计算机视觉和模式识别 作业3

*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）,默认对第1张图片进行处理。要对其他图片进行测试，执行./a.out ./Dataset/X.jpg即可。

Windows系统，在cmd中输入execute.exe ./Dataset/X.jpg。

二、实验过程

这次对A4纸的矫正大部分是基于上次的边缘和边缘点的检测。因此上次作业的正确性会大大影响这次作业的成功率。

一开始的时候，我就发现了上次作业中的几个Bug。对于竖直的直线来说，theta可能的值可能接近0，也可能接近PI，这就对直线去重那段程序带来了很多麻烦。由于我是先将所有直线大致按照theta进行排序，当theta比较接近时，则按照rho进行排序。这样，theta为0的线段会被排在前面，而theta接近PI的线段会排到后面。因此我就在去重之前，先将theta接近PI的线段移动到前面。但这里需要注意的是，在把直线移到前面的时候，要考虑rho的大小。一开始没有考虑的这个问题，所以Dataset2/10.jpg一直不能成功去重。

还有一个问题就是，对于缺了角的A4纸，很容易在Hough变换中会把缺角的边检测到。因此要进行错误线段的排查。这里我用的方法是：如果一条线段与他前面和后面的线段的theta值相差都过大时，则认为这条线段是错误检测的线段，因此要删去。

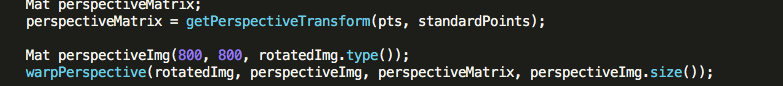
将上次作业中的Bug改正后，才进入到这次作业的编程之中。

首先，我的目标是将所有的A4纸都旋转到正的方向。因此考虑使用排序后的第一条线段的theta值所对应的角度进行旋转。但这样会产生一个问题，对于Dataset中的第二张和第四张图片，他们旋转后都会呈水平方向，因此还要对他们顺时针旋转90度。而对于那些本身就是横着放的图片，则也要顺时针旋转90度，但这样做的话，有些图片的会超出原来的边界，因此我考虑将它们缩小一定的倍数。以上的旋转都是调用opencv中的



函数来实现的。其中还要注意的是旋转的中心是A4纸的中心，不然也很容易在旋转的过程中图片超出边界。对图像旋转完成后，也需要对四个坐标点进行相应的变换，这里利用了

之后，对转换后的点进行排序，这样可以得到一个稳定的顺序，即左上，右上，左下，右下。然后设定四个与之对应的标准点，然后调用

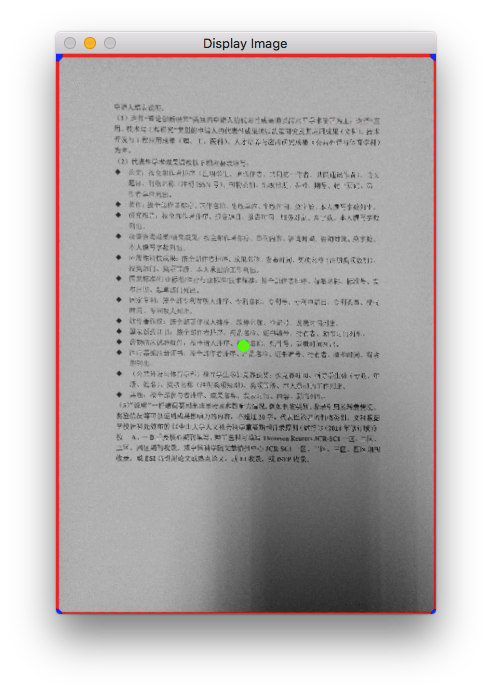


完成矫正。

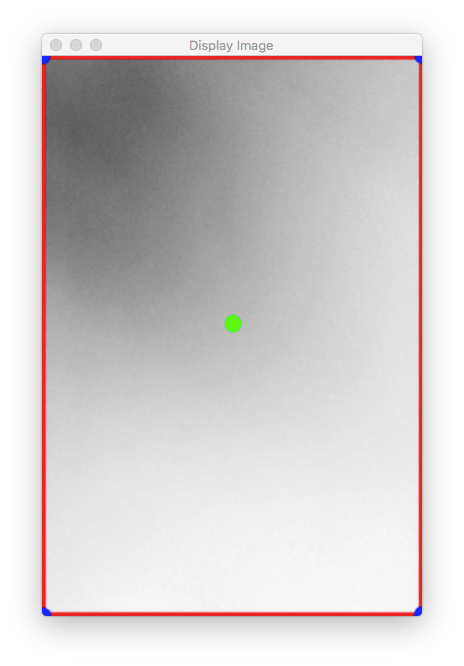
最后定义一个Rect,将图片裁剪出来并显示。

三、实验结果

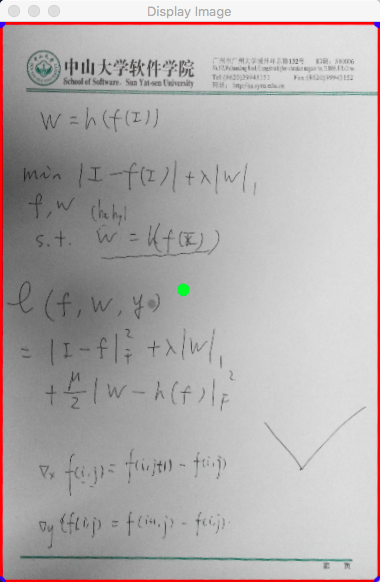
Dataset/1.jpg



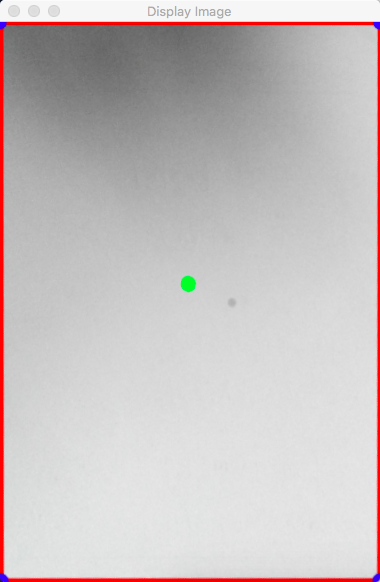
Dataset/2.jpg



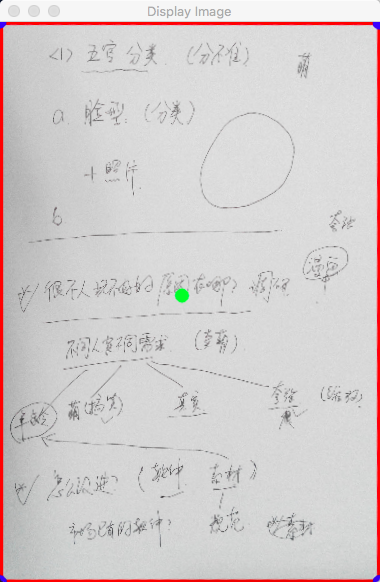
Dataset/3.jpg



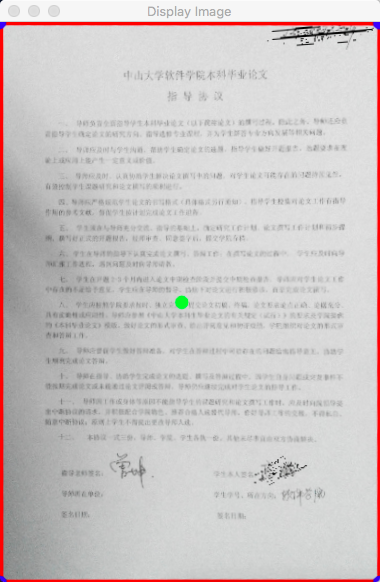
Dataset/4.jpg



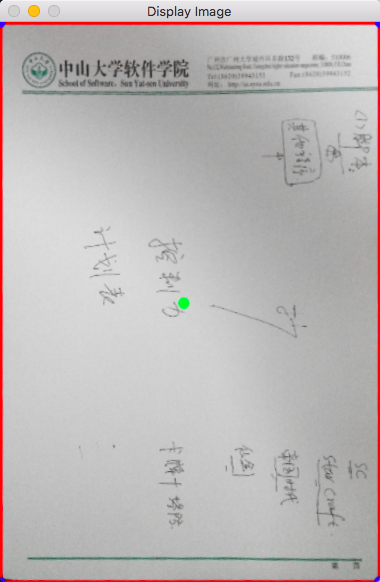
Dataset/5.jpg



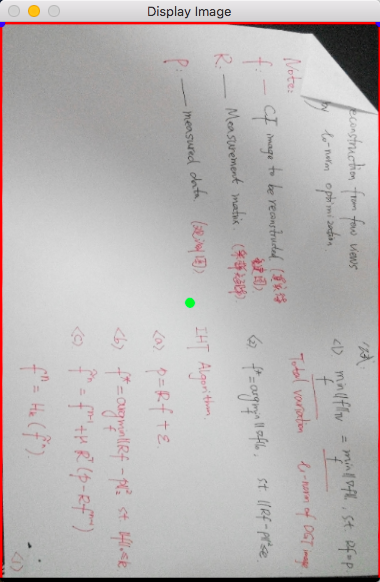
Dataset/6.jpg



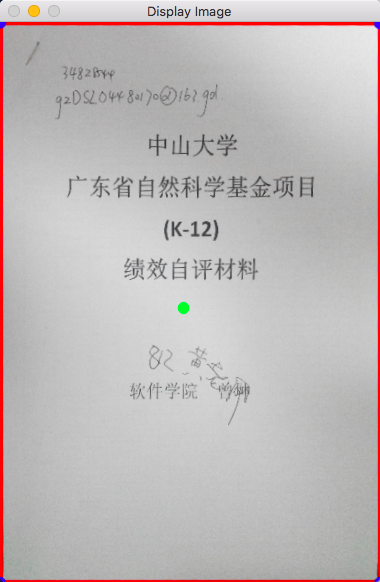
Dataset2/1.jpg



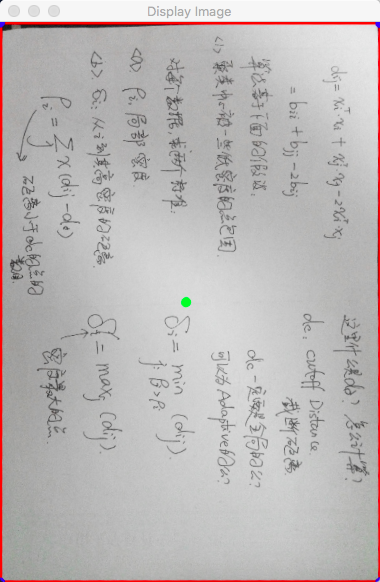
Dataset2/2.jpg



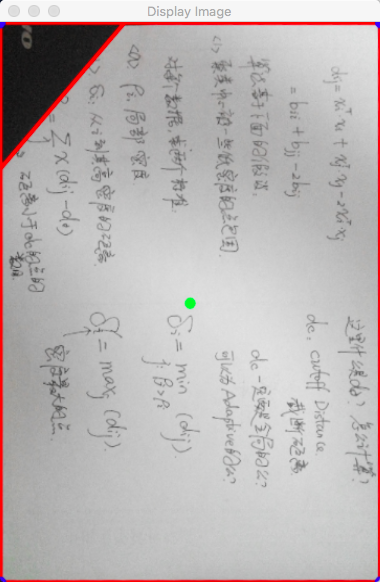
Dataset2/3.jpg



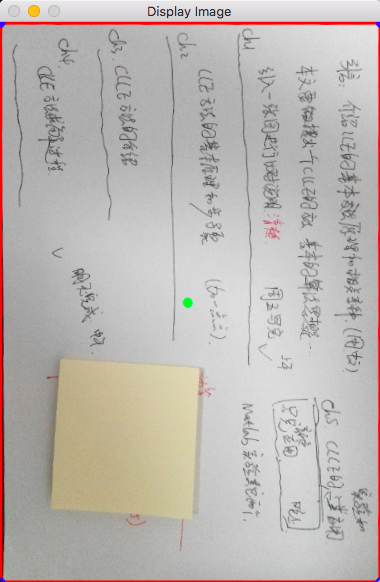
Dataset2/4.jpg



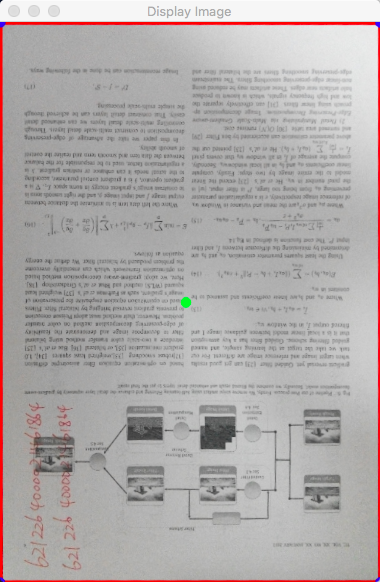
Dataset2/5.jpg



Dataset2/6.jpg



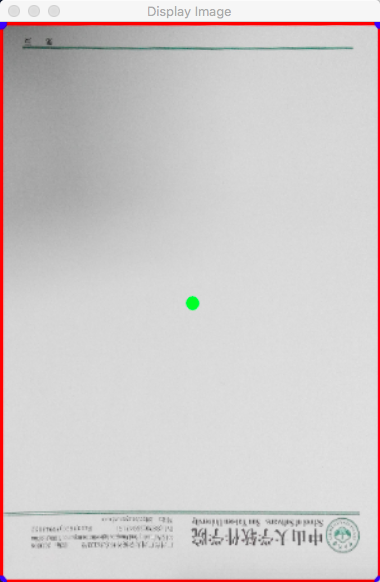
Dataset2/7.jpg



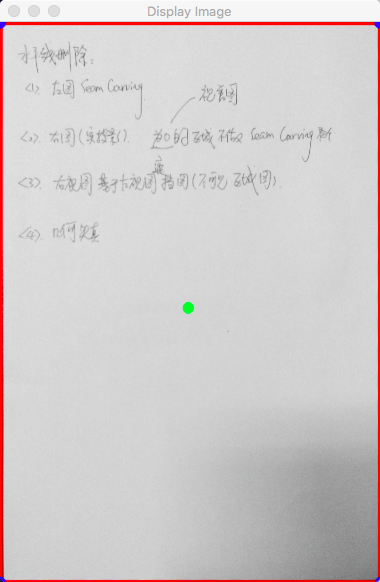
Dataset2/8.jpg



Dataset2/9.jpg



Dataset2/10.jpg



参考资料：

[1] warpAffine

<http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/warp_affine/warp_affine.html>

[2] transform

<http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html>

[3] transform函数中的Assertion failed解决办法

<https://groups.google.com/forum/?fromgroups=#!topic/android-opencv/CPDMJsmYVBI>

[4] 图像裁剪

<http://stackoverflow.com/questions/8267191/how-to-crop-a-cvmat-in-opencv>

[5] perspective transform原理

<http://math.stackexchange.com/questions/296794/finding-the-transform-matrix-from-4-projected-points-with-javascript>