

Traitement d'image

(1 séance = 1h30)

Raoul HATTERER

8 février 2019

Table des matières

1	Codage RVB et niveau de gris	1
2	Image de départ	2
3	Comment lire un pixel	2
4	Comment écrire un pixel	2
5	Que fait le programme suivant ?	3
6	Passage d'une image en niveaux de gris (codé RVB sur 3 octets)	4
7	Passage d'une image en niveau de gris (luminance codée sur 1 seul octet)	4
8	Récréation ou challenge ?	6
8.1	Créer une image en négatif	6
8.2	Diagonale	6

1 Codage RVB et niveau de gris

- Aller sur [colors RGB](#) et tester ce que l'on obtient si l'on remplace chacune des valeurs R, V et B d'un pixel par la moyenne des sous-pixels.
- Essayer pour plusieurs couleurs.

2 Image de départ



FIGURE 1 – Image de départ

3 Comment lire un pixel

Après avoir fait quelques recherches sur ce qu'est un "pixel", voyons comment lire le pixel de coordonnées (100,250).

```
from PIL import Image
img = Image.open("pomme.jpg")
r,v,b=img.getpixel((100,250))
print("canal rouge : ",r,"canal vert : ",v,"canal bleu : ",b)

('canal rouge : ', 19, 'canal vert : ', 88, 'canal bleu : ', 192)
```

4 Comment écrire un pixel

```
from PIL import Image
img = Image.open("pomme.jpg")
img.putpixel((5,5),(255,0,0))
img.show()
```

Question : *Identifier où se trouve l'origine de l'image.*

5 Que fait le programme suivant ?

```
from PIL import Image
img = Image.open("pomme.jpg")
largeur_image,hauteur_image=img.size
for y in range(hauteur_image):
    for x in range(largeur_image):
        rouge,vert,bleu=img.getpixel((x,y))
        nouveau_rouge=vert
        nouveau_vert=bleu
        nouveau_bleu=rouge
        img.putpixel((x,y),(nouveau_rouge,nouveau_vert,nouveau_bleu))
img.show()
img.save("pommeMystere.jpg")
```

On analyse le code ci-dessus qui servira de base pour le défi suivant.



FIGURE 2 – Résultat du programme mystère

6 Passage d'une image en niveaux de gris (codé RVB sur 3 octets)

Après avoir fait quelques recherches sur les "images en niveaux de gris", écrivez un programme qui transforme une "image couleur" en une "image en niveaux de gris".

Petite astuce qui pourrait vous aider : en Python pour avoir une division entière (le résultat est un entier), il faut utiliser l'opérateur `//` à la place de l'opérateur `/`

Remarque : On donne l'algorithme aux élèves (ou on le construit avec eux); ils doivent alors programmer le passage d'une image couleur à une image en niveaux de gris.

```
from PIL import Image
img = Image.open("pomme.jpg")
largeur_image=480
hauteur_image=300
for y in range(hauteur_image):
    for x in range(largeur_image):
        rouge,vert,bleu=img.getpixel((x,y))
        nouveau_rouge=(vert+bleu+rouge)//3
        nouveau_vert=(vert+bleu+rouge)//3
        nouveau_bleu=(vert+bleu+rouge)//3
        img.putpixel((x,y),(nouveau_rouge,nouveau_vert,nouveau_bleu))
img.show()
img.save("pommegrise.jpg")
```

7 Passage d'une image en niveau de gris (luminance codée sur 1 seul octet)

```
1 from PIL import Image
2 img = Image.open("pomme.jpg").convert("L")
3 img.show()
4 img.save("pommegriseL.jpg")
```

Comparer la taille des différents fichiers. Conclure.



FIGURE 3 – Pomme Linux en niveaux de gris (codé RVB)



FIGURE 4 – Image en niveaux de gris (luminance L)

8 Récréation ou challenge ?

8.1 Créer une image en négatif



FIGURE 5 – Négatif

8.2 Diagonale

Créer le programme qui garde l'image d'origine au-dessus d'une diagonale et qui transforme en niveaux de gris en-dessous de celle-ci.



FIGURE 6 – Pomme coupée