

TD05 : Solution EXERCICE 2 / 3

```
#####  
#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()  
kp_ref, ds_ref = detector.detectAndCompute(image_ref,mask)  
#####  
#INITIAL POSITION OF THE VIRTUAL OBJECT  
corners=helper_surf.keypoints2numpyarray(kp_ref)  
K=helper.create_K()  
corners_3d=helper.pixels2camera(corners,K,Z=500)  
square_3d=helper.create_square(corners_3d,size=60)  
square_3d_h=helper.coord2homogeneous(square_3d)  
  
#####  
#NEW POSE  
for i in range(1,14):  
    #Current image  
    image_current = cv2.imread('pose_logo_'+str(i)+'.png')  
    #DESCRIPTORS  
    keypoints, descriptors = detector.detectAndCompute(image_current,None)  
    #MATCHING  
    bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)  
    matches = bf.match(ds_ref,descriptors)  
    matches = sorted(matches, key = lambda x:x.distance)  
    #SUBSET OF MATCHING KEYPOINTS (FROM REF AND CURRENT)  
    kp_ref_subpart=[]  
    kp_current=[]  
    for m in matches[0:10]:  
        tmp_kp=kp_ref[m.queryIdx]  
        kp_ref_subpart+= [tmp_kp]  
        tmp_kp=keypoints[m.trainIdx]  
        kp_current+= [tmp_kp]  
    corners_2d=helper_surf.keypoints2numpyarray(kp_ref_subpart)  
    corners_3d=helper.pixels2camera(corners_2d,K,Z=500)  
    corners2d_current=helper_surf.keypoints2numpyarray(kp_current)  
  
    #TRANSFORMATION (POSE) ESTIMATION: 3D CORNERS IN REF IMAGE TO 2D CORNERS IN CURRENT IMAGE  
    M_ext=helper.compute_extrinsic_matrix(corners_3d, corners2d_current, K,True)  
    #PROJECTION  
    P=np.dot(K,M_ext)  
    registered_square=helper.world2pixels(square_3d_h,P)  
    #DISPLAY  
    image_current=cv2.drawKeypoints(image_current,kp_current)  
    cv2.polylines(image_current, [np.int32(registered_square)], True, (0, 0, 255), 8)  
    cv2.imshow('frame',image_current)  
    cv2.imwrite('registration_'+str(i)+'.png',image_current)  
    cv2.waitKey()
```