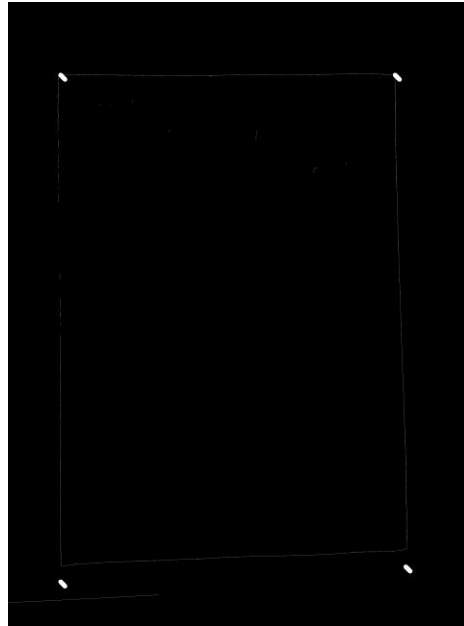
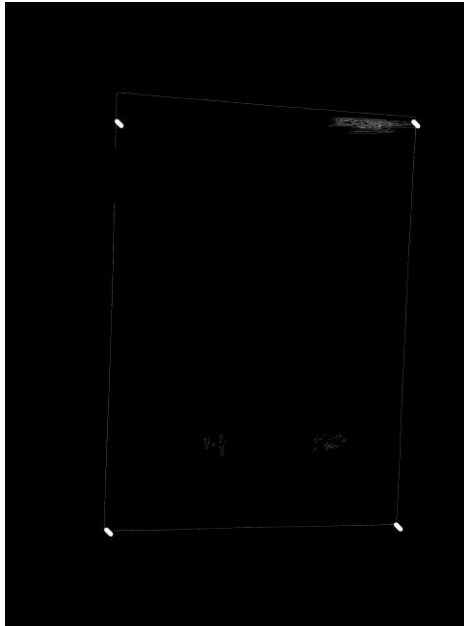


## 测试用例报告

(1) 测试用例：TA 给的 17 张图片进行测试，暂不针对其他图片进行测试

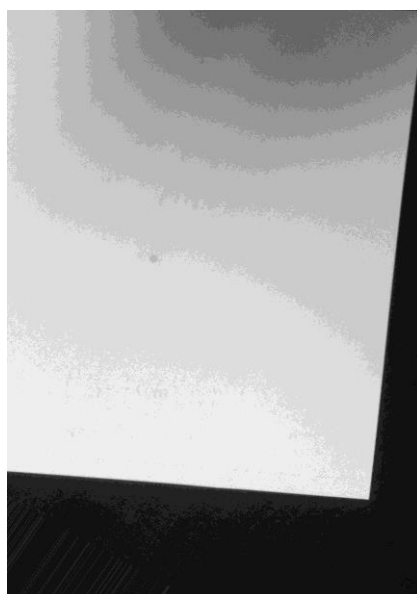
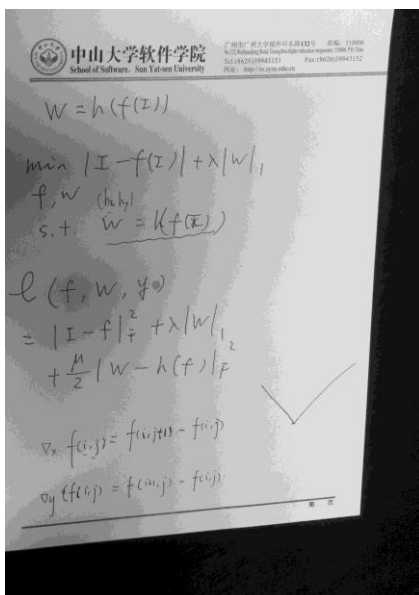
(2) 测试结果：首先测试纸张角点的正确性（测试结果在 FindPoints 文件夹中），在所有的图片中，基本没有问题，但是有两张图片测试的角点并不是很正确：



其原因是在进行角点检测的时候，出现了不规则的线干扰，即进行边缘检测的时候，噪声没有很好地去掉造成的。同时也与程序中的部分代码也有关系，在遭遇干扰线的时候，程序进行了取平均操作，因此才有了角点的偏移情况

(3) 进行完角点检测后，接下来就是进行单应矩阵操作，然后进行插值，最后得出我们需要的结果（测试结果在 Result 文件夹下），发现部分图片结果并不理想：

(转为 pdf 后图片可能出现光晕效果，是因为 pdf 灰度层级似乎比较低)



但是，上述的图片角点位置都是十分准确的，应该不会出现如此巨大的失误。其原因是，单应矩阵的效果并不是很好，在图片形变不规则的情况下，在进行矩阵操作时，似乎效果没有想象中好。但是，在程序中，由于时间原因，使用的是 `opencv` 的单应矩阵求解函数，因此很难进行分析。改进方法为自己花时间进行单应矩阵求解，或者是使用其他比较好的方法进行操作，避免出现这么大的失误。

（4）本程序有非常大的局限性，因为在进行高斯滤波的时候，实际上高斯核只能针对特殊大小的图片进行滤波，因此对于其他图片，可能效果不理想（貌似 **TA** 是用平板拍的图片，不然不会这么大的）。改进方法为，对图片进行归一化操作，也就是对大小不一的图片，都统一大小后进行操作，这样才能够避免图片不同造成高斯核不能够理想工作的影响。同时，对于单应矩阵的问题，后期还需改进变换方法，是程序能够更好地适应情况。