

## TD05 : Solution EXERCICE 1 / 3

### Ex1.1

```
import cv2 ;import numpy as np
#####
#IMAGES
image1 = cv2.imread('reference_image_logo.png')
image2 = cv2.imread('pose_logo_1.png')

#####
#FEATURES
detector = cv2.xfeatures2d.SURF_create()
keypoints1, descriptors1 = detector.detectAndCompute(image1,None)
keypoints2, descriptors2 = detector.detectAndCompute(image2,None)

#####
# MATCHES
bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
matches = bf.match(descriptors2,descriptors1)
matches = sorted(matches, key = lambda x:x.distance)
matches = matches[:30]

#####
#DISPLAY
image_display=np.zeros((image1.shape[0],2*image1.shape[1]))
image_display = cv2.drawMatches(image1,keypoints1,image2,keypoints2,matches, image_display)
cv2.imshow('frame',image_display)

cv2.waitKey()
cv2.destroyAllWindows()
```

### Ex1.2

```
#####
#REFERENCE IMAGE
image_ref = cv2.imread('reference_image_logo.png')
mask=cv2.imread('reference_image_logo_mask.png',0) #0: pour un channel
detector = cv2.xfeatures2d.SURF_create()
keypoints_ref, descriptors_ref = detector.detectAndCompute(image_ref,mask)

#####
#NEW IMAGES (NEW POSES)
image_display=np.zeros((image_ref.shape[0],2*image_ref.shape[1]))
for i in range(1,14): #'pose_1','pose_2',...,'pose_14'
    #READING
    filename='pose_logo_'+str(i)+'png'
    image = cv2.imread(filename)
    #DETECTOR
    keypoints, descriptors = detector.detectAndCompute(image,None)
    #MATCHING WITH REFERENCE IMAGE
    bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
    matches = bf.match(descriptors_ref ,descriptors)
    matches = sorted(matches, key = lambda x:x.distance)
    #DISPLAY
    image_display=cv2.drawMatches(image_ref, keypoints_ref,image, keypoints, matches[:20],image_display)
    cv2.imshow('frame',image_display)
    cv2.imwrite("matching_"+str(i)+".png",image_display)
    cv2.waitKey()
    cv2.destroyAllWindows()
```