

TP 8 Incrustation d'image

I Premières manipulation sur les images

I.1 Préparation du répertoire de travail

Lors de la lecture d'un fichier par python, il est nécessaire que le fichier à lire soit placé dans le répertoire de travail de python. Pour que cela soit le cas, suivre les instructions:

1. Ouvrir pyzo et ouvrir un **nouveau** fichier (file puis new)
2. Sauvegarder ce nouveau fichier python dans le dossier de votre choix.
3. Copier dans ce dossier les images 'chicago.png' et 'fond vert.png' situées dans les documents en consultation de la classe.
4. Dans le menu run de pyzo, cliquer sur **run file as script**

Le but du TP est d'incruster le personnage de l'image 'fond vert.png' sur l'image 'chicago.png'

I.2 Création d'une image couleur RVBT, utilisations des tableaux numpy

1 Recopier et exécuter les lignes suivantes:

```
import numpy as np
A=np.zeros((2,3))
print(A)
print(A[0][1])
print(A)
print(len(A),len(A[0]))
```

L'instruction `np.zeros` permet de créer un tableau deux dimensions dont les éléments sont nuls. Ce tableau peut s'utiliser comme les tableaux définis sous forme de listes de listes.

Tapez l'instruction suivante dans le shell:

A

On constate que A n'est pas une simple liste de listes mais un **array**. Un cours ultérieur présentera tous les avantages procurés par l'utilisation de cette structure de données (en particulier le calcul vectoriel).

2 Recopier et exécuter les lignes suivantes:

```
T=np.zeros((60,120,4))
print(len(T),len(T[0]),len(T[0][0]))
print(T[0][0])
print(T[10][10][3])
import matplotlib.pyplot as plt
plt.imshow(T)
plt.axis('off')
plt.show()
```

```

    puis

for i in range(60):
    for j in range(40):
        T[i][j]=[0,0,1,1]
    for j in range(40,80):
        T[i][j]=[1,1,1,1]
    for j in range(80,120):
        T[i][j]=[1,0,0,1]
plt.imshow(T)
plt.axis('off')
plt.show()

```

On constate que l'instruction `np.zeros` permet de créer un tableau trois dimensions. Lorsque la troisième dimension de ce tableau vaut 4, il peut représenter une image en couleurs.

- Sa première dimension est le nombre de lignes de l'image
- Sa deuxième est le nombre de colonnes de l'image
- Sa troisième dimension est 4.

Pour tout pixel (i,j) , $T[i][j]$ est un tableau à une dimension de longueur 4. Si $T[i][j]$ vaut $[r,v,b,t]$, alors r est le niveau de rouge du pixel (i,j) , v est le niveau de vert du pixel (i,j) , et b est le niveau de bleu du pixel (i,j) et t est le niveau de transparence du pixel (i,j) .

Les lignes sont numérotées de haut en bas et les colonnes de gauche à droite; les niveaux de couleurs sont (ici) des réels (plus précisément des flottants) compris entre 0 et 1.

Dans la suite, on prendra systématiquement $t=1$

$[r,v,b,t]=[1,0,0,1]$ correspond au rouge

$[r,v,b,t]=[0,0,1,1]$ correspond au bleu

$[r,v,b,t]=[1,1,1,1]$ correspond au blanc.

3 Faire afficher le drapeau italien.

II Manipulation d'images existantes

II.1 Chargement et affichage d'une image couleur avec python

Les instructions suivantes permettent de charger l'image 'chicago.png'

```

import matplotlib.pyplot as plt #importation de la bibliothèque pyplot
image1=plt.imread ("chicago.png") #chargement d'une image sous forme de tableau numpy

```

Q1: Exécuter ces instructions. Quelle est la taille du tableau à trois dimensions obtenu?

Q2: Afficher cette image avec matplotlib.

Q3: Quelle est le niveau de rouge du pixel en haut à gauche? le niveau de vert du pixel en bas à droite?

II.2 Modification d'images

Pour éviter de modifier une image, on peut effectuer une copie du tableau lui correspondant de la manière suivante:

```
import copy
nouv_image=copy.deepcopy(image1)
```

On peut ensuite modifier `nouv_image` sans modifier `image1`

Q4: Ecrire une fonction `filtrage(image,C)` où `image` est un tableau numpy correspondant à une image, `C` vaut 0,1 ou 2 et correspond au numéro d'une couleur et qui renvoie un nouveau tableau dans lequel le niveau de couleur `C` vaut 0 (couleur absente).

Q5: Supprimer le rouge de l'image chicao. Vérifier le en affichant l'image.

II.3 Incrustation d'image

Q6: Définir une tableau numpy `image2` correspondant à l'image 'fond vert.png' et afficher cette image.

Nous allons créer une nouvelle image qui "superpose" à l'image chicao le personnage de l'image fond vert. La méthode est la suivante.

- On crée une nouvelle image copie de chicao.
- On parcourt l'image fond vert (correspondant à l'argument `avant` de la fonction dans Q9) lorsqu'un pixel de cette image n'est pas vert, on transfère les niveaux de couleurs de ce pixel à la même position dans la nouvelle image.

Comme tous les pixels verts du fond vert ne sont pas exactement identiques, on va choisir un pixel de référence de couleur verte et on considèrera que les pixels dont les niveaux de couleur sont proches de ce pixel de référence sont aussi vert. On utilise pour cela la fonction distance suivante:

Q7: Ecrire une fonction `distance(P,Q)` d'arguments deux listes de longueur 4 `P` et `Q` qui renvoie la distance de `P` à `Q` (considérés comme des points de l'espace) définie comme suit:

Si $P=[rP,vP,bP,tP]$ et $Q=[rQ,vQ,bQ,tQ]$, la distance de `P` à `Q` est $\sqrt{(rQ-rP)^2+(vQ-vP)^2+(bQ-bP)^2}$ (ne tient pas compte de la transparence).

Q8: Ecrire une fonction `incrustation(avant,arriere,i_reference,j_reference,seuil)` d'arguments deux tableaux `avant` et `arriere` correspondant à deux images couleurs de mêmes dimensions et un nombre `seuil`, les valeurs `i_reference` et `j_reference` correspondant à la ligne et la colonne de référence. La fonction renverra un tableau correspondant à l'incrustation de l'image `avant` sur l'image `arriere` comme décrit précédemment en plaçant les pixels de l'image `avant` qui sont distants du pixel de référence de plus que la valeur `seuil` dans une copie de l'image `arriere` à la position qu'ils occupaient dans l'image avant.

Q9: Incruster l'image fond vert sur l'image chicao.

III Autour du drapeau du Japon

Les informations nécessaires pour la suite figurent ici: https://fr.wikipedia.org/wiki/Drapeau_du_Japon

Q10: Créer une image correspondant au drapeau du japon.

Q11: Créer une image correspondant au drapeau de l'armée impériale du japon (on pourra utiliser les fonction `acos` et `asin` de la bibliothèque `math`).

Correction

Q1: #Le resultat est (257, 322, 3) 227 lig 332 col 3 couleurs

```
image1=plt.imread ("chicago.png") # un tableau numpy est cree
len(image1) #Le resultat est 257 lignes
len(image1[0]) #Le resultat est 322 colonnes
len(image1[0][0])
image1[0][0][3]# niveau de rouge du pixel (0,0)
image1[256][321][1]# niveau de vert du pixel en bas droite
```

Q2:

```
plt.imshow(image1) #Affichage en utilisant une figure matplotlib
plt.axis('off')#Les axes sont supprimees
plt.show()
```

Q3:

```
image1[0][0][0]# niveau de rouge du pixel (0,0)
image1[256][321][1]# niveau de vert du pixel en bas droite
```

Q4:

```
def filtrage(couleur,image):
    """ couleur vaut 0,1 ou 2 designe la couleur rouge, vert, bleu a filtrer"""
    import copy
    nouv_image=copy.deepcopy(image) #on ne modifie pas l'image de depart
    n,p=len(image),len(image[0])
    for i in range(n):
        for j in range(p):
            nouv_image[i][j][couleur]=0
    return nouv_image
```

Q5:

```
IM=filtrage(1,image1)
plt.imshow(IM)
plt.axis('off')
plt.show()
```

Q6:

```
image2=plt.imread ("fond vert.png")
plt.imshow(image2)
plt.axis('off')
plt.show()
```

Q7:

```
def distance(P,Q): #les elements des tableaux sont des uint8
    d=(Q[0]-P[0])**2+(Q[1]-P[1])**2+(Q[2]-P[2])**2
    return d**(1/2)
distance(image2[250,315],image2[125,160])
```

Q8:

```
def incrustation(avant,arriere,i_reference,j_reference,seuil):
    import copy
    image=copy.deepcopy(arriere)
    n,p,q=len(image2),len(image2[0]),len(image1[0][0])
    i0,j0=i_reference,j_reference
    for i in range(n):
        for j in range(p):
            if distance(avant[i][j],avant[i0][j0])>seuil:
                image[i][j]=copy.deepcopy(avant[i][j])
    return image
```

Q9:

```
reference=image2[10,10]
seuil=0.1
image=incrustation(image2,image1,reference,seuil)
plt.imshow(image)
plt.axis('off')
plt.show()
```

Q10:

```
T=np.zeros((600,1000,4))
for i in range(600):
    for j in range(1000):
        T[i][j]=[1,1,1,1]
        if (i-300)**2+(j-500)**2<40000:
            T[i][j]=[1,0,0,1]
plt.imshow(T)
plt.axis('off')
plt.show()
```

Q11: