计算机视觉和模式识别 作业5

*13331231*

*孙圣*

*计应2班*

一、使用说明

MAC OSX系统：通过sh execute.sh直接编译运行即可（需要安装opencv）,默认对dataset第1张图片进行分离前后景。要对其他图片进行测试，执行./a.out dataset/X即可。

Windows系统，在cmd中输入execute.bat对第1张图片进行分离前后景。或者执行execute.exe dataset/X.jpg。

二、实验过程

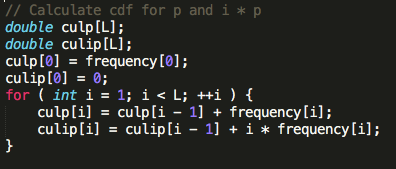
这次的实验相对简单，只需要按照OTSU算法一步步实现即可。

首先将图片读入，转换为灰度图。之后根据opencv官网提供的教程绘制灰度直方图并连接成线，便于验证所得到的阈值是否正确。

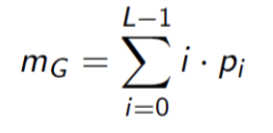
OTSU算法：

首先遍历图像中的每一个像素点，统计出0-255各个灰度出现的次数，再除以总的像素点，得到相应的频率。

由于需要对0-255之间的灰度逐个进行遍历，选出能够使得类间方差最大的灰度值，在这其中需要不断对p和i\*p求和加总。因此考虑计算一个累积分布函数(cdf)来减少计算的时间，达到用少量空间换取时间的效果，时间复杂度从O(n^2)降为O(n)。

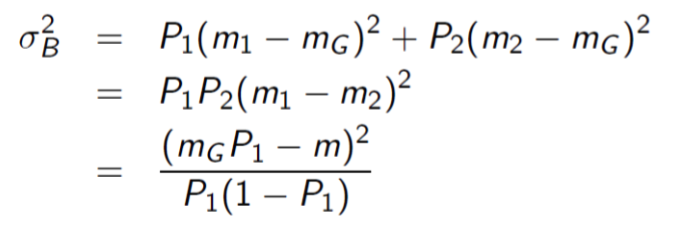
因此定义两个大小为256的数组，用来保存累积分布函数，从1开始遍历，每次迭代数组的值都为前一个值加上该像素点所求得的值：

之后，对于每一个灰度进行迭代，根据公式，mG为



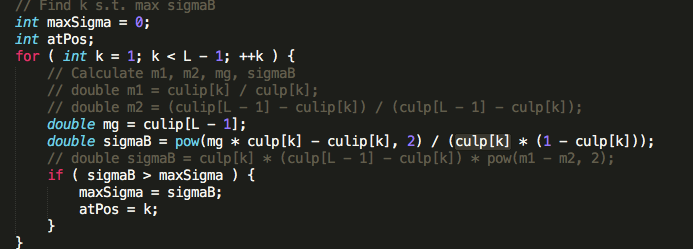
而根据之前的定义，mg其实就是culip[L - 1]

接下来，需要计算类间方差，公式如下：



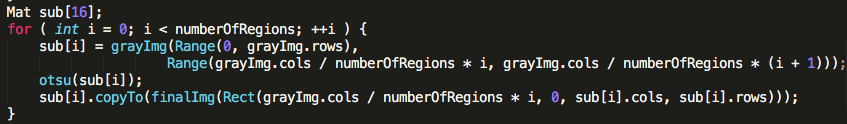
这里，m为culip[k]，P1为culp[k]，都可以在O(1)的时间内得到。

最后再判断是否是最大的方差，进行更新：



完成了基本的OTSU算法之后，对各个图像进行测试，发现大部分图像都能成功地分离前后景，除了第3和14张因为存在多个波峰导致部分信息显示不清。

因此考虑对图像进行分割（水平或竖直），分割成小的部分进行OTSU之后再拼接起来。竖直分割部分的主要代码如下：



经过这样的分割处理之后，效果有所提升，具体分析见实验结果部分。

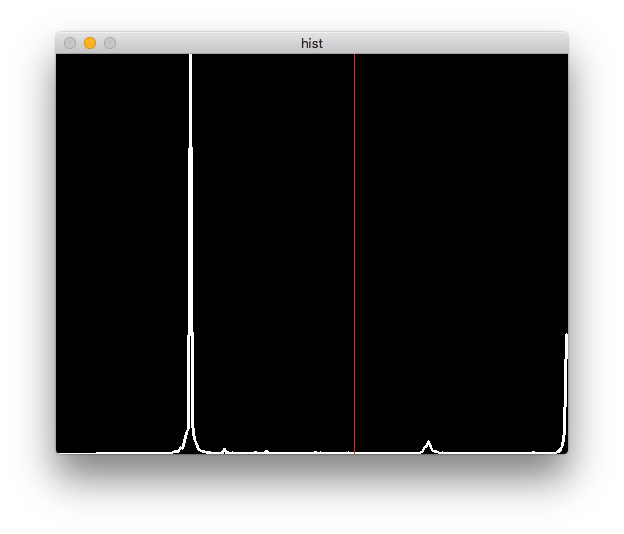
三、实验结果

大部分图像都能准确分离前后景，只有图3和图14会出现问题，先将图片切割后再使用OTSU能够改善相应的问题。

1.jpg：



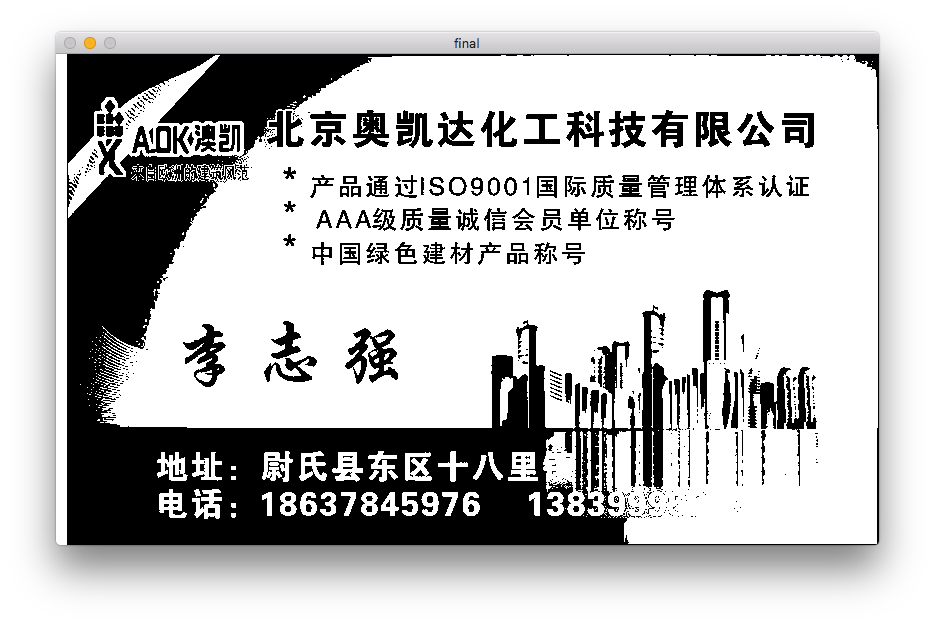
直方图为标准的双峰，因此能准确划分：



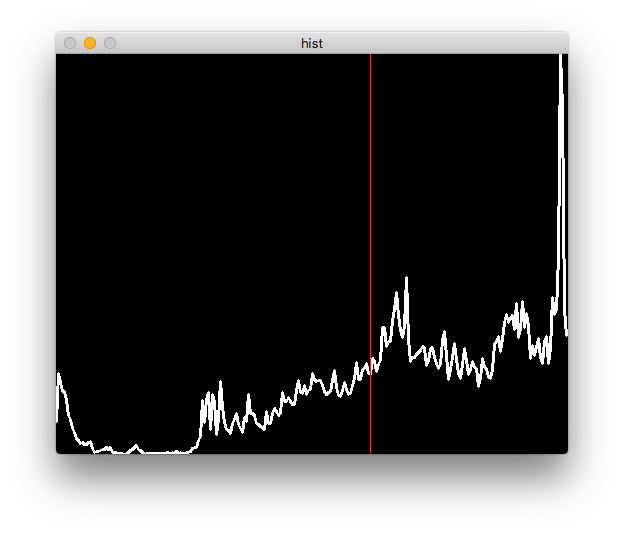
2.jpg



3.jpg

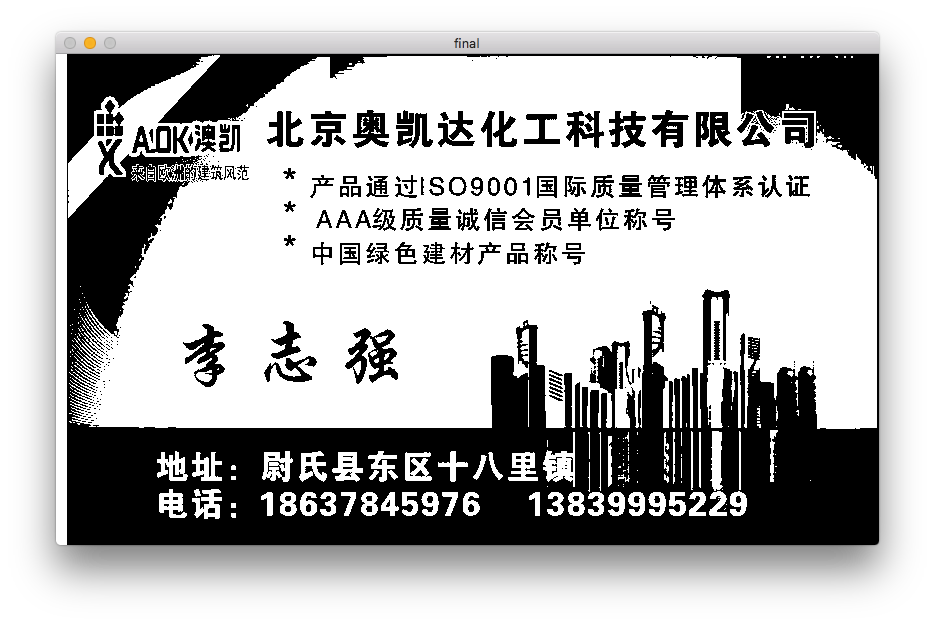


由于直方图不是标准的双峰图，因此对OTSU算法造成了很大的影响，导致右下角电话号码部分显示不完整：

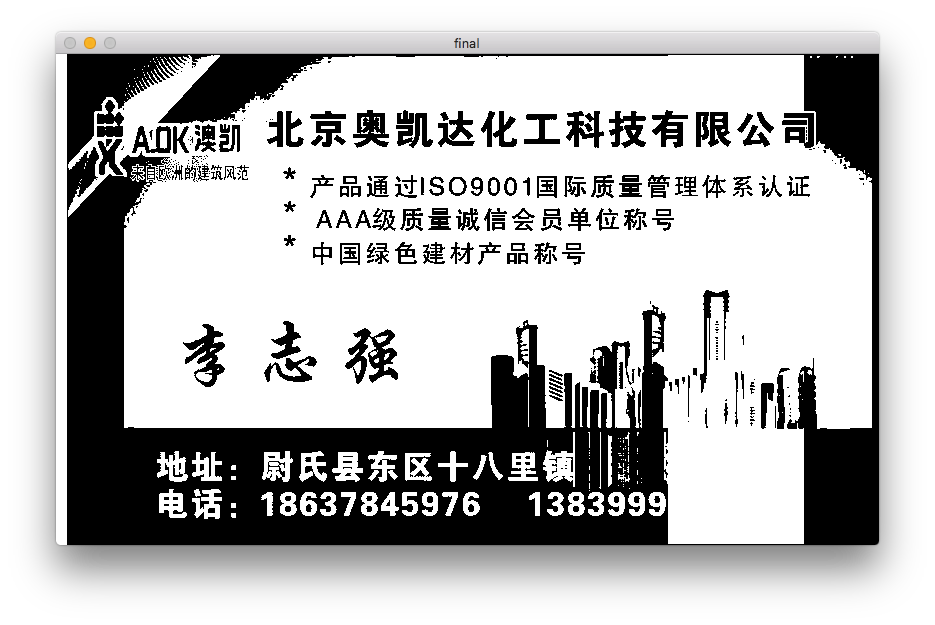


考虑对图像进行切割，之后进行OTSU。

竖直切割6份，效果不错：



但是，切割的份数不能太多，然很容易被局部的噪声所干扰，例如，竖直切割12份的效果如下：



4.jpg



5.jpg



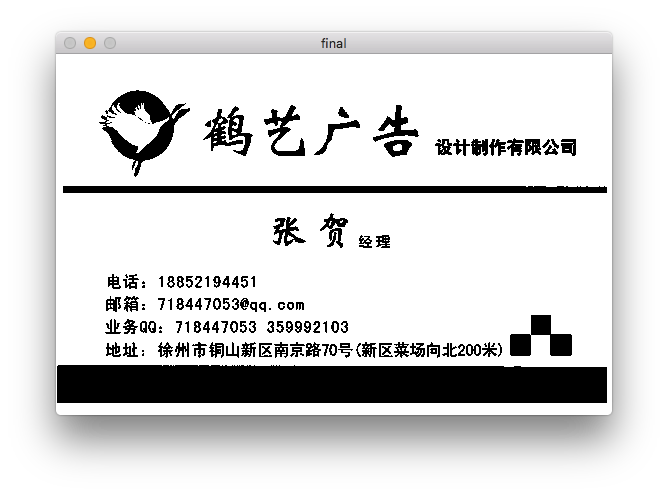
6.jpg



7.jpg



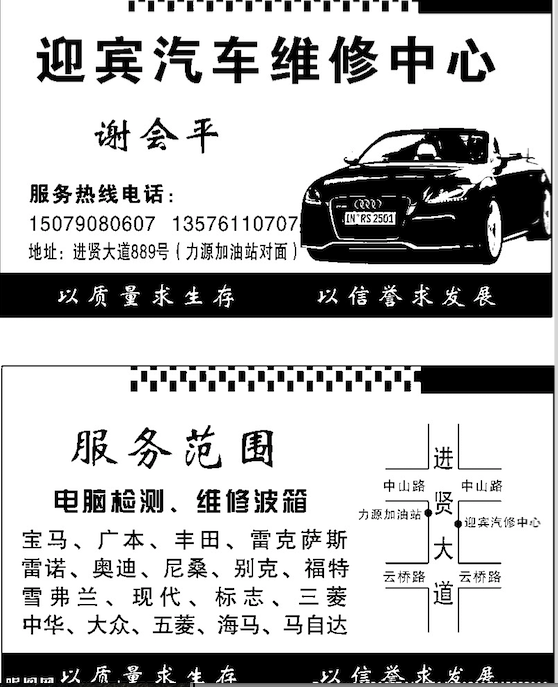
8.jpg



11.jpg



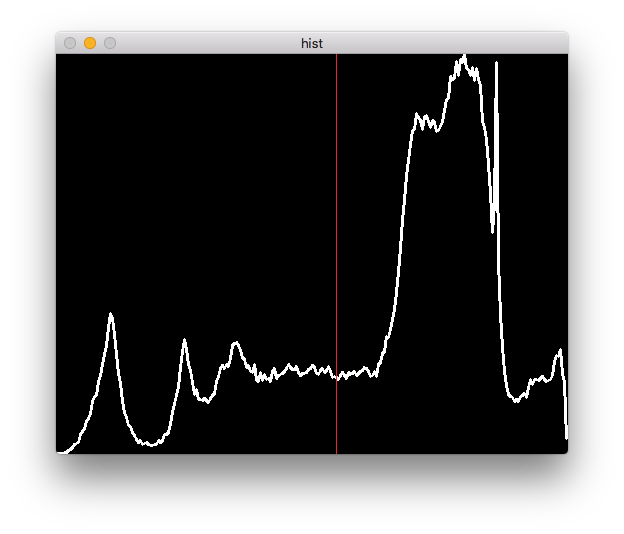
9.jpg



10.jpg



虽然这张图的频率图也不是标准的双峰图，但是在中间有一段平坦的波谷，因此也能够准确分离前后景：



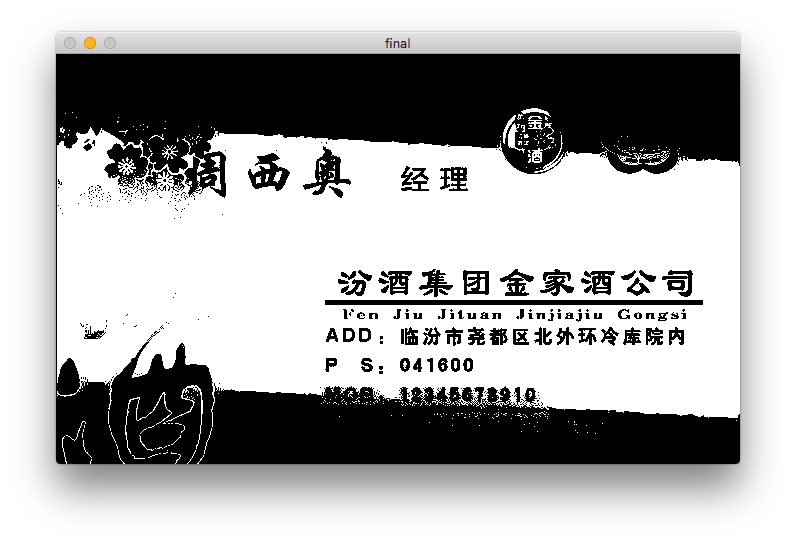
12.jpg



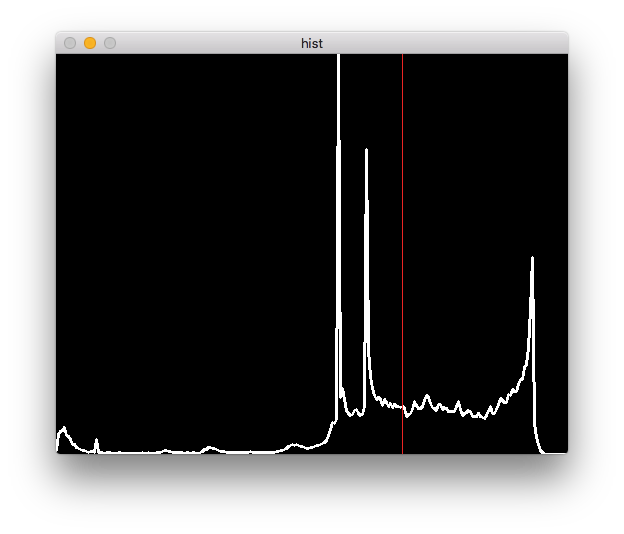
13.jpg



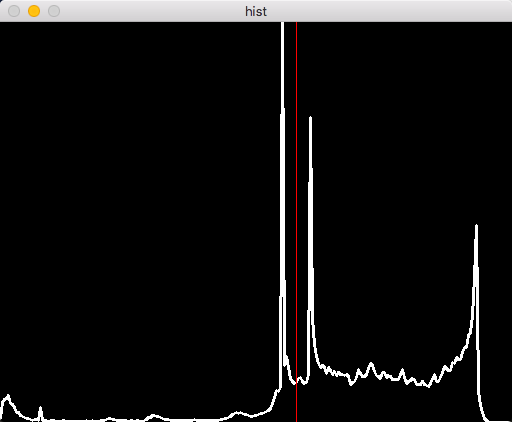
14.jpg



这张图的频率图有三个波峰，所找到的阈值在相对较宽的波谷处。由于图片下方的背景颜色较深，和Email处文字的黑色比较接近，因此Email也被隐去了：



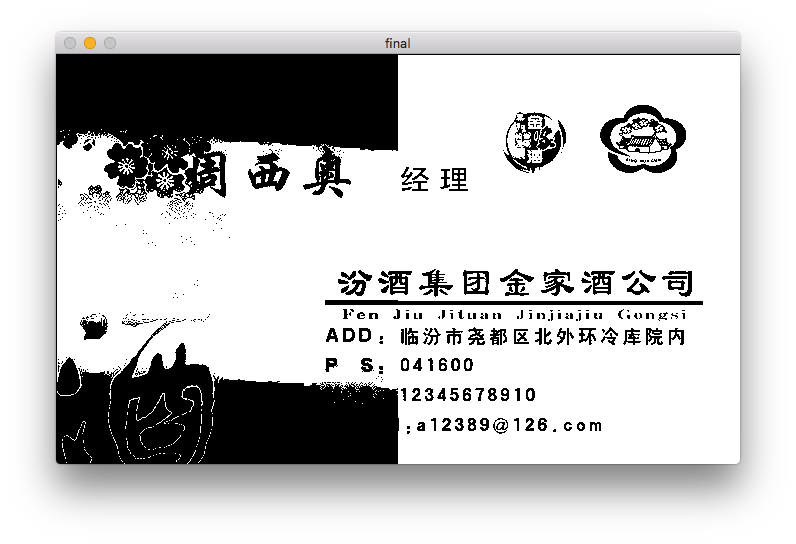
尝试手动修改阈值，使得阈值在两个靠的比较近的波峰之间，效果如下：



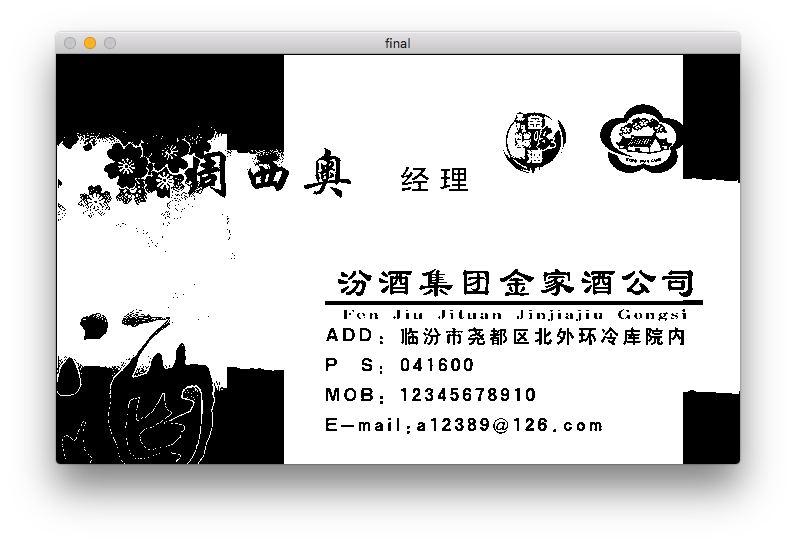


虽然这样做可以达到相应的效果，但是如果对每一张图都手动调节参数显然是不可取的。因此改用切割图片的方法。

竖直切割2份，尽管Email的地址能清楚显示，但左侧Email关键词还是被隐去了：



竖直切割12份，此时前后景分离的效果的还是比较好的：



水平切割3份：



水平切割6份：



这两种切割方法都能够很好的保留图片上的文字，而花纹等都被去除。可能的原因是名片中的文字也是横向排列的。

参考资料：

[1] opencv获得像素点的值

<http://stackoverflow.com/questions/21287082/accessing-certain-pixels-intensity-valuegrayscale-image-in-opencv>

[2] 绘制频率分布图

<http://docs.opencv.org/2.4/modules/imgproc/doc/histograms.html?highlight=calchist#calchist>

[3] 绘制频率分布图

<http://opencvexamples.blogspot.com/2013/10/histogram-calculation.html>