Digital Image Processing 2019 Spring

B04902083 Hsiang-Hsu Chuang

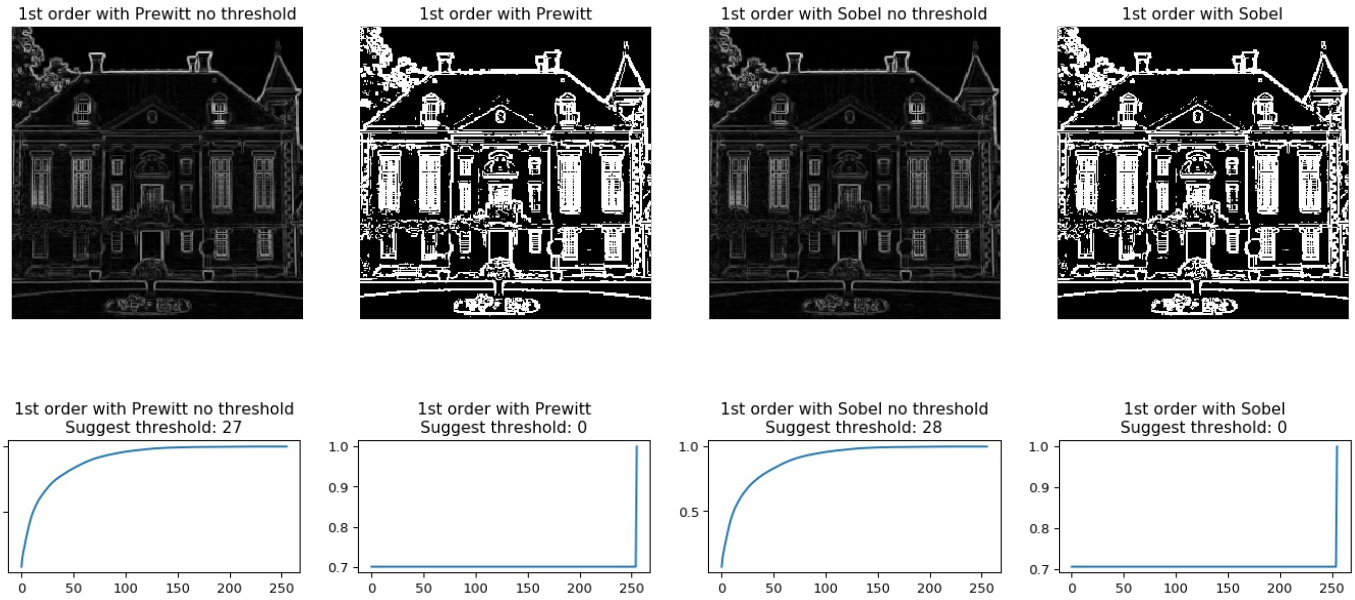
PROBLEM 1: EDGE DETECTION

對三張圖片做 edge detection，一張原圖，一張有 uniform noise, 一張有 impluse noise

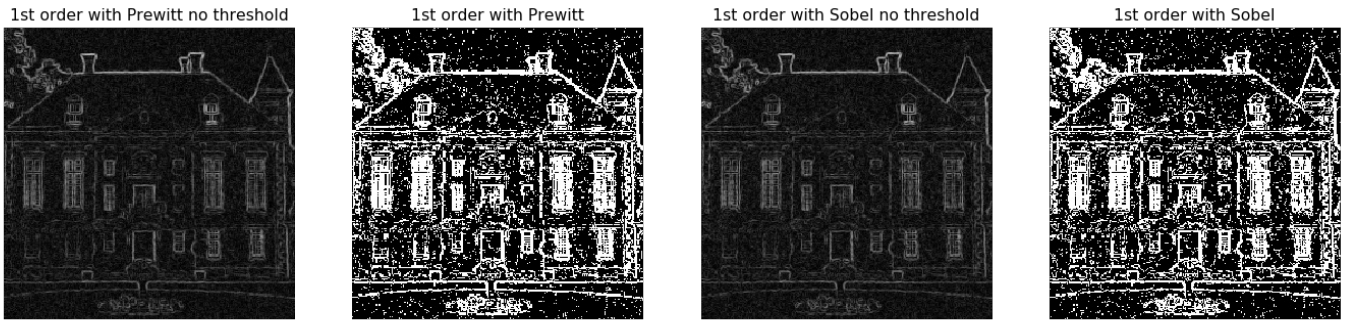
Discussion of results

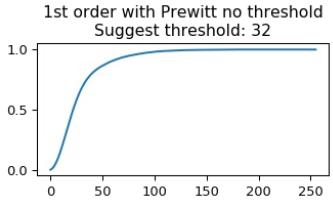
1. Perform 1st order edge detection and output the edge map as E1
   都是先套上 mask，然後先生成原圖計算 cdf > 0.7 的地方是多少，在以此作為 threshold，Prewitt 和 Sobel 兩個 mask 比較後覺得沒有差太多。

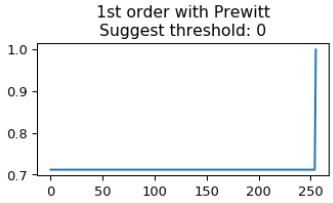
smaple1

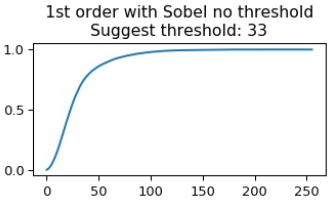


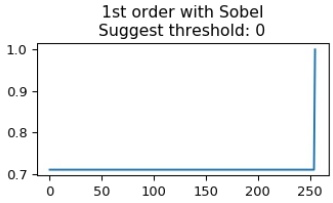
smaple2



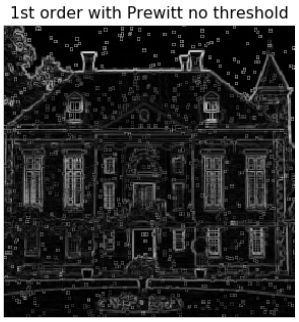

1st order with Prewitt no threshold
Suggest threshold: 32

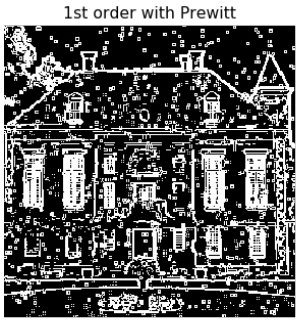

1st order with Prewitt
Suggest threshold: 0


1st order with Sobel no threshold
Suggest threshold: 33

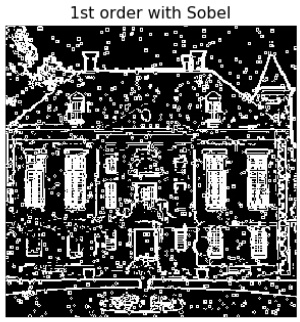

1st order with Sobel
Suggest threshold: 0

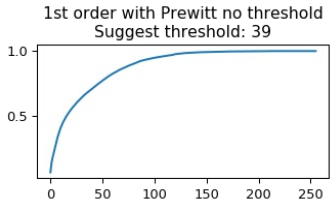
smaple3

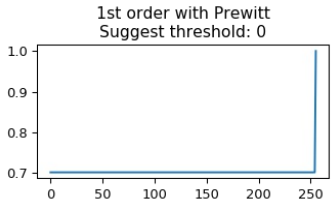

1st order with Prewitt no threshold

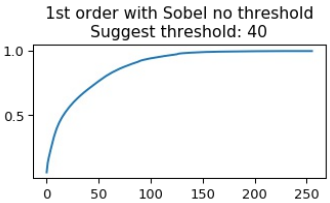

1st order with Prewitt

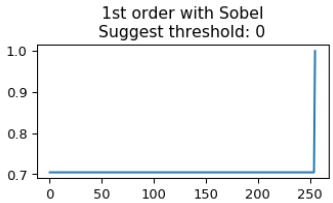

1st order with Sobel no threshold


1st order with Sobel


1st order with Prewitt no threshold
Suggest threshold: 39

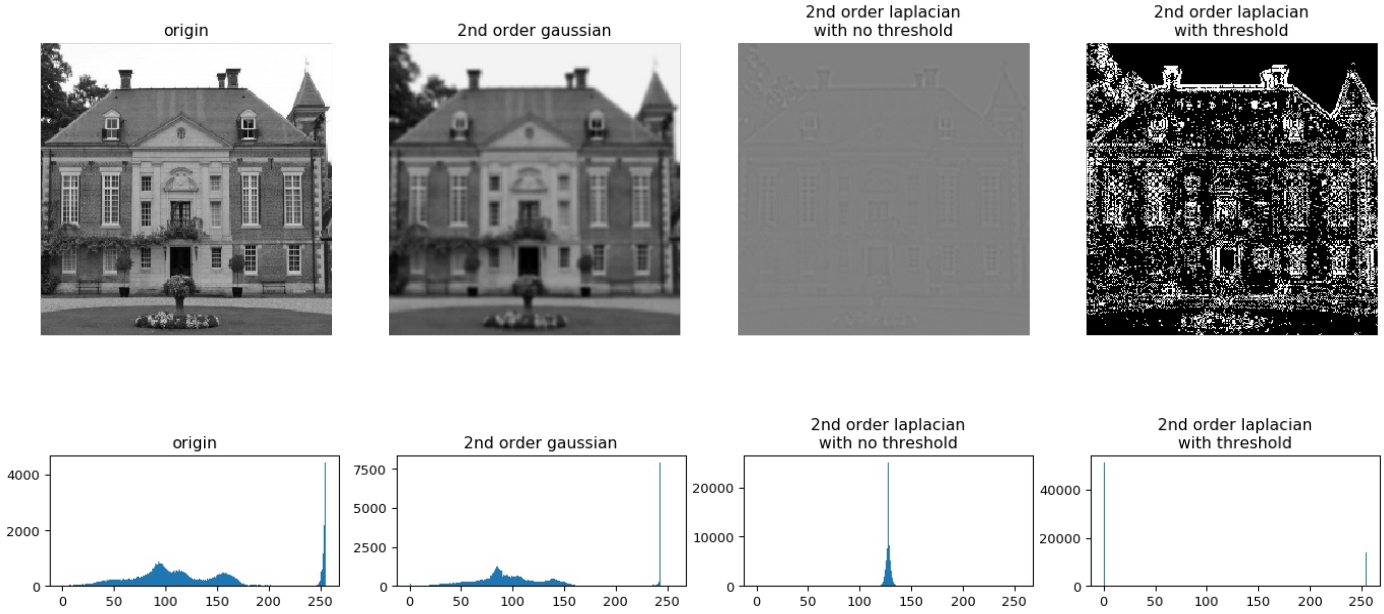

1st order with Prewitt
Suggest threshold: 0


1st order with Sobel no threshold
Suggest threshold: 40

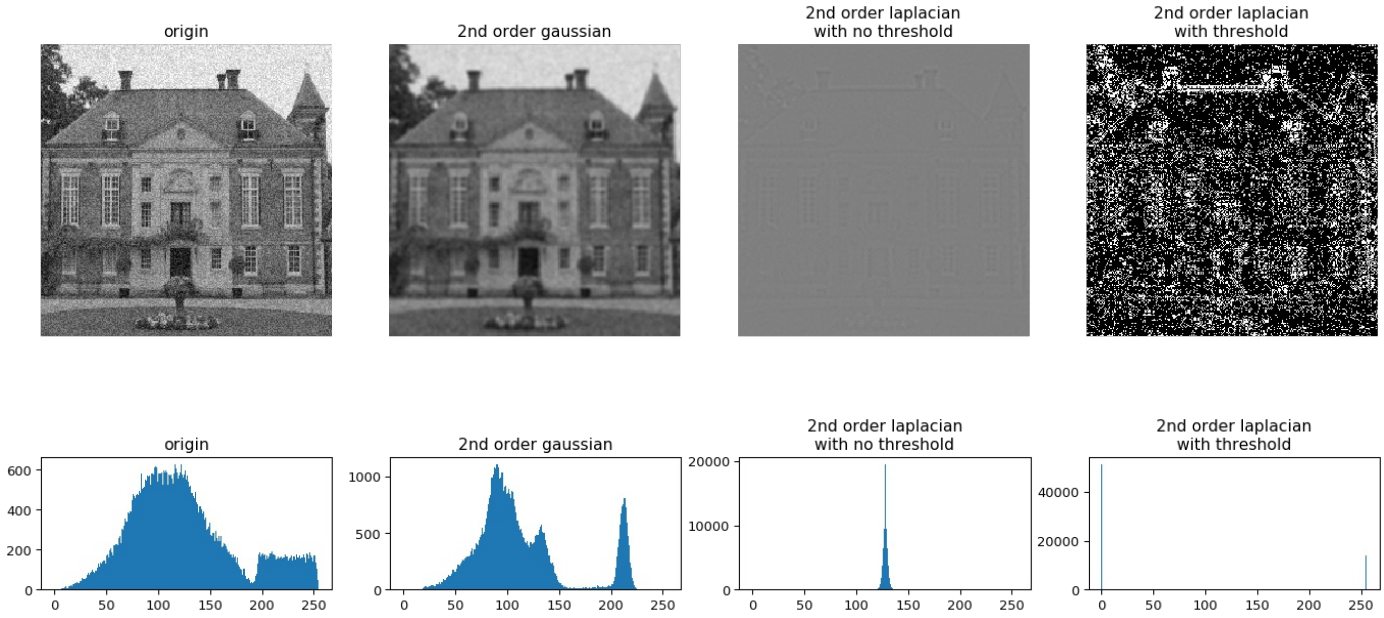

1st order with Sobel
Suggest threshold: 0

2. Perform 2nd order edge detection and output the edge map as E2.
   套用 LoG 的做法，先做 gaussian 在做 laplacian，然後就遇到難關，threshold 不管設多少出來的效果都很差，後來把 laplacian 平移 128 後生成 histgram，發現 沒有可以切 threshold 的地方...。後面的 zore crossing 偵測 zore point 的 mask 並用 2x2, 3x3, 4x4 效果也都很差...。
   gaussian filter 的 x,y sigma 都設成 1.4，threshold 都為 3，出來的圖勉強可以看到輪廓。

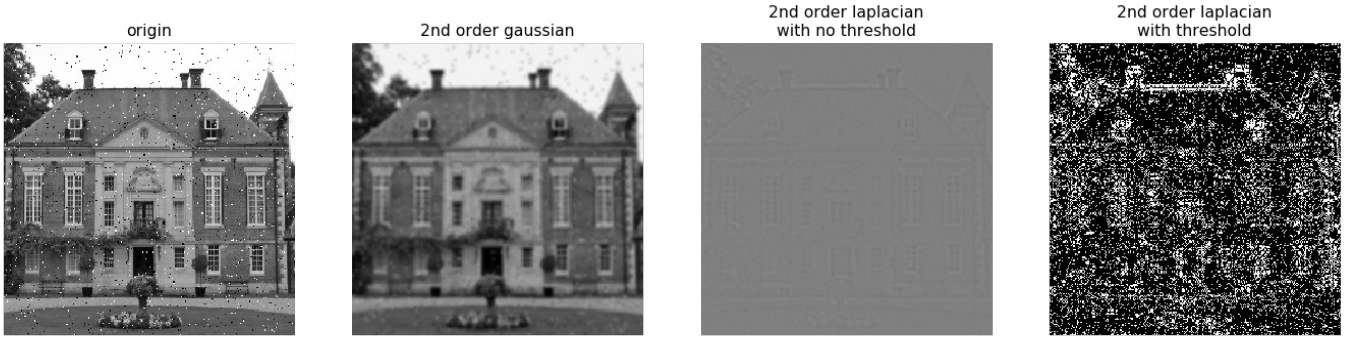
smaple1

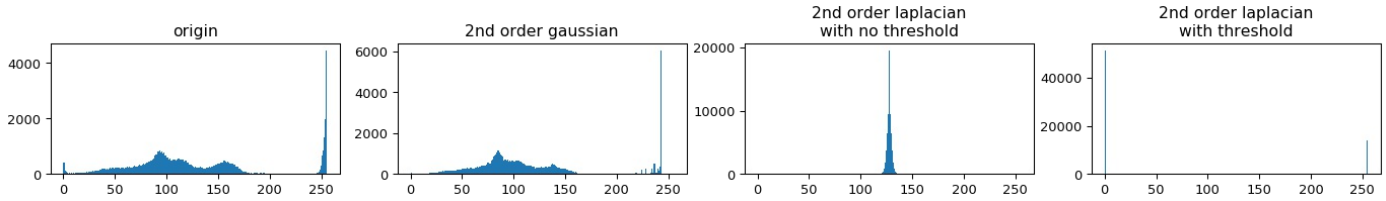


smaple2



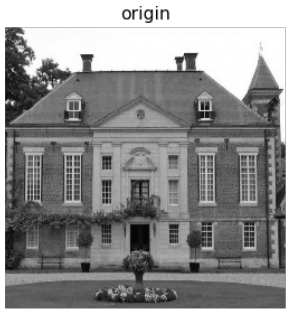
smaple3

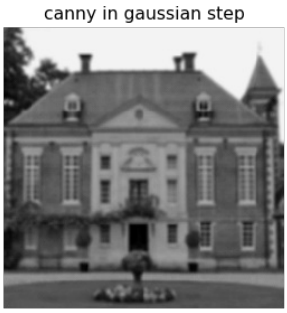

origin
2nd order gaussian
2nd order laplacian
with no threshold
2nd order laplacian
with threshold

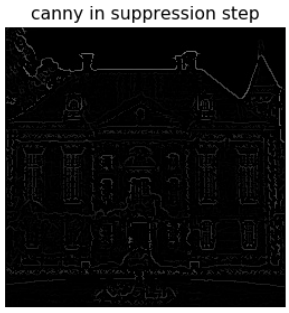

origin
2nd order gaussian
2nd order laplacian
with no threshold
2nd order laplacian
with threshold
0
2000
4000
6000
10000
20000
40000
0
50
100
150
200
250

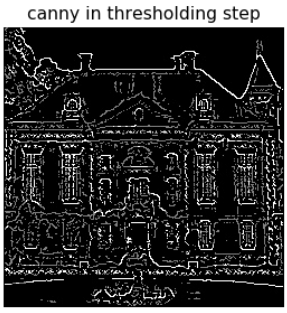
3. Perform Canny edge detection and output the edge map as E3.
   仿造講義(lecture3.p24)，對每個步驟都生成圖片來對照，在寫的過程中發現 threshold 區間不能設太小，不然 connected component 階段會沒有東西可以連接。
   gaussian filter 的 x,y sigma 都設成 1.4，threshold 為 8,30，在有 noise 的情況下也可以去除大部分的 noise 不被影響。

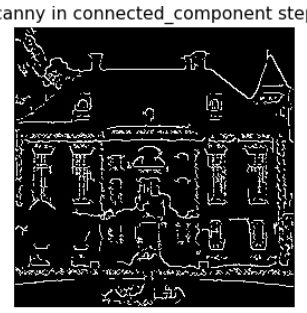
smaple1



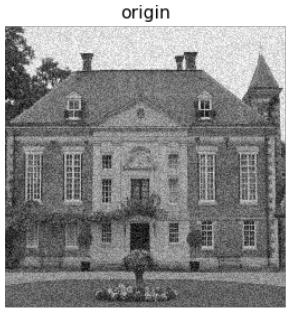


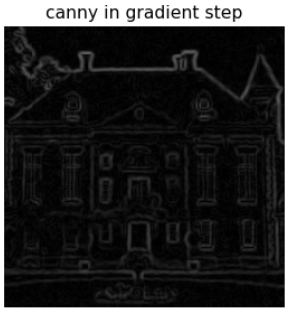


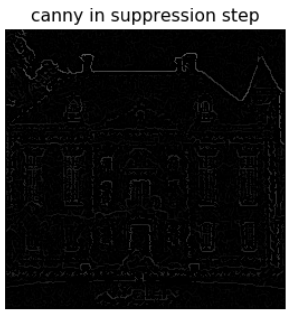


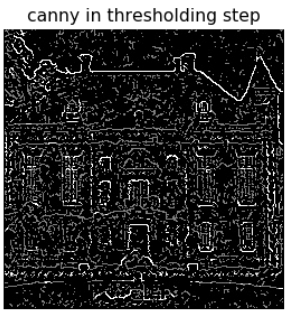


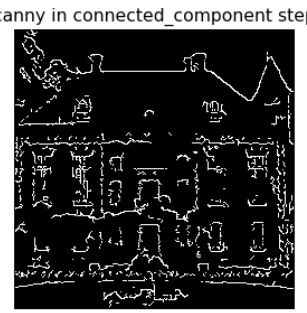
smaple2











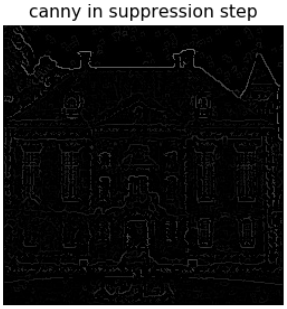
smaple3

origin

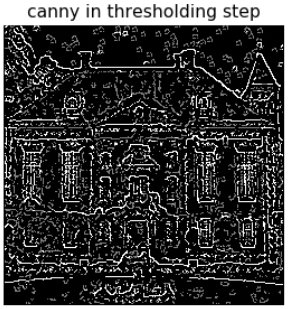
canny in gaussian step

canny in gradient step

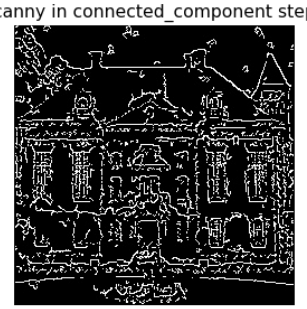
canny in suppression step



canny in thresholding step



canny in connected_component step



PROBLEM 2: GEOMETRICAL MODIFICATION

對圖片做 edge crispening 和 warping!

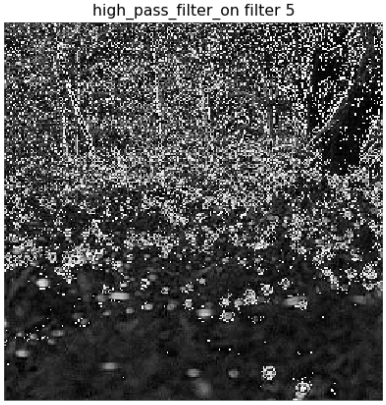
Discussion of results

1. Perform edge crispening on I2 and denote the result as C. Show the parameters and provide some discussions on the result as well.
   比較上課講到的兩種 filter，分別是 5 和 9，但效果都差強人意

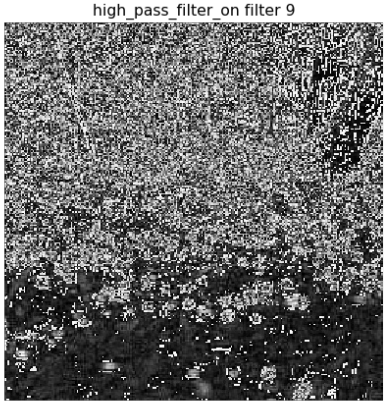
   ```
   5:          9:
    0 -1  0    -1 -1 -1
   -1  5 -1    -1  9 -1
    0 -1  0    -1 -1 -1
   ```

   另外一個是 unsharp，調整 c，覺得 4/5 的結果最好。往上，會太接進原圖，往下會因為 low pass 的比例變多而模糊。

   high_pass_filter_on filter 5

   high_pass_filter_on filter 9

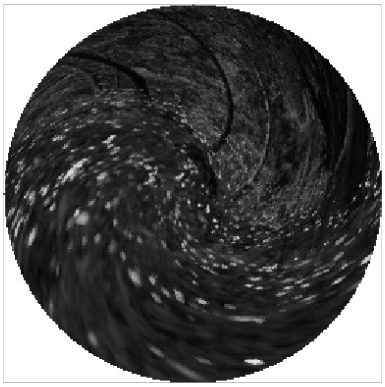
   

   unsharp masking

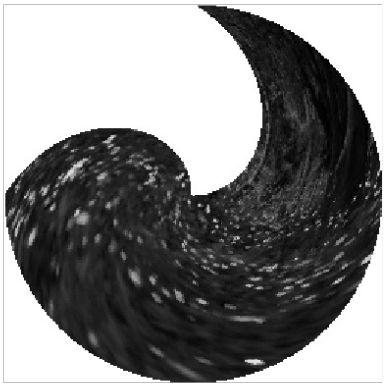
   

2. Design a warping method to produce D from C. As shown in Fig. 2(b), D is a swirled disk with diameter of 256 pixels.
   一開始看到原圖，想說是裁切成圓型後按照離圓心距離越遠 rotate 的角度越大的方式進行，結果如 warping，但很快的觀察到有 1/4 的圓轉之後不太一樣，test1 是差不多的部分， test2 是差很多得部分。推測不一樣的地方是只用別種方式旋轉後在拼貼。不過想不出此種旋轉方式...

   warping

   warping test1

   warping test2

```
//translate
int x = i - IMG_SIZE / 2, y = j - IMG_SIZE / 2, r = IMG_SIZE / 2;
//rotate theta
double theta = -PI / 2 * (1 - sqrt(x * x + y * y) / 128);

temp(i, j) = 255;
if (x * x + y * y <= r * r) {
    // inverse rotate to backward
    int u = get_u(x, y, theta), v = get_v(x, y, theta);
    temp(i, j) = this->image(get_u(x, y, theta), get_v(x, y, theta));
}

int get_u(int x, int y, double theta) {
    return cos(theta) * x + sin(theta) * y + IMG_SIZE / 2;
}
int get_v(int x, int y, double theta) {
    return cos(theta) * y - sin(theta) * x + IMG_SIZE / 2;
}
```

Note:

histogram 的產生方式是使用 python 的 matplotlib 和 numpy，使用 jupyter 來當顯示界面，jupyter export 的 html 檔為 homework2_problem1.html 和 homework2_problem2.html