

Un lièvre en 3D

Le tableau ci-contre est une œuvre d'Albrecht Dürer peintre allemand du début du XVI^e siècle. Chacun d'entre nous appréciera la prouesse technique qui consiste à « croquer » ainsi un lièvre, la précision du geste de l'artiste qui souhaite poil par poil coller le plus précisément possible à la nature.

Dans ce texte, notre intention est de partir de cette image numérique du tableau pour construire une image en 3D. Le procédé que nous allons utiliser consiste à obtenir **un anaglyphe** du grec ana « du bas vers le haut » et de glyphe « ciselure ».



Par souci d'efficacité, nous n'allons pas nous pencher sur tout ce qui concerne le traitement par le cerveau humain d'un anaglyphe mais nous limiter à sa construction. Il va s'agir pour nous de décortiquer en trois étapes le processus de construction et d'utiliser tout au long de ce processus des programmes développés en Python.

Utilitaire pour programmer en Python et enregistrer l'image de départ

Le logiciel EduPython

Ce logiciel est téléchargeable à l'adresse suivante : <https://edupython.tuxfamily.org/> . Ce logiciel offre un environnement éducatif pour se former à l'utilisation du logiciel Python.



L'enregistrement de l'image de départ

Nous enregistrons sur notre ordinateur l'image numérique fournie ci-avant dans ce texte au format PDF. Pour ce faire, il suffit de :

- Cliquer droit sur l'image du tableau de Van Gogh et de copier l'image ;
- Coller ensuite cette image dans un éditeur de texte ;
- Cliquer droit à nouveau sur l'image et l'enregistrer sous le format jpeg dans le dossier dans lequel le logiciel EduPython est enregistré sur votre disque C:. On convient de donner à ce fichier le nom du tableau : lièvre.

Remarque : Il sera utile par la suite de fournir au logiciel l'adresse de l'image à traiter, pour ce faire, il suffira de cliquer droit sur le fichier contenant l'image et de prendre l'adresse dudit fichier dans les propriétés de l'image :

C:\EduPython\lièvre.jpg .

Il nous faut à présent ouvrir la bibliothèque Image qui contient – entre autres - plusieurs programmes utiles pour parcourir ou définir

une image numérique. Nous allons aussi affecter une adresse à l'image du lièvre et ouvrir cette image. Voici les instructions à entrer :

```
from PIL import Image
a="C:\EduPython\lièvre.jpg"
im=Image.open(a)
```

Notre image numérique du lièvre à traiter s'appelle à présent im.

Une première image en rouge

Nous allons effectuer un traitement de l'image de départ afin d'obtenir une image uniquement dans les tons de rouge. Pour ce faire :

- Nous passons en revue cette image pixel par pixel et relevons pour chaque pixel son code couleur **(r,g,b)** ;
- Nous allons conserver uniquement le niveau de rouge de chaque pixel, nous remplaçons ainsi son code couleur par le code **(r,0,0)**.

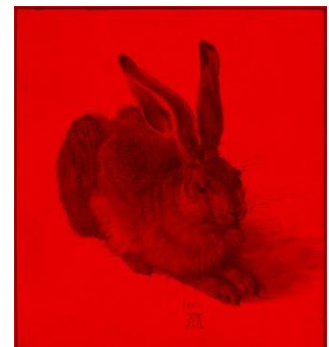
Par exemple, si le code couleur d'un pixel est (122, 56 ,180), on ne retiendra que le 122. Le code couleur attribué au nouveau pixel sera (122, 0 , 0). Ce qui donnera un ton de rouge pour ce pixel.

Programmons à présent le parcours de notre image et la modification de tous ses pixels.

```
def rouge(image):
    (c,l)=image.size
    imagearrivee=Image.new(image.mode,image.size)
    for x in range(c):
        for y in range(l):
            pixel=image.getpixel((x,y))
            p=(pixel[0],0,0)
            imagearrivee.putpixel((x,y),p)
    imagearrivee.save("C:\Edupython\photo rouge.JPG")
```

Le résultat du programme est une nouvelle image numérique placée à l'adresse suivante : C:\Edupython\photo rouge.JPG. Pour appliquer cette fonction à notre image du lièvre, il suffit de prendre comme argument im. C'est-à-dire taper : rouge(im). Nous pouvons aller rechercher cette nouvelle image dans le dossier Edupython.

Voici ci-contre le résultat obtenu, vous remarquerez que lorsque l'on observe cette photographie numérique avec des lunettes 3D en fermant l'un ou l'autre de vos deux yeux :



✚ En conservant uniquement l'œil gauche ouvert, vous voyez la photographie car le filtre rouge laisse passer la couleur rouge ;

✚ En conservant uniquement l'œil droit ouvert, vous voyez un cadre noir car le filtre cyan ne laisse pas la couleur rouge.

Une deuxième image en cyan

Rappelons que la couleur cyan est la couleur complémentaire de la couleur rouge. Nous allons à présent effectuer un traitement de l'image en tout point analogue au présent mais il va s'agir d'obtenir in fine une image uniquement dans les tons de vert et de bleu. Pour ce faire :

- Nous passons en revue l'image de départ pixel par pixel et relevons pour chaque pixel son code couleur **(r,g,b)** ;
- Nous conservons uniquement le niveau de vert et le niveau de bleu de chaque pixel, nous remplaçons ainsi son code couleur par le code **(0,g,b)**.

Par exemple, si le code couleur d'un pixel est (122, 56 ,180), on ne retiendra que le 56 et le 180. Le code couleur attribué au nouveau pixel sera (0 , 56 , 180). Ce qui donnera une couleur obtenue en mélangeant un ton de vert et un ton de bleu pour ce pixel.

Programmons à présent le parcours de notre image et la modification de tous ses pixels.

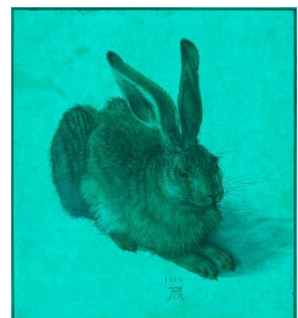
```
def cyan(image):  
    (c,l)=image.size  
    imagearrivee=Image.new(image.mode,image.size)  
    for x in range(c):  
        for y in range(l):  
            pixel=image.getpixel((x,y))  
            p=(0,pixel[1],pixel[2])  
            imagearrivee.putpixel((x,y),p)  
    imagearrivee.save("C:\Edupython\photo_cyan.JPG")
```

Pour appliquer cette fonction à notre image du lièvre, il suffit de prendre comme argument im. C'est-à-dire taper : cyan(im).

Nous pouvons à présent aller rechercher le résultat dans le dossier Edupython.

A l'identique de ce nous avons fait précédemment, prenons des lunettes 3D pour regarder le résultat obtenu. On remarque cette fois-ci que :

- + En conservant uniquement l'œil gauche ouvert, on voit un cadre noir car le filtre rouge ne laisse passer ni les composantes de vert et ni celles de bleu ;
- + En conservant uniquement l'œil droit ouvert, vous voyez la photographie car le filtre cyan laisse passer les composantes de vert et celles de bleu.

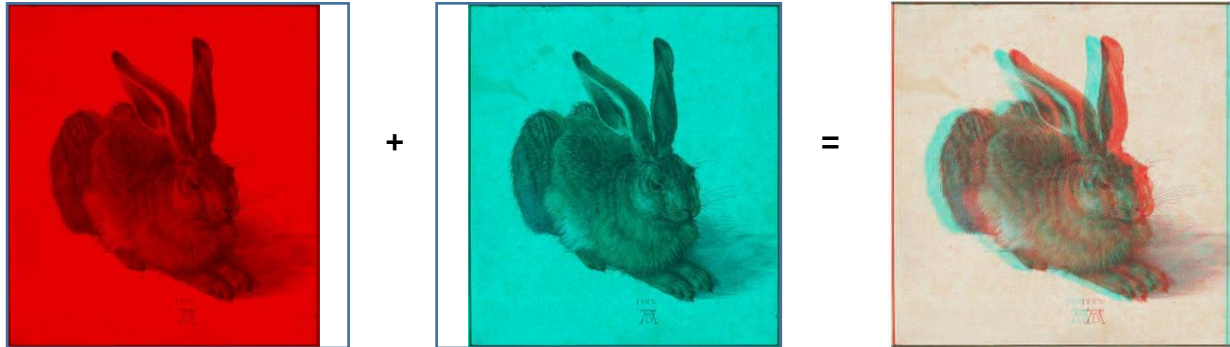


Superposition des deux images de façon décalée

La construction de l'anaglyphe est basée sur le fait de superposer à présent les deux images avec un petit décalage. Pour ce faire, nous allons :

- + Ajouter à l'image en rouge un bord blanc à droite ;
- + Ajouter à l'image en cyan un bord blanc à gauche ;
- + Superposer enfin les deux images ainsi obtenues.

Autrement dit, le schéma ci-dessous montre ce que nous allons progressivement réaliser.



Ajout à l'image en rouge d'une bordure à droite

Nous allons définir une nouvelle image en ajoutant 20 colonnes à droite à l'image en rouge. Nous remplissons ensuite ces vingt colonnes uniquement avec des pixels de couleurs blanches.

```
def ajoutborddroit(image):  
    (c,l)=image.size  
    imagearrivee=Image.new(image.mode,(c+20,l))  
    for x in range(c):  
        for y in range(l):  
            pixel=image.getpixel((x,y))  
            imagearrivee.putpixel((x,y),pixel)  
    for x in range(c,c+20):  
        for y in range(l):  
            imagearrivee.putpixel((x,y),(255,255,255))  
    imagearrivee.save("C:\Edupython\photo augmentée droit.JPG")  
ajoutborddroit(im)
```

Pour appliquer cette fonction à notre image en rouge, il nous faut l'ouvrir et lui donner un nom. Il suffit ensuite d'appliquer la fonction ajoutborddroit en prenant comme argument ce nom.

```
a1="C:\Edupython\photo rouge.JPG"  
imrouge=Image.open(a1)  
ajoutborddroit(imrouge)
```

Ajout à l'image en cyan d'une bordure à gauche

Nous allons définir une nouvelle image en ajoutant 20 colonnes à gauche à l'image en cyan. Nous remplissons ensuite ces vingt colonnes uniquement avec des pixels de couleurs blanches.

```
def ajoutbordgauche(image):
    (c,l)=image.size
    imagearrivee=Image.new(image.mode,(c+20,l))
    for x in range(20):
        for y in range(l):
            imagearrivee.putpixel((x,y),(255,255,255))
    for x in range(20,c+20):
        for y in range(l):
            pixel=image.getpixel((x-20,y))
            imagearrivee.putpixel((x,y),pixel)
    imagearrivee.save("C:\Edupython\photo augmentée gauche.JPG")
```

Pour appliquer cette fonction à notre image en cyan, il nous faut l'ouvrir et lui donner un nom. Il suffit ensuite d'appliquer la fonction ajoutbordgauche en prenant comme argument ce nom.

```
a2="C:\Edupython\photo cyan.JPG"
imcyan=Image.open(a1)
ajoutborddroit(imcyan)
```

Superposition des deux images bordées

Nous superposons à présent les deux images bordées que nous avons obtenues. Le fait d'avoir ajouté un bord à droite pour l'image en rouge et un bord à gauche pour l'image en cyan va nous permettre d'obtenir ainsi le décalage souhaité entre l'image en rouge et l'image en cyan.

La superposition des deux images va consister à :

- ✚ Définir une nouvelle image qui a pour dimensions les dimensions communes aux deux images à superposer ;
- ✚ Parcourir en même temps les deux images à superposer et relever le code couleur de chaque pixel pour chacune des images ;
- ✚ Prendre comme code couleur du pixel de la nouvelle image, le maximum de la composante rouge, le maximum de la composante verte et le maximum de la composante bleue.

Par exemple, si les codes couleurs relevés sont (122, 56 ,180) et (74 , 221 , 101), le code couleur attribué au nouveau pixel sera (122 , 221 , 180).

```
def superpose(image1,image2):
    (c,l)=image1.size
    imagearrivee=Image.new(image1.mode,(c,l))
    for x in range(c):
        for y in range(l):
            pixel1=image1.getpixel((x,y))
            pixel2=image2.getpixel((x,y))
            r=max(pixel1[0],pixel2[0])
            g=max(pixel1[1],pixel2[1])
            b=max(pixel1[2],pixel2[2])
            imagearrivee.putpixel((x,y),(r,g,b))
    imagearrivee.save("C:\Edupython\photo résultat.JPG")
```

Il nous faut à présent lancer la production de notre image 3D. Pour ce faire, nous allons ouvrir les deux images bordées et leur donner un nom, puis appliquer la fonction de superposition de ces deux images en appliquant la fonction superpose.

```
ag1="C:\Edupython\photo augmentée droit.JPG"
im1=Image.open(ag1)
ag2="C:\Edupython\photo augmentée gauche.JPG"
im2=Image.open(ag2)
superpose(im1,im2)
```

Le résultat de cette superposition est disponible dans le fichier EduPython. L'image porte le nom photo résultat. L'effet 3D dépend de la distance entre les pupilles. Pour modifier cet effet, il suffit de modifier le décalage entre les deux images superposées c'est-à-dire de remplacer le nombre de 20 colonnes ajoutées en bordure par un autre nombre.



Conclusion

L'effet 3D dépend de la distance entre les pupilles. Pour modifier cet effet, il suffit de modifier le décalage entre les deux images superposées c'est-à-dire de remplacer le nombre de 20 colonnes ajoutées en bordure par un autre nombre.

Vous pouvez à présent appliquer ces fonctions à toute image de votre choix.

Jean-Alain Roddier
IA-IPR de mathématiques
Académie de Clermont-Ferrand