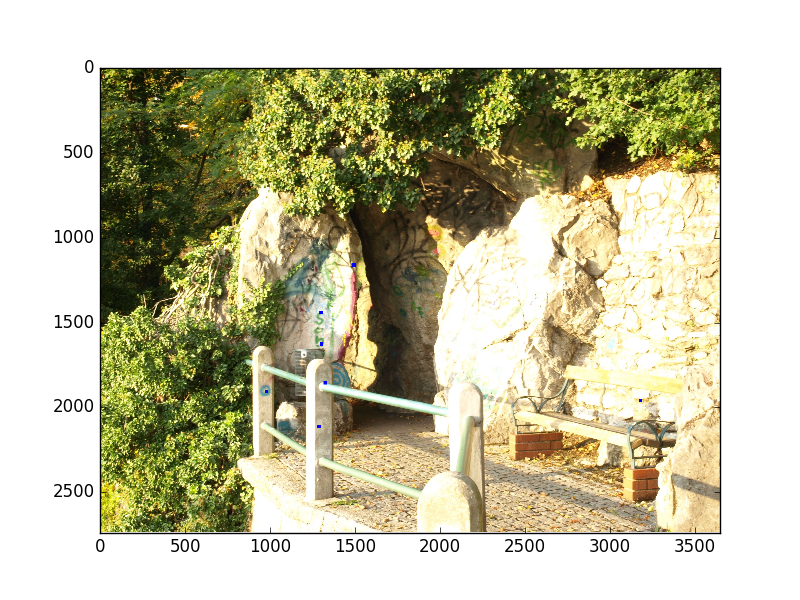
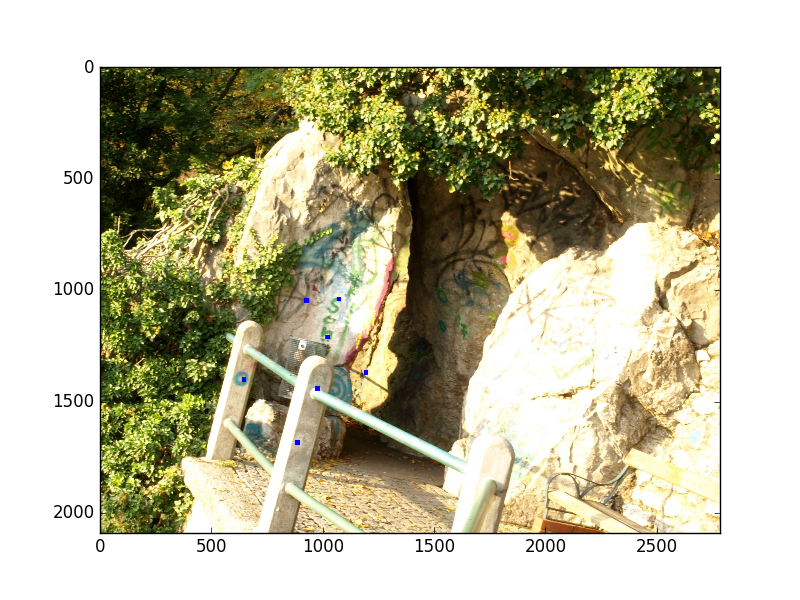
**图像配准作业说明**

**（以标注4个点为例）**

**一．手动标点：**

蓝点为标记点





**二．输出两幅图中对应点的坐标：**

图Ａ选取的坐标：

([[1963,3176,1],

[2117,1286,1],

[1448,1299,1],

[1860,1323,1],

[1632,1301,1],

[1168,1492,1],

[1911,977,1]]

)

图Ｂ依次对应的坐标

([[1371,1193,1],

[1684,886,1],

[1043,1071,1],

[1442,976,1],

[1212,1021,1],

[1049,926,1],

[1403,645,1]])

**三．计算转换矩阵：**

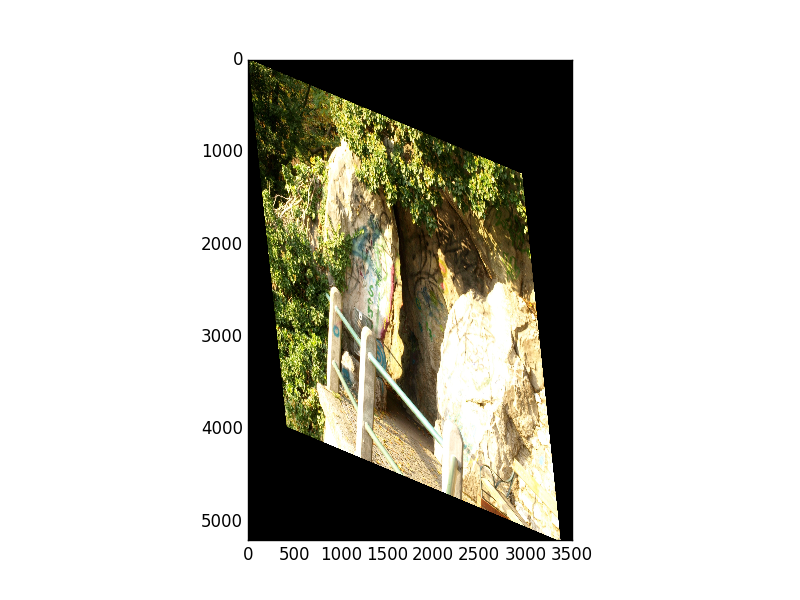
H=

[[ 1.05993237 0.19937949 -0.12700399]

[ 0.44537477 1.90374003 -0.72988393]

[ 0. 0. 1. ]]

**四．输出转换之后的图像：**



**五．代码示例：**

from \_\_future\_\_ import division, print\_function

from PIL import Image

from pylab import \*

import cv2

import copy

def highlight(coord, img):

im = copy.deepcopy(img)

for c in range(7):

x,y = coord[:,c][:2]

x=int(x)

y=int(y)

for i in range(-10,10):

for j in range(-10, 10):

im[x+i,y+j]=[0,0,255]

return im

img1 = Image.open('./Image A.jpg')

img2 = Image.open('./Image B.jpg')

img1 = np.array(img1)

img2 = np.array(img2)

Q=np.array([[1963,3176,1],[2117,1286,1],[1448,1299,1],[1860,1323,1]

,[1632,1301,1],[1168,1492,1],[1911,977,1]], dtype=np.float)

P=np.array([[1371,1193,1],[1684,886,1],[1043,1071,1],[1442,976,1],

[1212,1021,1],[1049,926,1],[1403,645,1]], dtype=np.float)

Q=Q.T

P=P.T

tmp1=np.matmul(Q,P.T)

tmp2=np.matmul(P,P.T)

tmp2=np.matrix(tmp2)

H=np.matmul(tmp1,tmp2.I)

print(H)

H = H[:2,:]

res = cv2.warpAffine(img2, H,(3500, 5200))

imshow(res)

show()

# im1=highlight(Q,img1)

# imshow(im1)

# show()

# im2=highlight(P,img2)

# imshow(im2)

# show()

**六．心得体会：**

图像配合中最关键的步骤是特征点的选取，这关系到仿射矩阵Ｈ的计算，也就决定了最后匹配的效果。而在本次作业中，由于特征点的选取是比较随性的，仅凭肉眼选定，所以即使是在像素数值上都存在着一定的差异，更不用说对于图像特征的表达水平了。所以，在做图像配准的工作当中，一定要精准找到特征点才行。