

% 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): first order;Bsecond order;Bcanny edge detection
% Implementation 1: Orthogonal gradient generation
% M-file name: first_order.m , second_order.m , canny.m ,
rmunifnoise.m
% Usage: edge detection
% Output image: first_order.raw , secoond_order.raw , canny.raw
% Parameters: first_order thershold:50 , secon_order thershold:7 ,
canny
% thershold :20 , k=2
%execute code: first_order("sample1.raw")
%               second_order("sample1.raw")
%               canny("sample1.raw")
```

- 我的作法處要是從一階微分,二階微分,以及canny來判斷哪個方法的edge detection最好
- "sample1.raw"
- "first_order.raw","second_order.raw","canny.raw"
- 我覺得以結果來比較的話,first order的偵測結果來不錯,但是雜訊會比較多。Second order處理雜訊的結果會比較好,但是偵測到的便藉比較沒有那麼清楚。Canny 的輸出結果則是介於兩者之間。

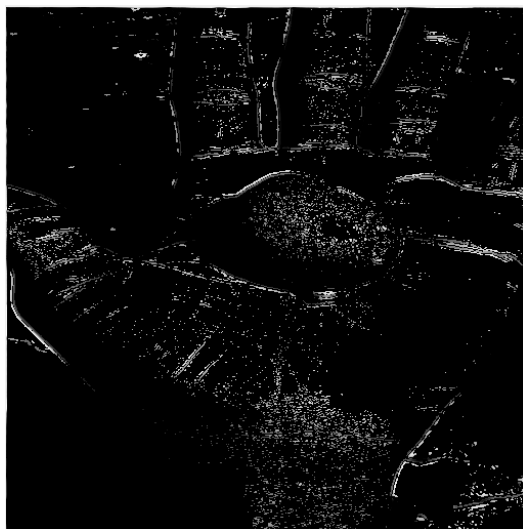
sample1.raw



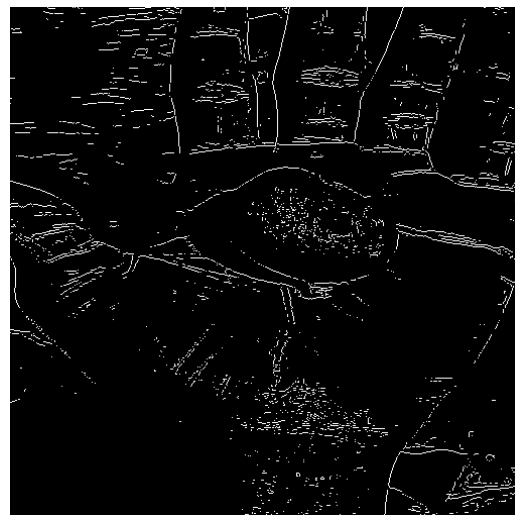
first_order.raw



second_order.raw



canny.raw



```
% Problem 1(b): edge detection for noising image
% Implementation 1: detecting the edge when image has noising signal
% M-file name: first_order.raw
% Usage: edge detection
% Output image: first_order.raw
% Parameters: first_order thershold:50
%execute code:first_order("sample2.raw")
```

- a. 這一題一開始的想法是要先把斜條紋雜訊濾掉之後,再做 edge detection,但是試過許多方法之後,包括上次作業寫道的 low pass filter 以及 median filter 都沒辦法有效率的濾掉雜訊
- b. "sample2.raw"
- c. "first_order.raw"
- d. 這題雖然找不到適合的方法濾掉斜條紋,但是只階做 edge detection 卻可以達到跟沒有斜條紋照片一樣的效果,且其中邊界找得最漂亮的就是一題用的 first order。

sample2.raw



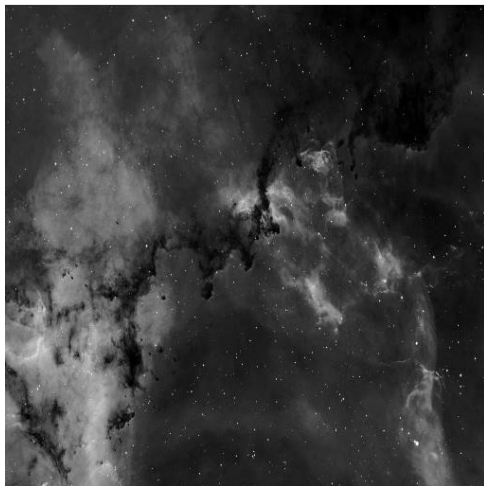
first_order.raw



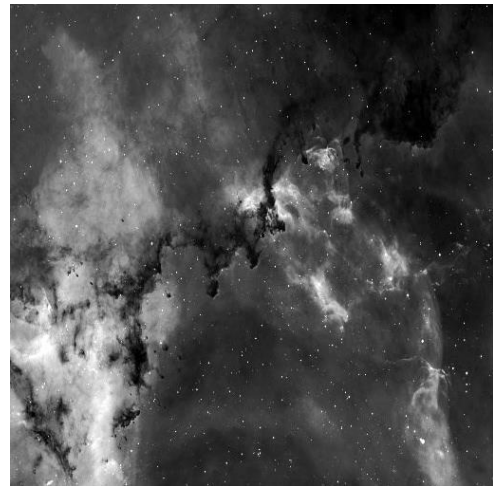
```
% Problem 2(a):edge crispening
% Implementation :make the original image more apparently
% M-file name: edge_crispening.m
% Usage: edge crispening
% Output image: "H.raw"
% Parameters: thershold 50
%execute code: edge_crispening("sample3.raw")
```

- a. 這議題的想法主要是先把原本的圖的邊界找出來,再疊加到原圖上,就可以讓原圖的輪廓更加明顯
- b. "sample3.raw"
- c. "edge crispening.raw"
- d. 從照片的輸出結果來看,把邊界疊加到原圖上確實可以達到讓原圖更明顯的效果,其中,邊界的 pixel 我都把它拿來 $\times(1/3)$ 再疊加上去,不讓會讓原圖變得太亮。

sample3.raw



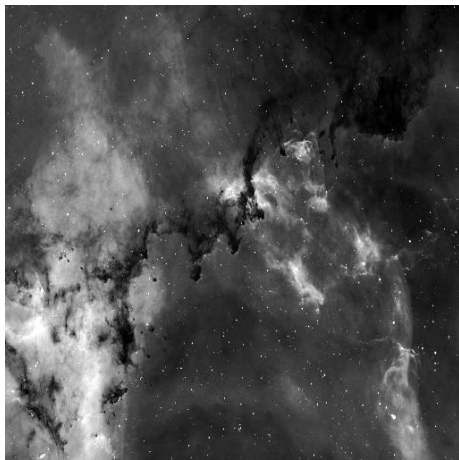
edge crispening.raw



```
% Problem 2(b):wrapping a image
% Implementation : performin wrapping as a geometrical modification
% M-file name: warping.m
% Usage: :warping
% Output image: "warping.raw"
% Parameters: no
%execute code:warping("edge_crispening.raw")
```

- a. 這題一開始的想法是覺得邊界的變化很像 sine,於是就 google 有關 warping sine 的相關網站,做法大概是在原本的位置+位置的 sine 函數。
- b. "edge_crispening.raw"
- c. "warping.raw"
- d. 使用了添加 sine 函數的方法後可以得到類似於題目要求的圖案,效果也還不錯。

edge_crispening.raw



warping.raw

