

西安交通大学

数字图像处理报告

图像配准

姓名 姜彦凯

班级 自动化少 61

学号 2140506083

提交日期 2019. 3. 5

# 数字图像处理-图像配准

自动化少 61 姜彦凯

西安交通大学电气学院 710049 陕西 西安

## 摘要:

本次报告利用 python 进行程序编写, 首先利用调用鼠标 API, 获得两张图共 14 个点的坐标, 方便之后计算转换矩阵。利用函数(或者矩阵运算), 得到转换矩阵, 最后对原图片和现在的图片进行转换从而得到了配准之后的图像

## 一. 手动标点:

利用 python 进行标定点, 调用鼠标 API 进行标点生成图像的坐标。



## 二. 输出两幅图中对应点的坐标:

### Image\_A points:

1197,1698

1303, 1449

2206, 2113

2066,1076

1813,914

1391,1844

1265,1718

### Image\_B points:

910,1254

1074,1042

1774,1913

1903,873

1704,653

1055,1445

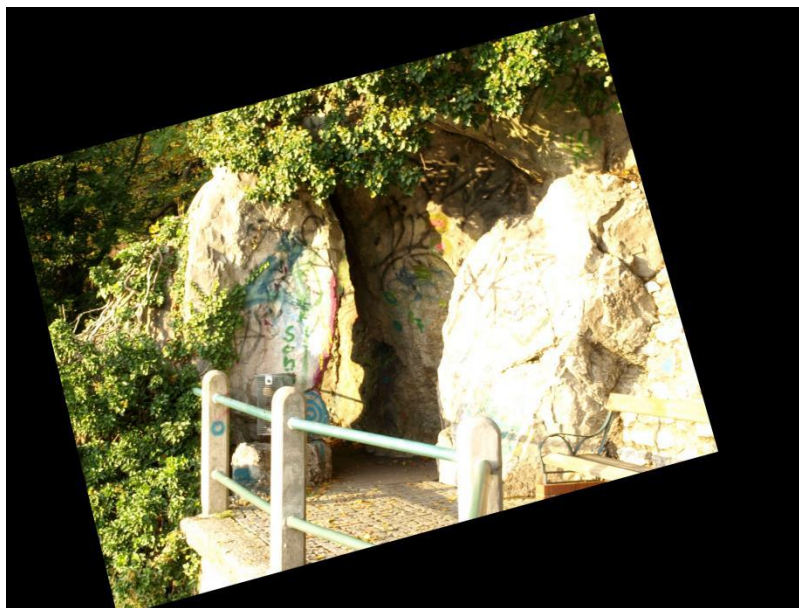
971,1292

### 三. 计算转换矩阵:

```
[[ 9.63924574e-01  2.26599556e-01  1.82037395e+01]
 [-2.52425495e-01  9.33668639e-01  7.32247849e+02]
 [ 4.11981284e-06 -1.43197617e-05  1.00000000e+00]]
```

### 四. 输出转换之后的图像:

设置输出图片大小为 3648\*2736, 之后再进行恢复即可



### 五. 代码示例:

```
1.         #-*- coding: utf-8 -*-
2.
3.
4.         import cv2
5.         import numpy as np
6.         from PIL import Image
7.         def gen_point(event,x,y,flags,param):
8.             if event == cv2.EVENT_LBUTTONDOWN:
9.                 print (x,y)
10.        img1 = cv2.imread("C:\\Users\\lenovo\\Desktop\\2\\Image A.jpg")
11.        cv2.namedWindow('image', 2)
12.        cv2.setMouseCallback('image',gen_point)
13.        while(1):
14.            cv2.imshow('image',img1)
15.            if cv2.waitKey(20) & 0xFF == 27:
16.                break
17.        cv2.destroyAllWindows()
18.        im = np.array(Image.open("C:\\Users\\lenovo\\Desktop\\2\\Image B.jpg"))
19.        dsize=(3648,2736)
```

```
20.     A = np.array([[1197,1698], [1303, 1449], [2206, 2113],[2066,1076],[1813,914],[1391,1844],[1265,1718]])
21.     B = np.array([[910,1254],[1074,1042],[1774,1913],[1903,873],[1704,653],[1055,1445],[971,1292]])
22.     h, s = cv2.findHomography(B, A, cv2.RANSAC)
23.     print(h)
24.     book = cv2.warpPerspective(im, h, dsize)
25.     book = cv2.cvtColor(book, cv2.COLOR_RGB2BGR)
26.     cv2.imwrite('LLL.jpg',book)
```

## 六. 心得体会:

因为大家都在使用 `matlab` 进行编程, 对于取点工作也是很好就可以进行, 在这次实验中我查阅了大量的资料, 最后在一篇博客上发现了一些处理手动取点的方法, 并加以实践, 实现了具体功能。然而在对于转换矩阵以及图像变换的工程中, 我尝试了从原理上对算法进行复现, 但是最后的结果并不尽人意, 校准图片的结果并不是很好, 而是产生了一些偏离。所以就使用了 `openCV` 自带的函数库, 并加以幅值最后便完成了最后的图像配准。总之, 这次实验让我的编程水平得到了明显的提高, 今后我也会继续努力。

## 七. 参考文献:

《数字图像处理(第三版)》 [美]冈萨雷斯 电子工业出版社

《OpenCV3 编程入门》 毛星云 电子工业出版社