数字图像处理图像配准作业

赵子瑞 自动化钱61班 2019年3月5日

摘要

本文是数字图像与视频处理的第二次作业,通过理论学习和代码实践,完成了以下任务:

通过手动标点,得到图像中7个位置的坐标,进行输出并保存,计算仿射变换矩阵,通过仿射变换矩阵计算得到变换后的图像,输出图像。

本次实验较为简单,建立在第一次实验的基础之上,提高了动手能力和对 图像处理的理论理解。

关键词: 图像配准 图像处理 仿射变换 矩阵计算

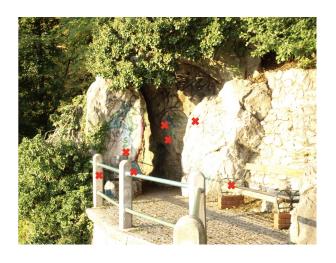
1 项目任务 2

1 项目任务

要求根据已给的两幅图像,在各幅图像中随机找出7个点,计算出两幅图像之间的转换矩阵H,并且输出转换之后的图像。注:已给图像分别为ImageA和ImageB。

2 手动标点

本次实验中标定的点如图1所示



(a) 原图像标定点选择



(b) 变换后图像选择

图 1: 标定点选择

3 输出两幅图中对应点的坐标

得到的坐标输出如下。

Image A 的特征点如下。

(977.693542, 1919.11292)

(1286.59680, 1624.91931)

(1374.85486, 1867.62903)

(1749.95166, 1316.01611)

(1786.72583, 1448.40320)

(2139.75806.1249.82263)

(2581.04834, 2022.08069)

Image B选取的特征点坐标如下。

(639.600830, 1397.73987)

(1015.80042, 1201.21777)

(1032.64514, 1453.88916)

(1537.98792, 1010.31049)

(1537.98792, 1161.91333)

(1931.03223, 1066.45972)

(2150.01416, 1925.54236)

4 计算转换矩阵

转换矩阵的计算可以由公式1得到:

$$H = Q \cdot P^T \cdot (P \cdot P^T)^{-1} \tag{1}$$

,其中,P为原图像的特征点组成的矩阵,Q为变换后对应的特征点组成的矩阵,H为变换矩阵。

经过计算,变换矩阵为:

5 输出在转换后的图像

```
\begin{pmatrix} 0.958704864 & -0.263552481 & 208.122426 \\ 0.267151126 & 0.967908948 & -720.247315 \\ 0 & 0 & 1 \end{pmatrix}
```

5 输出在转换后的图像

变换后的图像如图2所示。



图 2: 变换后的图像

6 代码示例

```
import cv2
import matplotlib.pyplot as plt
import base64
import struct
import numpy as np
from scipy import interpolate
from pylab import *
from PIL import Image

class basic_cv_tool:
```

7 心得体会 5

```
def __init__(self , ImageName):
12
                    self.ImageName = ImageName
13
           def ImageRead(self, ImageName):
                   img = cv2.imread(ImageName)
                    return img
           def interest_point_choosing(self, ImageName):
                   img = array (Image.open (ImageName))
                   imshow (img)
                    fea_point = ginput(7)
                    fea_point = np.float32(fea_point)
                    fea_point = np.column_stack((fea_point, \)
                    array([1,1,1,1,1,1,1])))
                    return fea-point
           def Getting_H_Matrix(self, img_points_1, img_points_2):
                    H_{\text{matrix}} = ((img_{\text{points}} - 2.transpose()) \setminus
                    .dot(img_points_1)).dot(np.linalg.inv( \ \ \ \ \ \ \ \ \ \ )
                    (img_points_1.transpose()).dot(img_points_1)))
                    print(H_matrix)
                    return H_matrix[:2]
  if __name__ = '__main__':
           image\_1\_name \ = \ `../../ \ homework2/Image\_A.jpg \ `
           image_2_name = '../../homework2/Image_B.jpg'
           tool = basic_cv_tool(image_1_name)
           image = tool.ImageRead(image_1_name)
           print(image.shape)
42
           img1 = tool.interest_point_choosing(image_1_name)
           print("interest_point_of_image_A_is",img1)
           img2 = tool.interest_point_choosing(image_2_name)
           print("interest_point_of_image_B_is",img2)
          M = tool. Getting_H_Matrix (img1, img2)
           img = cv2.warpAffine(image, M, (image.shape[1], image.shape[0]))
           print(img.shape)
           cv2.imwrite('test.jpg', img)
```

7 心得体会

本次作业,通过实践,我深入了解了图像配准的相关基础知识,同时也提高了动

7 心得体会 6

手能力,加深了我对仿射变换的理解和掌握程度,也让我拥有了解决实际问题的基本功。同时,代码能力也得到了有效锻炼,我的python能力得到了有效锻炼,同时也学习了opencv的很多基本功能。我自行组织的cvtoolbox在我的github上维护,欢迎访问https://github.com/1989Ryan/Digital-Image-Processing-Project进行了解.