

## 實用技巧| OpenCV快速計算直線/網格線交點坐標(附源碼)

原創 Color Space OpenCV與AI深度學習 今天

收錄於話題

#OpenCV

27個 >



### 導讀

本期將介紹使用OpenCV用兩種不同的方法實現快速查找計算直線/網格線交點坐標。

## 直線交點計算思路與常用方法

直線交點的計算這裡列舉幾個比較常用的方法：

- ①在知道直線方程的前提下(或知道直線上一點和直線角度)，聯立方程求解交點坐標(注意數學坐標系和圖像坐標系的關係)；
- ②不知道直線方程，通過檢測直線的方法(例如霍夫變換)計算直線方程，然後聯立方程求解交點坐標；

第一種方法當然是最理想的情況，這樣計算的交點也最準確。但是大多數情況我們並不能預先知道直線方程。第二種方法在干擾少的情況比較適用，當然我們還要考慮OpenCV的直線檢測方法可能會有重複直線被檢測，還需要做剔除操作。

綜上來看，前面兩種方法並不是非常好用，下面來介紹兩種簡單實用的方法：

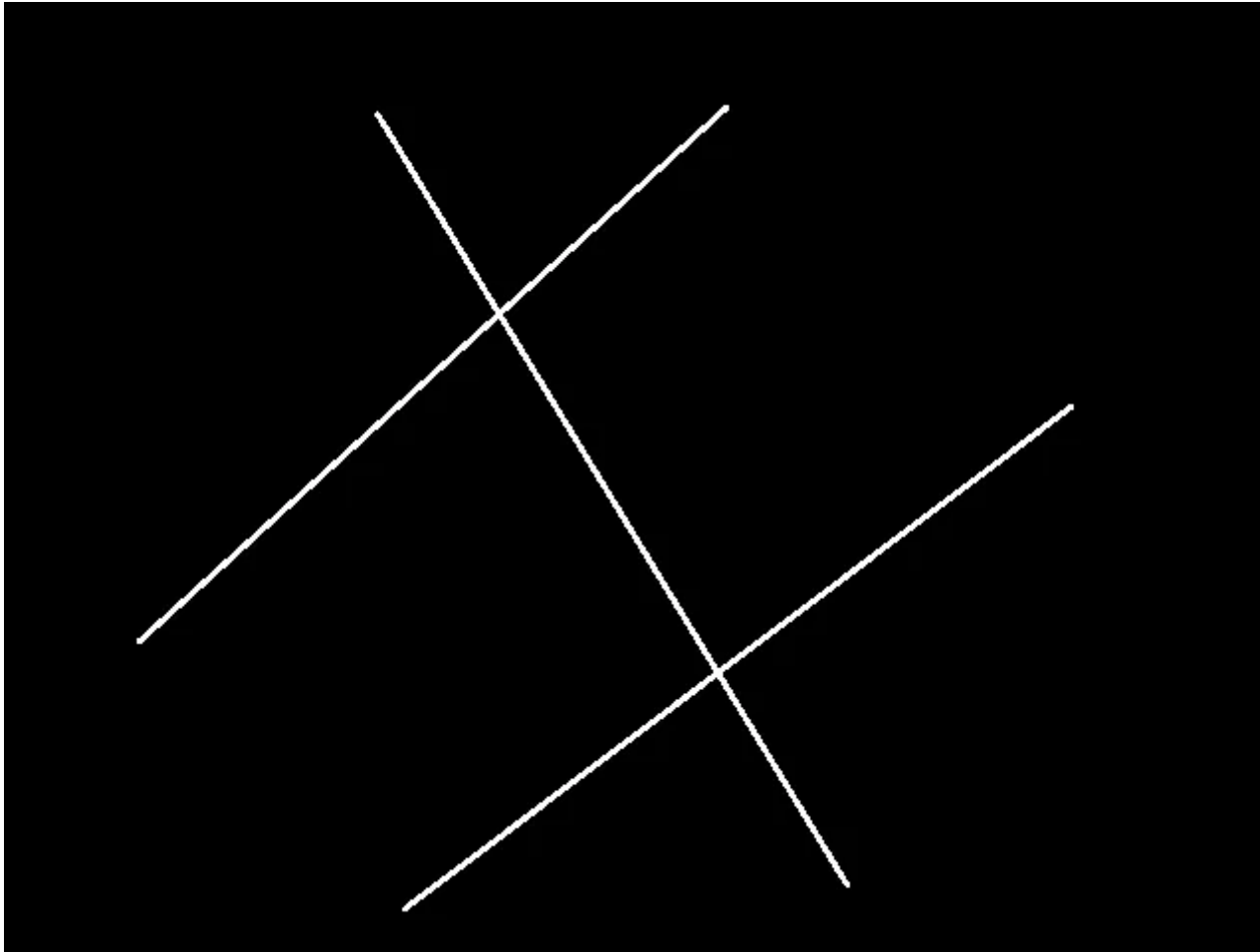
- ③ 利用角點檢測方法粗定位角點，然後通過形態學操作計算出直線交點位置；
- ④利用直線交點出特徵，使用形態學操作直接定位直線交點位置。

本文使用方法③和方法④，設置合適的參數即可檢測到直線交點位置。

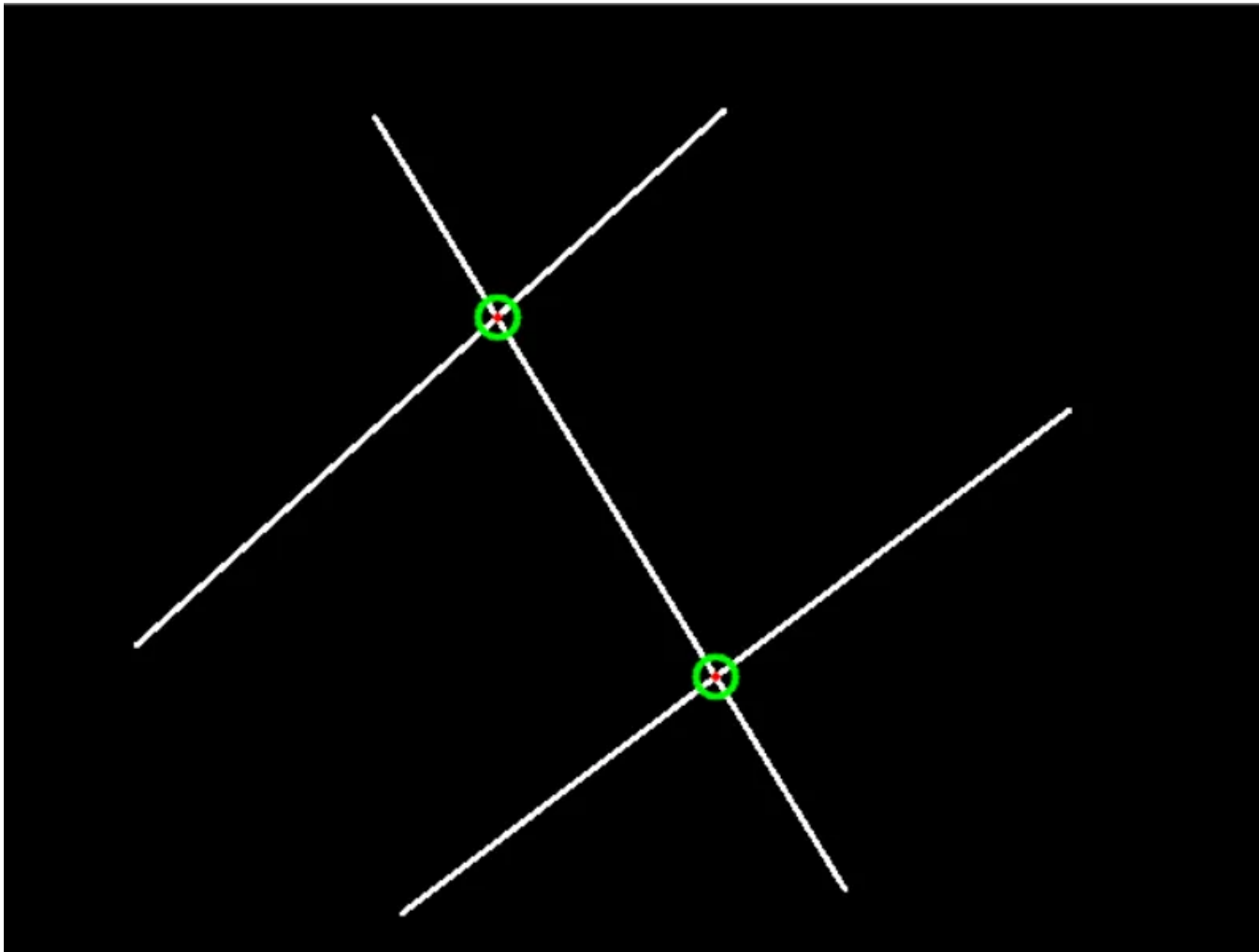
## 效果展示

本文演示圖片包含2張，一張是簡單的三條直線相交，另一張是多條直線相交形成網格，分別計算它們的交點位置並標識，效果如下：

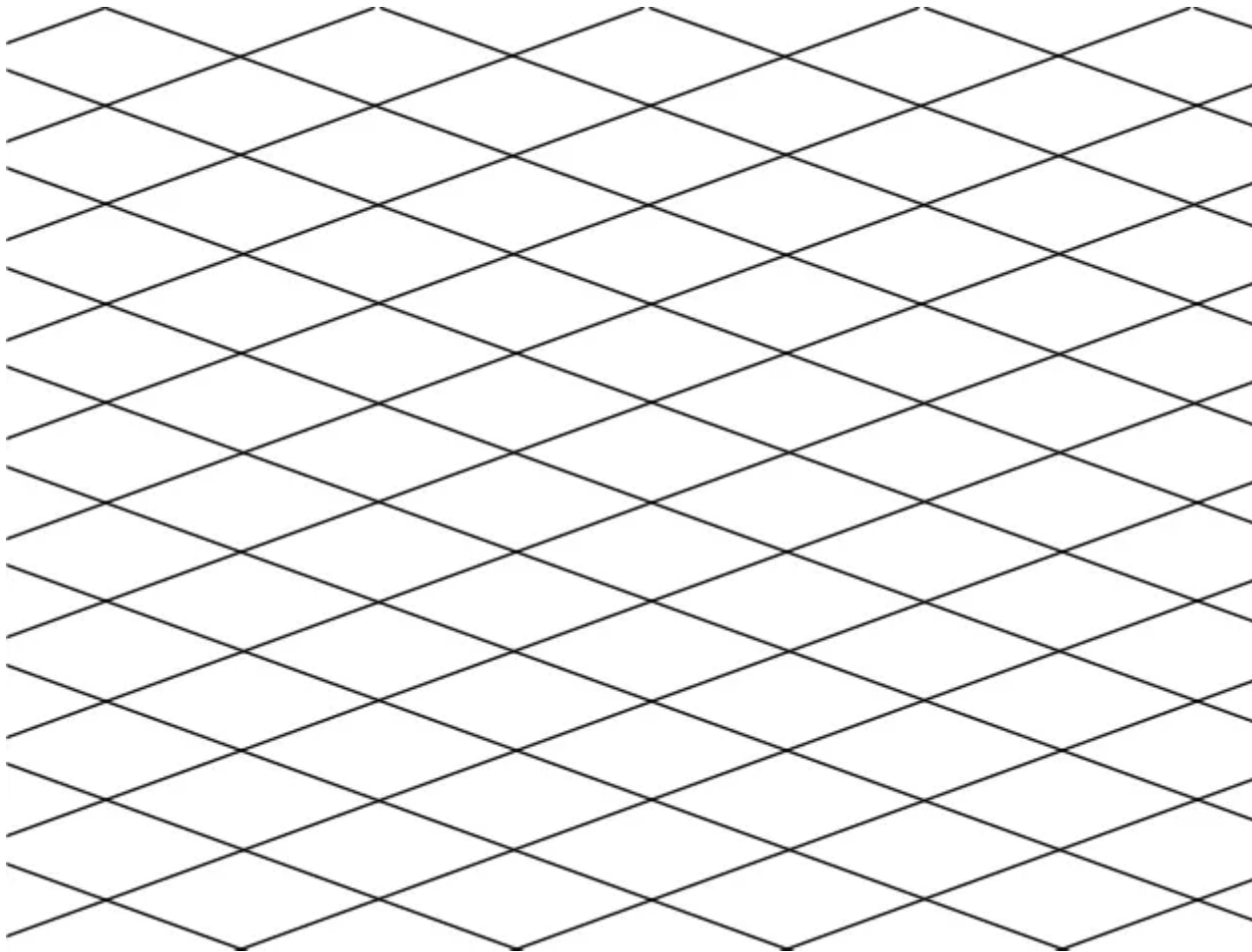
原圖：



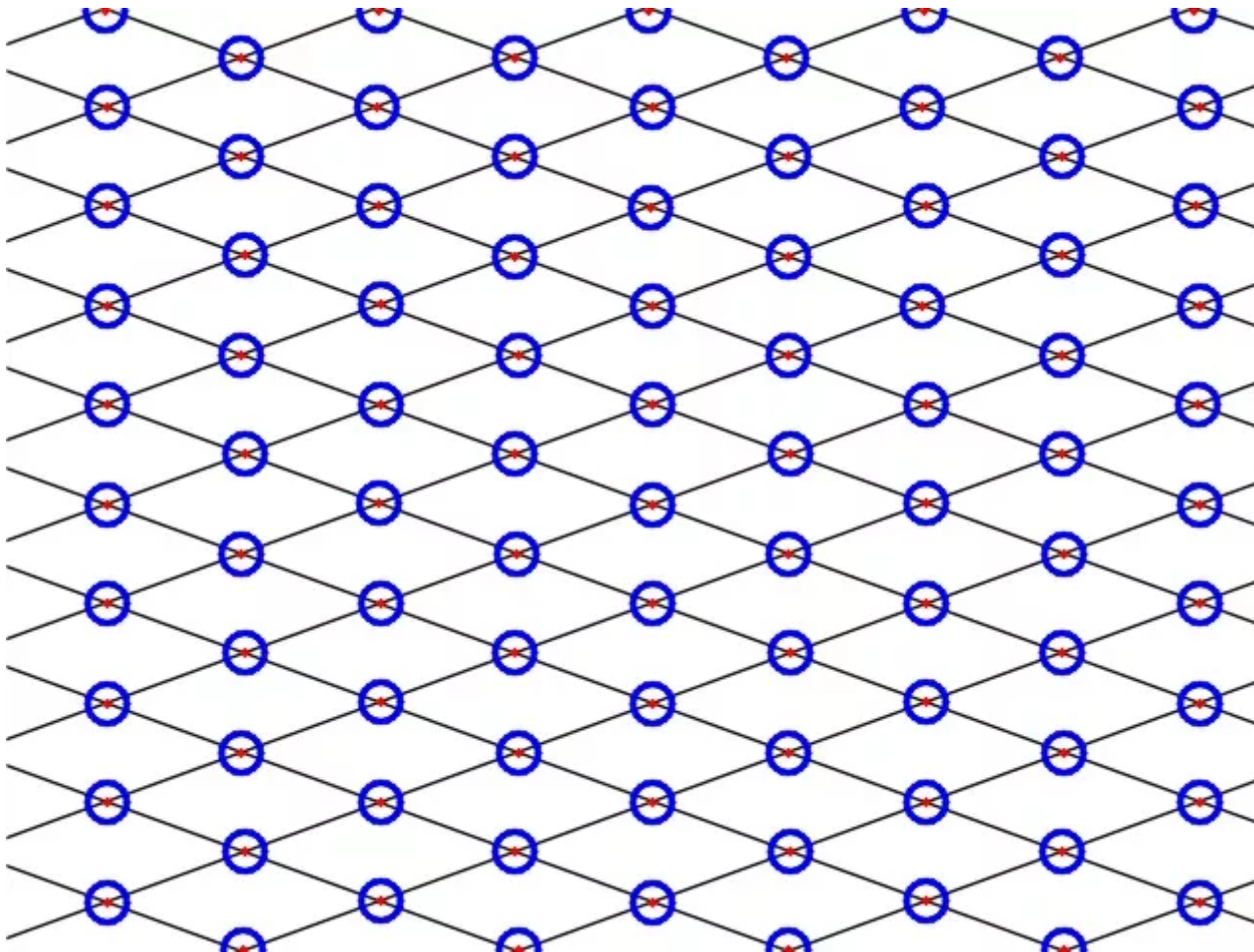
結果圖：



原圖：



結果圖：



## 實現步驟與對應代碼

方法③實現步驟與代碼：

(1) Harris角點檢測+閾值處理；

```
1 import cv2
```

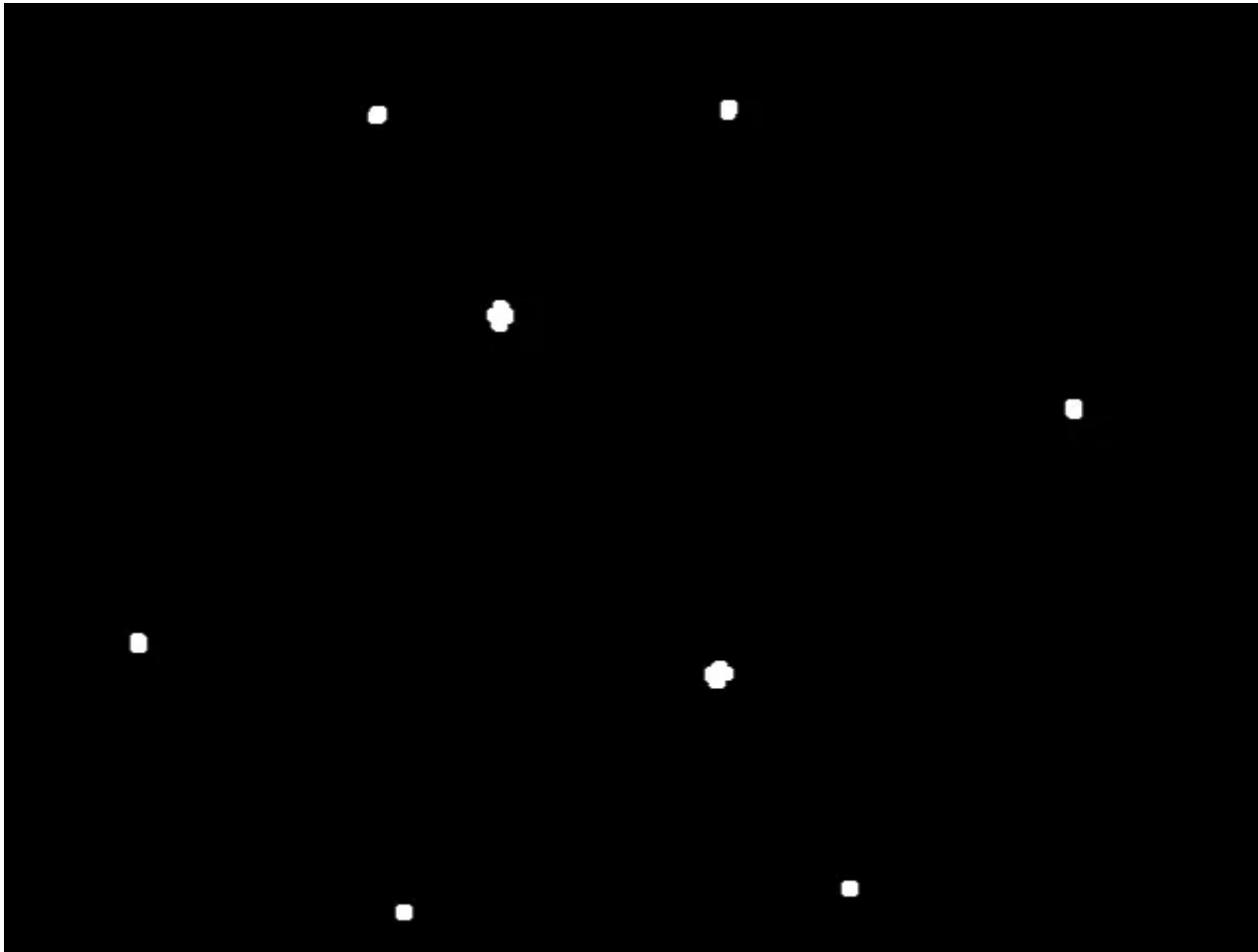
```
2 import numpy as np
3
4 img = cv2.imread('1.png')
5 gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
6 mask = np.zeros(gray.shape,np.uint8)
7 dst = cv2.cornerHarris(gray,2,3,0.04)
8 ret,thresh = cv2.threshold(dst,0.01,255,cv2.THRESH_BINARY)
9 cv2.imshow('thresh',thresh)
```



(2) 形態學膨脹(將角點連接起來)；

```
1 k1=np.ones((5,5), np.uint8)#指定膨脹核大小
2 mask = cv2.morphologyEx(mask, cv2.MORPH_DILATE, k1)
```

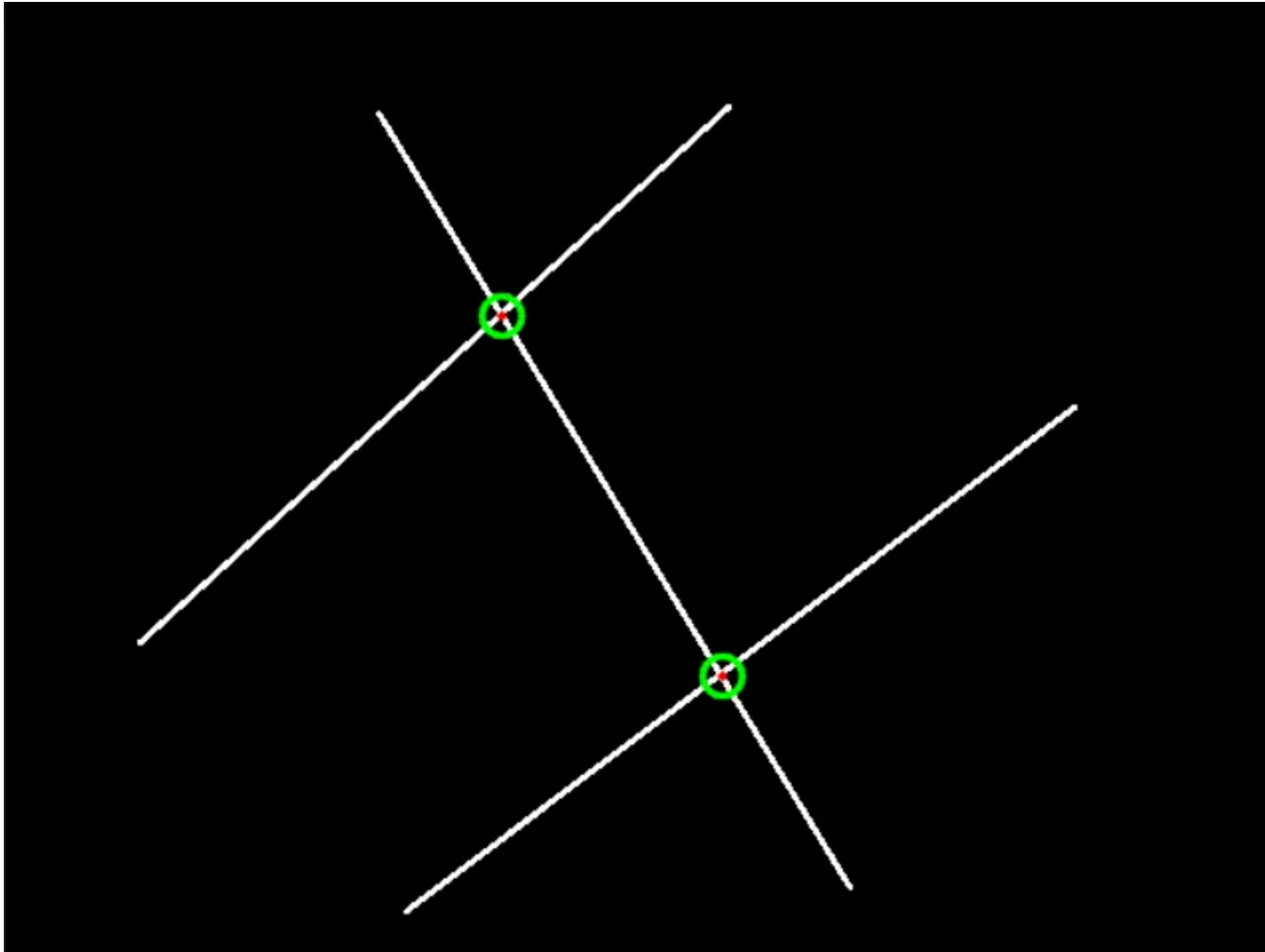




(3)輪廓篩選， $w > 10$  and  $h > 10$ ，直線交點位置輪廓更大。

```
1 contours,hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
2 for cnt in contours:
3     (x, y, w, h) = cv2.boundingRect(cnt)
4     if w > 10 and h > 10:
5         cx = int(x + w / 2)
6         cy = int(y + h / 2)
```

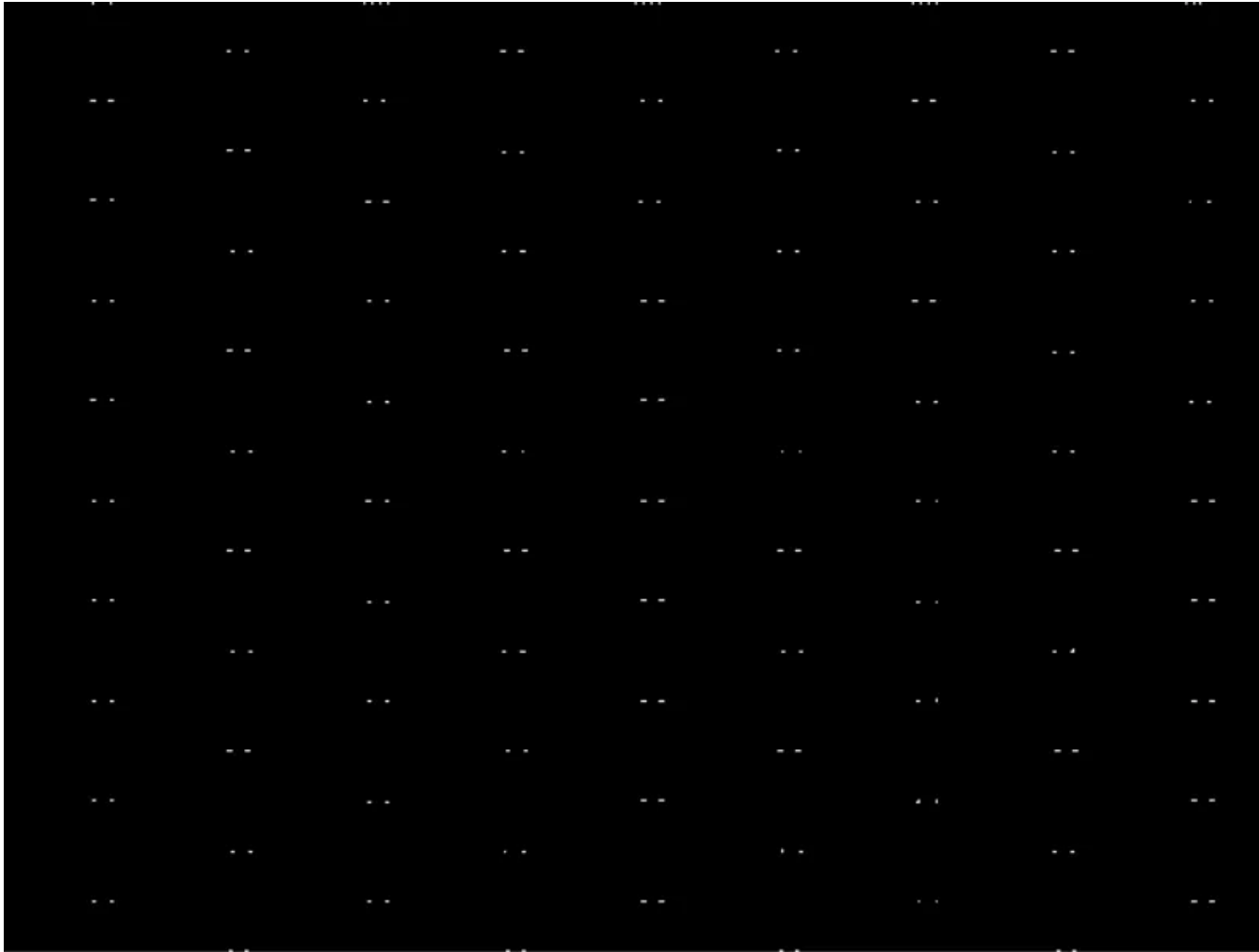
```
7 cv2.circle(img,(cx, cy),10,(0,255,0),2, cv2.LINE_AA)  
8 cv2.circle(img,(cx, cy),2,(0,0,255),-1, cv2.LINE_AA)
```

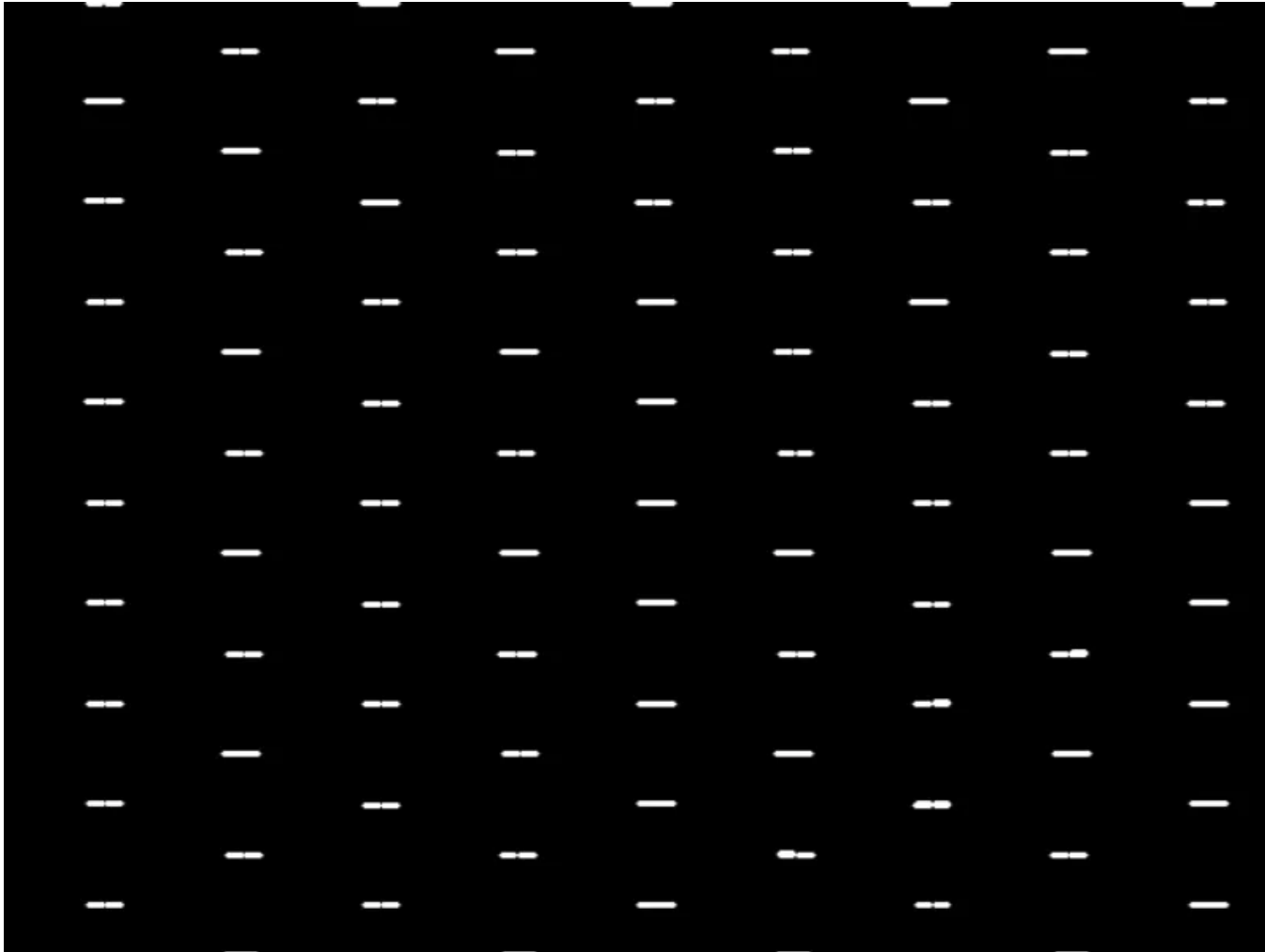


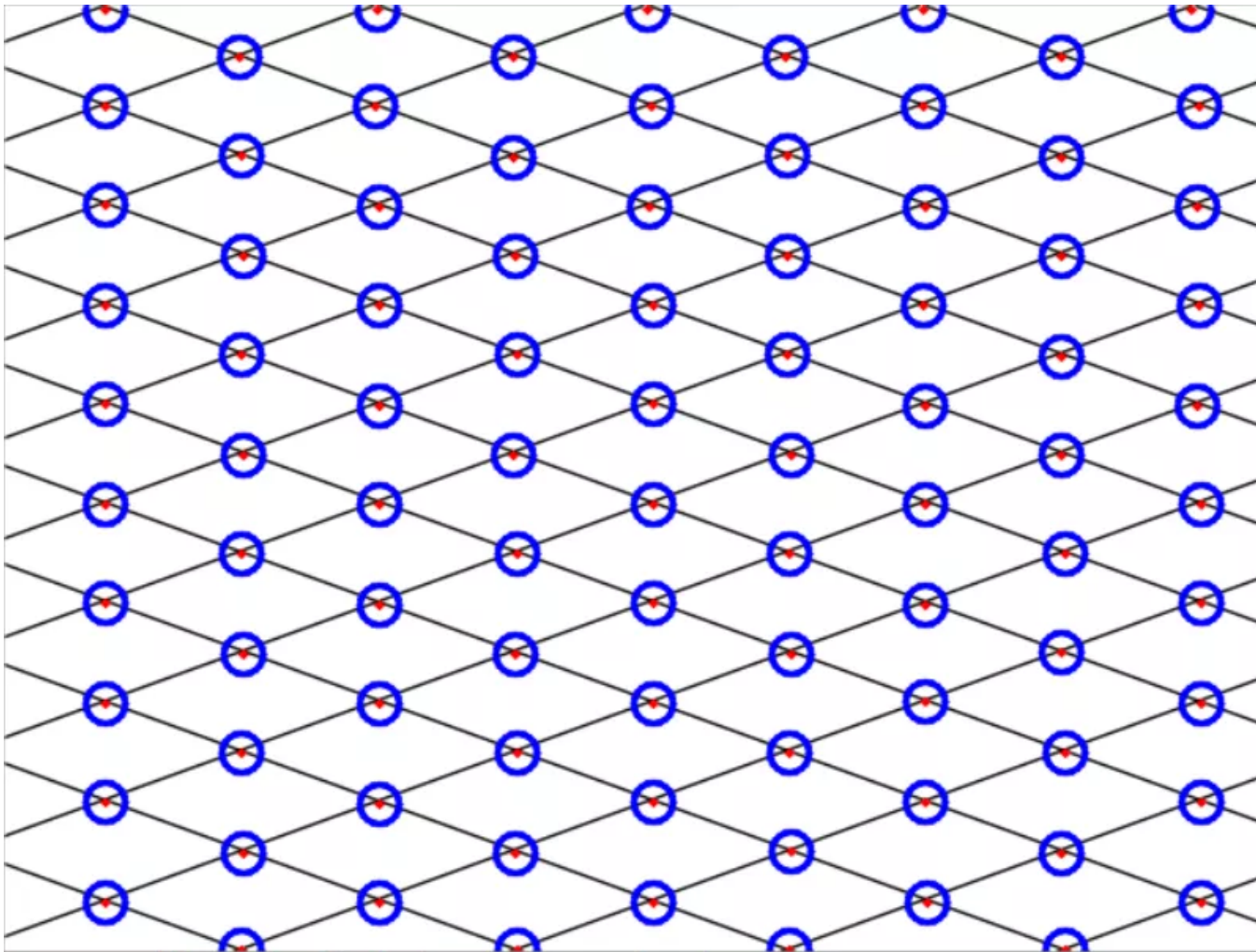
對於圖2，代碼修改兩處參數：

```
1 k1=np.ones((1,7), np.uint8)  
2 .....
```

```
3 if w > 1 and h > 1:  
4 .....
```





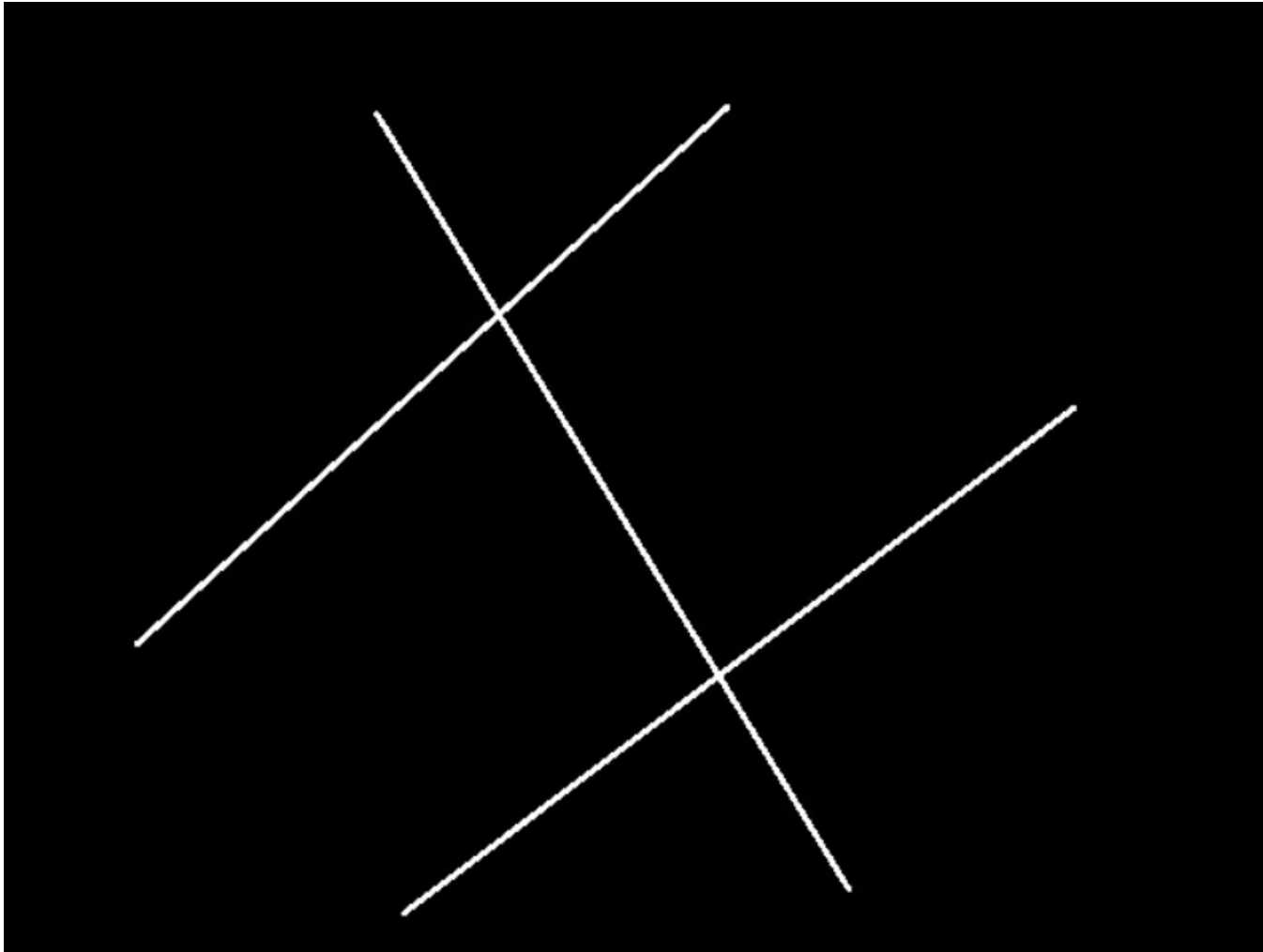


方法④實現步驟與代碼：

(1) 固定閾值二值化凸顯直線為白色，背景為黑色；

```
1 import numpy as np
2 import cv2
3
4 img= cv2.imread('1.png')
```

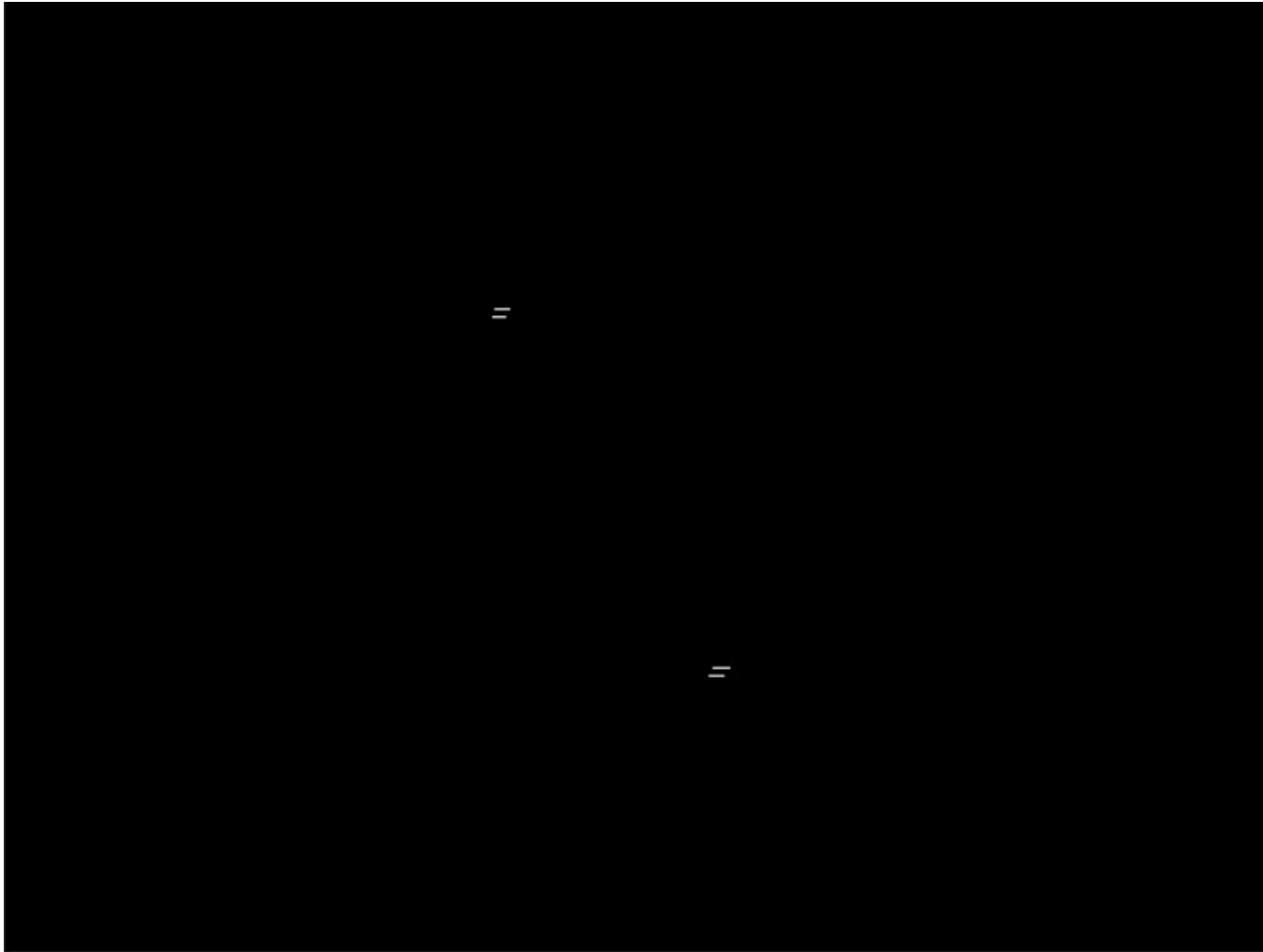
```
5 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
6 ret, thres= cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY)
7 cv2.imshow('thres', thres)
```



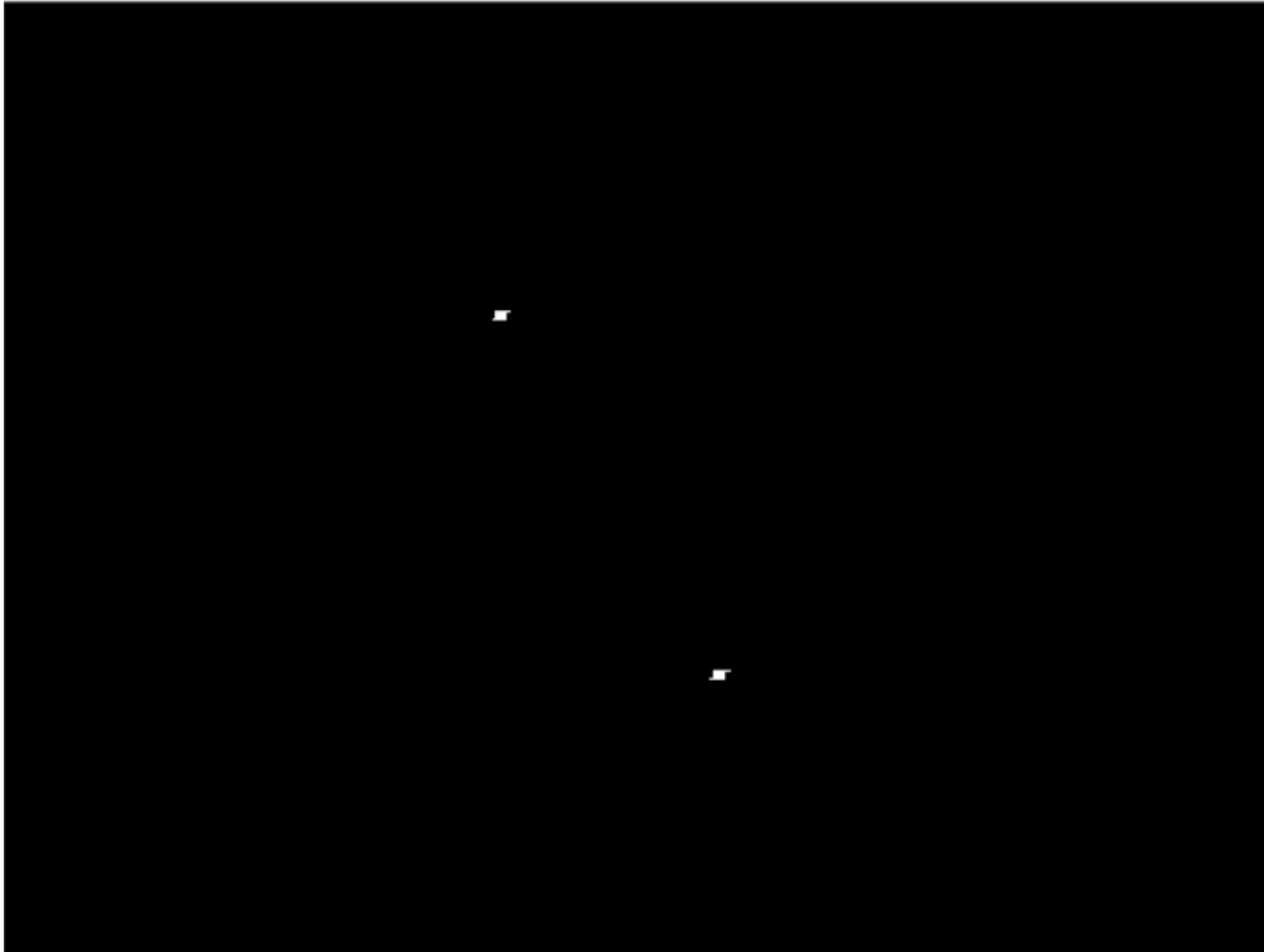
(2)指定核做開運算，再做閉運算；

```
1 k1=np.ones((1,7), np.uint8)
```

```
2 k2=np.ones((7,1), np.uint8)
3
4 mask = cv2.morphologyEx(thres, cv2.MORPH_OPEN, k1)
5 cv2.imshow('open', mask)
6
7 mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, k2)
8 cv2.imshow('close', mask)
```



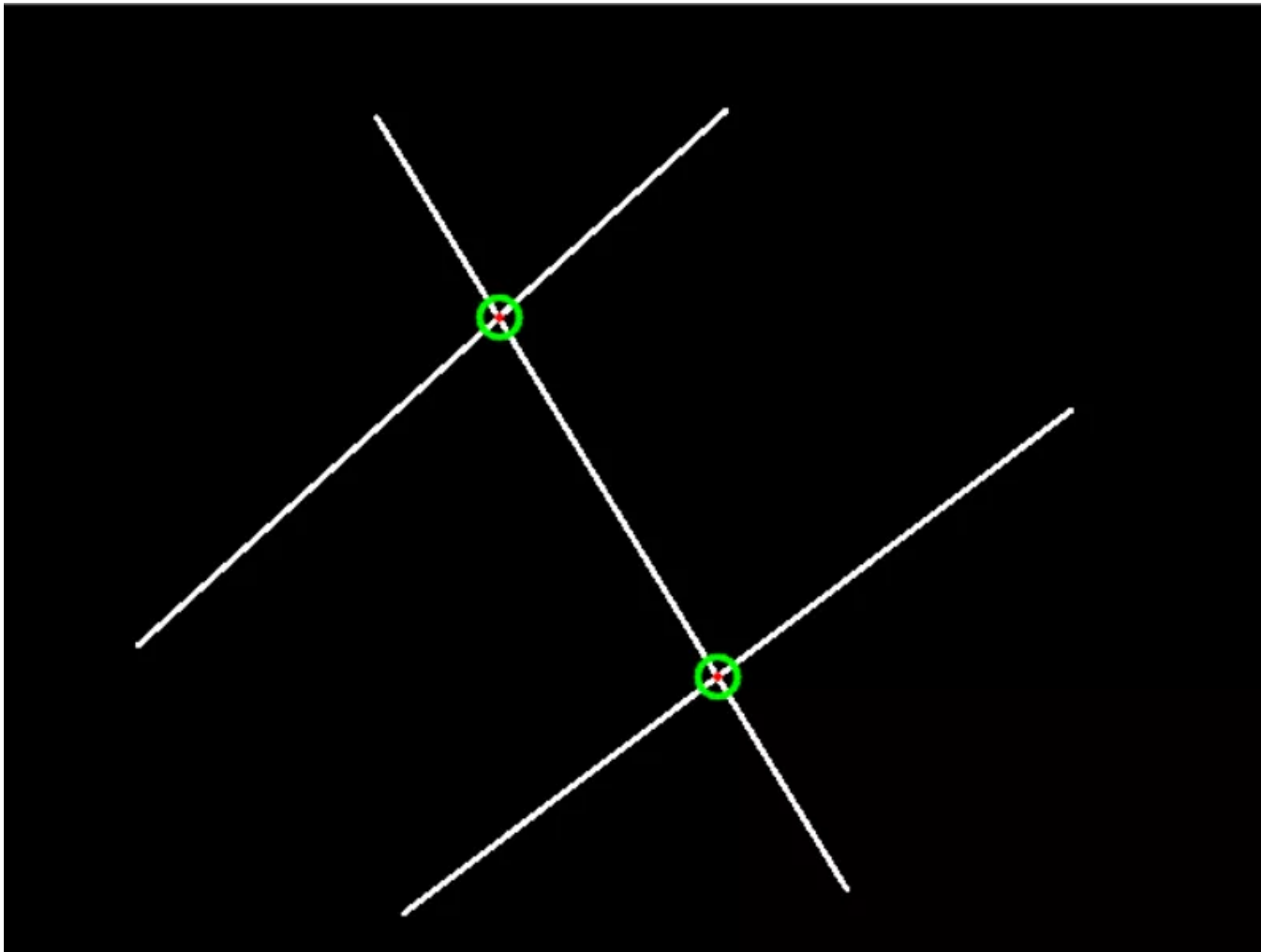




(3)輪廓查找與中心計算標示；

```
1 contours,hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
2 for cnt in contours:
3     (x, y, w, h) = cv2.boundingRect(cnt)
4     #cv2.rectangle(img, (x, y),(x+w,y+h),(0, 255, 0), 2)
5     cx = int(x + w / 2)
```

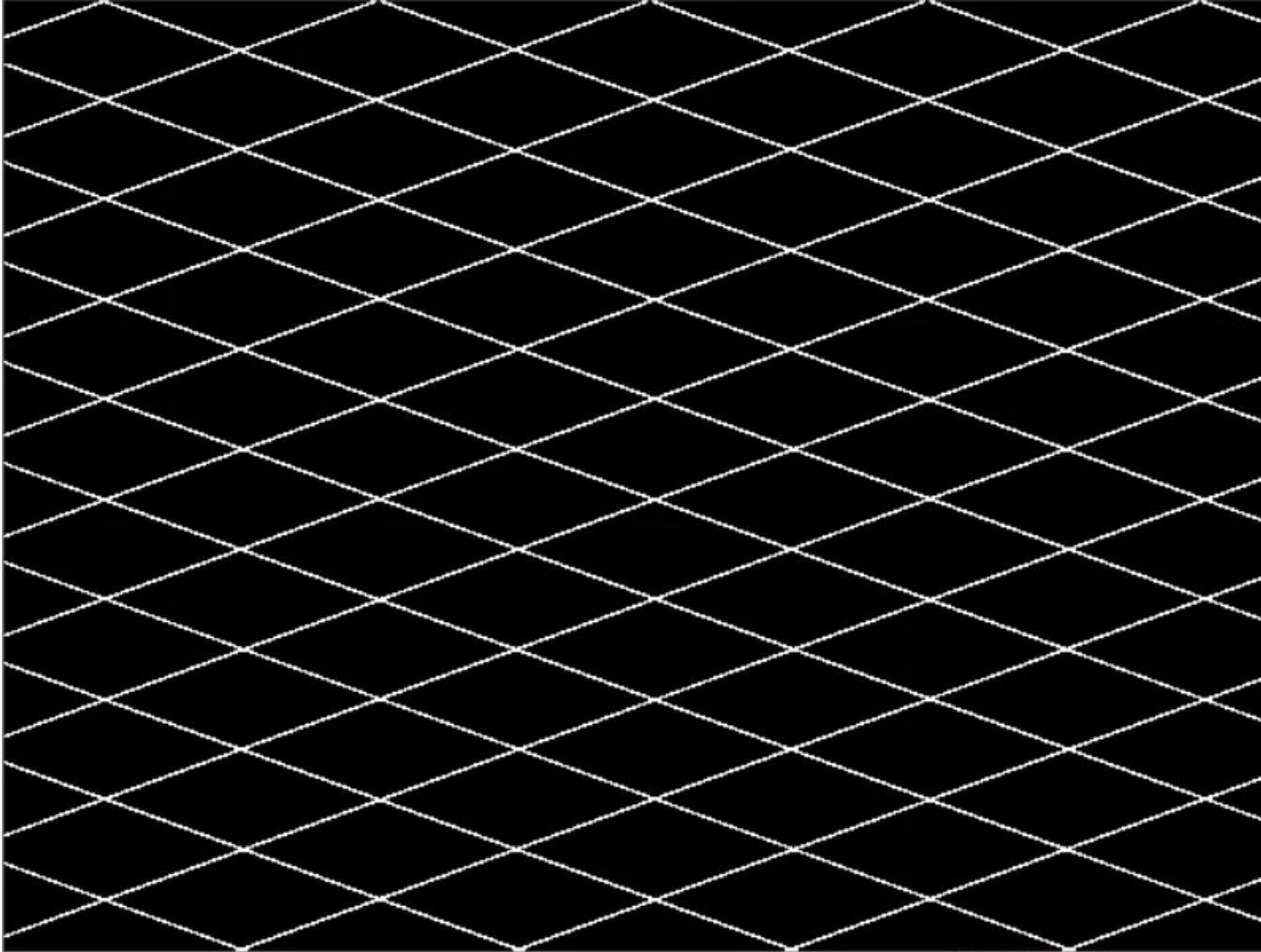
```
6  cy = int(y + h / 2)
7  cv2.circle(img,(cx, cy),10,(0,255,0),2, cv2.LINE_AA)
8  cv2.circle(img,(cx, cy),2,(0,0,255),-1, cv2.LINE_AA)
```



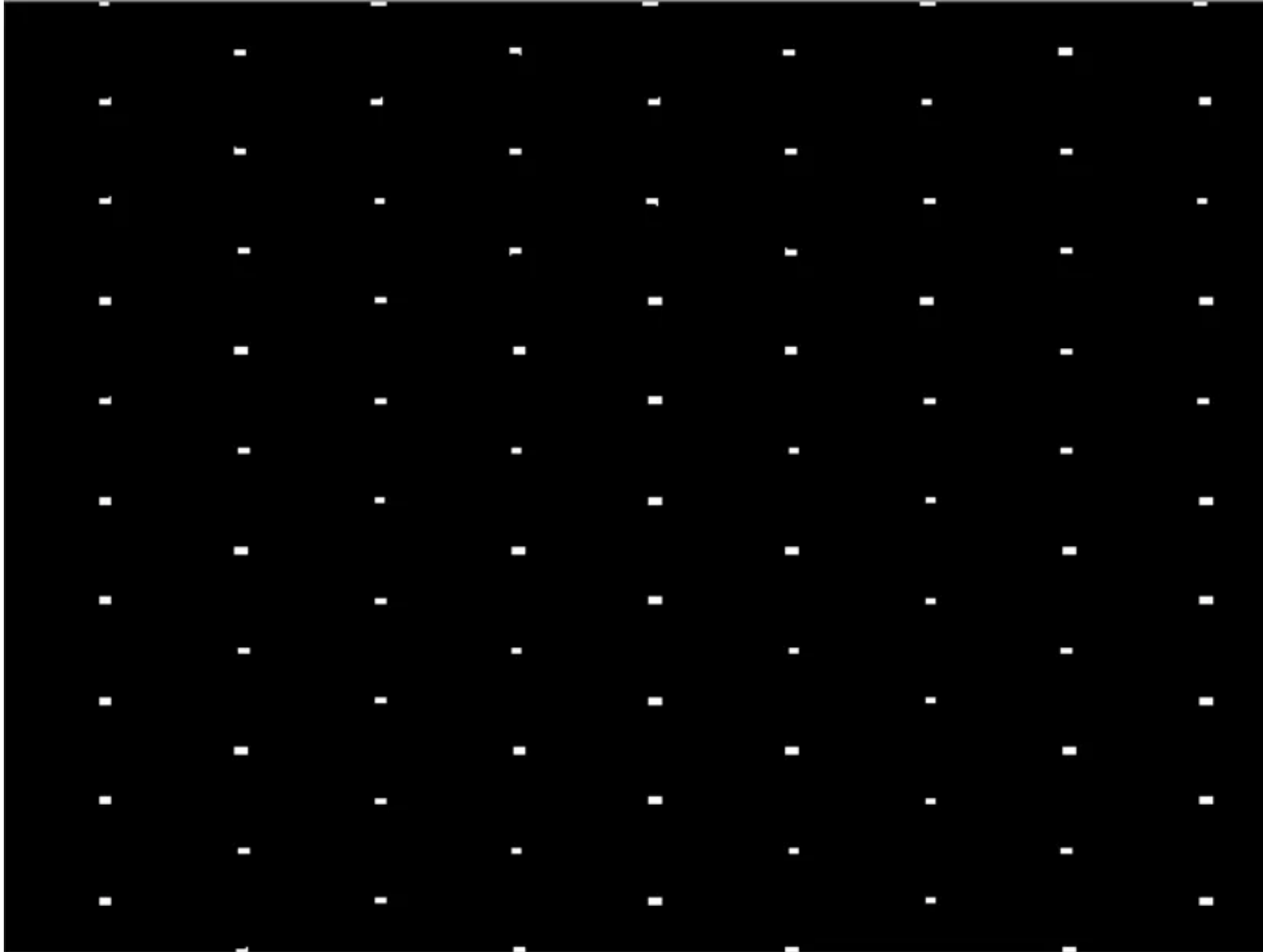
對於圖2，代碼修改兩處參數：

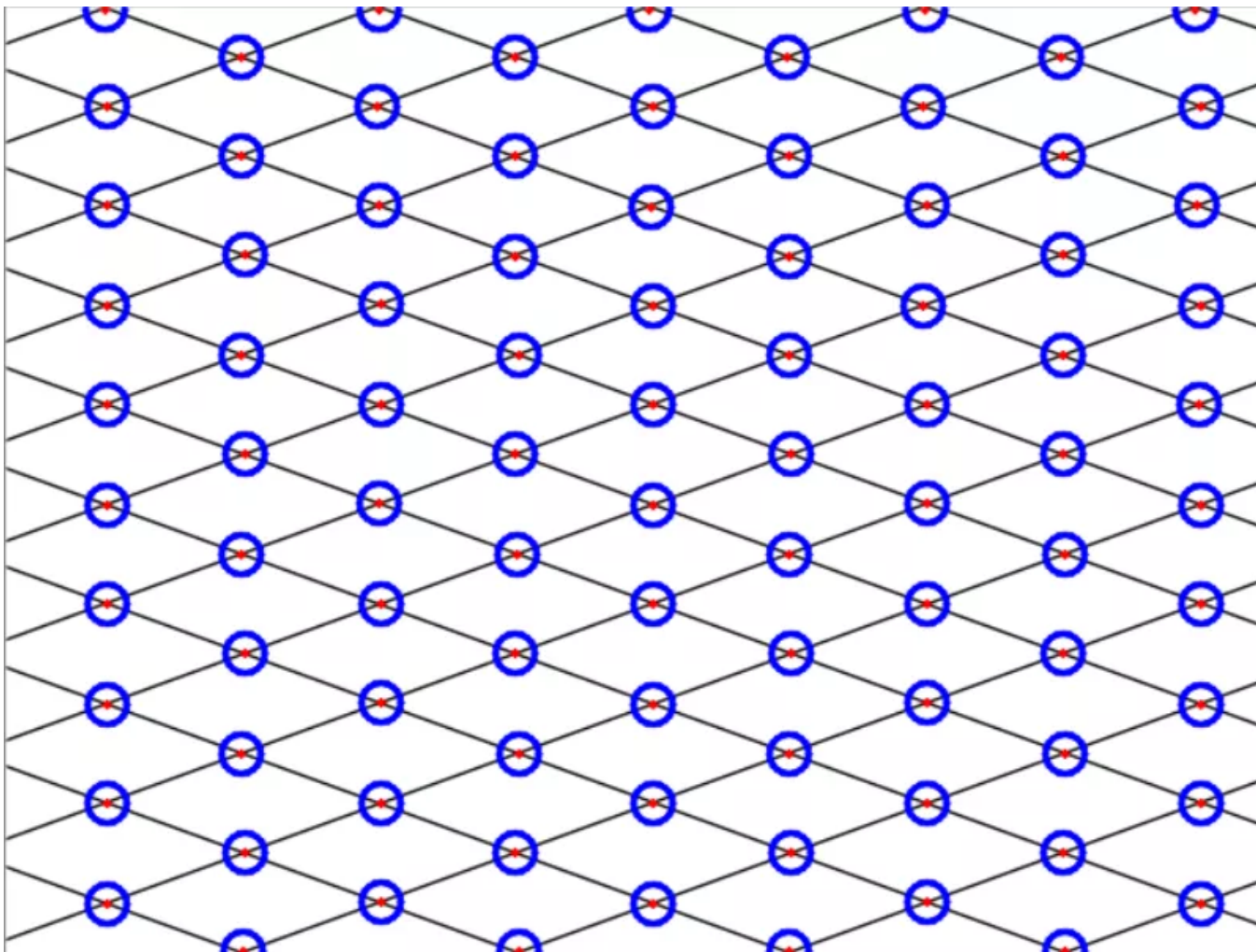
```
1  ret, thres= cv2.threshold(gray, 150, 255, cv2.THRESH_BINARY_INV)
```

```
2 .....  
3 k1=np.ones((3,1), np.uint8)  
4 k2=np.ones((1,7), np.uint8)
```









此處涉及到的形態學技巧原來有介紹過，可以參看下面鏈接查看：

[OpenCV形態學處理使用技巧與應用演示](#)

結尾語：

文章實現的步驟並非最優，但基本檢測效果還OK，大家跟著上述步驟和代碼完全可以自己做出來。當然如果你懶得動手，那就加入知識星球獲取吧，那里為你準備了完整源碼和測試素材，還有C++源碼。

更多實用文章請移步至--[精華文章](#)--[專題文章分類](#)。

更多視覺圖像處理相關內容，請長按關注：[OpenCV與AI深度學習](#)。



## OpenCV與AI深度學習

致力於OpenCV、Halcon、人工智能、深度學習相關內容的學習、交流、分享！

99篇原創內容



公眾號

覺得有用，麻煩給個贊和在看



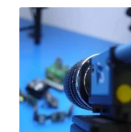
收錄於話題# OpenCV·27個 >

下一篇·手把手教你用OpenCV做人臉口罩佩戴檢測(附詳細步驟+代碼) >

喜歡此內容的人還喜歡

### 基於正樣本的表面缺陷檢測

新機器視覺



### 乾貨| 使用圖像分割來做缺陷檢測的一個例子

新機器視覺



## 基於OpenCV實戰：3步實現圖像降噪

小白學視覺

