

CVPR 第四次作业

学 院 名 称 : 数据科学与计算机学院

学 生 姓 名 : 谢元新

学 号 : 14331311

专 业 （ 班 级 ） : 14 软件工程三（5）班

总体思路:

首先要得到 A4 纸的边缘, 采用 canny 边缘检测算法, 得到边缘, 要注意的是对 sigma 和 threshold 两个参数的取值。

接着要得到边缘的直线方程。在第一步的基础上用霍夫变换, 得到每条边缘的直线方程。要注意的是对点的投票值的取值。

要得到每个点的坐标, 我的做法是类似于投票法, 当然也可以利用第二步的直线方程, 但是这样要考虑的东西比较多, 所以偷了个小懒, 不过图片经过下采样之后像素点并不是很多, 效率也很高。

各个参数意义:

in_thresh 主要用于去除噪声, 低于像素值低于此的不进行霍夫变换

out_thresh 取值区间为 0~1, 在霍夫空间中与最大点的比值低于此的不认为是一条直线

theta_difference 两条直线的 theta 值差距在此之间认为两条直线可能相等

offset_difference 两条直线的 offset 值差距在此之间认为两条直线可能相等

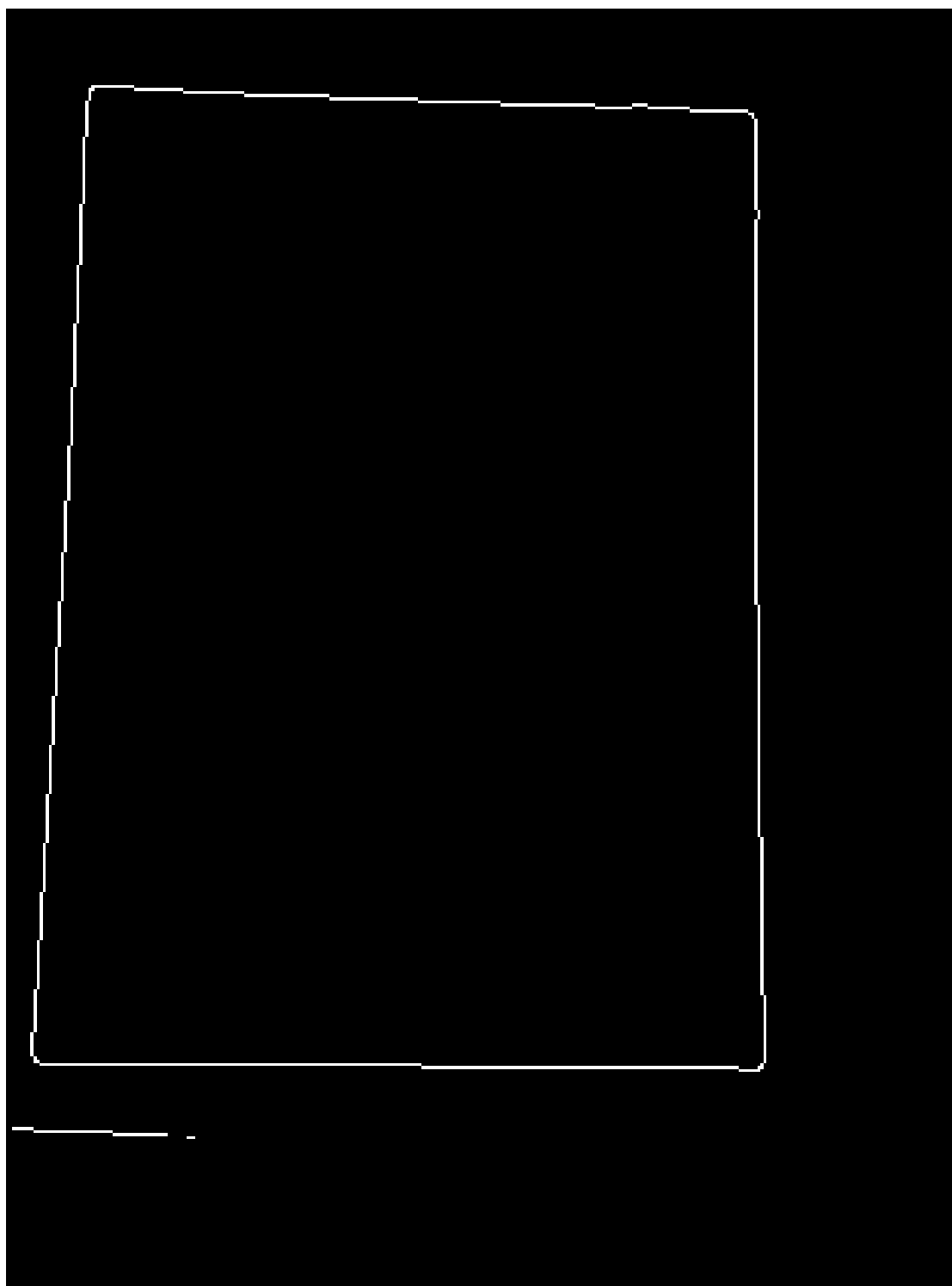
non_maximum_suppression 对霍夫空间进行取点的时候, 对某个点的周围的几个点组成的点阵进行判断, 取出点阵中值最大的点, 避免直线重复

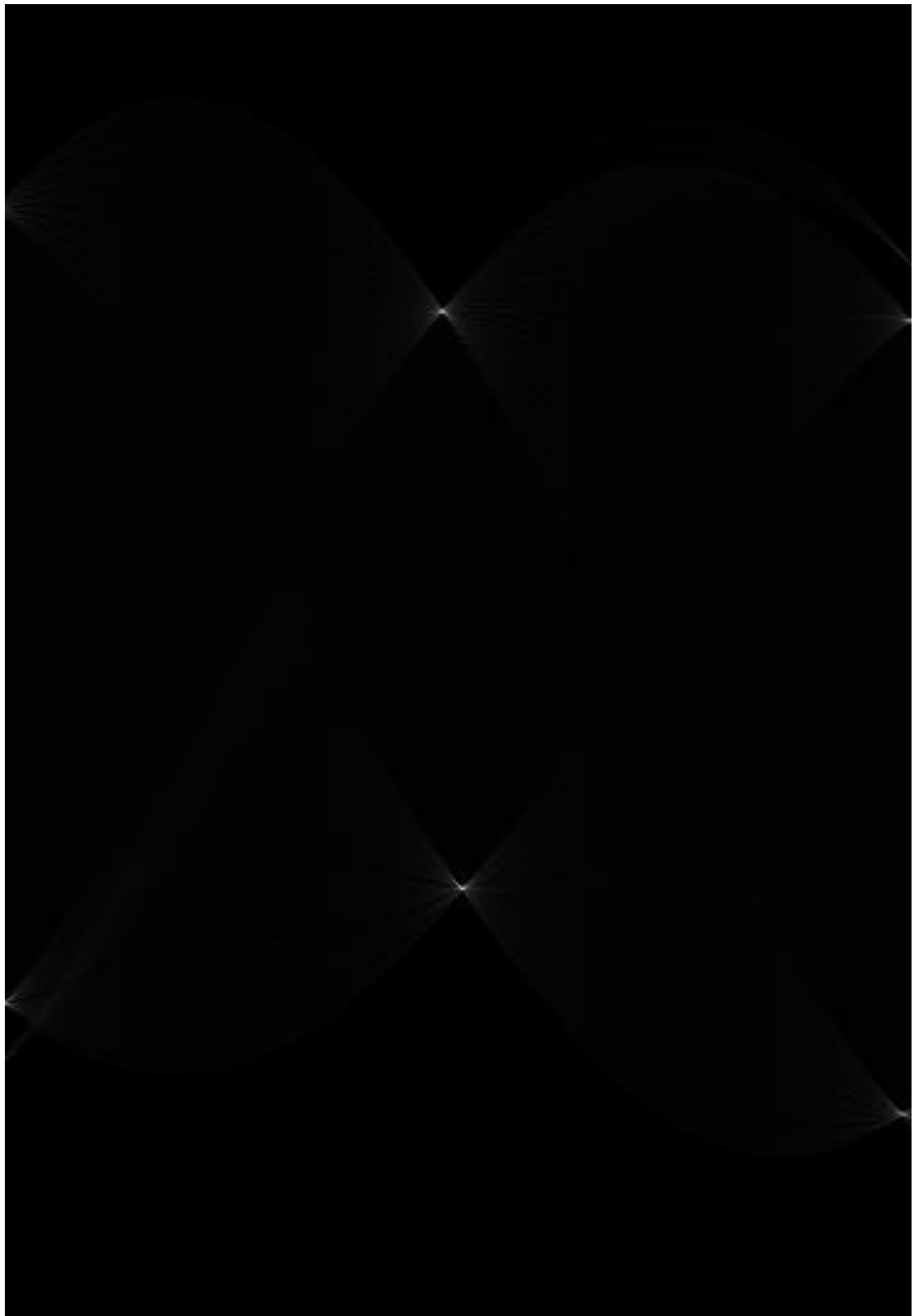
思考如何运行得更快:

1. 将图片进行适当的下采样, 可有效降低运行时间
2. 确定合理的梯度步长
3. 计算点的位置的时候不用投票法, 直接进行计算

第一张图片的运行结果:

参数取值: sigma=2.0, threshold=10.0, in_thresh=223.0, out_thresh=0.50, theta_difference=0.2, offset_difference =5, non_maximum_suppression=0



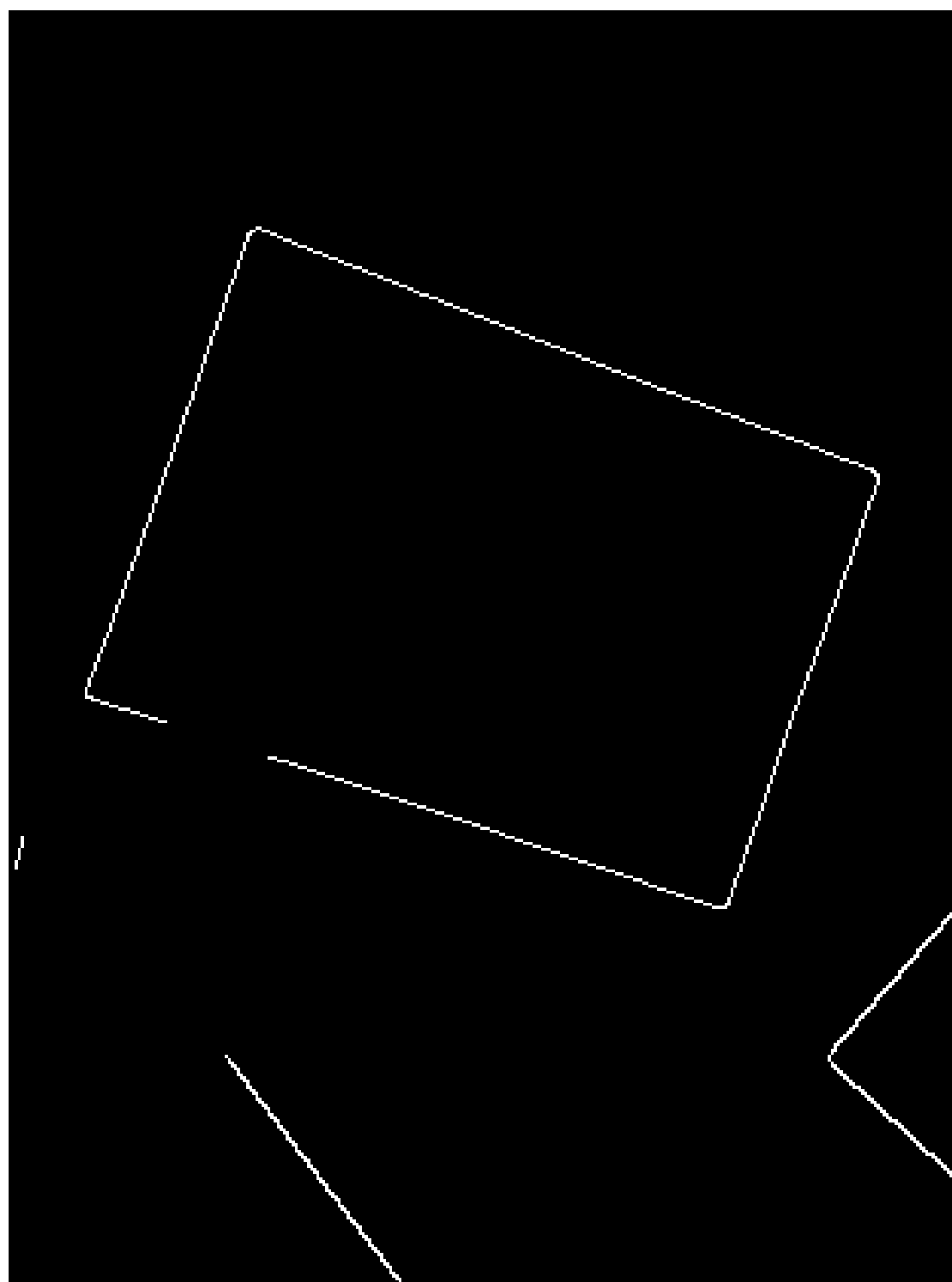



```
linear equation:
0.0610485 * Y + 0.998135X = 27.974
-0.999962 * Y + 0.00872664X = -345.184
-4.37114e-008 * Y + 1X = 244.535
-0.999048 * Y + 0.0436195X = -24.4253

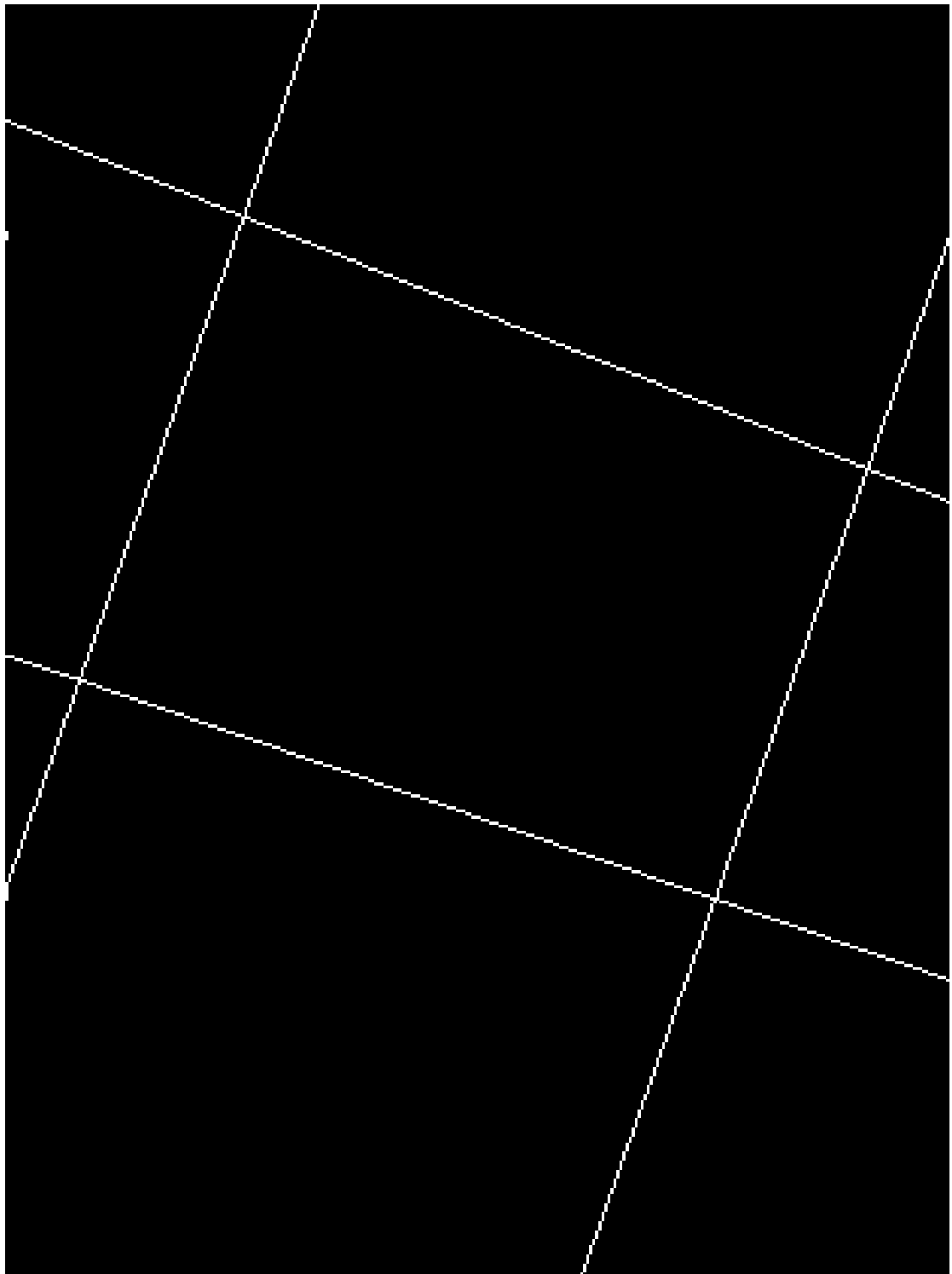
the conners coordinate:
<6,345>
<26,25>
<244,35>
<244,347>
```

第二张图片:

参数取值: sigma=2.0, threshold=15.0, in_thresh=223.0, out_thresh=0.75, theta_difference=0.2, ,
offset_difference =5, non_maximum_suppression=0





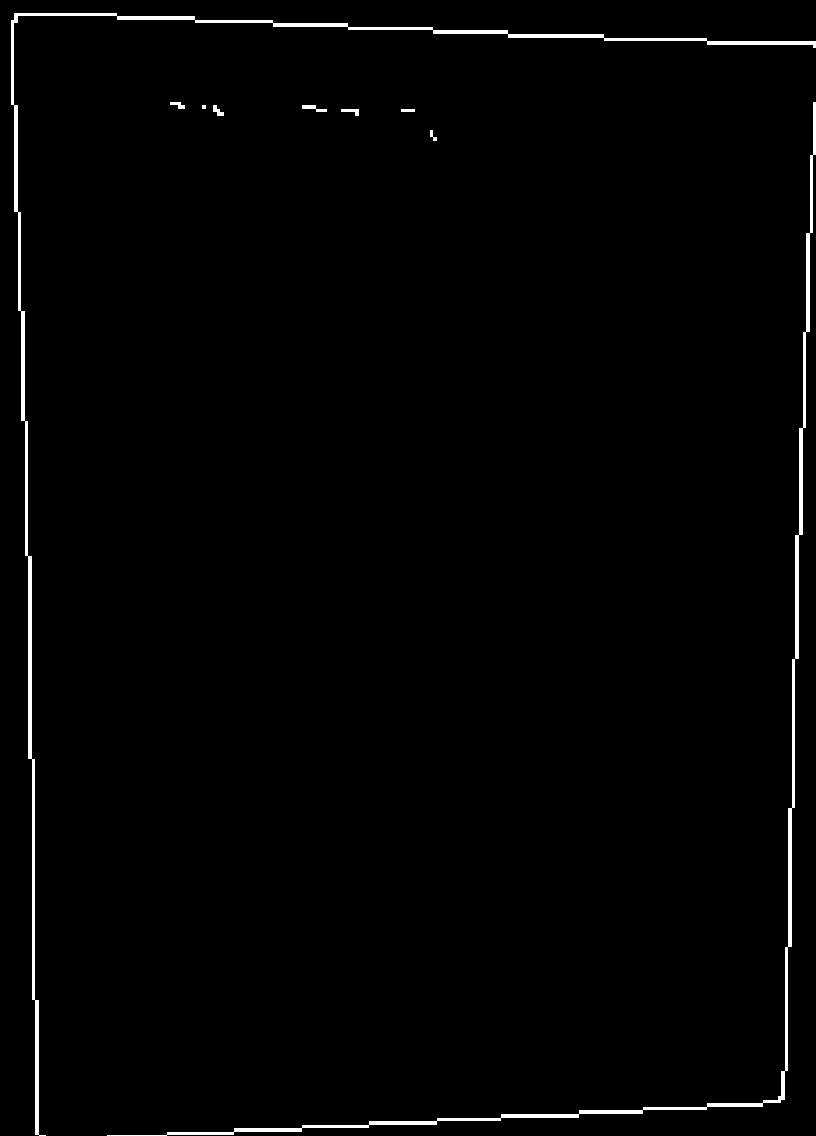


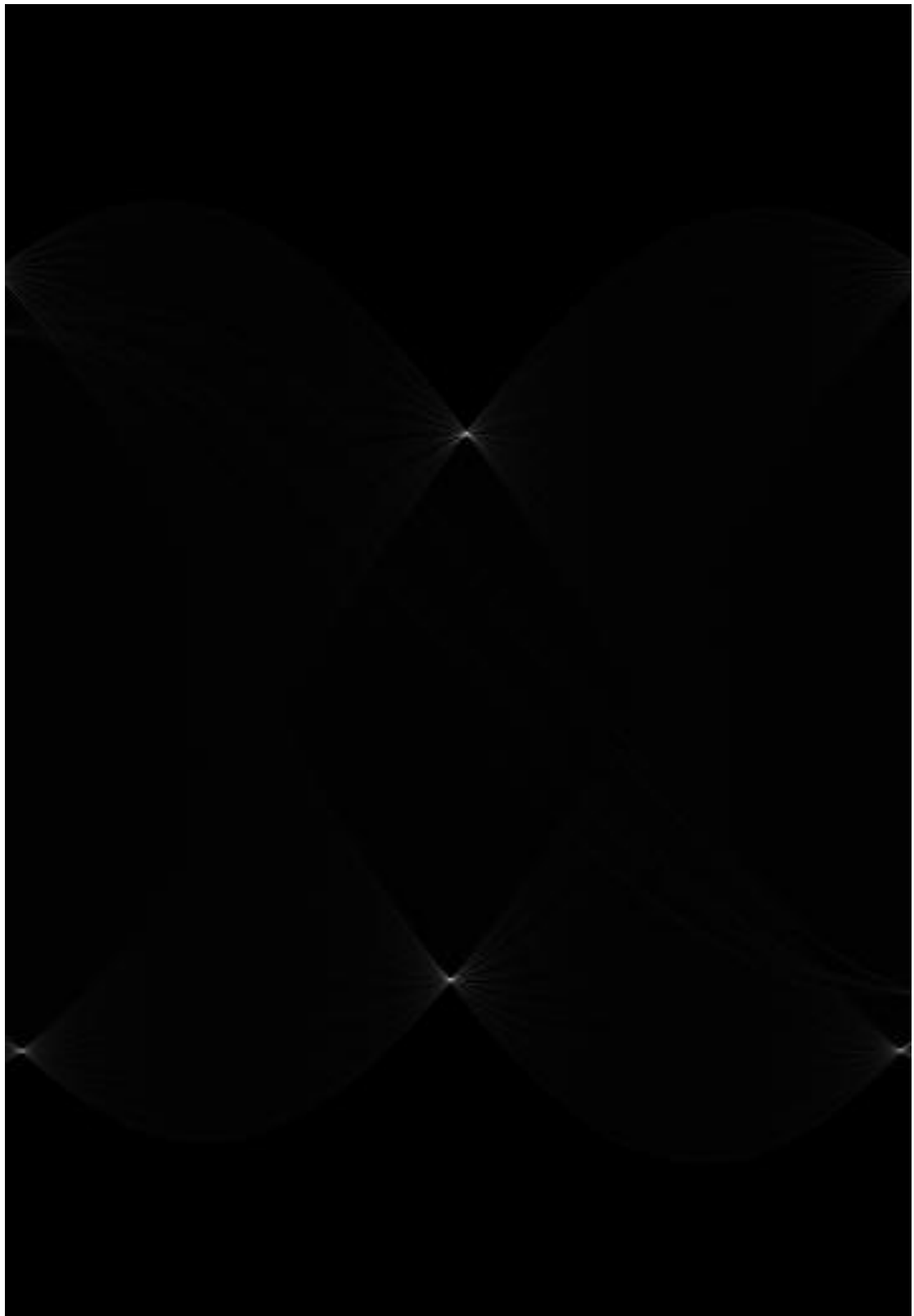
```
linear equation:
0.333807 * Y + 0.942641X = 97.6046
-0.945519 * Y + 0.325568X = -203.292
-0.927184 * Y + 0.374607X = -35.7305
0.333807 * Y + 0.942641X = 319.692

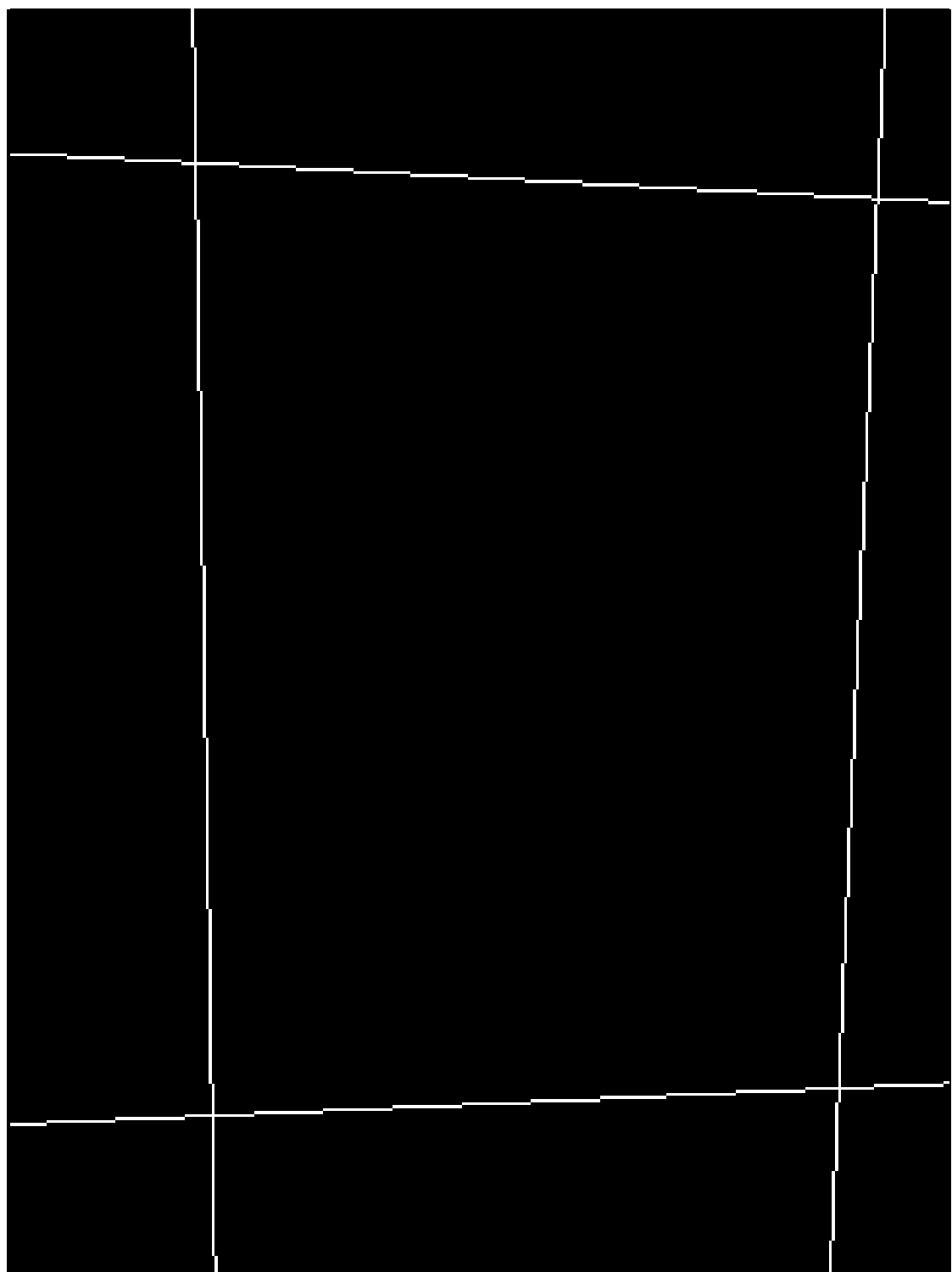
the conners coordinate:
<24,223>
<78,70>
<234,295>
<284,153>
```

第三张图片:

参数取值: sigma=2.0, threshold=10.0, in_thresh=223.0, out_thresh=0.50, theta_difference=0.2, ,
offset_difference =5, non_maximum_suppression=0





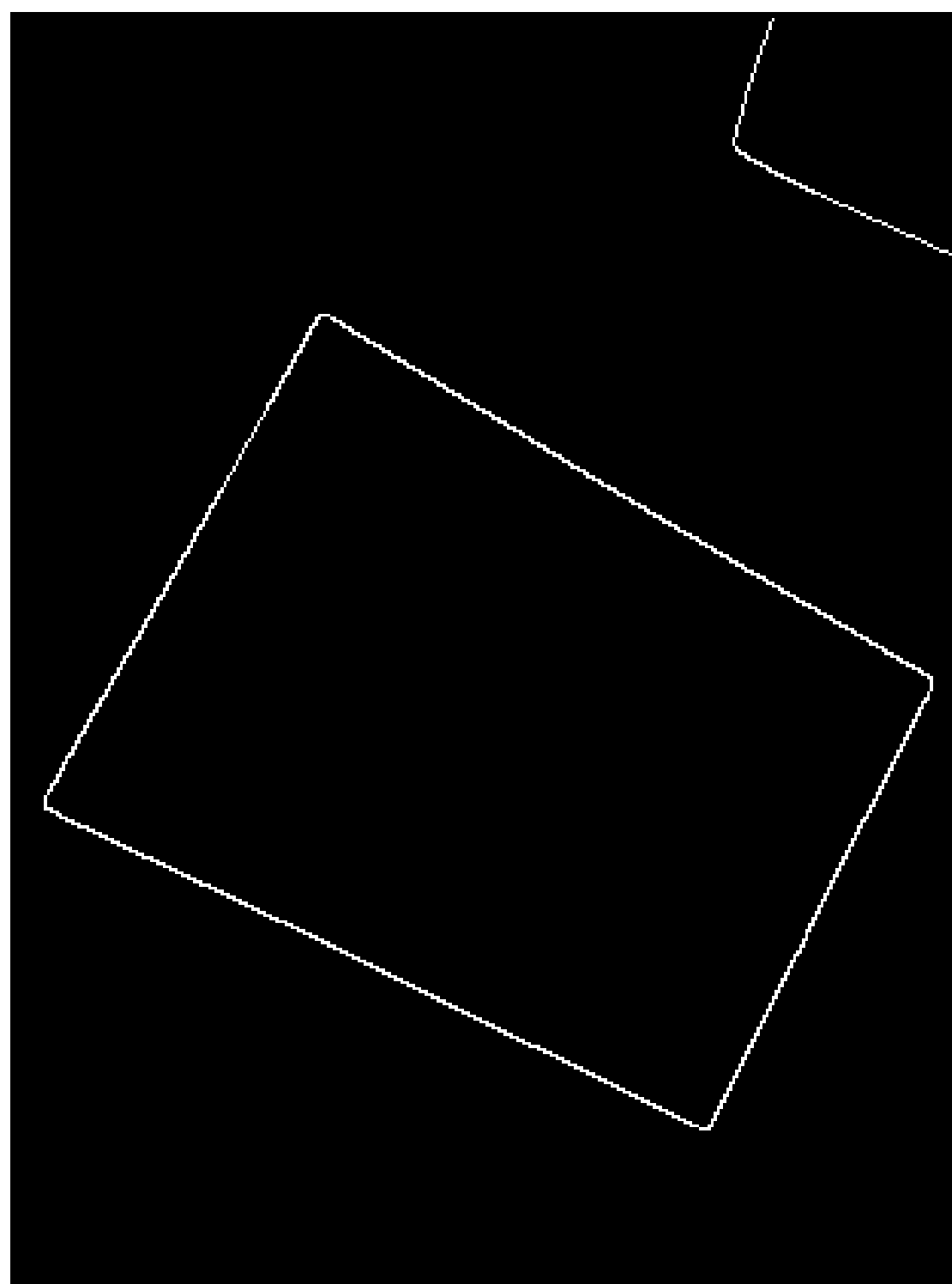


```
linear equation:
-0.0174524 * Y + 0.999848X = 60.7753
0.0436195 * Y + 0.999048X = 290.561
0.999048 * Y + 0.0436194X = 370.165
-0.99863 * Y + 0.052336X = -47.9875

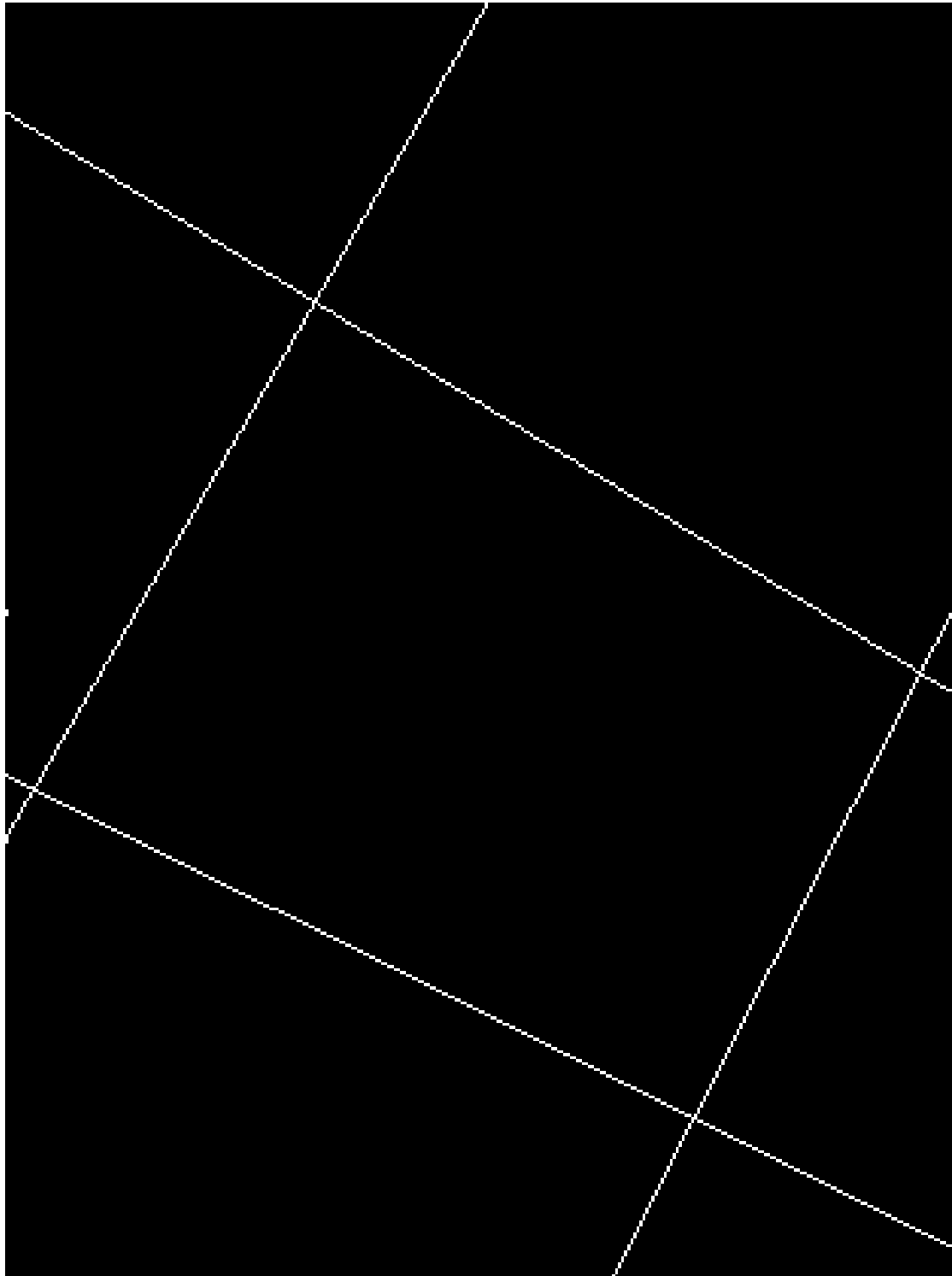
the conners coordinate:
<61,51>
<67,367>
<275,358>
<288,63>
```

第四张图片:

参数取值: sigma=2.0, threshold=10.0, in_thresh=223.0, out_thresh=0.50, theta_difference=0.2, ,
offset_difference =5, non_maximum_suppression=0





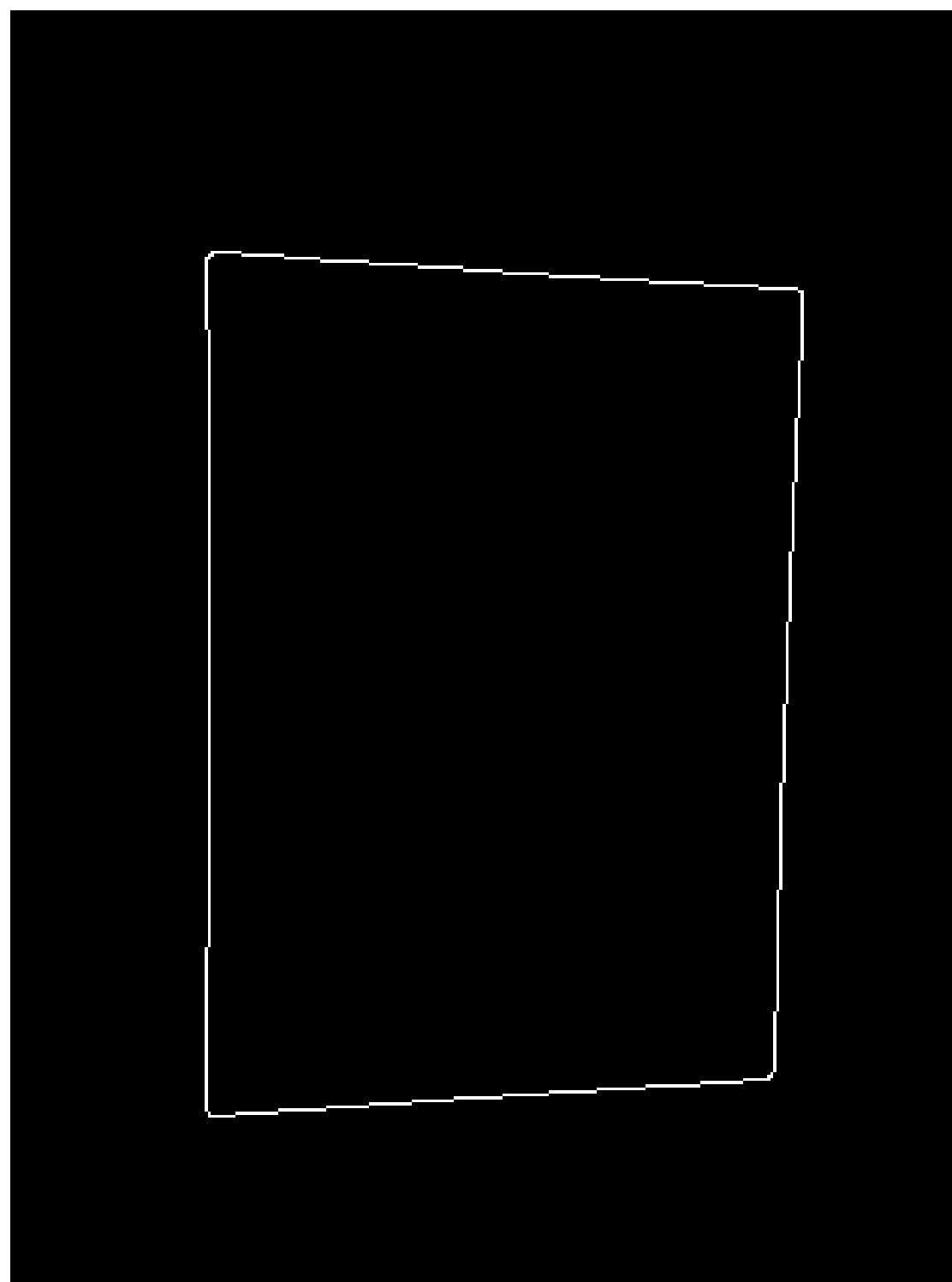


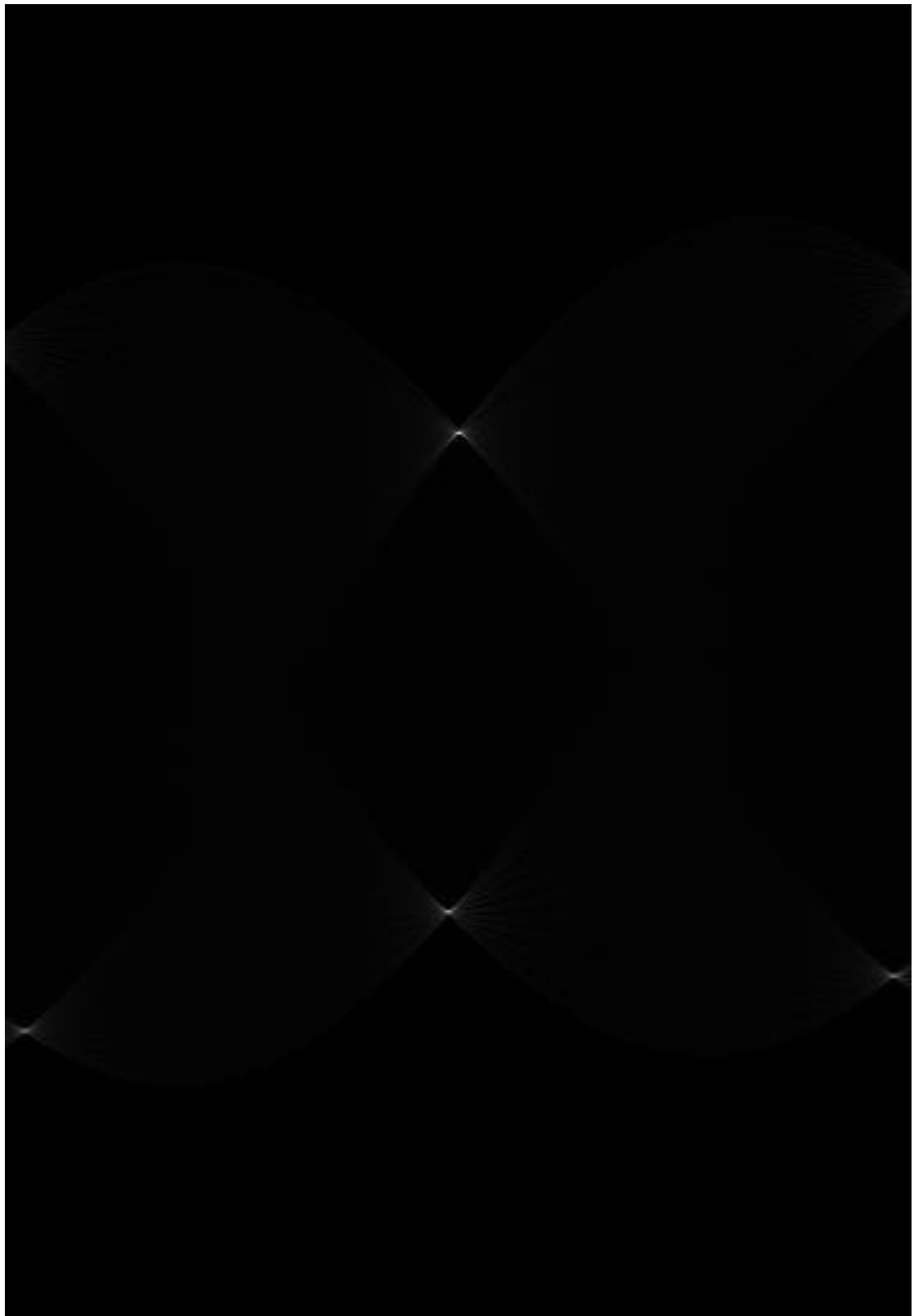
```
linear equation:
-0.894934 * Y + 0.446198X = -227.872
0.5 * Y + 0.866025X = 137.56
-0.85264 * Y + 0.522499X = -31.0185
0.453991 * Y + 0.891007X = 368.888

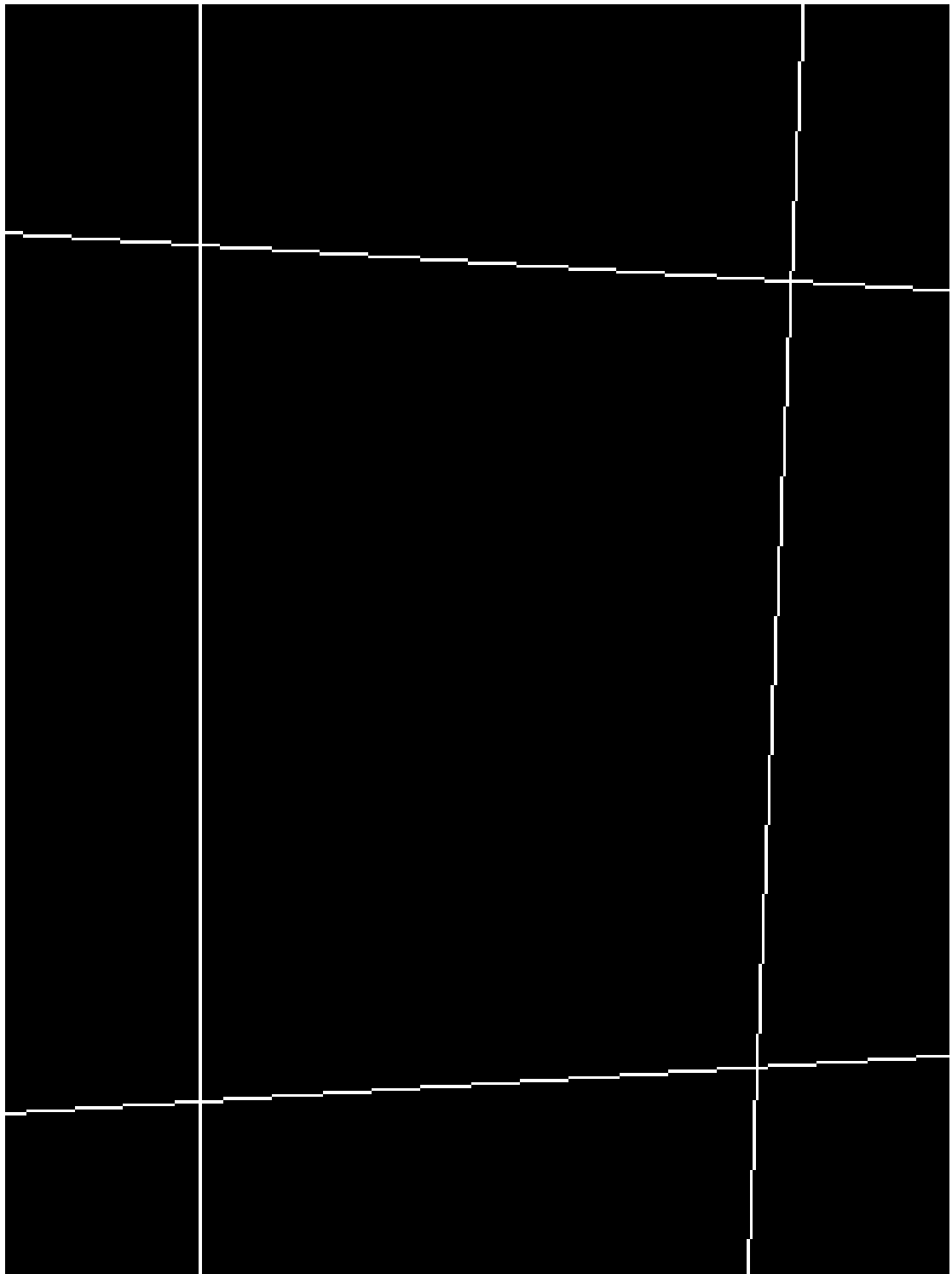
the conners coordinate:
<9,259>
<102,98>
<227,367>
<301,220>
```

第五张图片:

参数取值: sigma=2.0, threshold=15.0, in_thresh=223.0, out_thresh=0.75, theta_difference=0.1, ,
offset_difference =3, non_maximum_suppression=1





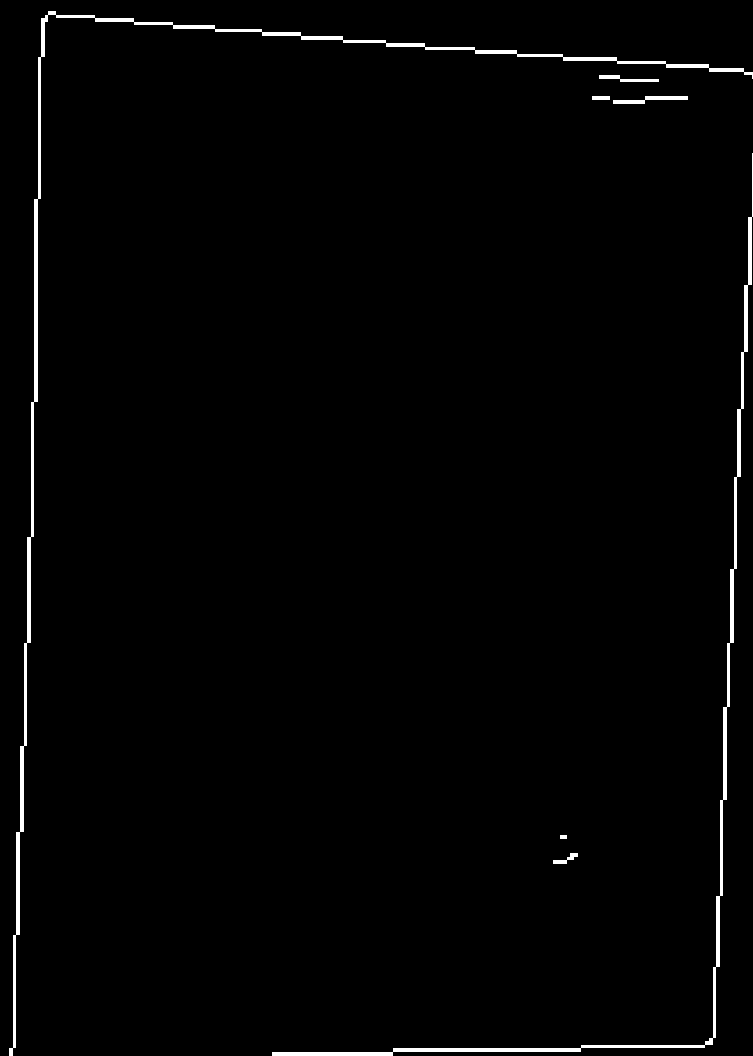


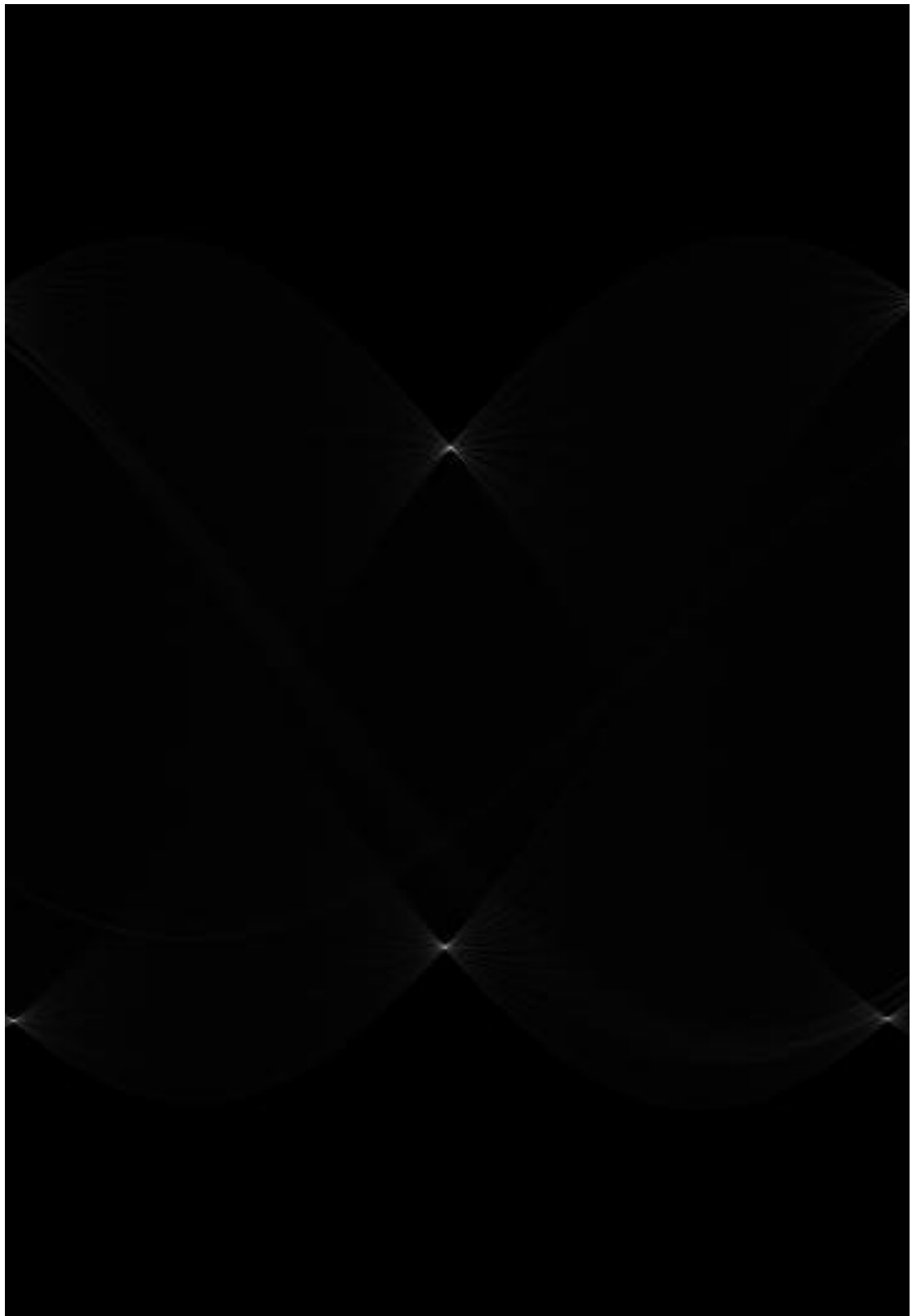
```
linear equation:
-4.37114e-008 * Y + 1X = 64.464
0.0436195 * Y + 0.999048X = 263.55
-0.998135 * Y + 0.0610486X = -75.5358
0.998135 * Y + 0.0610485X = 365.689

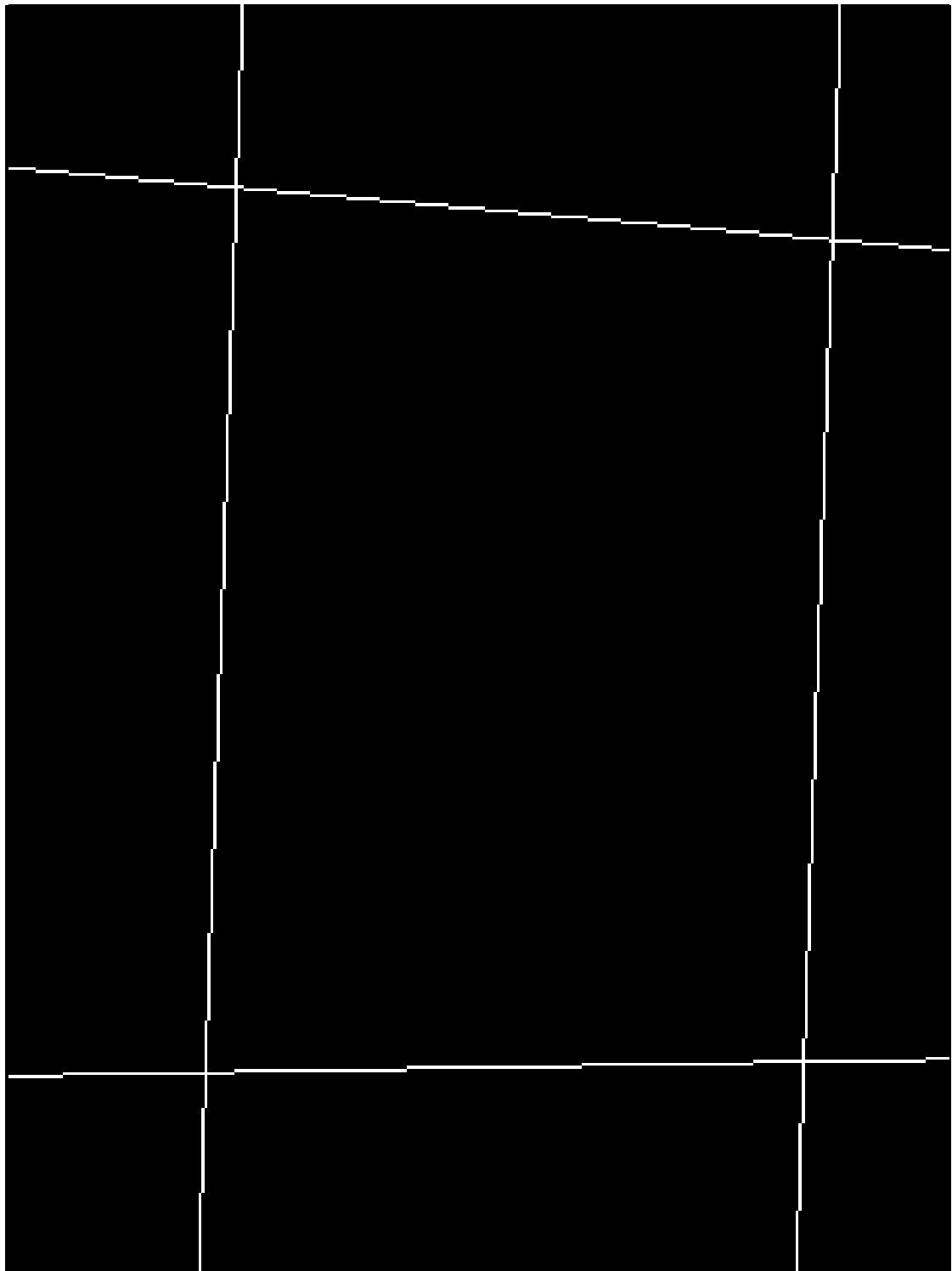
the conners coordinate:
(64,79)
(64,362)
(248,351)
(259,91)
```

第六张图片:

参数取值: sigma=2.0, threshold=10.0, in_thresh=223.0, out_thresh=0.50, theta_difference=0.2, ,
offset_difference =5, non_maximum_suppression=0







```
linear equation:
0.0348995 * Y + 0.999391X = 77.7003
0.0348995 * Y + 0.999391X = 275.778
-0.996195 * Y + 0.0871559X = -54.049
0.999848 * Y + 0.0174524X = 355.247

the conners coordinate:
<65,354>
<75,60>
<263,350>
<273,78>
```

总结：这次的作业很多坑，我做了很久很久，原因如下：

1. 一开始没有将图片进行下采样，导致每次跑出结果要等很久很久。
2. 要对每张图片的每个处理过程的每个参数进行调教，特别痛苦。
3. 我是在 Windows 下做的作业，没有 makefile 的参与，导致每次修改参数后进行编译要等一分钟