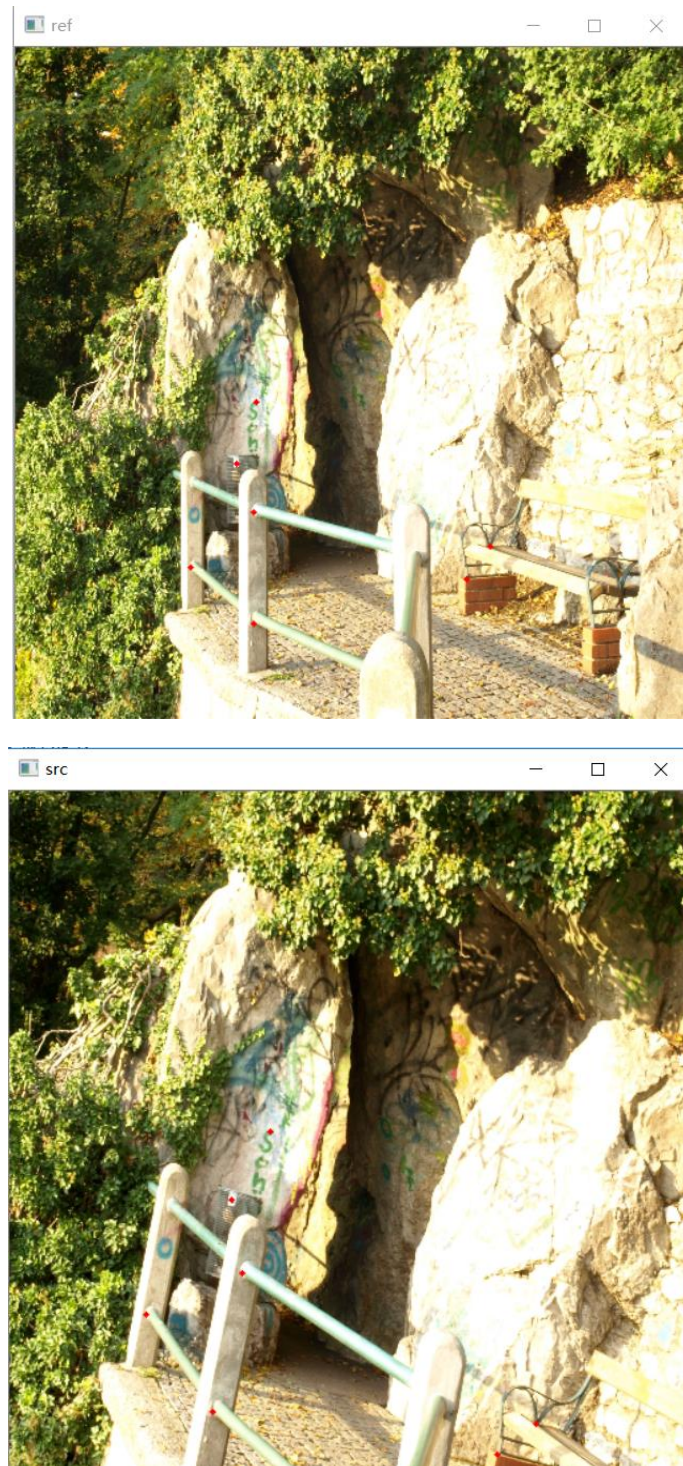


图像配准作业

自动化少 61 董浩 2140506111

一、手动标点：

以 Image A 为基准图，以 Image B 为待配准图，每张图片选取 7 个特征点，如下图中红点所示。由于原图片太大，运行程序后图片显示不完整，无法选取特征点，所以先把两张图片分别缩小到 512*512 大小。



二、输出两幅图中对应点的坐标

```
ref: [133, 396]
src: [102, 392]
ref: [181, 439]
src: [152, 465]
ref: [343, 405]
src: [365, 497]
ref: [361, 380]
src: [394, 474]
ref: [183, 270]
src: [195, 255]
ref: [181, 354]
src: [174, 361]
ref: [168, 317]
src: [166, 306]
```

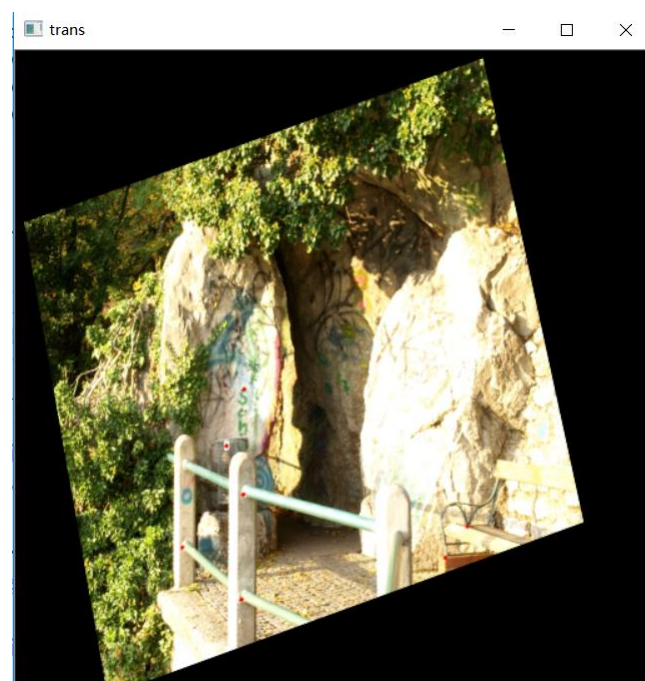
如上图所示，共选七对特征点，其中 ref 从 Image A 中选取，src 从 Image B 中选取。

三、计算转换矩阵

```
homo: [[ 7.20523393e-01  1.24656324e-01  7.04474479e+00]
 [-2.56248431e-01  7.00589457e-01  1.37366253e+02]
 [ 1.15379798e-05 -7.35506586e-05  1.00000000e+00]]
```

如上图所示，提供 OpenCV 的 findHomography 函数得到转换矩阵 H。

四、输出转换之后的图像



五、代码示例

```
import cv2
import numpy as np

img1 = cv2.imread('./Image A.jpg')
img2 = cv2.imread('./Image B.jpg')
img1_r = cv2.resize(img1, (512, 512), interpolation=cv2.INTER_CUBIC)
img2_r = cv2.resize(img2, (512, 512), interpolation=cv2.INTER_CUBIC)
ref = img1_r.copy()
src = img2_r.copy()

points1 = []
points2 = []

def mouse1(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.circle(ref, (x, y), 2, (0, 0, 255), -1)
        p = [x, y]
        cv2.imshow('ref', ref)
        points1.append(p)
        print('ref:', p)

def mouse2(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.circle(src, (x, y), 2, (0, 0, 255), -1)
        p = [x, y]
        cv2.imshow('src', src)
        points2.append(p)
        print('src:', p)

cv2.namedWindow('ref')
cv2.namedWindow('src')
cv2.imshow('ref', ref)
cv2.imshow('src', src)
cv2.setMouseCallback('ref', mouse1)
cv2.setMouseCallback('src', mouse2)
cv2.waitKey(0)
print('continue?')
a = input('random:')

pts1 = np.float32(points1).reshape(-1, 1, 2)
pts2 = np.float32(points2).reshape(-1, 1, 2)
homo, mask = cv2.findHomography(pts2, pts1, 0)
```

```
print('homo:', homo)
trans = cv2.warpPerspective(src, homo, ref.shape[0:2])

cv2.imshow('trans', trans)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

六. 心得体会

因为需要手动选特征点，所以需要对 **opencv** 如何检测点击鼠标这个动作以及检测后的回调函数的使用方法有一定了解才行。手动选取特征点之后通过 **opencv** 自带的 **findHomography** 函数求得 **H**，然后通过 **warpPerspective** 函数配准即可。所以如果遇到一些自己之前没有遇到过的问题，多 **google** 一下总可以找到解决方法。