

## 计算机视觉作业三

姓名：李宁

学号：2120180459

专业：计算机科学与技术

### 问题一

#### 一、问题描述

给定两张图片 HW3Pic1.jpg 和 HW3Pic2.jpg。要求按照一定原理将两张图片拼接为一张图片。具体要求如下。

- 1、介绍所采用的拼接方法的拼接原理；
- 2、给出评估拼接结果质量的评价标准；
- 3、实现拼接算法并显示拼接结果图像；
- 4、对拼接结果给出分析说明；

#### 二、基本思路

##### 1、图片拼接问题

如果要想实现两张图片的简单拼接，首先要找到两张图片中多个相似的点（至少四个），然后通过这些相似的特征点计算一张图片可以变换成另一张图片的变换矩阵（homography 单应性矩阵），通过该矩阵，可以把一张图片变换后放到另一张图片的相应位置，即将两张图片中相似的相似点重合在一起，从而对两张图片实现简单的全景拼接

##### 2、图像融合问题

这里的图像融合，是在实现图像的简单拼接之后，对图片重叠部分像素值的重新计算，从而使拼接图像看起来有一个很好的平滑过渡。

对于计算重叠区域的像素值，本次实验通过实现一个图像的线性渐变这种简单粗暴的方式来完成。对于重叠区域，靠近左边的部分，让左边图像内容显示多一些，靠近右边的部分，让右边图像的内容显示多一些。假设 $\alpha$ 表示像素点横坐标到左右重叠秋雨边界横坐标的距离，则新的像素值可以由下列公式表示出来：

$$newpixel = left\_pixel \times (1 - \alpha) + right\_pixel \times \alpha$$

如此，便实现了一个简单的图像融合效果。关于其他一些更好的融合方法如 multi-band 融合这里不再一一讲解。

##### 3、图像拼接质量评价方法

一般图像融合质量的评价方法分为主观和客观两类：主观方法呵客观方法。

其中客观方法有分两类：无参考图像评价方法（如信息熵）和有参考图像评价方法（如均方根误差）。

本次实验中，采用的是计算图像信息熵的方法，图像的信息熵的计算方法如下。

$$E = - \sum_{i=0}^{L-1} p_i \log_2 p_i$$

其中  $L$  表示图像的灰度级别， $P_i$  表示灰度值  $i$  像素占总像素比例。 $E$  越大表示融合图像信息量越大。

### 三、原理及算法

综上所述思路所述，完成两张图片较好的融合大致需要两个步骤：

- ① 找到两张图片中的相似特征点，计算变换矩阵；
- ② 变换一张图片到另一张图片的合适位置，重新计算重叠区域新的像素值；

具体的算法流程如下：

- 1) 读取两张图片 HW3Pic1.jpg 和 HW3Pic2.jpg，分别赋值给变量 `img_src1` 和 `img_src2`；
- 2) 分别对两张图片向上、下、右三个方向进行扩展补 0，其中向右扩展为图像的宽；
- 3) 将两张图片转换为灰度图，提取并描绘出相似的特征点（ORB）和特征点描述符；
- 4) 匹配两张图片相应的特征点，根据这些点获取 Homograph 单应性图像变换矩阵  $M$ ；
- 5) 对 `img_src2`（程序中为 `pt_src2`）即 HW3Pic2.jpg 进行单应性图像变换，获得变换后的图像 `img_wrap`；
- 6) 对两张图片 `img_src1` 和 `img_wrap` 按列进行遍历，对列像素存在与否进行‘与’操作，获得重叠部分的左边界 `left` 和右边界 `right`；
- 7) 对 `img_src1` 和 `img_wrap` 按像素位置进行遍历，对于边界 `left` 左边的部分赋图像 `img_src1` 的像素值，对边界 `right` 右边的部分赋图像 `img_wrap` 的像素值，对于中间的部分，进行图像线性渐变操作；
- 8) 获取融合后的图片，计算融合图片的信息熵。

### 四、实验结果与分析

#### 1、实验结果

如下，是两张原图与融合图片的展示结果。



图 1 HW3Pic1.jpg



图 2 HW3Pic2.jpg



图 3 融合图像

融合图像的信息熵为: 7.497983503276919

```
g' '--client' '--host' 'localhost' '--port' '55517' 'e:\Project
The information entropy of the result is 7.497983503276919
PS E:\Project\VSCodeProject>
```

图 4 信息熵结算结果图

## 2、实验结果分析

本次实验在各个不同的图像处理阶段分别给出了结果显示。

1) 原图像转换为灰度图进行拼接操作, 如图 5。

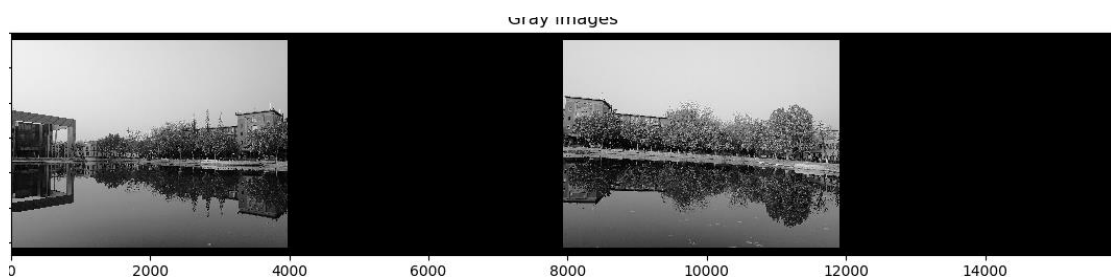


图 5 扩展后图像灰度图拼接结果

2) 获得 ORB 特征点后的图像拼接结果, 如图 6 所示。



图 6 提取特征点后的拼接图像

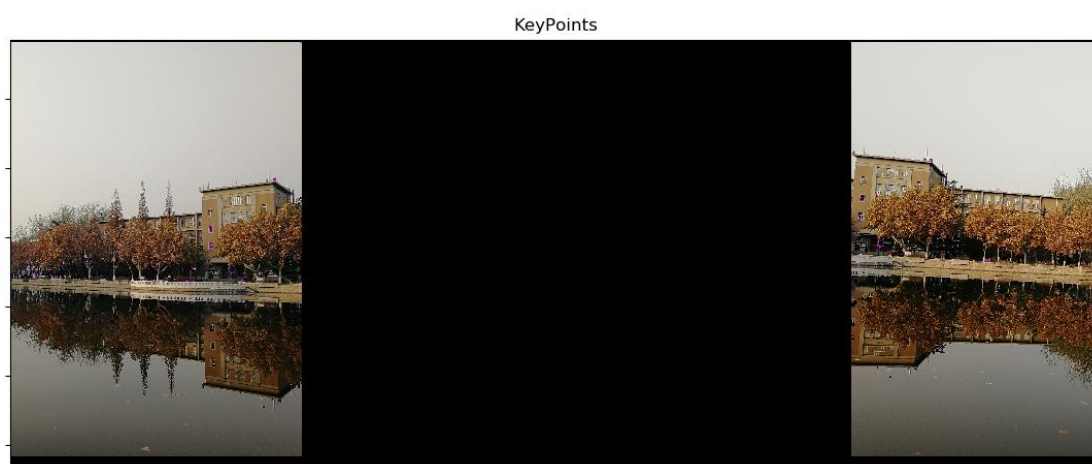


图 7 图 6 的放大结果（粉色点为特征点）

这里，实验中所采用的特征点提取方法是 ORB 算子提取方法，使用该算子的原因：第一是因为 SURF 和 SIFT 算子存在版权问题；第二是因为 ORB 算子较其他特征算子速度快。

ORB 采用 FAST 算法来检测特征点。FAST 核心思想就是找出那些卓尔不群的点，即拿一个点跟它周围的点比较，如果它和其中大部分的点都不一样就可以认为它是一个特征点。

FAST 具体计算过程如下：

- ① 从图片中选取一个像素点  $P$ ，要判断它是否是一个特征点。首先把它的密度（即灰度值）设为  $I_p$ 。
- ② 设定一个合适的阈值  $t$ ：当两个点的灰度值之差的绝对值大于  $t$  时，认为这两个点不相同。
- ③ 考虑该像素点周围的  $N$  个像素。
- ④ 如果  $N$  个点中有连续的  $n$  个点都和  $I_p$  不同，则认为  $P$  点是一个角点。

而 ORB 算子在 FAST 算法的基础上减少了上述④中  $n$  的个数，从而进一步加快了速度。

3) 特征匹配效果图，如图 8 所示。

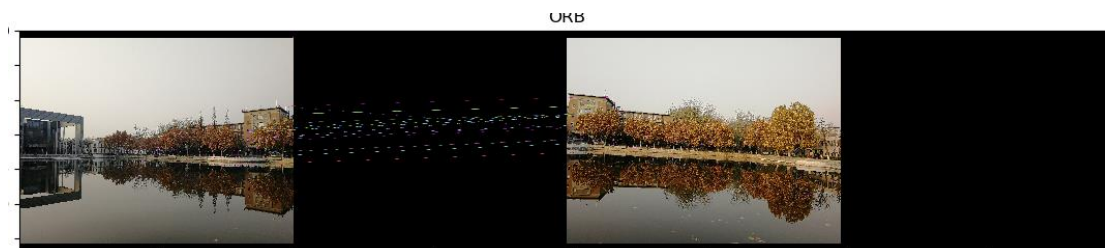


图 8 ORB 特征匹配效果图

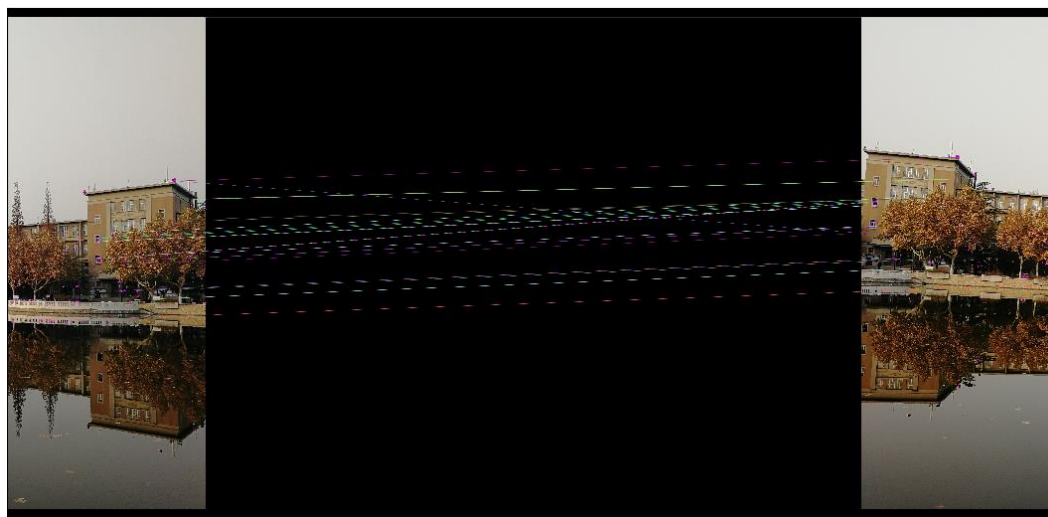


图 9 图 8 匹配效果放大图

4) HW3Pic2.jpg 经过单应性变换矩阵处理后的效果，如图 10 所示。



图 10 Hw3Pic2 单应性变换效果图

综合所有实验分析可知，完成两张图片较好融合的关键有两点：

- ① 有较好的方法获取单应性变换矩阵，将一张图片变换到另一张图片的相应位置；
- ② 两张图片位置对应后，对它们交叉重叠的部分能够进行较好的融合处理，本次作业中使用了简单的线性渐变方法来实现该操作。

