

计算机视觉 课程实验报告

学号：201900130151	姓名：莫甫龙	
-----------------	--------	--

实验题目：图像拼接

实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

这个实验在 python 中可以直接调用 stitcher 这个类来实现，并且可以进行多图拼接。实现效果如下：



然后我自己也实现了一下 stitcher 的过程，首先先计算出两张图的关键点和特征向量

```
def detectAndDescribe(self, img):  
    sift = cv.xfeatures2d.SIFT_create()  
    (kps, des) = sift.detectAndCompute(img, None)  
    kps = np.float32([kp.pt for kp in kps])  
    return (kps, des)
```

接着获得理想的匹配点，然后筛选掉一部分，再利用 opencv 自带的函数来实现 RANSAC 然后计算出单应性矩阵。

```

def matchKeyPoints(self, kp1, kp2, des1, des2, ratio, reprojThresh):
    matcher = cv.DescriptorMatcher_create('BruteForce')
    matches = matcher.knnMatch(des1, des2, 2)

    # 获取理想匹配
    good = []
    for m in matches:
        if len(m) == 2 and m[0].distance < ratio * m[1].distance:
            good.append((m[0].trainIdx, m[0].queryIdx))

    # 透视变换
    if len(good) > 4:
        # 获取关键点的坐标
        src_pts = np.float32([kp1[i] for (_, i) in good])
        dst_pts = np.float32([kp2[i] for (i, _) in good])

        # 通过两个图像的关键点计算变换矩阵
        (H, mask) = cv.findHomography(src_pts, dst_pts, cv.RANSAC, reprojThresh)

        # 返回最佳匹配点、变换矩阵和掩模
        return (good, H, mask)
    # 如果不满足最少四个 就返回None
    return None

```

接着就是进行图像拼接，然后输出：

```

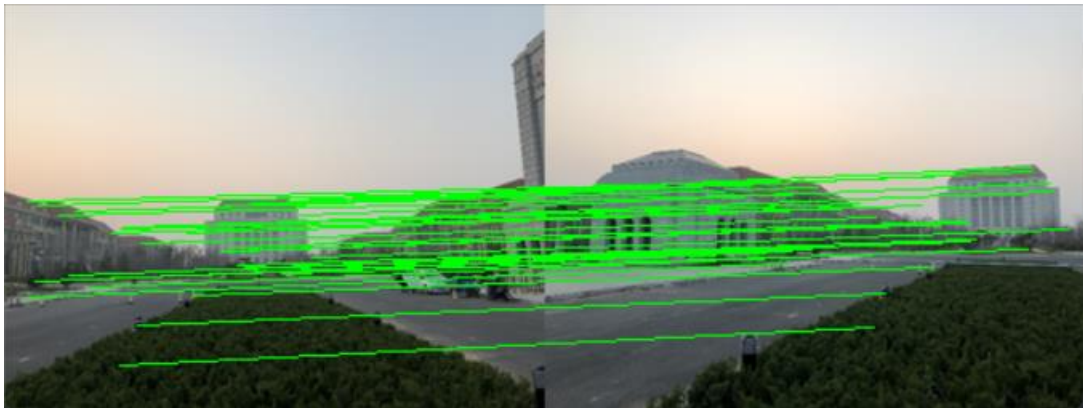
(good, H, mask) = R
# 对img1透视变换，M是ROI区域矩阵，变换后的大小是(img1.w+img2.w, img1.h)
result = cv.warpPerspective(img1, H, (img1.shape[1] + img2.shape[1], img1.shape[0]))
# 将img2的值赋给结果图像
result[0:img2.shape[0], 0:img2.shape[1]] = img2

if showMatches:
    vis = self.drawMatches(img1, img2, kp1, kp2, good, mask)
    return (result, vis)

return result

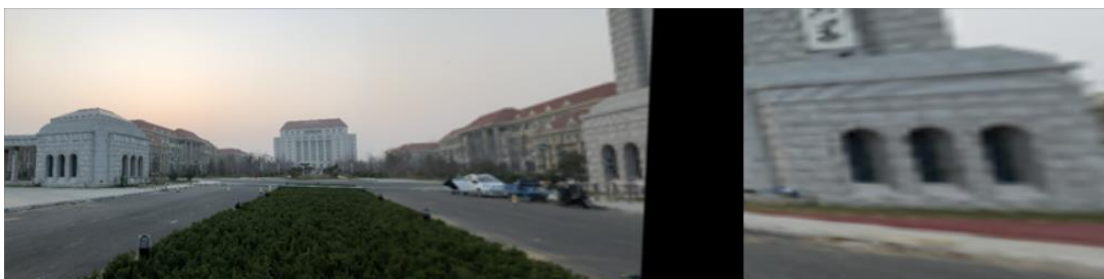
```

结果如下：



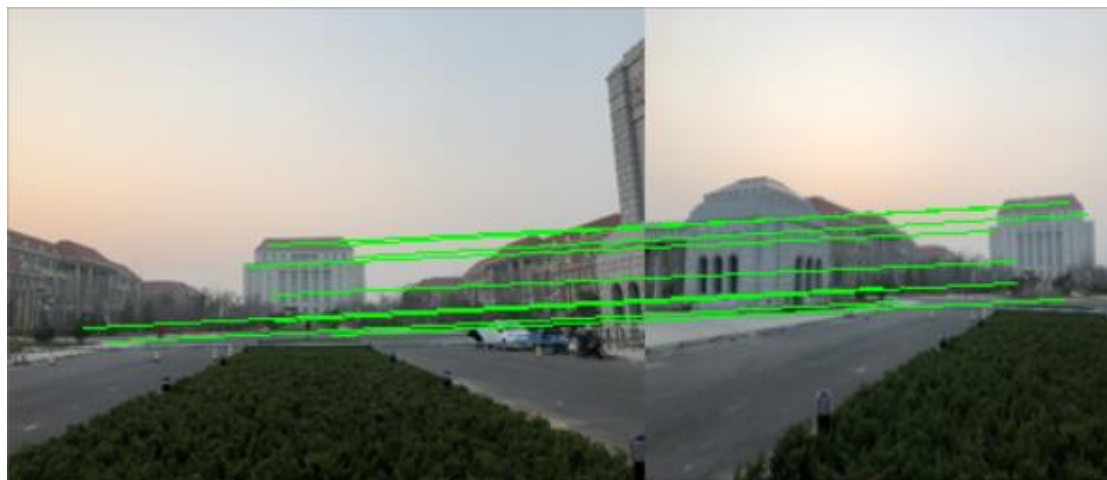


但是如果进行三张图的拼接，就会出现以下情况：



可以看到实现效果十分差，但是特征点的匹配是对应上的，后面尝试了很多都无法解决。我认为这可能是因为第三副图和前面得到的结果图有旋转的关系。

对于不同尺寸的图片（即相机位移），是可以很好地解决的：





#### 结果分析与体会：

这个实验实现了两张图的拼接，但是没有实现多张图的。