

数字图像处理

图像配准

姓名：王浩然

班级：自动化 61

学号：2160504020

提交时间：2019/3/4

摘要：图像配准在医学等图像领域是一项重要的技术，图像配准是将两幅或多幅图像进行匹配、叠加的过程，通过两张图片中对应的相同点，计算出放射变换矩阵，再将图片进行变换的过程。本次试验用 C++ 和 OpenCV 库进行图像配准功能。

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。

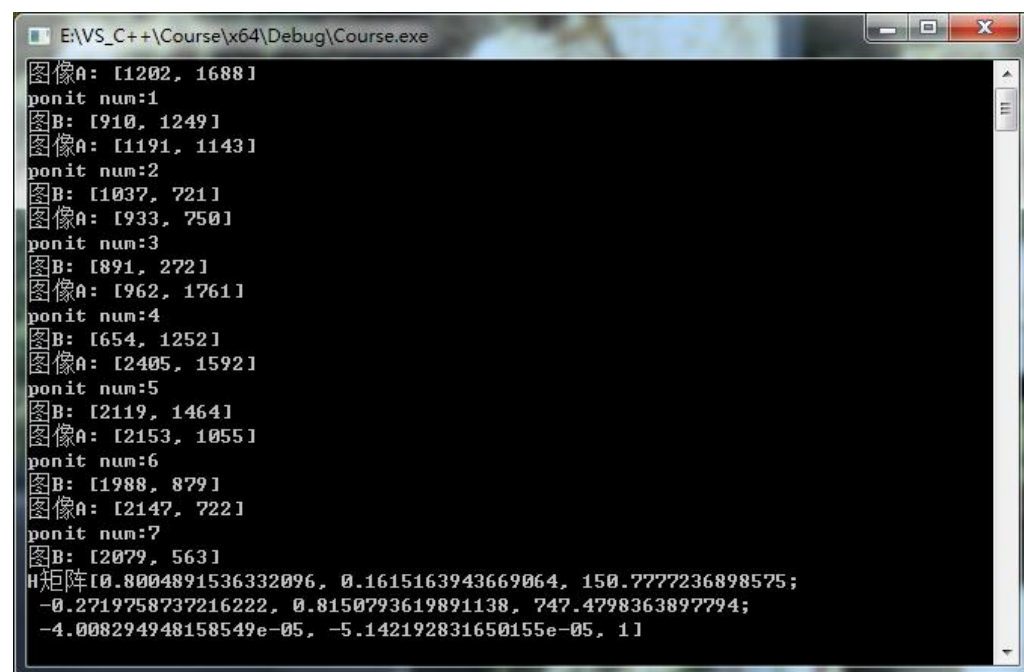
注：已给图像分别为 Image A 和 Image B。

图像配准的具体原理就是将图片中选中的相互匹配的点在两张图片 Image A 和 Image B 的坐标信息取出，经过计算变换矩阵 H ，将图片进行坐标变换的过程。

具体实现过程中由于要手动选择 7 点，故用到了事件 `CV_EVENT_LBUTTONDOWN`，当鼠标左键点击 Image A 或者 Image B 时，记录下当时鼠标位置即图片坐标并用 `circle` 函数在点击处定义一个红色的点标记点击位置，依次点击两张图片，将坐标信息打印出来，并将数据依次存入数组，根据两张图片的坐标信息用 `findHomography(Point2, Point1, CV_RANSAC)`；计算出 H 矩阵，再利用 H 矩阵和 `warpPerspective` 进行图片变换即可。

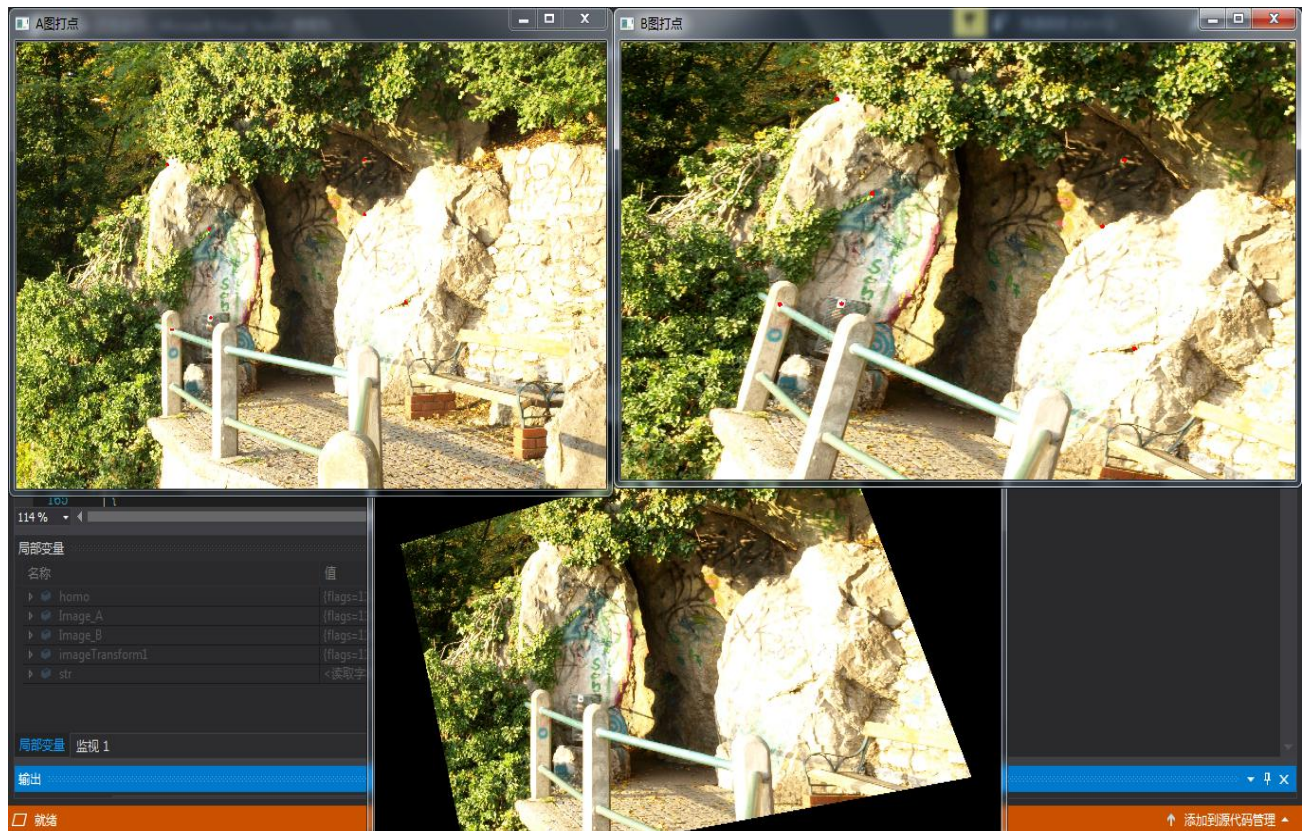
实验结果：

一，两张图片中对应点的坐标，以及 H 矩阵的计算结果：



```
E:\VS_C++\Course\x64\Debug\Course.exe
图像A: [1202, 1688]
point num:1
图B: [910, 1249]
图像A: [1191, 1143]
point num:2
图B: [1037, 721]
图像A: [933, 750]
point num:3
图B: [891, 272]
图像A: [962, 1761]
point num:4
图B: [654, 1252]
图像A: [2405, 1592]
point num:5
图B: [2119, 1464]
图像A: [2153, 1055]
point num:6
图B: [1988, 879]
图像A: [2147, 722]
point num:7
图B: [2079, 563]
H矩阵[0.8004891536332096, 0.1615163943669064, 150.7777236898575;
-0.2719758737216222, 0.8150793619891138, 747.4798363897794;
-4.008294948158549e-05, -5.142192831650155e-05, 1]
```

二，手动选择的点（红色）以及



三，代码示例

```
#include "stdafx.h"

#include <opencv2\opencv.hpp>

#include <opencv2\dnn.hpp>

#include <iostream>

using namespace std;

using namespace cv;

vector<Point2f> Point1, Point2;

Mat A_clone, B_clone;

int pnumber = 0;

void A_click(int event, int x, int y, int flags, void *ustc)
{
    if (event == CV_EVENT_LBUTTONDOWN)
```

```

{
    Point p = Point(x, y);

    circle(A_clone, p, 10, Scalar(0, 0, 255), -1);

    Point1.push_back(p);

    cout << "图像 A: " << p << endl;

    pnumber++;

    cout << "point num:" << pnumber << endl;

}
}

```

```

void B_click(int event, int x, int y, int flags, void *ustc)
{
    if (event == CV_EVENT_LBUTTONDOWN)
    {
        Point p = Point(x, y);

        circle(B_clone, p, 10, Scalar(0, 0, 255), -1);

        Point2.push_back(p);

        cout << "图 B: " << p << endl;

    }
}

```

```

int main()
{
    Mat Image_A, Image_B, homo, Transform;

    Image_A = imread("G:\\图像视频处理\\图像配准\\图像配准\\Image A.jpg");
    Image_B = imread("G:\\图像视频处理\\图像配准\\图像配准\\Image B.jpg");

    A_clone = Image_A.clone();
    B_clone = Image_B.clone();
}

```

```

namedWindow("A",0);

namedWindow("B",0);

imshow("A", A_clone);

imshow("B", B_clone);

setMouseCallback("A", A_click);

setMouseCallback("B", B_click);


waitKey();


homo = findHomography(Point2, Point1, CV_RANSAC);

cout << "H 矩阵"<<homo;

warpPerspective(Image_B, Transform, homo, Size(Image_A.cols, Image_A.rows));


namedWindow("A 图打点", 0);

namedWindow("B 图打点", 0);

namedWindow("变换图", 0);

imshow("A 图打点", A_clone);

imshow("B 图打点", B_clone);

imshow("变换图", Transform);


waitKey();

return 0;

}

```

四，心得体会

在实验过程中有不清楚如何区分两张图片与鼠标的事件结合的区分的时候，查到了 `setMouseCallback` 函数，顺利的解决了问题，以及点了后不显示的问题，也是在加了 `namedWindow` 控制窗口函数后才得到解决，OpenCV 库实在是解决而太多复杂的问题。