Workshop Comparaison d’images

OPENCV

I/ Prérequis :

- $> sudo apt-get install python3
- $> sudo apt-get install pip3
- $> pip3 install opencv-python

- Télécharger les images (.jpg) dans le dossier images

2/ Initialisation :

- Créer un nouveau fichier .py avec les import suivants :

```
import cv2 as cv # compare method
from matplotlib import pyplot as plt # draw graph
from skimage.measure import compare_ssim as ssim # ratio
```

3/ Programme :

- Charger les deux 1ères images (img1 et img2) à comparer : “./images/original_golden_bridge.jpg“ et “./images/sunburst.jpg “
- Initialiser le **SIFT DETECTOR** pour les deux images. (kp1, des1; kp2, des2)
- Initialiser les paramètres **FLANN**. (index_params, search_params, flann, matches)

Si les images ont la même taille on peut alors les comparer entre elles et afficher le nombre de “good match”.

Plus le ssim est proche de 1 plus les images semblent similaires entre elles

```
if img1.shape == img2.shape:
    print("The images have same size and channels")
    s = ssim(img1, img2, multichannel = True)
    print("SSIM: %.2f" % s)
    print("Keypoints 1ST Image: " + str(len(kp1)))
    print("Keypoints 2ND Image: " + str(len(kp2)))
```

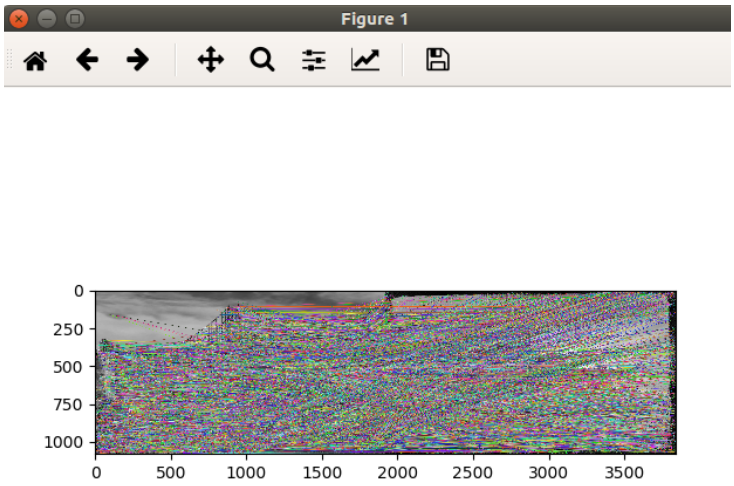
- Afficher la comparaison finale avec **PLT**

```
    img3 = cv.drawMatchesKnn(img1,kp1,img2,kp2,matches,None)
    plt.imshow(img3,),plt.show()
```

Après avoir exécuté le programme avec la commande:

“python3 nom_fichier.py”

Vous devriez avoir un résultat comme suit :




```
SSIM: 0.54
Keypoints 1ST Image: 6565
Keypoints 2ND Image: 20681
```

On ne peut pas réellement déduire si les deux images sont similaires ou non...

Comment pourrait-on rendre cette comparaison plus claire?

Si on revenait un peu plus haut avant de comparer la taille des deux images...

Grâce aux paramètres **FLANN** on pourrait garder que les bonnes comparaisons et avoir moins d'erreur de précision.

On aurait quelque chose qui ressemblerait à ça :

```
ratio = 0.7

good_points = []
matchesMask = [[0,0]

for i in range(len(matches))]
for i,(m,n) in enumerate(matches):
    if m.distance < ratio * n.distance:
        matchesMask[i]=[1,0]
        good_points.append(m)
```

On peut alors afficher le nombre de "*good match*".

```
if img1.shape == img2.shape:
        print("The images have same size and channels")
        s = ssim(img1, img2, multichannel = True)
        print("SSIM: %.2f" % s)
        print("Keypoints 1ST Image: " + str(len(kp1)))
        print("Keypoints 2ND Image: " + str(len(kp2)))
        print('matches : ' + str(len(good_points)))
        print("How good is the match: ", len(good_points) / number_keypoints * 100)
```

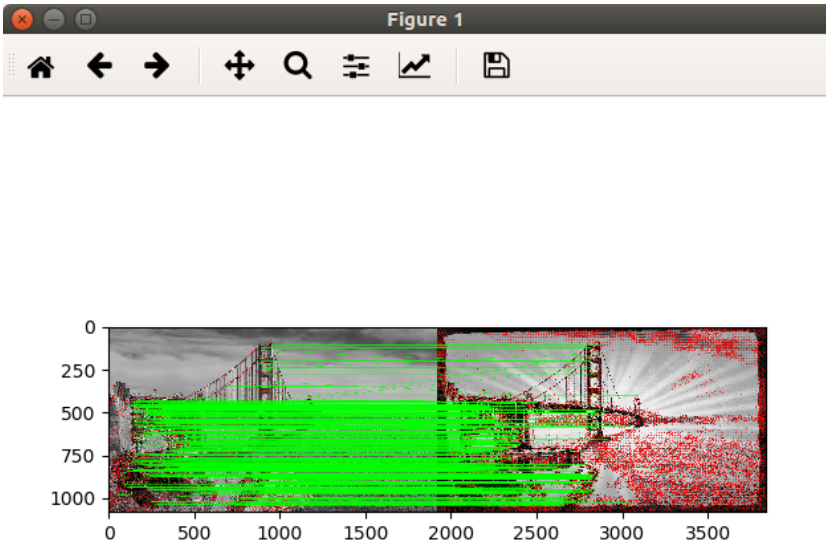
- Initialiser les paramètres nécessaires à l'affichage (draw_params) d'uniquement des points que l'on vient de calculer

- En affichant la comparaison finale des deux images...

```
        img3 = cv.drawMatchesKnn(img1,kp1,img2,kp2,matches,None,**draw_params)
        plt.imshow(img3,),plt.show()
```

… On aurait nu résultat bien différent !

“python3 nom_fichier.py”




```
SSIM: 0.54
Keypoints 1ST Image: 6565
Keypoints 2ND Image: 20681
matches : 843
How good is the match: 12.84
```

Le résultat est bien celui qu’on attendait.

Maintenant vous pouvez essayer avec les autres images du fichier et comparer les résultats. “./images/textured.jpg”, “./images/blu_filer.jpg”, “./images/duplicate.jpg”