

# 计算机视觉第六次作业

朱明杰 15331441

## 一. 实验任务

1. 实现全景图像缝合

## 二. 实验工具

Visual studio 2017  
CImg Library  
VIFeat

## 三. 算法流程

1. 球坐标系映射。

设直角坐标下图像坐标点为 $(x, y)$ ，球坐标下图像坐标点为 $(x', y')$ ，相机焦距为 $f$ ，伸缩因子为 $s$ （为简单起见可以设 $s = f$ ），那么球坐标系映射公式为：

$$x' = s \arctan \frac{x}{f}$$
$$y' = s \arctan \frac{y}{\sqrt{x^2 + f^2}}$$

在不知道相机参数的情况下， $f$ 的求得是非常困难的。解释具体可以见[这里](#)。好在第二个数据集中的图片保留了照片的 EXIF 信息，查看如下：

图片名称: 1.JPG	快门速度: 1/92s
图片路径: D:/college/Junior/计算机视觉...	光圈值: F3.1
图片类型: JPG	焦距: 8mm
图片尺寸: 1536 × 2048	感光度: ISO100
文件大小: 1.04MB	曝光补偿: 0.0
拍摄时间: 0000:00:00 00:00:00	闪光灯: 关
相机厂商: NIKON	白平衡:
设备型号: E990	拍摄模式: 自动曝光
相机镜头:	测光模式: 加权测光
快门次数:	

当然，要实现这个变换的话，我们还是要用到反向插值算法，先要求出上述变换的逆变换：

$$x = f \tan \frac{x'}{s}$$
$$y = f \frac{\tan \frac{y'}{s}}{\cos \frac{x'}{s}}$$

先利用正变换估计结果的图像范围，然后利用逆变换进行反向插值。在代码中，我们实现了该步骤。

可选方案：柱坐标变换，正向变换为

$$x' = s \arctan \frac{x}{f}$$
$$y' = s \frac{y}{\sqrt{x^2 + f^2}}$$

逆变换为

$$x = f \tan \frac{x'}{s}$$
$$y = f \frac{\frac{y'}{s}}{\cos \frac{x'}{s}}$$

## 2. 特征点检测——SIFT 特征。

SIFT 特征是图像的一个非常重要的局部特征。其利用不同尺度的高斯差分卷积核来构造高斯差分尺度空间，通过极值点检测与主方向分配来生成主方向。这里我们直接用开源库 VIFeat 的 SIFT 特征提取器来实现就可以了。对于每个 CImg 图像，我们需要把图像上的点转成 vl\_sift\_pix\*格式，然后利用滤波器 vlSiftFilt\*来实现 SIFT 特征的提取。每个 SIFT 特征被描述成一个 128 维的向量。

## 3. 两幅图像的 SIFT 特征点匹配。

对于两幅图像，先各自提取它们的 SIFT 特征点，然后将这些特征点进行配对。具体匹配方法可以用最近邻来实现。对于第一幅图像的每一个 SIFT 特征点，我们找第二幅图像的 SIFT 特征（128 维向量）距离最小的那个特征点，其中特征距离可以取成欧氏距离。最近邻问题的经典算法就是利用 kd 树来解决：先以第二幅图像的所有 SIFT 特征点构造一棵 kd 树，然后对于第一幅图像的每一个 SIFT 特征点，我们利用分支定界算法来找出该点的最近邻。VIFeat 库提供一个叫做 vlKDForest\*的数据结构来实现 kd 森林。为了简单起见，我把森林中树的个数设置成了 1。当然，不是所有最近邻都是好的。有些点就算跟它的最近邻匹配起来都很牵强。这里我参考了 [VIFeat MATLAB Toolbox](#) 的做法：当特征点和其最近邻的距离乘上个阈值不能大于其跟其他特征向量的距离时，匹配才算可取。阈值的默认值为 1.5。

## 4. 利用 RANSAC 求得变换矩阵。

在一般情况下，变换是一个射影变换，变换的矩阵是一个 3×3 的具有 8 个自由变量的矩阵。一对点可以产生两个方程，利用四对点就可以确定一个变换了。这一套方法和之前某次作业中求射影变换的做法一模一样。确定这个变换后，对上一步骤中的配对点逐一进行检验，将符合要求（距离不大于某个阈值，这里阈值设为 6）的点对最多的那个变换作为最终结果。

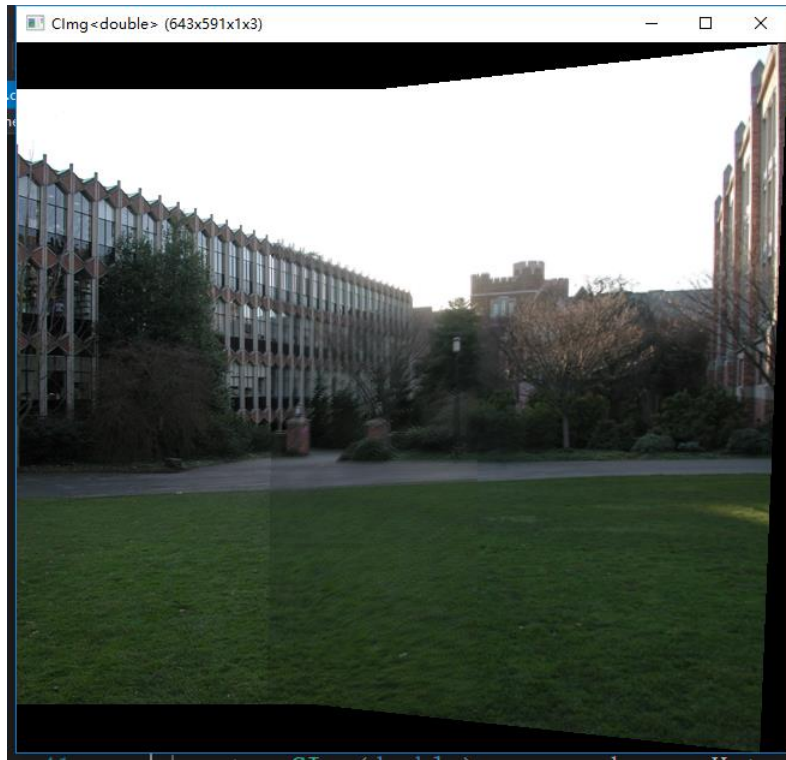
在没有做球坐标变换或柱坐标变换的时候，变换是一个射影变换；但是在做了球坐标变换或柱坐标变换之后，变换期望成为一个平移变换。

为方便起见，RANSAC 的试验次数设为 1000 次。

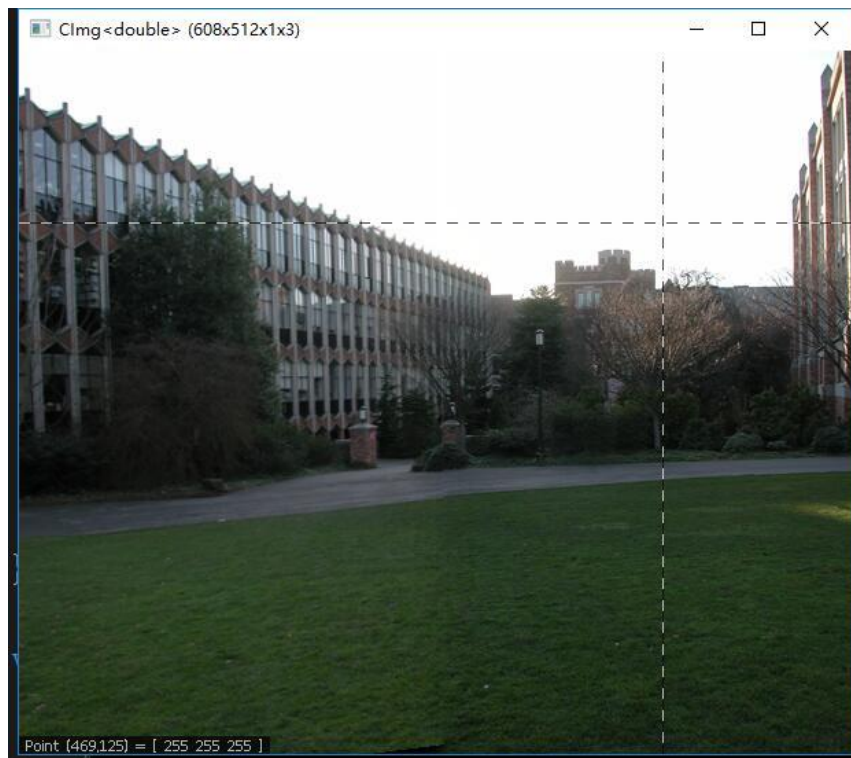
事实上，在 [VIFeat 的 sift-mosaic-code \(MATLAB\)](#) 中，做完 RANSAC 求得变换矩阵后，还存在着一个矩阵参数最优化的步骤：将最佳变换矩阵下对应的点都筛出来，将它们之间的距离和作为残差，来求一个使得残差最小化的矩阵作为参数优化后的矩阵。这里我们对射影变换没有做这一步，对求坐标变换或柱坐标变换做了这一步。

## 5. 图像的最终缝合。

这一步实现的最简单做法是将两个图像的非重叠部分设置成各自的部分，将重叠部分设置成两个图像各自的一半相加。不过这可能会造成存在两个明显边界的现象，给缝合后图像带来不自然的感觉：



这里，我们使用基于老师发给我们的论文《A Multiresolution Spline With Application to Image Mosaics》所实现的算法。在使用算法的时候，我们需要注意到：仅仅对两幅图像的重叠部分使用算法就好，其他各自的部分还是按照各自的来。我们把两幅图像的重叠部分的中线定为分界线，使用这个多分辨率的缝合算法就行了。这样我们的边界线就不那么明显：



虽然边界线看上去不是很明显，但是由于拍照时候的光照变化，使得图像看上去还是由多个图像缝合成的。可能事先要通过同态滤波的方法减小图片之间由光照变化所带来的亮度差异。

值得指出的是，之前参考链接所实验的最优缝合结果并不是基于这个多分辨率的缝合算法，而是 Smart-Cut+Feathering。具体这两种方法是干啥的，之后有空再说。

#### 四. 算法代码（CImg）实现

见压缩包下代码。

#### 五. 实验结果

缝合图像见工程文件下的 Dataset/1/stitch.bmp。

#### 六. 分析与评价

之前老师的意思是要在事先不告诉图像顺序的情况下，来做图像的缝合。如果是这样的话，那么图像缝合的整体算法框架不变，只是要先两两算出 RANSAC 方法的匹配正确率 ( $O(n^2)$ 次计算)，然后再将整个问题看成一个图，每个图像看成一个点，每对图像的匹配看成一条边，边权即为匹配正确率，这个图的 Hamilton 路径即为图像的缝合序列。可以实现这个过程，但是 Hamilton 路径求解是一个经典的 NP-完全问题，没必要平白无故增加算法的复杂度，而且图像顺序是不难获知的。