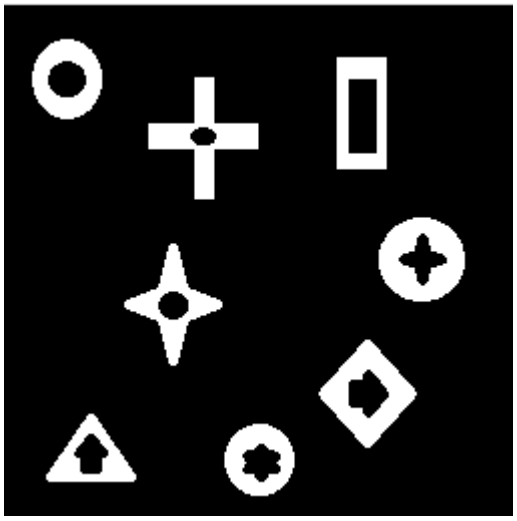
% DIP Homework Assignment 2
% 04 10, 2018 % Name: 吳睿哲
% ID #: r06921095
% email: r06921095@ntu.edu.tw
% 只要執行每題的 execute code 就可以得到該題需要的檔案或答案

```
% Problem 1(a): extract the object's boundary
% Implementation 1: erosion
% M-file name: readraw.m , writeraw.m , erosion.m
% Usage: boundary extraction
% Output image: erosion_boundary.raw
% Parameters: structuring element = logical([0 1 0 ; 1 1 1 ; 0 1 0]);
%execute code:  erosion("sample1.raw")
```

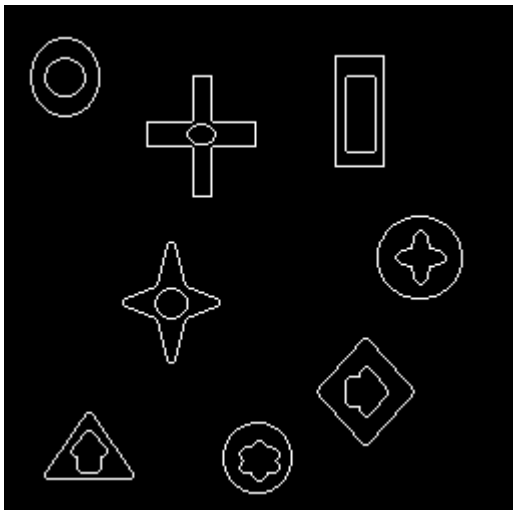
---

a. 這題的做法主要就是依照老師上課講的用 erosion 的方法讓原本的影像縮小,再用原本的影像減掉經過 erosion 後的 image,就可以得到此圖的 boundary
b. "sample1.raw"
c. "erosion_boundary.raw"
d. 我覺得這題的輸出結果是相當成功的,每個 object 的 boundary 都處理得相當仔細,可以完整地呈現原本 image 的 boundary。

Sample1.raw



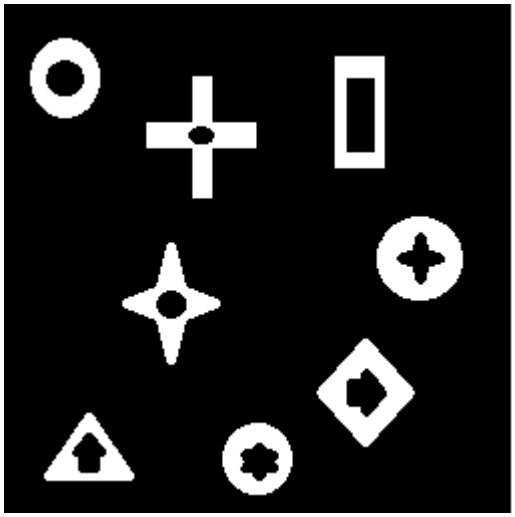
erosion_boundary.raw



```
--------------------------------------------------------------------------
% Problem 1(b): design an algorithm to count the object in the image
% Implementation : dilation
% M-file name: readraw.m , writeraw.m , dilation.m ,countobj.m
% Usage: count object
% Output the number of connected component
% Parameters: structuring element object=strel('square',3);
%execute code:countobj("sample1.raw")
--------------------------------------------------------------------------
```

a. 這題是先找出圖片中每個不等於 0 的區域,再用對他們做 dilation,藉此找到每個不同的 connected componrnt 的個數
b. “sample1.raw”
c. No(just the number of connected component)
d. 我覺得這題的卻點主要是在做 dilation 的時候速度太慢了,造成計算 connected component 的個數時候速度也很慢,主要是因為自己手寫 dilation 的方式不好,速度太慢,如果 dilation 的部分直接用 built in function 的話,幾乎可以瞬間就完成計算。

Samle1.raw



```
-------------------------------------------------------------------------
% Problem 1(c):find the skeleton of the image
% Implementation :skeletonziation
% M-file name: readraw.m , writeraw.m , skeletonizing
% Usage: find the skeleton of the image
% Output image: edge_crispening.raw
% Parameters: no
%execute code:skeletonizing("sample1.raw")
-------------------------------------------------------------------------
```

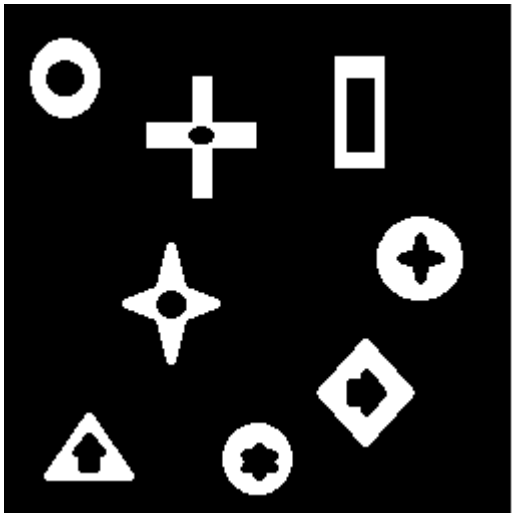
a. skeletonizing 的做法應該是用依照投影片講的方法,先覺調要對照片做什麼處理之後,依照 bound 數比照 lutup table 的 stage1 及 stage2,來得到結果,但此題我用了其他方法

b. "smple1.raw"

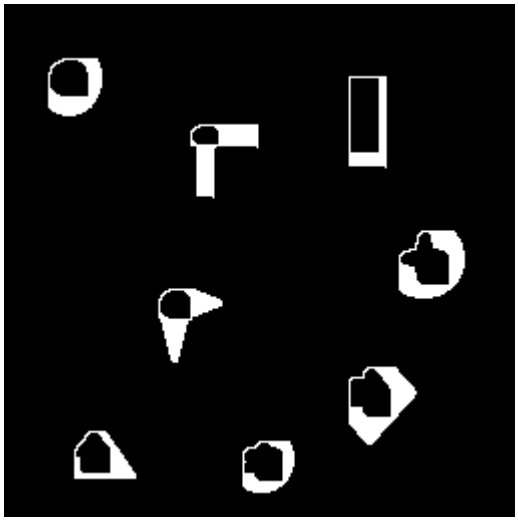
c. "skeletonizing.raw"

d. 這題原本我打算照投影片的做法來做,但是光是 key in lutup table 就要花上不少時間,所以覺得應該會有更有效率的方法,於是我參考了網路上其他人的做法,這種做法只包含了 lutup table 的一小部分,一樣適用 convolution 的方式找出周圍的 pixel,如果非零的數目介於 2 到 6 之間,且對角線 pixel 相乘的值等於零的話,就讓這個 pixel 的值零,但是由輸出可得知這種作法,應該沒有比直接查 lutup table 來的好。

Sample1.raw



skeletonizing.raw



```
--------------------------------------------------------------------------
% Problem 2(a):perform law's method to classify the texture
% Implementation :use law's method and energy computation to classify
the image
% M-file name: readraw.m , writeraw.m , law.m
% Usage: :classifit the different texture
% Output image: "classification.raw"
% Parameters:  8th filter in the law's method
%execute code:law("sample2.raw")
--------------------------------------------------------------------------
```

a. 這題的做法我是先用 law's method 裡面的 9 個 filter 對 sample2.raw 做 convolution,最作 energy computation,然後比照經過 9 個 filter 及 energy 後的 9 張 image 看哪張圖的 3 個 texture 色差比較明顯,再用 local histogram 來找 3 個 texture 的料度分布,設一個 thershold 來把三個 texture 分開

b. "sample2.raw"

c. "classification.raw"

d. 用這種分法雖然大部芬都可以區分的很好,但是會有一些雜訊,沒有辦法區分開來,向中間那個 texture 就會有兩個特別亮的點,沒有被區分為第二個 texture

Sample2.raw

classification.raw

```
-------------------------------------------------------------------------
% Problem 2(b):exchange the type of different pattern

%Implementation :use synthetic method to generrate the texture
% M-file name: synthesize.m , getxcorr2.m , xcorr2.m , example.m
example2.m, example3.m changepattern.m peform2b.m
% Usage: :generate the input texture into a biger size
% Output image: "synthesize.raw"
% Parameters: no
%execute code:perform2b
-------------------------------------------------------------------------
```

a. 這題的做法主要是用 synthesize 的概念去做 texture 的生成,先取每個 texture 的一小部分,再 syntesize 成 512*512 的 image,再用上一題 local hisogram 的方式找出邊界的部分,然後把生成的 texture paste 上去,且 paste 再原本不同的區域,達到交換的目的。

b. “sample2.raw”

c. “synethesize.raw”

d. 這議題我覺得我做的比較不好的地方是炫渦型 pattern 的生成,這應該跟原始丟進去要生成的 image 有關,但是不管我再怎麼切,生成的 image 都會有一些地方怪怪的,可能要再經過一些處理才會不容易發現破綻,界的處理也要再做一些補強。

Sample2.raw

synethesize.raw