HW#10

R09631050 生機碩一 詹閎棋

這次作業是做邊緣偵測

Zero-crossing on the following four types of images to get edge images.

首先要將原圖片的每個pixel乘上kernel的權重值,因為這次作業有不同size(3*3、11*11)的kernel,所以我先寫一個能按照輸入的kernel size擴增原圖片的函式。經過kernel處理過後,將每個pixel值與threshold做比較,若大於threshold就給1,小於負的threshold就給-1,其餘給0。

將結果存於新的矩陣,這部分是我花最多時間的地方,因為一開始未發現生成矩陣的資料型態,選擇未包含負數的資料型態,所以如果value是-1,會自動轉換成255,這在下一步的邊緣偵測函式中會產生問題。

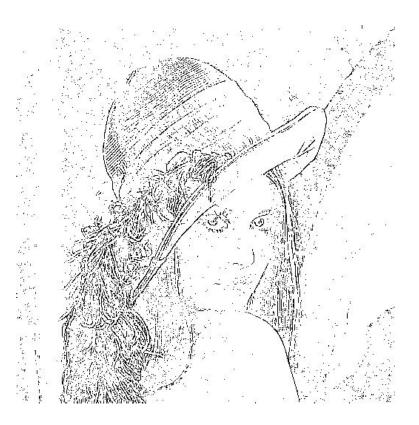
將上一步驟完成的矩陣代入我寫的zero_crossing函式,若原pixel value = 1,且周 **園8個鄰居有其中一個以上為-1**,就給定為0(black),其餘都給定為255,把結果 儲存為新矩陣即完成。

1.

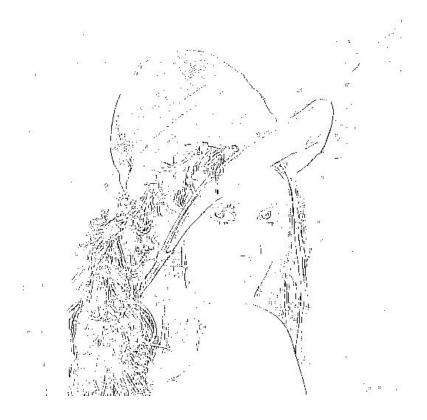
1.1 Laplacian_1 (threshold = 15)



1.2 Laplacian_2 (threshold = 15)



2. Minimum-variance Laplacian (threshold = 30)



3. Laplacian of Gaussian (threshold = 3000)



4. Difference of Gaussian, (use tk to generate D.O.G.) DoG (inhibitory $\sigma = 1$, excitatory $\sigma = 3$, kernel size=11) (threshold = 1)

