

作业三 地理卫星云图处理

吴政亿 151220129 nju_wzy@163.com

(南京大学 计算机科学与技术系, 南京 210093)

1 实验要求

对卫星云图进行处理，保留经纬线，剥离大陆线边界。允许应用所有matlab函数，尽可能的处理，最后实验评判由输出冗余点剔除率、目标像素点的保留率以及一个评分score来排序。

2 实验思路

2.1 图像分析

首先观察实验图像，实验图像是一个二值图像，并且观察image可以发现所有的经纬线都是严格按照八邻域连接的，并且经过边缘细化。

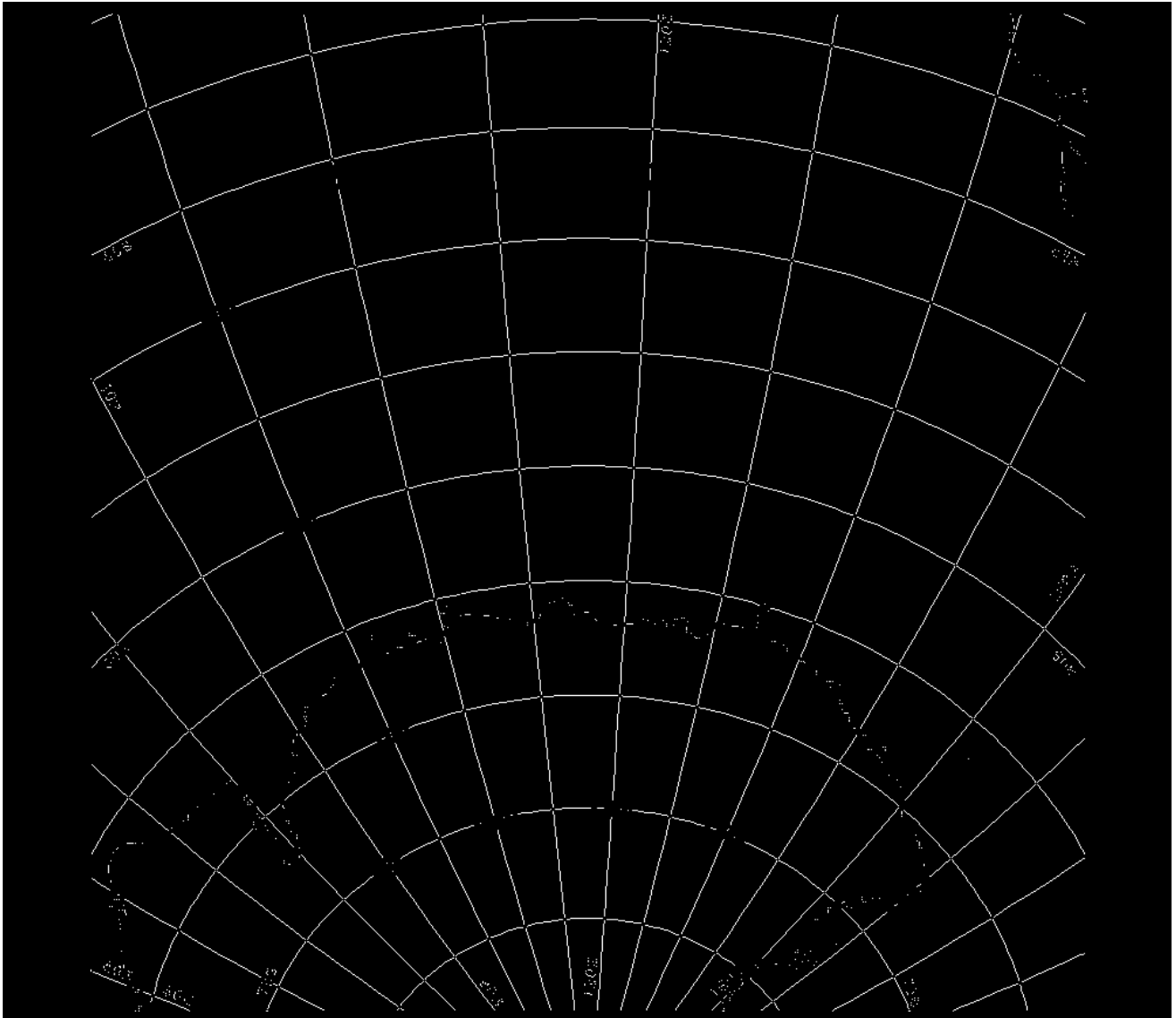
257	258	259	260	261	262	263	264	265	266	267	268	269	270
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	1	1	1	1	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0

2.2 一次卷积

因此，我首先尝试通过卷积将这些边界线筛选出来，这样，我先设计了一个判断出长度为3的边界的卷积核（卷积和为0）：

$$kernel\ 1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

应用这个卷积核对1.png进行处理得到的结果为

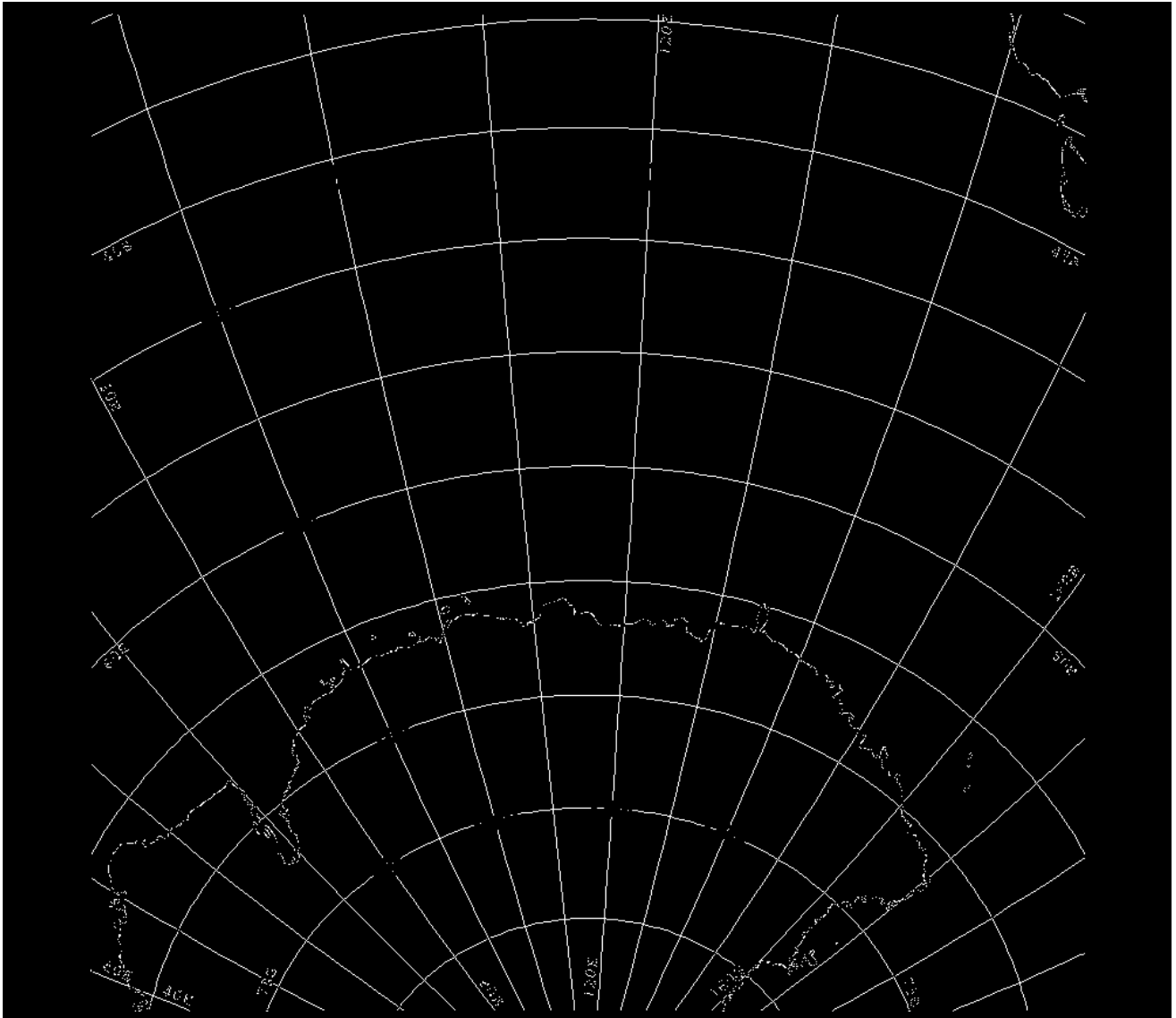


可以发现经纬线除了交汇点外，大陆线被清除为离散的点，部分的字并没有去除掉。分数如下：

```
score    = 1.1600e+04
score1    = 0.9931
score2    = 0.9991
```

为了加上两点交汇的点，我对原图像再一次卷积判断出具有交汇处的点（卷积和为0），并加入其中，结果如下：

$$kernel\ 2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



可以发现虽然交汇处的点被加入其中，但是大陆边界的线被割裂的较为离散，此时分数为：

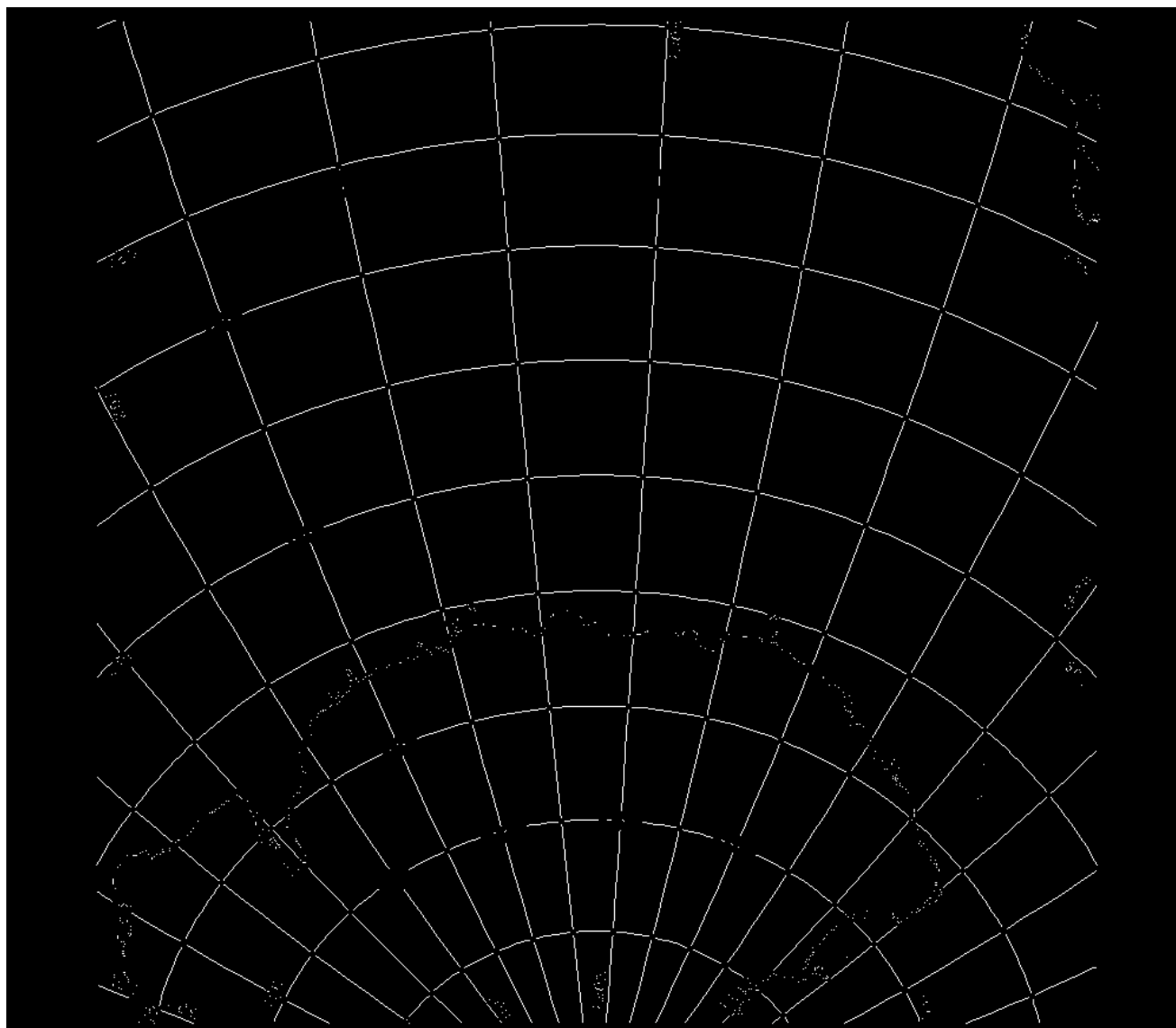
```
score    = 1.1601e+04
score1   = 0.9931
score2   = 0.9992
```

可见此时分数相比上面上升了不少，但是我觉的这个效果反倒是下降了，所以分数并不是很好的评判标准。

为了更好的去除大陆边界线，我将卷积核的大小由3*3变为5*5，卷积核如下：

$$kernel\ 3 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & 2 & -2 & 2 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

得到的结果如下：

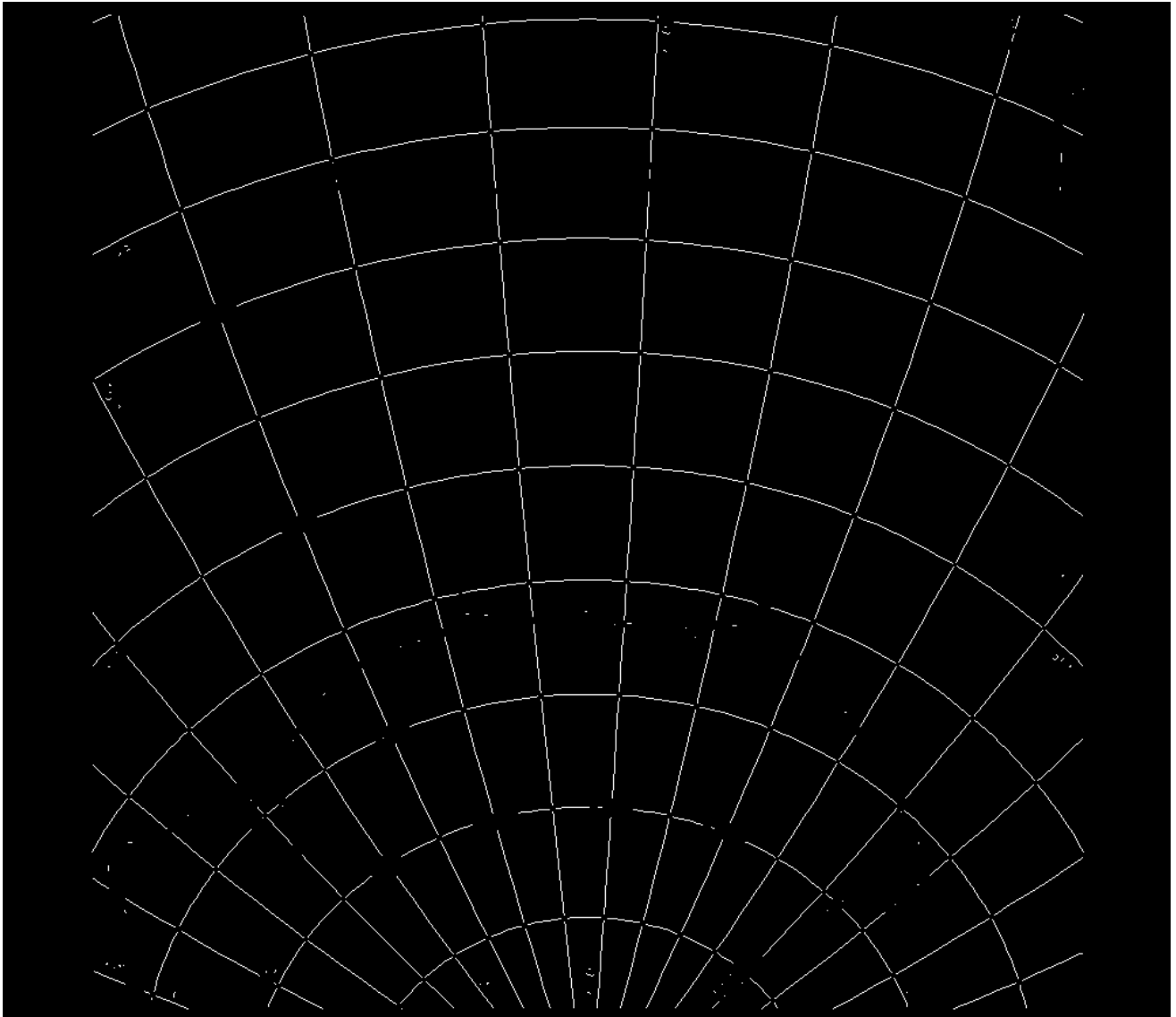


```
score    = 1.1602e+04
score1    = 0.9931
score2    = 0.9992
```

直观感受，看起来在交界处去除的点相比与kernel1要多了4个，但是对与边界线的去除效果得到了加强。

2.3 多次卷积

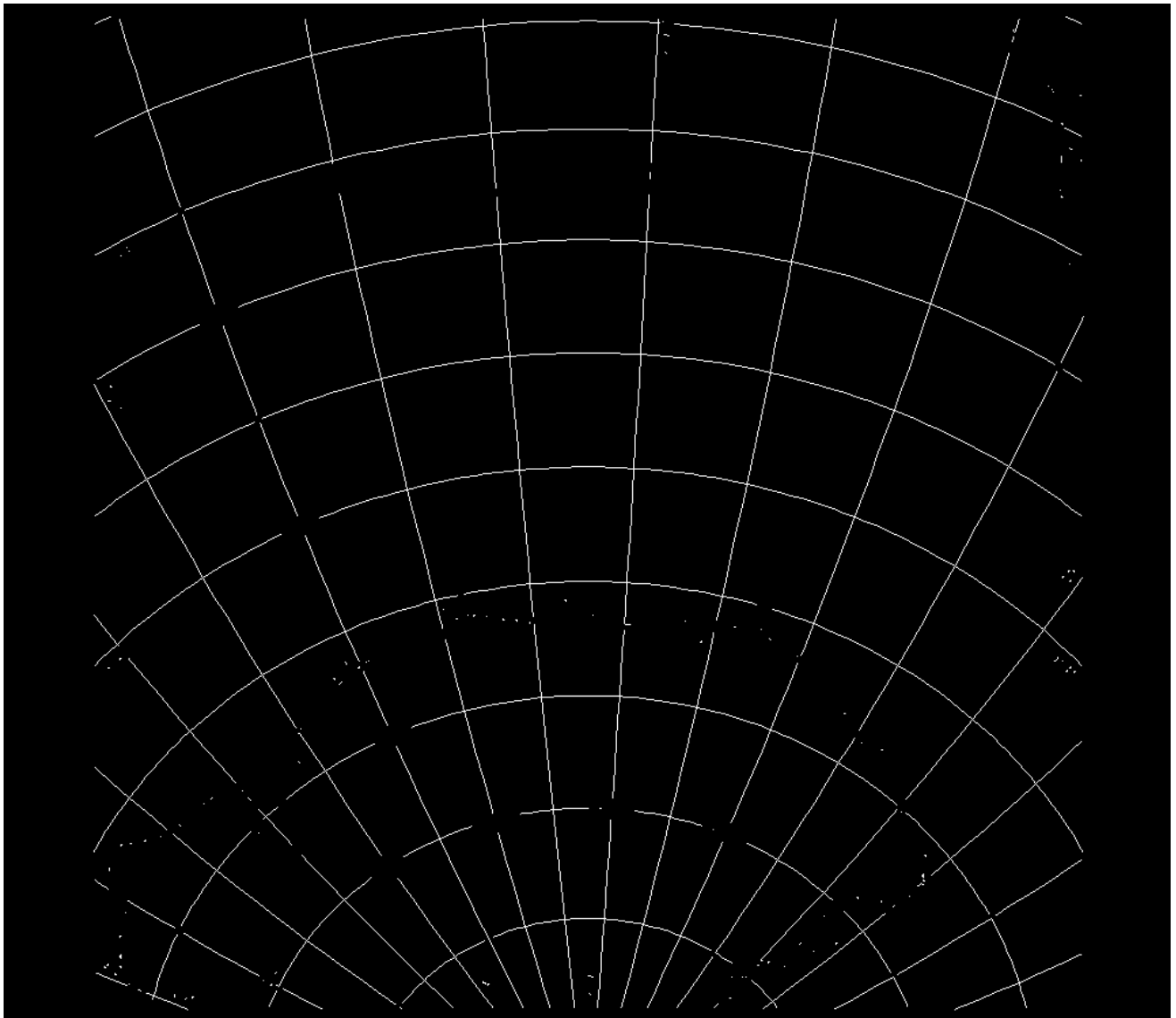
根据卷积核的变化，可以发现效果较为相似，因此我增加了卷积的次数，由一次更改为两次，kernel1的结果如下：



```
score    = 1.1600e+04  
score1   = 0.9931  
score2   = 0.9991
```

在效果更佳明显的情况下，分数竟然没有变化。

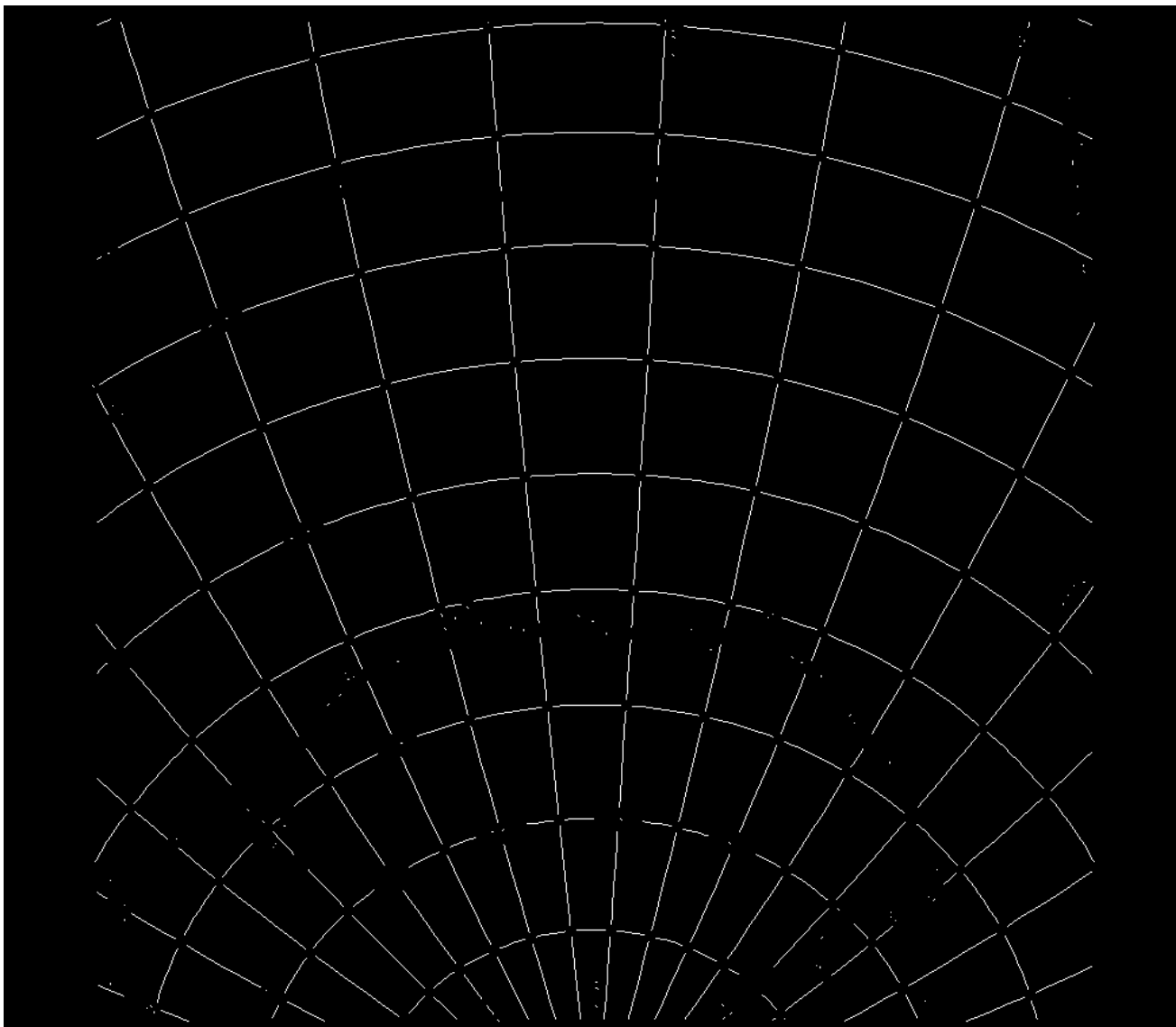
同样的，我们对kernel2进行二次卷积操作，效果同样明显并且保留的大陆线更多，但是由于kernel保留的数据较多，因此我进行了第三次卷积，结果如下：



```
score    = 1.1602e+04  
score1   = 0.9931  
score2   = 0.9992
```

在效果更佳明显的情况下，分数竟然没有变化。

对kernel3进行二次卷积，实验结果如下：



```
score    = 1.1588e+04
score1    = 0.9931
score2    = 0.9980
```

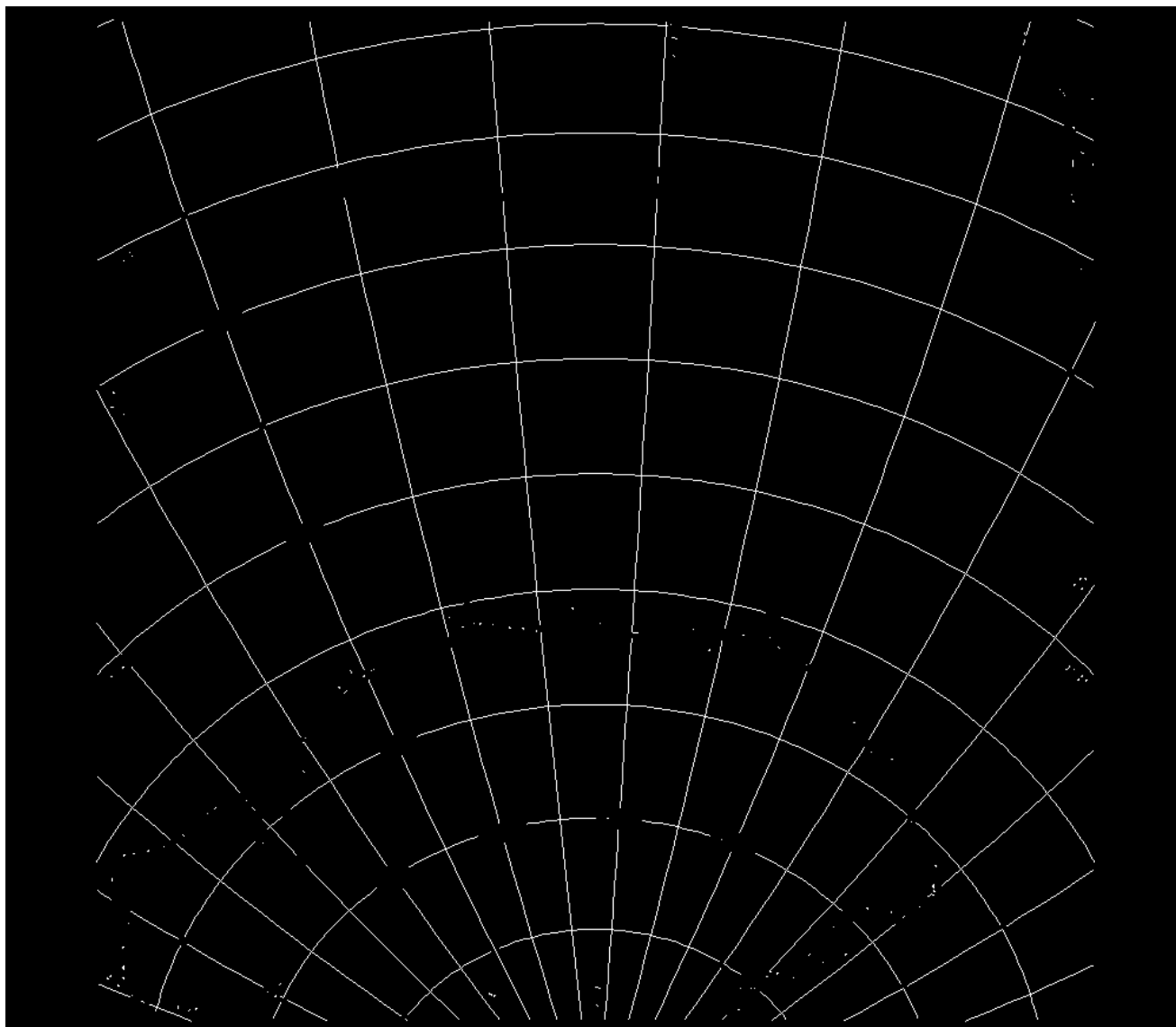
可以发现，实验结果在保留了经纬线的情况下，进一步的剔除了大陆边界线，虽然评分有所降低，但是效果up。

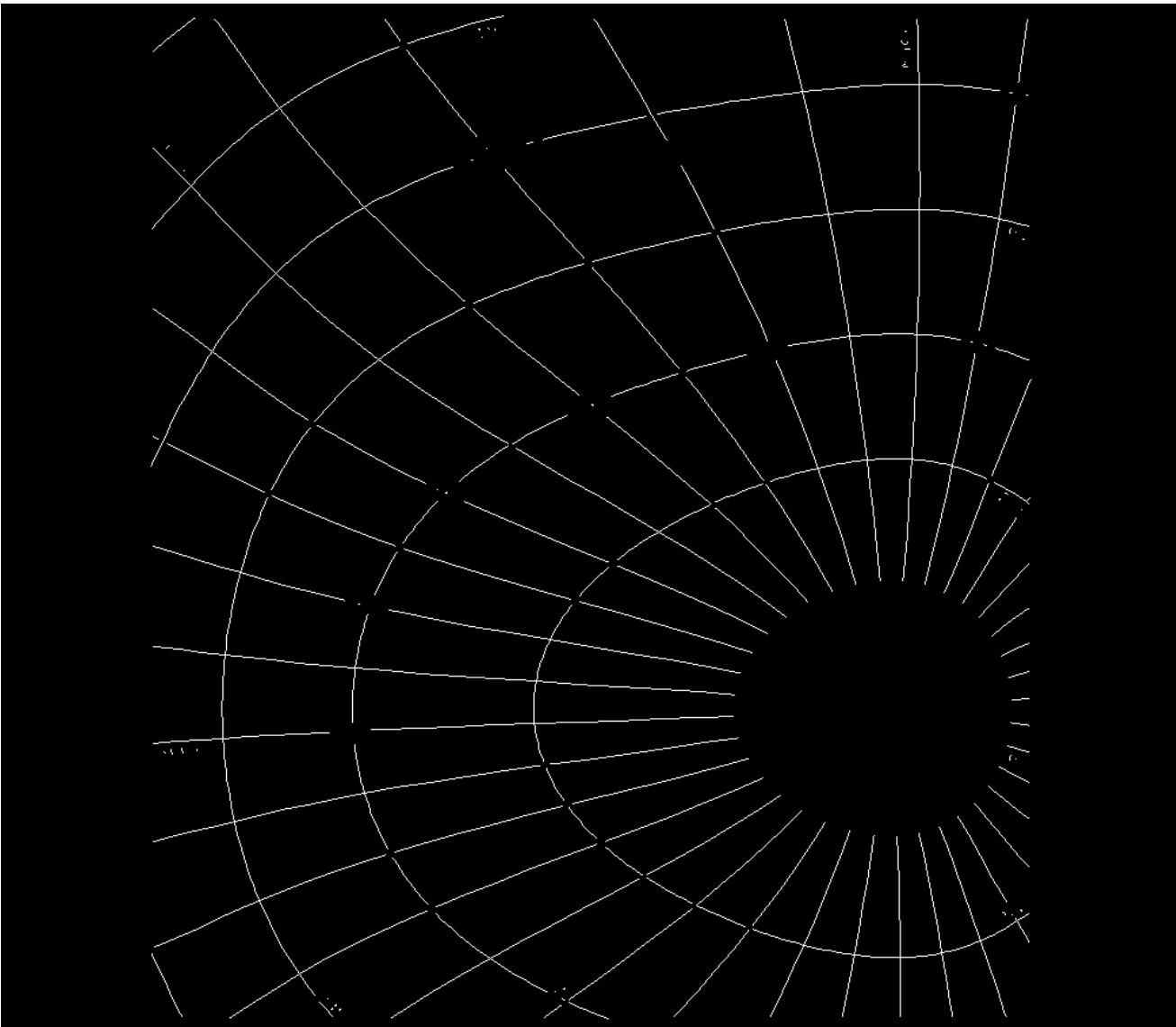
2.4 对比总结

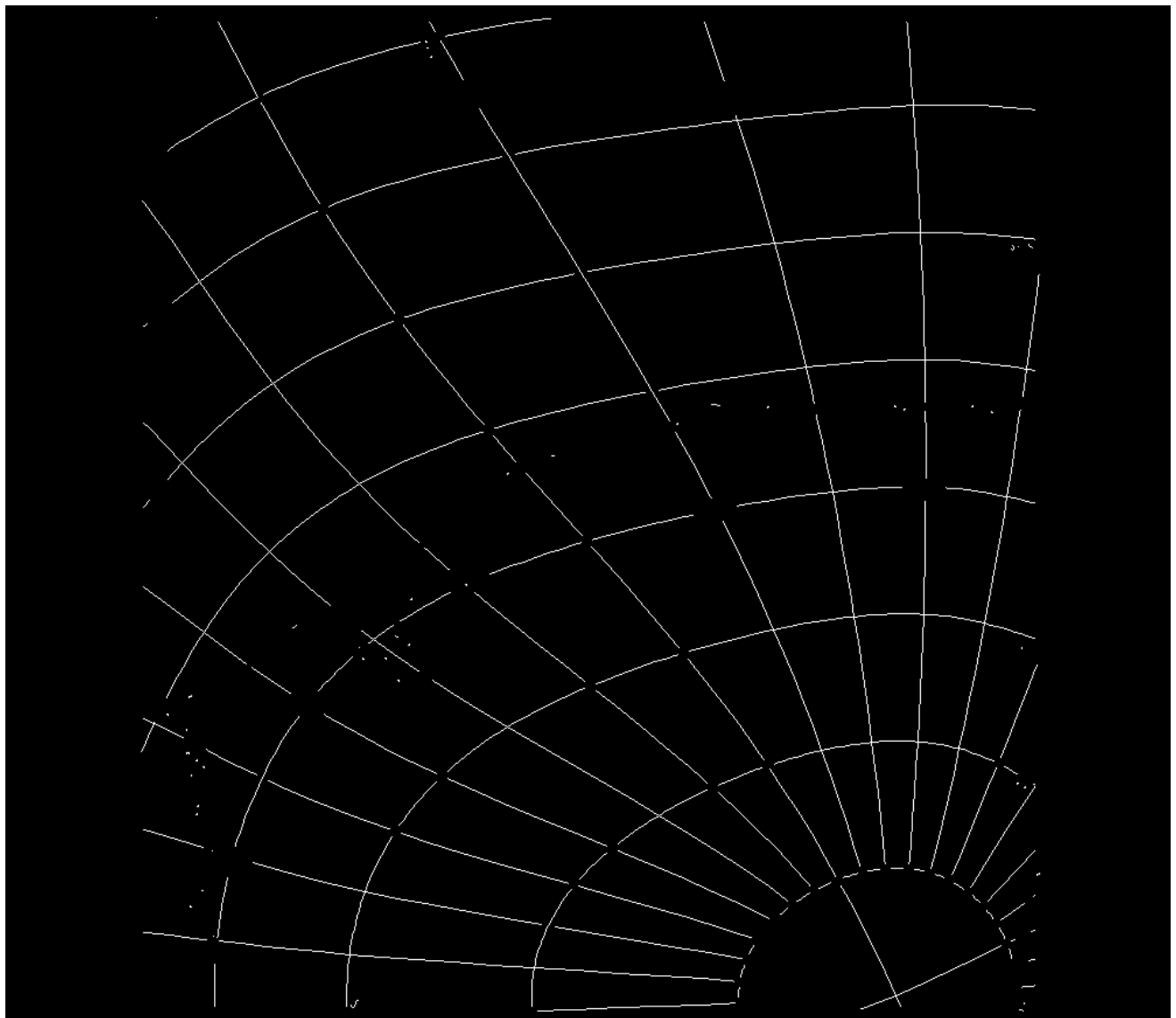
总的来说，进行多次卷积，可以将大陆线进一步的剔除，但是代价就是经纬线交汇处的缺口会越来越大。在对比总结之中，kernel1与kernel2配合的效果最佳，因为可以保证交汇了的经纬线不会有缺口。下图为各图经过3~4次卷积的结果，存在放 `./image` 下。卷积次数由 `imageprocessing_test.m` 中显示。

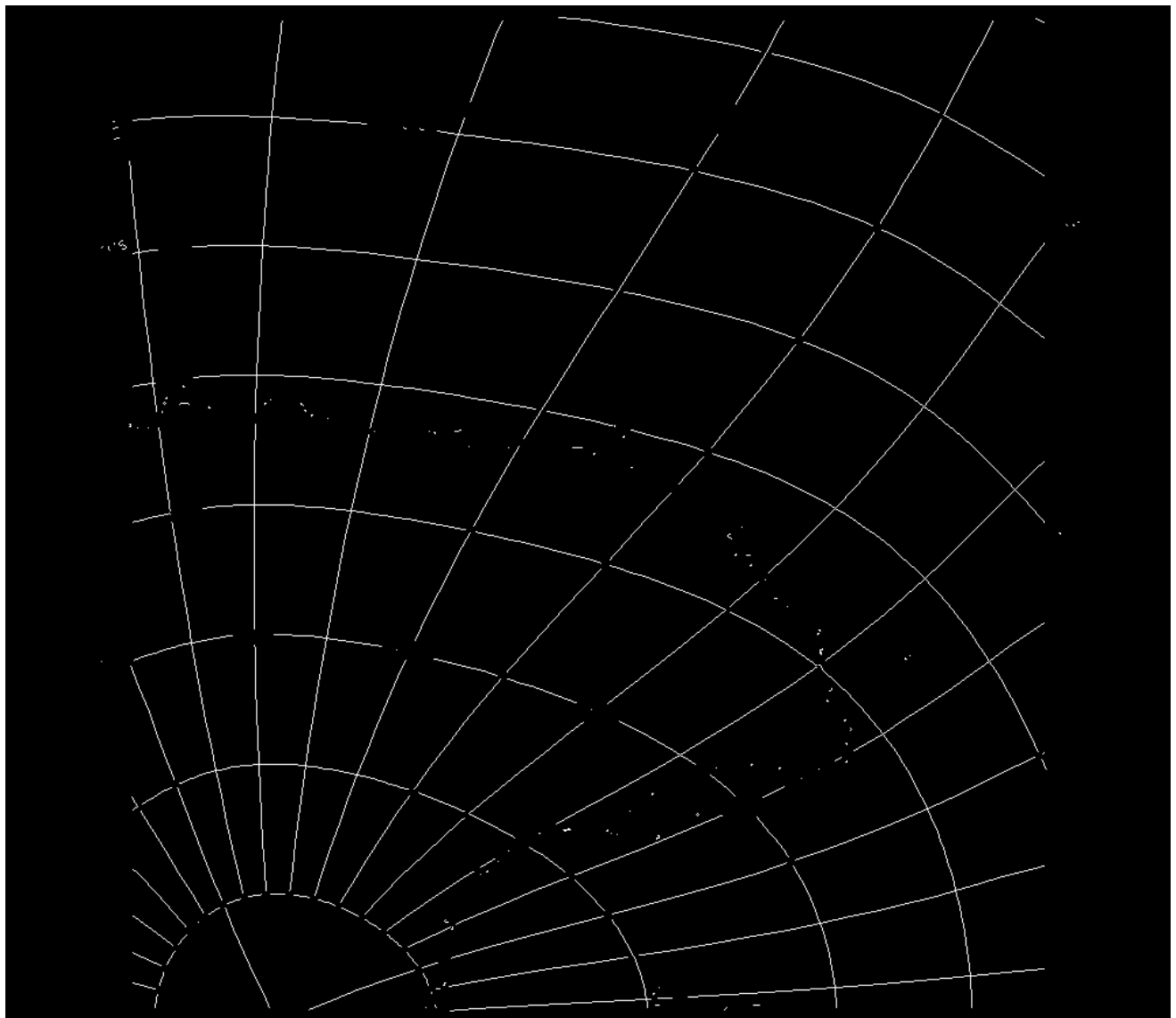
```
processed_image = my_imageprocessing(bw); % 卷积一次
processed_image = my_imageprocessing(processed_image); % 卷积二次
processed_image = my_imageprocessing(processed_image); % 卷积三次
processed_image = my_imageprocessing(processed_image); % 卷积四次
```

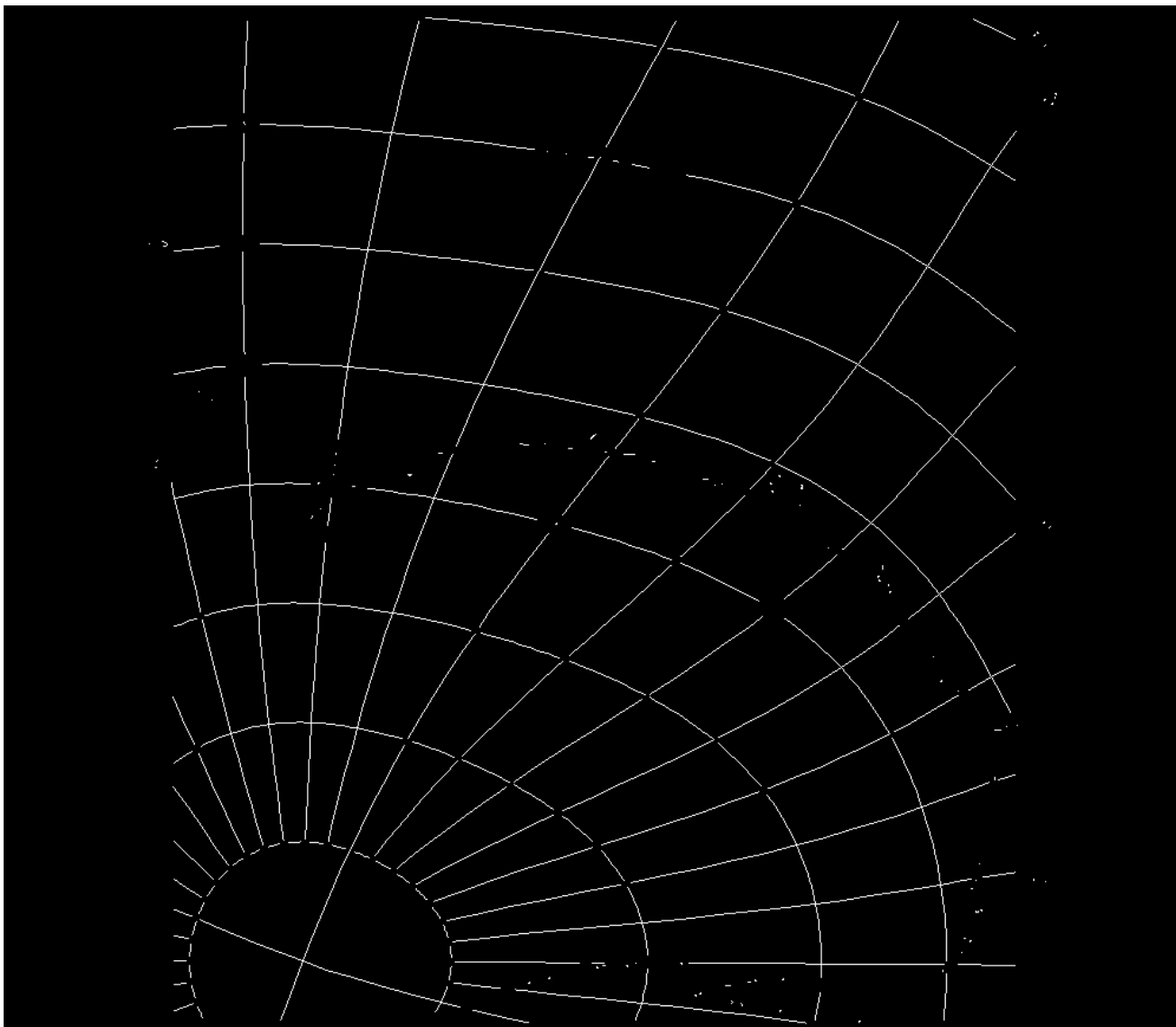
2.5 实验结果











3 实验体会

这次实验尝试了许多方法，从霍夫变换，到边界追踪（由于满足八连通，可以完整追踪下来，但是难以去除经纬线），到最后的卷积，发现我的方法越厉害，结果越辣鸡，最终选择九九归一，反而取得了不错的结果，本次实验由于不像前两次实验一样方向明确，费了不少时间精力，虽然我真的很像像群里说的打表一样（我从第二次实验就说想打表了嘤嘤嘤），但是考虑到报告没法写还是放弃了emmm，而且还要默默的抠图。