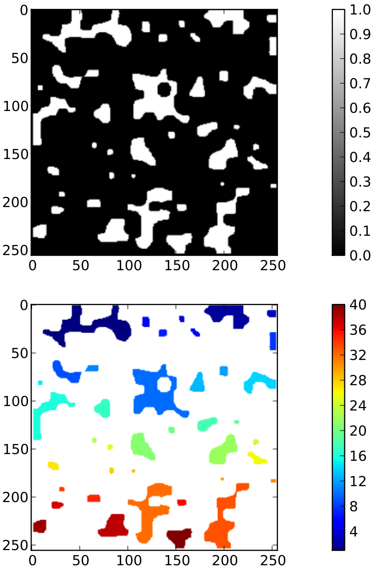


```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
im = Image.open('../Slides/figures/image.jpg')
channels = im.split()
z = np.array(channels[0])
seuil = 150.
zs = z > seuil
zse = ndimage.morphology.binary_erosion(zs, structure=np.ones((3,3)))
zsd = ndimage.morphology.binary_erosion(zse, structure=np.ones((3,3)))
fig = plt.figure()
plt.clf()
fig.add_subplot(2, 1, 1)
plt.imshow(zss, origin = "upper", cmap = cm.gray)
plt.colorbar()
fig.add_subplot(2, 1, 2)
plt.imshow(zse, origin = "upper", cmap = cm.gray)
plt.colorbar()
plt.show()
```

Plan

1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Comptage

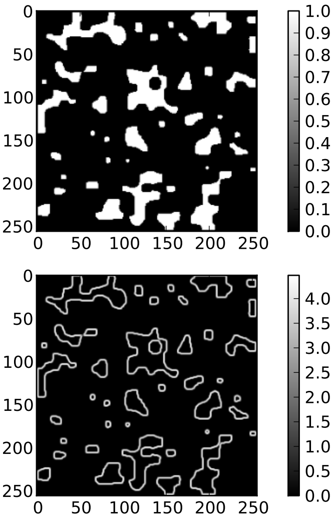


```
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
im = Image.open(’../Slides/figures/image.jpg’)
channels = im.split()
z = np.array(channels[0])
seuil = 150.
zs = z > seuil
zse = ndimage.morphology.binary_erosion(zs, structure=np.ones((3,3)))
zsd = ndimage.morphology.binary_erosion(zse, structure=np.ones((3,3)))
zl, nombre = ndimage.measurements.label(zsd) # On compte les zones
zl = np.where(zl == 0, np.nan, zl)
fig = plt.figure()
plt.clf()
fig.add_subplot(2, 1, 1)
plt.imshow(zsd, origin = "upper", cmap = cm.gray)
plt.colorbar()
fig.add_subplot(2, 1, 2)
plt.imshow(zl, origin = "upper", cmap = cm.jet)
plt.colorbar()
plt.show()
```

Plan

1. Ouverture/Fermeture
2. Opérations basiques
3. Filtrage
4. Histogramme
5. Seuillage
6. Érosion / Dilatation
7. Comptage
8. Contours

Contours



```python
from PIL import Image
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import cm
from scipy import ndimage
im = Image.open('../Slides/figures/image.jpg')
channels = im.split()
z = np.array(channels[0])
seuil = 150.
zs = z > seuil
zse = ndimage.morphology.binary_erosion(zs, structure=np.ones((3,3)))
zsd = np.float64(ndimage.morphology.binary_erosion(zse, structure=np.ones((3,3))))
zgx = ndimage.sobel(zsd, axis=0, mode='constant')
zgy = ndimage.sobel(zsd, axis=1, mode='constant')
zsob = np.hypot(zgx, zgy)
```