

数字图像处理-图像配准

姓名：王宇桁

班级：自动化 61

学号：2160504021

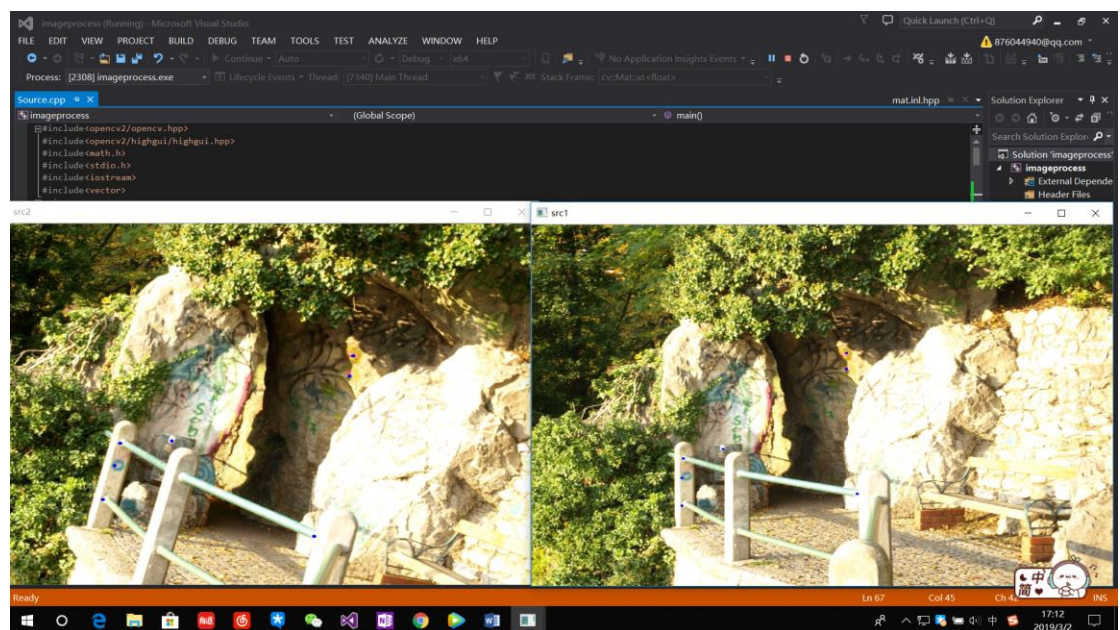
提交时间：2019/3/4

摘要：本次作业进行图像配准的处理工作，通过两张图片中对应的相同点，计算出仿射变换矩阵，再将图片进行变换从而实现图像配准。本次作业使用 C++ 和 Opencv 进行图像配准功能。

题目：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。

由于题目中要求使用鼠标在图中选取 7 个参考点，因此使用到了 opencv 中 mousecallback 函数以及鼠标事件等功能，通过鼠标的信息获取函数可以得到相应的点的位置信息，以下为实验内容：

一. 手动标点：



如图在图中选取 7 个相同的点，用于计算变换矩阵。

二. 输出两幅图中对应点的坐标：

其中，第一张图（左图）七个点的坐标分别为：

fixedPoints3 =

1270	644
1399	615
1595	556
1257	909
764	1834
881	1816
1809	1635

第二张图（右图）七个点的坐标分别为：

movingPoints3 =

1769	950
1917	942
2128	942
1700	1202
971	1976
1093	1976
2038	2043

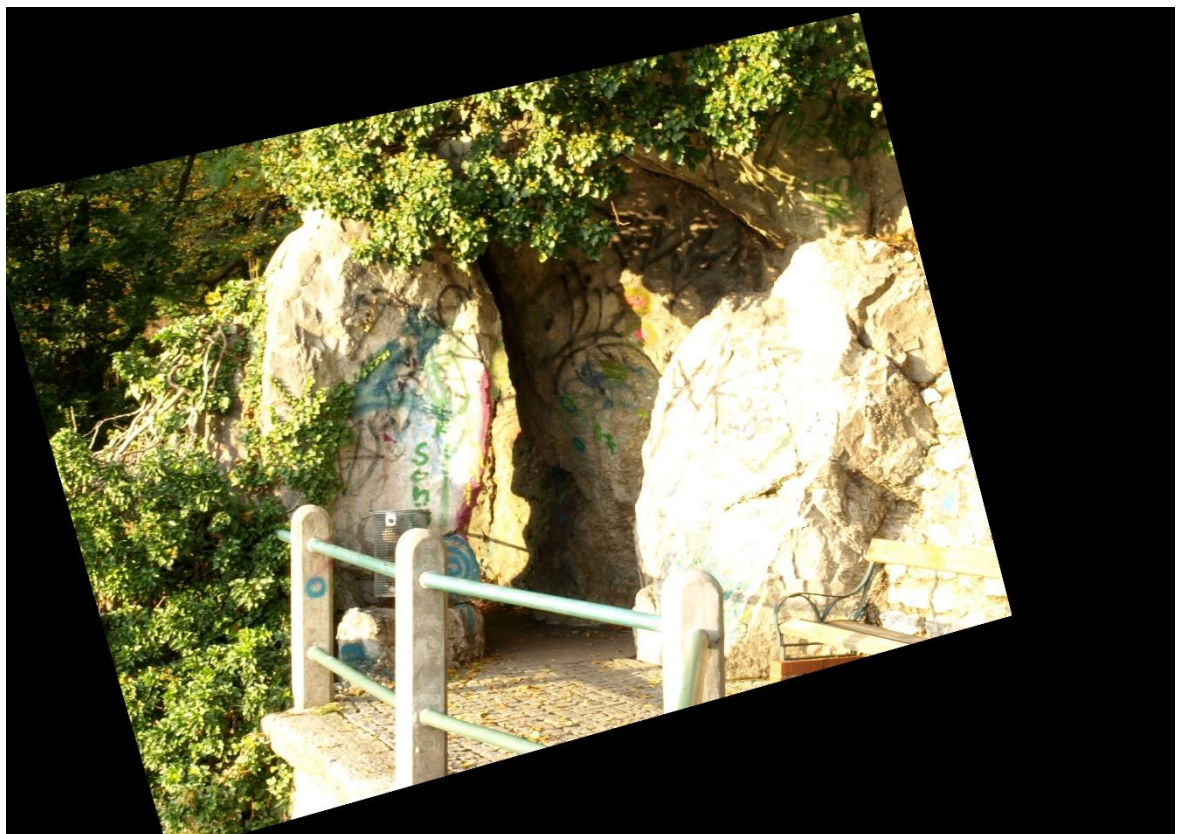
三. 计算转换矩阵:

由上面得到的 7 组对应点, 通过使用 Opencv 中的 `findHomography` 函数可以计算出 3×3 变换矩阵 h :

1.07695	0.28957	-82.9044
-0.220474	1.06449	634.286
3.44771e-005	1.21983e-005	1

四. 输出转换之后的图像:

通过 Opencv 中的 `warpPerspective` 函数可以将第一张图片进行旋转与第二张图进行对应, 其效果如图:



五. 代码示例:

```
#include<opencv2/opencv.hpp>
#include<opencv2/highgui/highgui.hpp>
```

```

#include<math.h>
#include<stdio.h>
#include<iostream>
#include<vector>
using namespace cv;
using namespace std;

Mat src1, src2;
Point curpoint = Point(-1, -1);
vector<Point>points1;
vector<Point>points2;
void onmouse1(int event, int x, int y, int flags, void*ustc)
{
    if (event == CV_EVENT_LBUTTONDOWN)
    {
        curpoint = Point(x, y);
        points1.push_back(curpoint);
        circle(src1, curpoint, 10, Scalar(255, 0, 0), CV_FILLED);
        waitKey(5);
        imshow("src1", src1);
    }
}

void onmouse2(int event, int x, int y, int flags, void*ustc)
{
    if (event == CV_EVENT_LBUTTONDOWN)
    {
        curpoint = Point(x, y);
        points2.push_back(curpoint);
        circle(src2, curpoint, 10, Scalar(255, 0, 0), CV_FILLED);
        waitKey(5);
        imshow("src2", src2);
    }
}

int main()
{
    src1 = imread("E:\\课件\\图像\\图像配准\\Image A.jpg");
    Mat copy1 = src1.clone();
    cvNamedWindow("src1",0);
    setMouseCallback("src1", onmouse1, 0);
    imshow("src1", src1);
    waitKey(0);
    imwrite("E:\\课件\\图像\\图像配准\\Image APoint.jpg",src1);
}

```

```

src2 = imread("E:\\课件\\图像\\图像配准\\Image B.jpg");
Mat copy2 = src2.clone();
cvNamedWindow("src2", 0);
setMouseCallback("src2", onmouse2, 0);
imshow("src2", src2);
waitKey(0);
imwrite("E:\\课件\\图像\\图像配准\\Image BPoint.jpg", src2);
Mat h=findHomography(points2, points1);
Mat trans;
warpPerspective(copy2, trans, h, copy1.size());
cvNamedWindow("trans", 0);
imshow("trans", trans);
for (int i = 0; i < h.rows; i++)
{
    for (int j = 0; j < h.cols; j++)
    {
        cout << h.at<double>(i, j) << ' ';
    }
    cout << '\n';
}
for (int i = 0; i < points1.size(); i++)
{
    cout << points1[i].y << ' ' << points1[i].x << '\n';
}

for (int i = 0; i < points1.size(); i++)
{
    cout << points2[i].y << ' ' << points2[i].x << '\n';
}
imwrite("E:\\课件\\图像\\图像配准\\trans.jpg", trans);
waitKey(0);
return 0;
}

```

六. 心得体会:

这一次的图像配准作业主要思路是通过提取两张旋转角度不同的图片中相对应的多组特征点,并根据他们在各自图片中的位置坐标信息进而求出图像变换矩阵,从而能将两个旋转角度不同的图像转换为一角度相同的图像,从而实现图像的配准工作。

通过作业,对上课所学的内容有了一个更清楚的认识。