

高等影像處理

作業#4

姓名：蘇柏凱

學號：111c71007

指導老師：李曉祺、蔣欣翰

1.1

Discussion

1.1 centroid of shape100_1 x:51.5147 y:47.4853

1.1 centroid of shape100_2 x:50 y:49

1.1 centroid of shape100_3 x:56.9125 y:48.8565

1.1 centroid of shape100_4 x:50.8094 y:48.7252

1.1 centroid of shape100_5 x:54.9452 y:49.3637

1.1 centroid of shape100_6 x:51.0667 y:48.913

1.1 centroid of shape100_7 x:46.9613 y:47.8572

1.1 centroid of shape100_8 x:52.0208 y:47.736

1.2

Discussion

Shape_100_1

p=1,q=0 : -8.90106e-07 p=0,q=1 : 1.72622e-09 p=2,q=0 : 3.50147e+08
p=0,q=2 : 3.50147e+08 p=1,q=1 : 458078 p=2,q=1 : -8.58721e+06
p=1,q=2 : 8.58721e+06 p=0,q=3 : 1.61234e+06 p=3,q=0 : -1.61234e+06

Shape_100_2

p=1,q=0 : 0 p=0,q=1 : 0 p=2,q=0 : 3.58507e+08 p=0,q=2 : 3.58507e+08 p=1,q=1 : 0
p=2,q=1 : 0 p=1,q=2 : 0 p=0,q=3 : 0 p=3,q=0 : 0

Shape_100_3

p=1,q=0 : 2.30126e-07 p=0,q=1 : -3.28509e-09 p=2,q=0 : 2.33544e+08
p=0,q=2 : 2.33131e+08 p=1,q=1 : 741022 p=2,q=1 : 3.8544e+06
p=1,q=2 : 2.18289e+09 p=0,q=3 : 2.32395e+07 p=3,q=0 : -2.1848e+09

Shape_100_4

p=1,q=0 : -4.43582e-07 p=0,q=1 : 9.60233e-09 p=2,q=0 : 3.74722e+08
p=0,q=2 : 3.76839e+08 p=1,q=1 : -892062 p=2,q=1 : -1.9958e+06
p=1,q=2 : 9.77327e+07 p=0,q=3 : 4.45125e+06 p=3,q=0 : -8.46929e+07

Shape_100_5

p=1,q=0 : -6.89808e-07 p=0,q=1 : 1.26565e-08 p=2,q=0 : 4.32067e+08
p=0,q=2 : 7.27276e+08 p=1,q=1 : 749147 p=2,q=1 : -2.68045e+06
p=1,q=2 : 3.24757e+09 p=0,q=3 : 1.57586e+07 p=3,q=0 : -1.24211e+09

Shape_100_6

p=1,q=0 : -5.25422e-07 p=0,q=1 : 1.27275e-08 p=2,q=0 : 4.01783e+08
p=0,q=2 : 4.01648e+08 p=1,q=1 : 1.08448e+06 p=2,q=1 : 1.04261e+07 p=1,q=2 : -
3.71115e+07 p=0,q=3 : -1.27443e+07 p=3,q=0 : 1.70814e+07

Shape_100_7

p=1,q=0 : 4.5985e-07 p=0,q=1 : -2.72466e-08 p=2,q=0 : 3.99117e+08
p=0,q=2 : 5.38861e+08 p=1,q=1 : 662968 p=2,q=1 : -2.131e+07 p=1,q=2 : -
2.89849e+07 p=0,q=3 : 1.02531e+07 p=3,q=0 : 9.34652e+06

Shape_100_8

p=1,q=0 :5.10288e-07 p=0,q=1 :-1.14305e-08 p=2,q=0 :5.18732e+08
p=0,q=2 :5.12901e+08 p=1,q=1 :-867139 p=2,q=1 :-3.90911e+06 p=1,q=2 :-
3.65569e+07 p=0,q=3 :1.24179e+07 p=3,q=0 :1.79102e+07

2.1

Figure



box_smooth_3x3_slate.png



box_smooth_5x5_slate.png



gaussian_smooth_3x3_slate.png



gaussian_smooth_5x5_slate.png



box_smooth_3x3_slate_noise.png



box_smooth_5x5_slate_noise.png



gaussian_smooth_3x3_slate_noise.png



gaussian_smooth_5x5_slate_noise.png

Discussion

1	1	1
1	1	1
1	1	1

Box smooth 3x3

1/25

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Box smooth 5x5

1/16

1	2	1
2	4	2
1	2	1

Gaussian blur 3x3

1/273

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Gaussian blur5x5

Box smoothing 是一種簡單的平均方法，適用於快速降低圖像中的噪點，但可能會導致圖像細節的丟失。

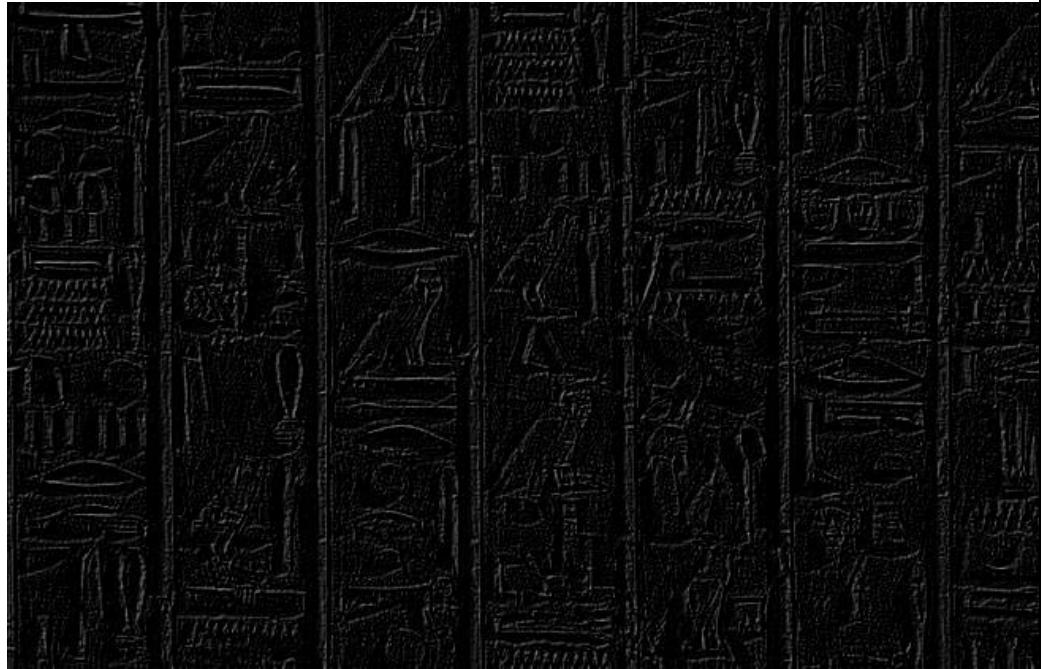
高斯模糊過濾是一種更複雜的過濾方法，通過考慮像素之間的空間關係來減少噪點，可以保留更多的圖像細節。

在沒有噪音的圖片上兩者的表現差異沒有到太大，**box smoothing** 會讓圖片較為模糊，高斯模糊可以保留更多的細節;filter 越大 **box smoothing**、高斯模糊會越模糊一些。

而在有噪音的圖片上高斯模糊可以明顯看出來效果比起 **box smoothing** 好上許多，

2.2

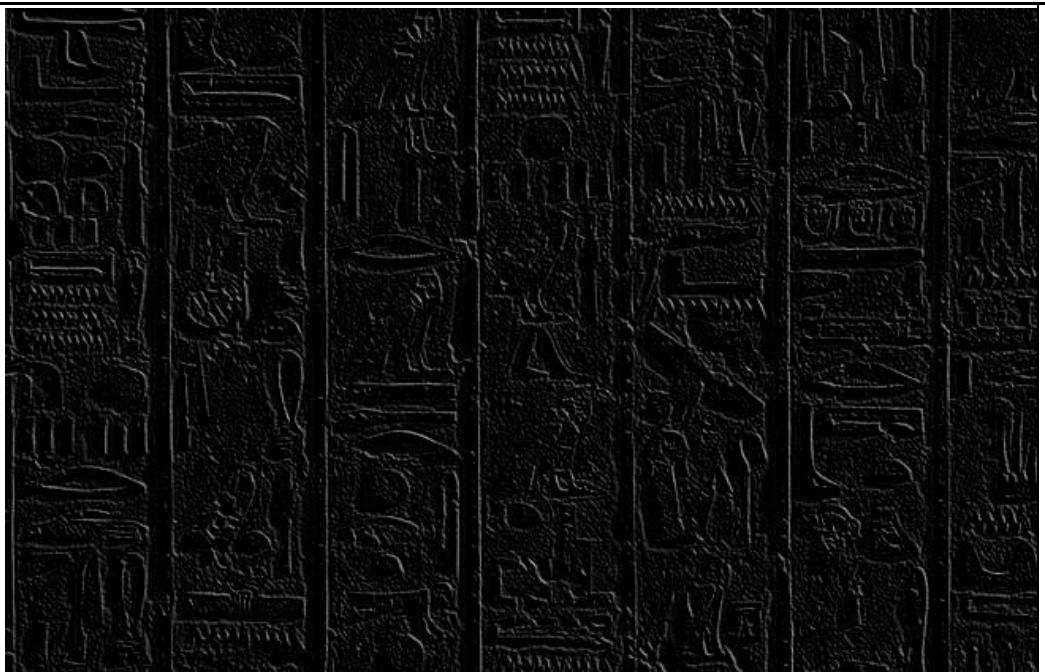
Figure



robert_45_slate.png



robert_45_slate_noise.png



robert_n45_slate.png



robert_n45_slate_noise.png



robert_combine_slate.png



robert_combine_slate_noise.png

Discussion

Robert45 filter:

0	0	0
0	1	0
0	0	-1

Robert -45 filter:

0	0	0
0	0	1
0	-1	0

$$|G| = \sqrt{G_x^2 + G_y^2}$$

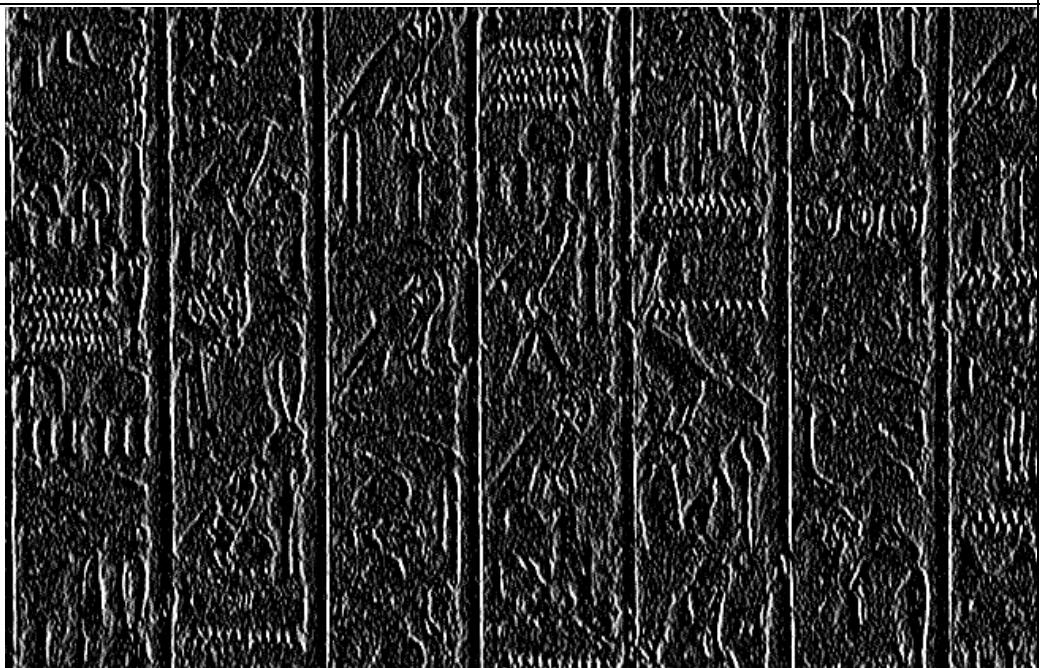
Roberts 交叉運算子非常簡單和快速，但是也很容易受到雜訊的影響。它在檢測弱邊緣或對比度較低的邊緣時效果不佳。

2.3

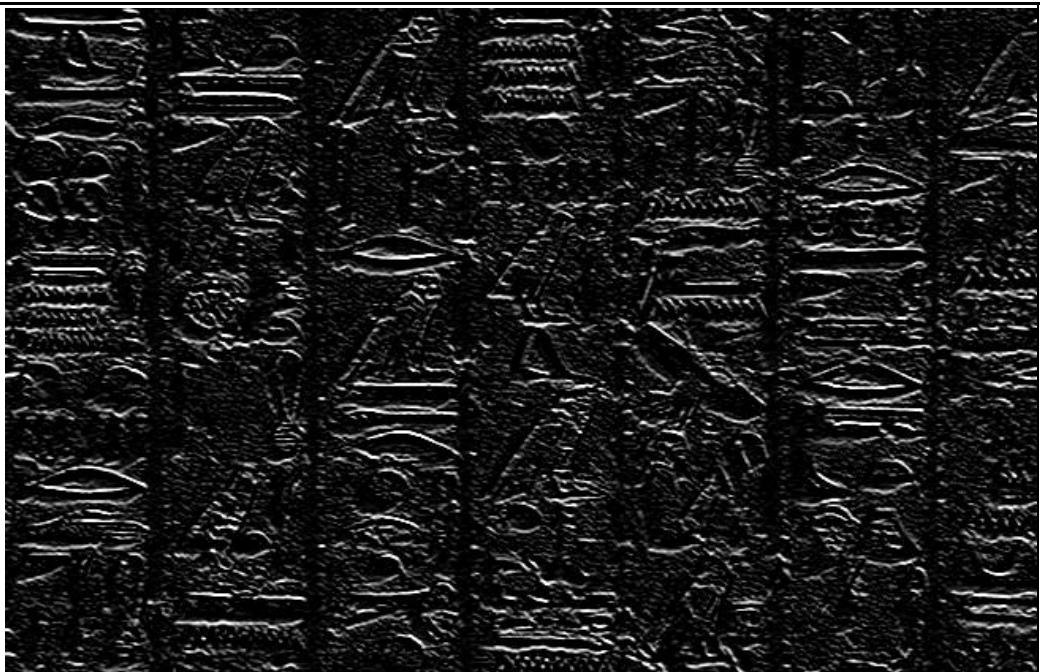
Figure



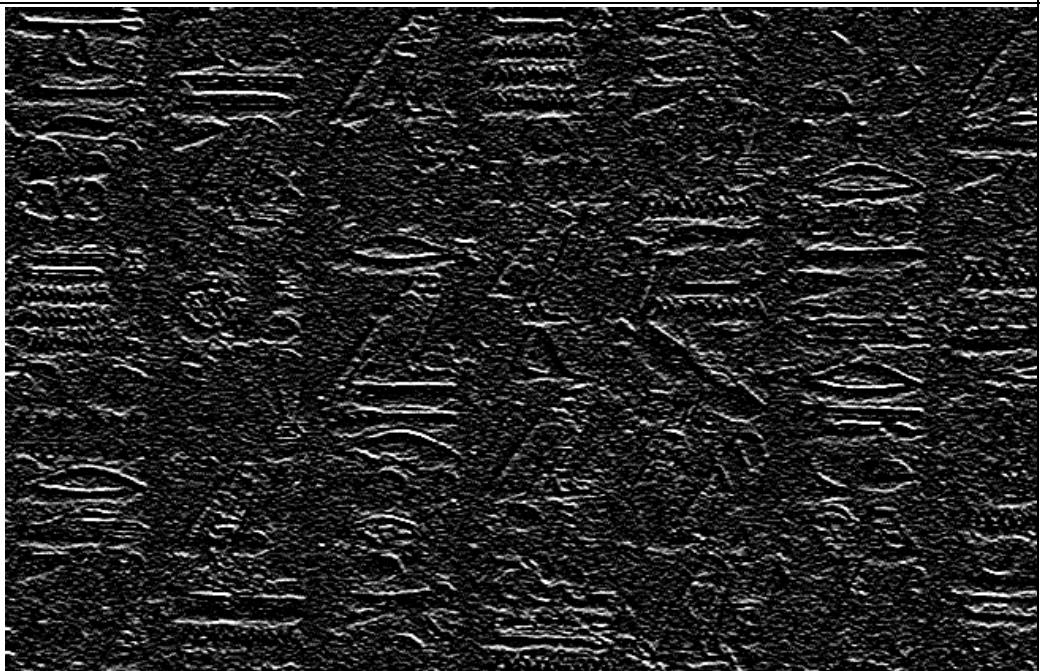
prewitt_0_slate.png



prewitt_0_slate_noise.png



prewitt_90_slate.png



prewitt_90_slate_noise.png



prewitt_combine_slate.png



prewitt_combine_slate_noise.png

Discussion

0° filter: 90° filter

$$[-1 -1 -1] \quad [-1 0 1]$$

$$[0 0 0] \quad [-1 0 1]$$

$$[-1 -1 1] \quad [-1 0 1]$$

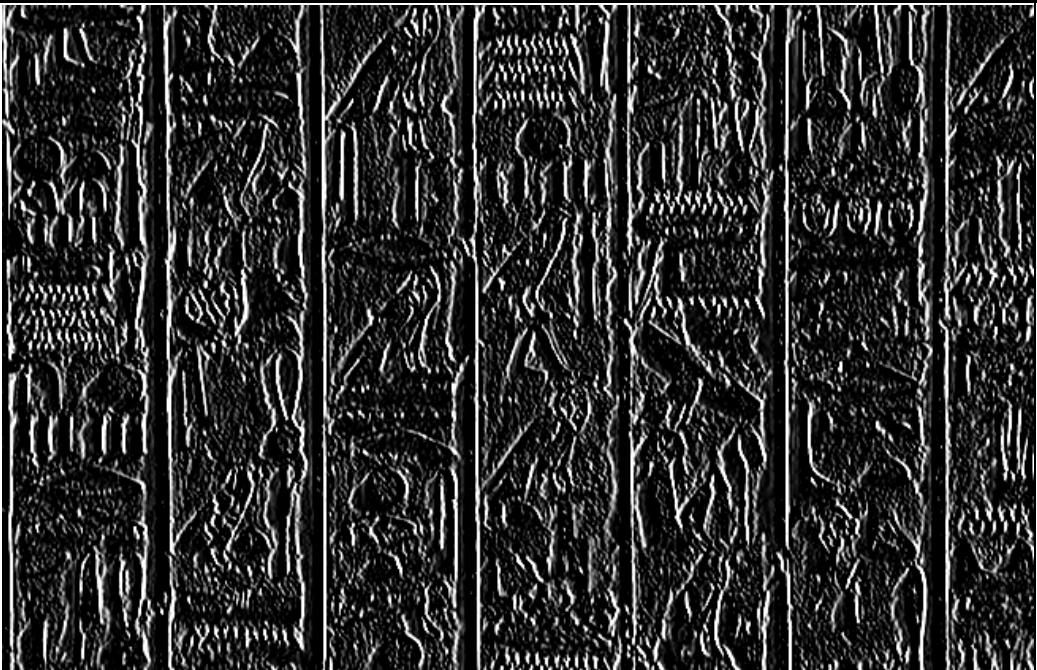
$$|G| = |G_x| + |G_y|$$

與 Sobel 不同的是，Prewitt 直接將兩個方向梯度的絕對值相加，而不是平方和開根號。

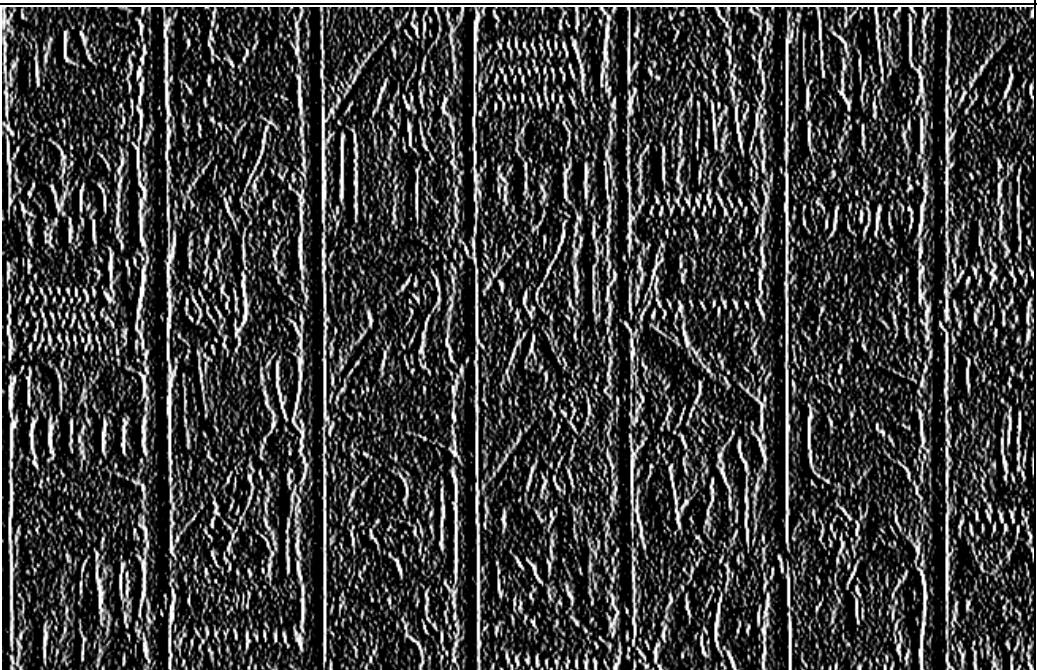
與 Sobel 類似，Prewitt Kernels 只考慮了兩個主方向的梯度。這使得它對斜方向邊緣的檢測比較弱。同時也比較容易受到雜訊的影響，在超出邊界時的處理採用 mirroring mask processing，而在數值超過 255 時則設為 255，小於 0 則設為 0。

2.4

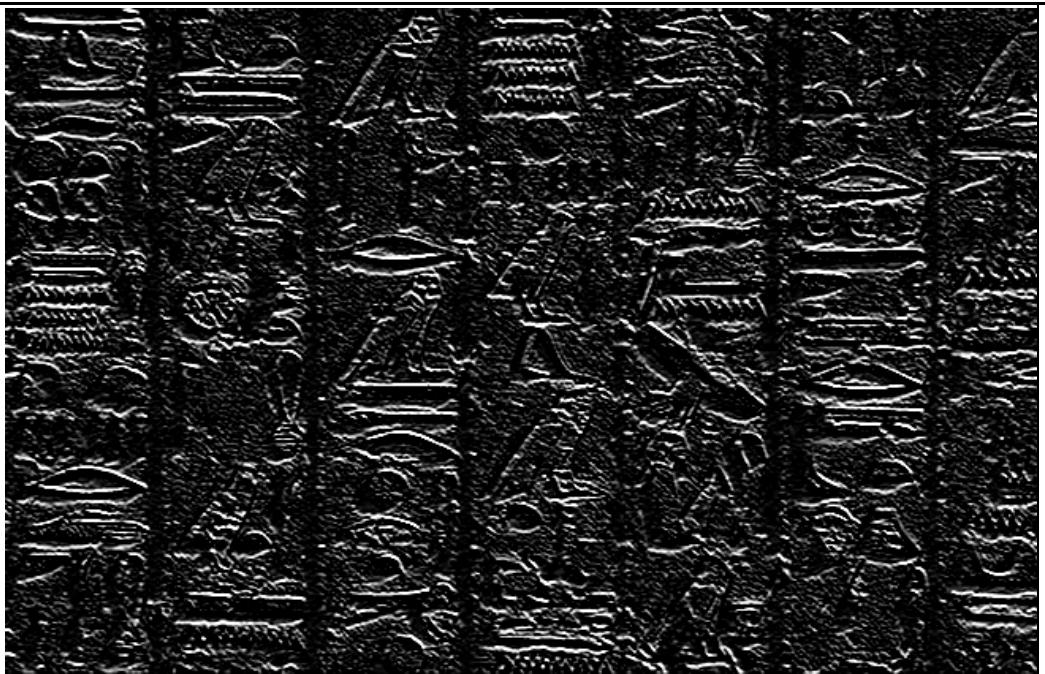
Figure



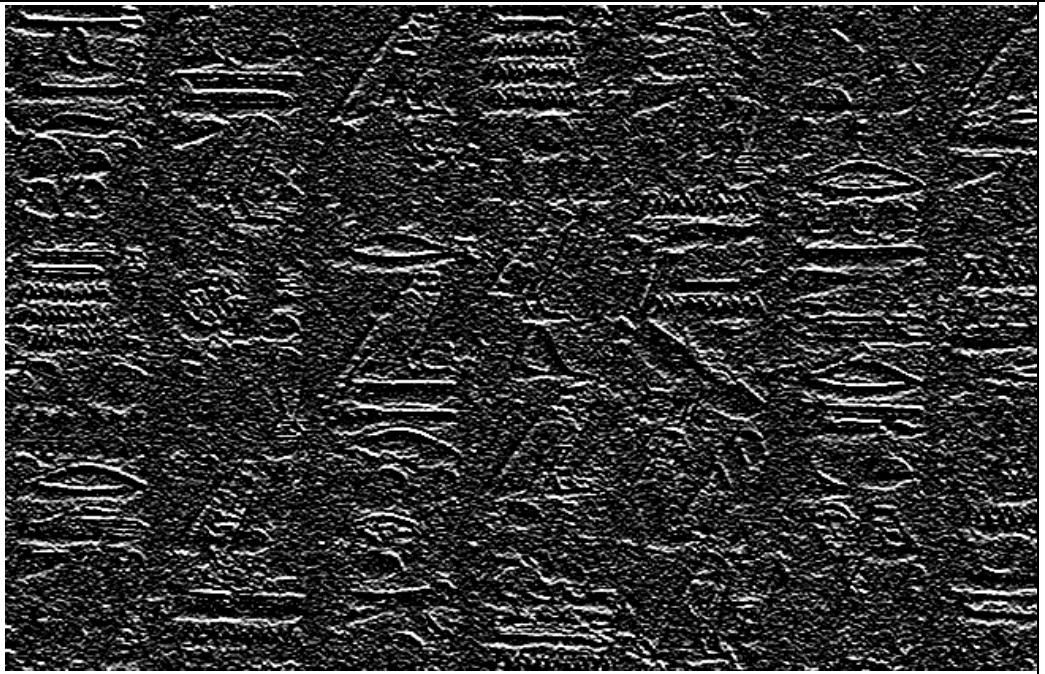
sobel_0_slate.png



sobel_0_slate_noise.png



sobel_90_slate.png



sobel_90_slate_noise.png



sobel_combine_slate.png



sobel_combine_slate_noise.png

Discussion

0° Sobel filter 90° Sobel filter:

$$\begin{array}{ll} [-1 \ 0 \ 1] & [-1 \ -2 \ -1] \\ [-2 \ 0 \ 2] & [0 \ 0 \ 0] \\ [-1 \ 0 \ 1] & [1 \ 2 \ 1] \end{array}$$

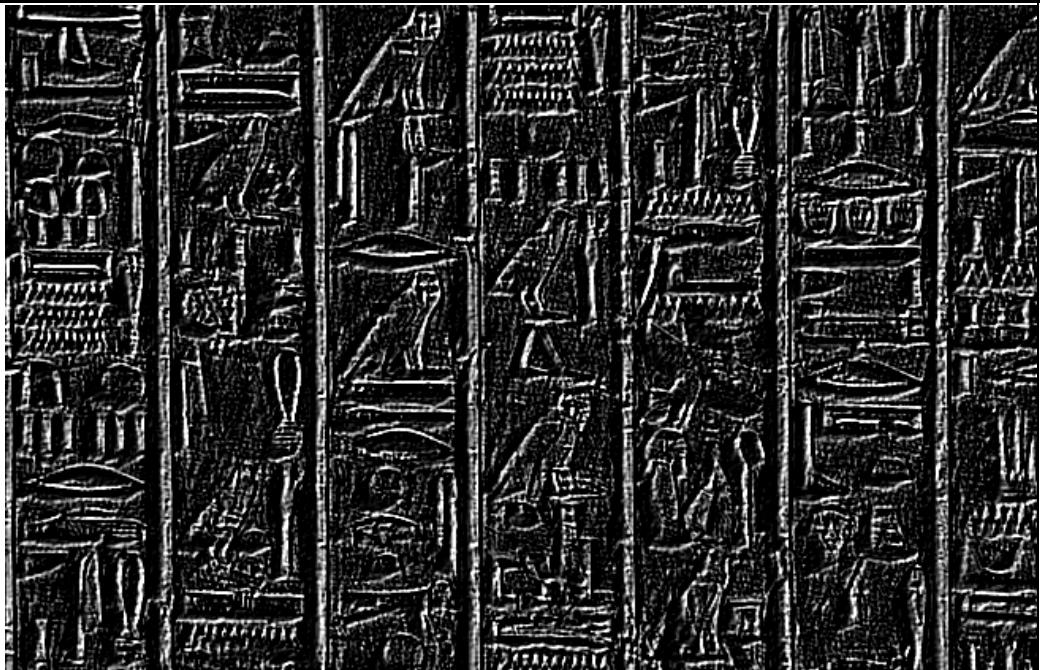
$$|G| = \sqrt{G_x^2 + G_y^2}$$

Sobel 只考慮了兩個正交方向的梯度，所以對斜方向邊緣的檢測效果較差。
同時也易受雜訊影響，在超出邊界時的處理採用 mirroring mask processing，

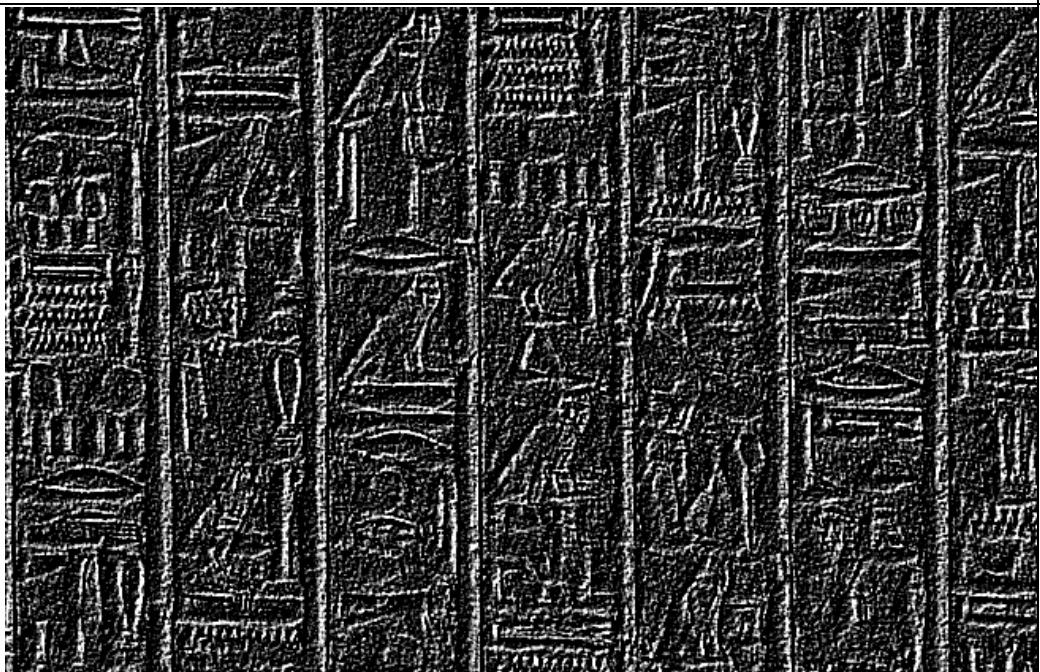
而在項數值超過 255 時則設為 255，小於 0 則設為 0。

2.5

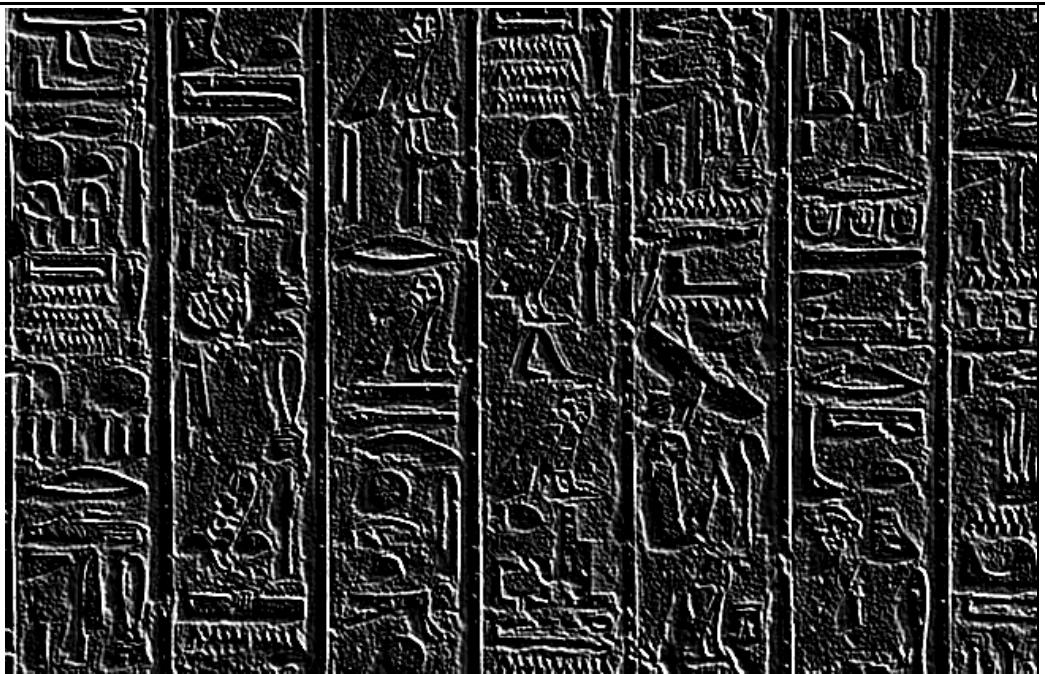
Figure



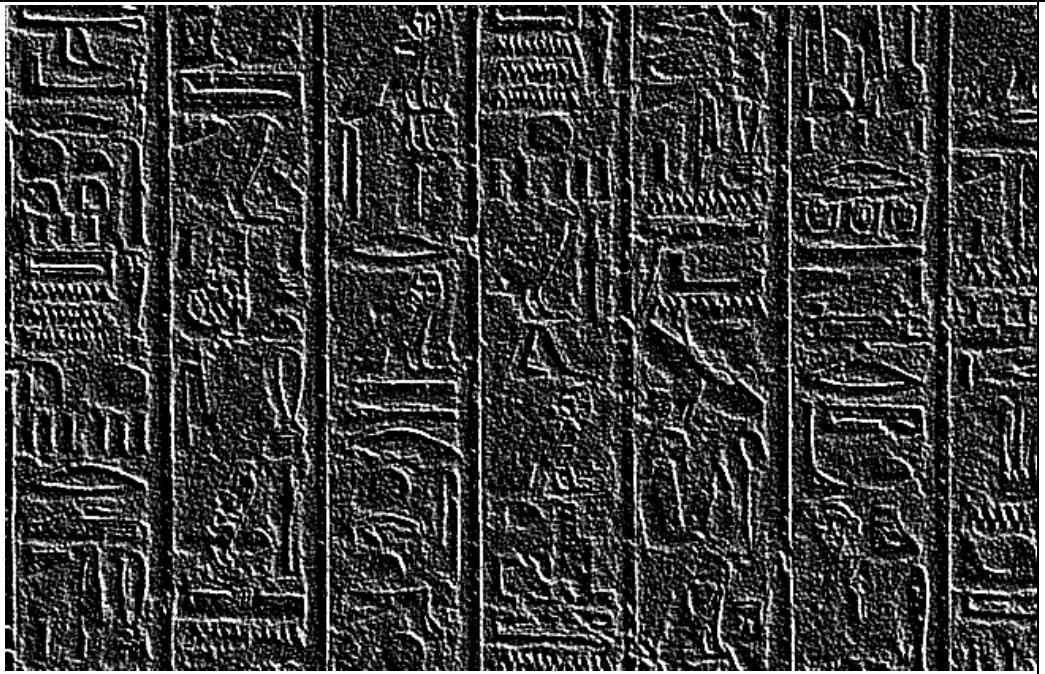
sobel_45_slate.png



sobel_45_slate_noise.png



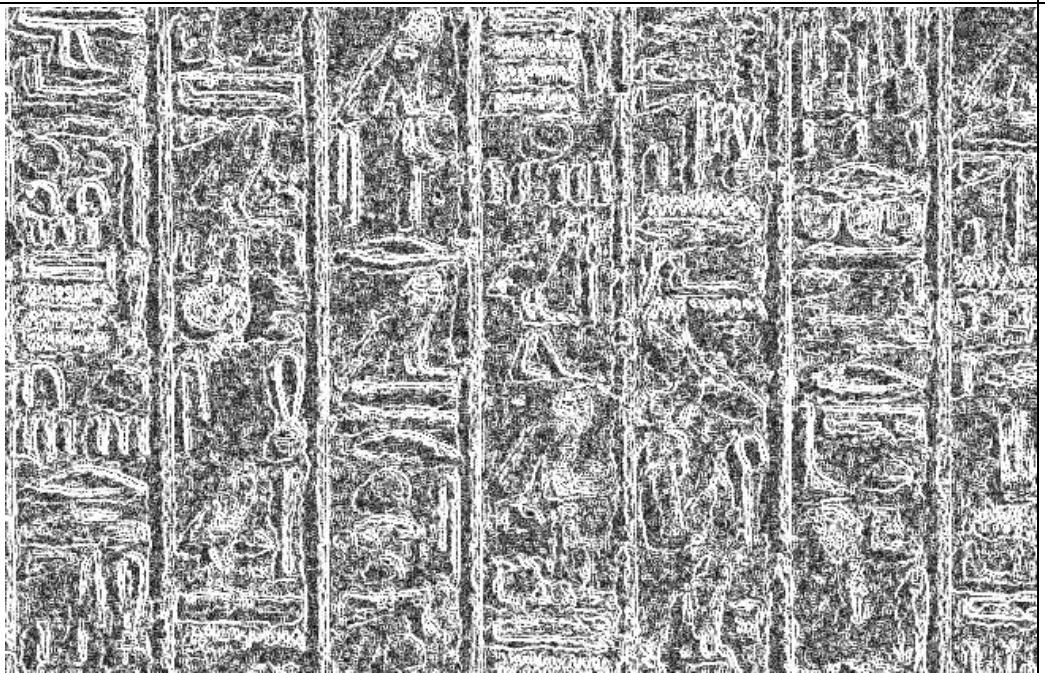
sobel_n45_slate.png



sobel_n45_slate_noise.png



sobel45_combine_slate.png



sobel45_combine_slate_noise.png

Discussion

-45 度 filter:

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

+45 度 Sobel filter:

$$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

這兩個 filter 針對-45 度和+45 度方向進行了優化

$$|G| = \sqrt{(G_{-45})^2 + (G_{+45})^2}$$

這可以合併斜線方向的邊緣資訊，檢測更多的邊緣。

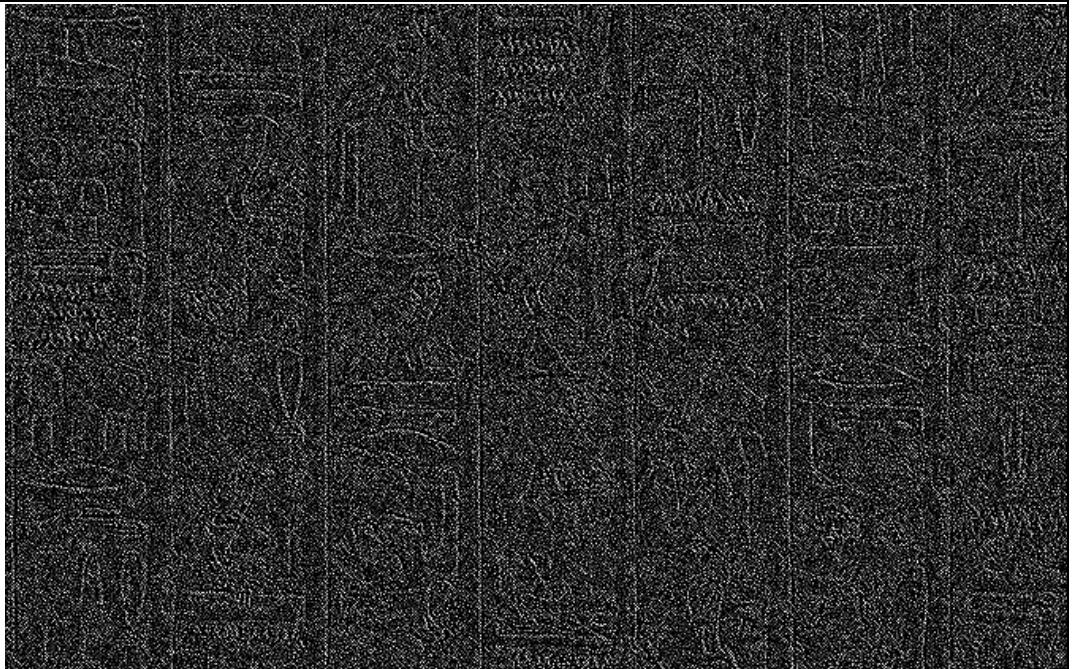
與只有 0 度和 90 度兩個標準 Sobel 濾波器相比，斜的兩個方向使得對邊緣方向更加敏感，可以檢測到更多斜方向的邊緣，減少誤差，但是在縱橫方向的資訊則較為缺乏，在超出邊界時的處理採用 **mirroring mask processing**，而在項數值超過 255 時則設為 255，小於 0 則設為 0。

2.6

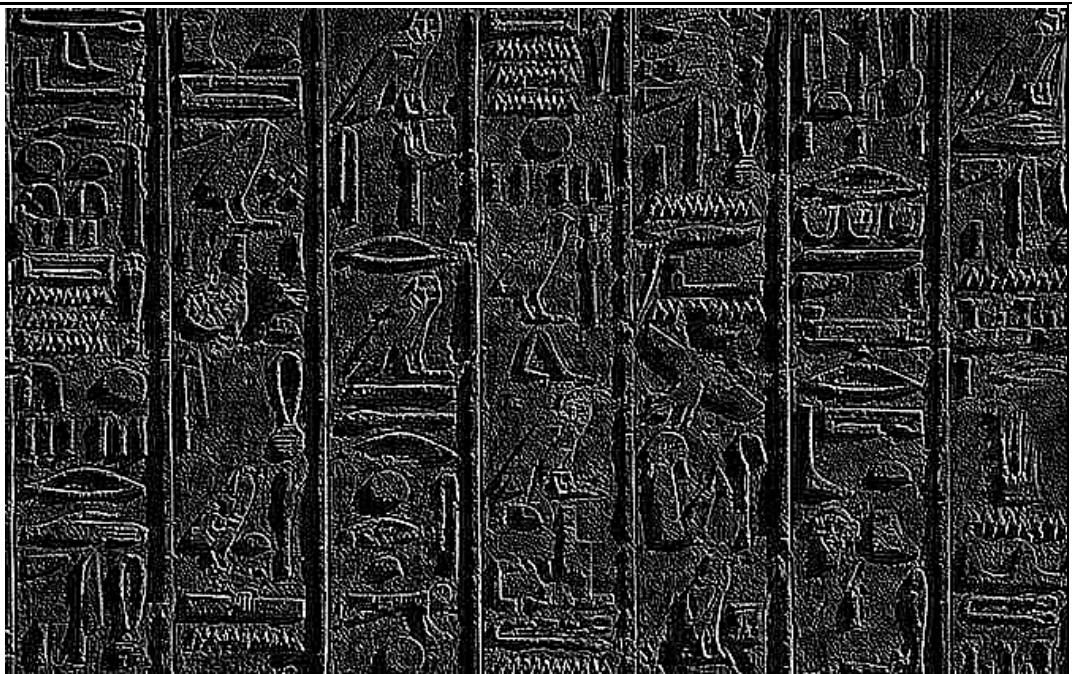
Figure



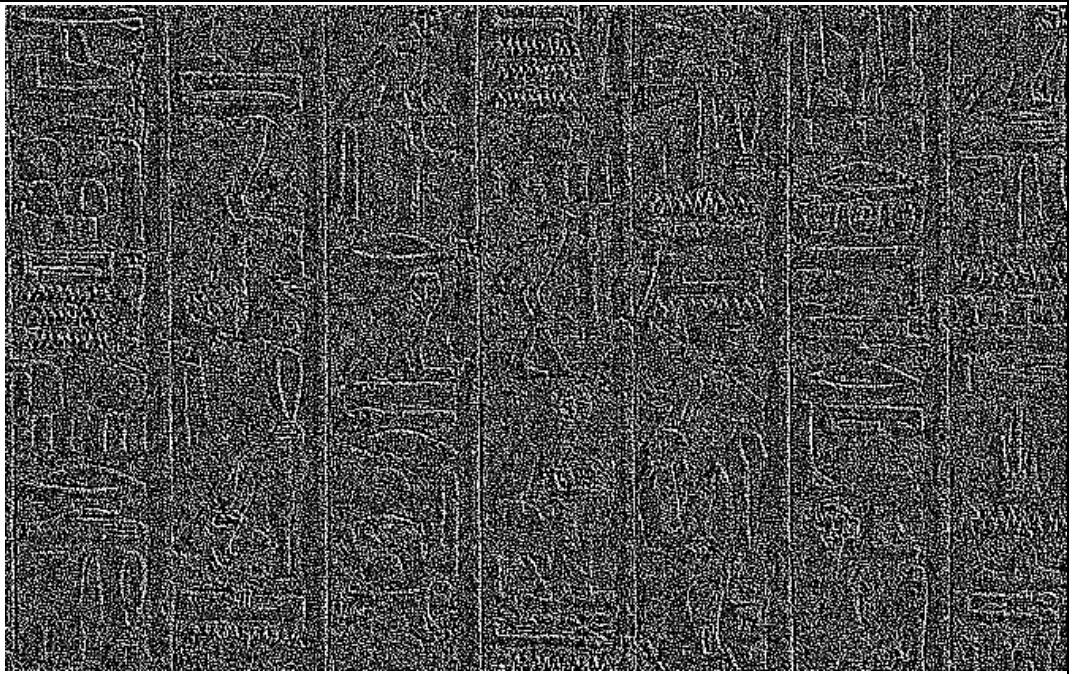
laplacian_4_slate.png



laplacian_4_slate_noise.png



laplacian_8_slate.png



laplacian_8_slate_noise.png

Discussion

4 neighbor – Laplacian filter:

$$[0 \ -1 \ 0]$$

$$[-1 \ 4 \ -1]$$

$$[0 \ -1 \ 0]$$

8 neighbor – Laplacian filter:

$[-1 -1 -1]$

$[-1 8 -1]$

$[-1 -1 -1]$

4 neighbor 只考慮上下左右的 pixel，所以對對角邊緣的檢測效果較差。而 8 neighbor 還考慮了對角線方向，可以檢測到更多的邊緣，4 neighbor 與 8 neighbor 都極怕雜訊的影響，而 8 更容易受雜訊的影響，在超出邊界時的處理採用 mirroring mask processing，而在項數值超過 255 時則設為 255，小於 0 則設為 0。

2.7

Discussion

Roberts 只能檢測少量的邊緣，會有較多的缺失

Prewitt 考慮了水平和垂直兩個方向，斜邊緣檢測效果差

Sobel 與 **Prewitt** 相似考慮了水平和垂直兩個方向，斜邊緣檢測效果差

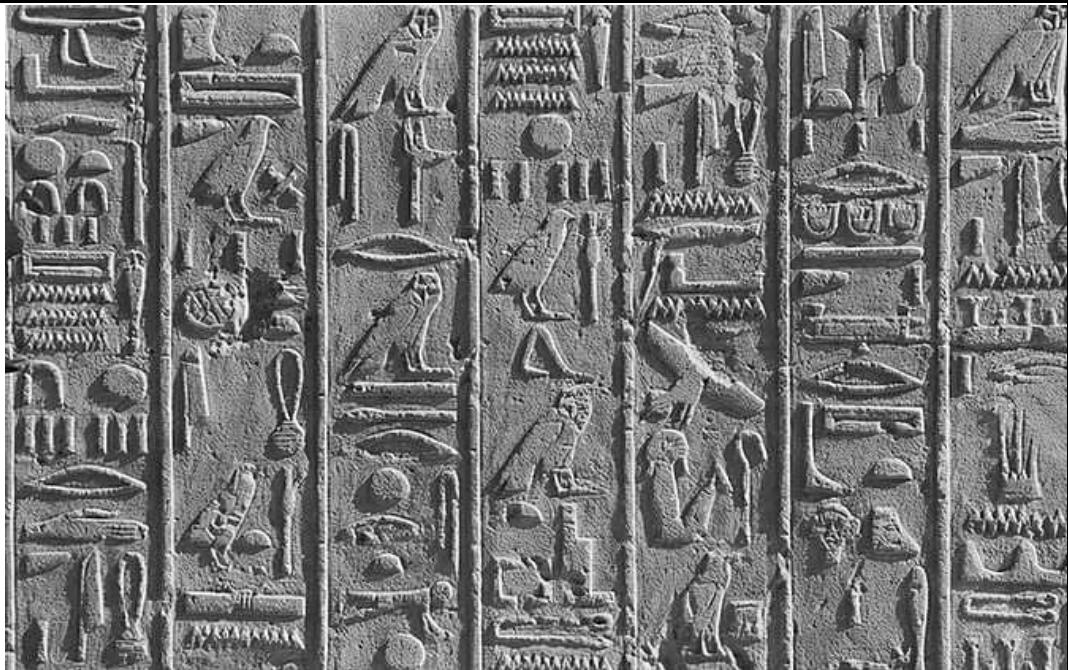
Sobel 45 僅考慮到斜線，缺乏垂直、水平資訊

Laplacian 檢測到邊緣，但沒有方向資訊

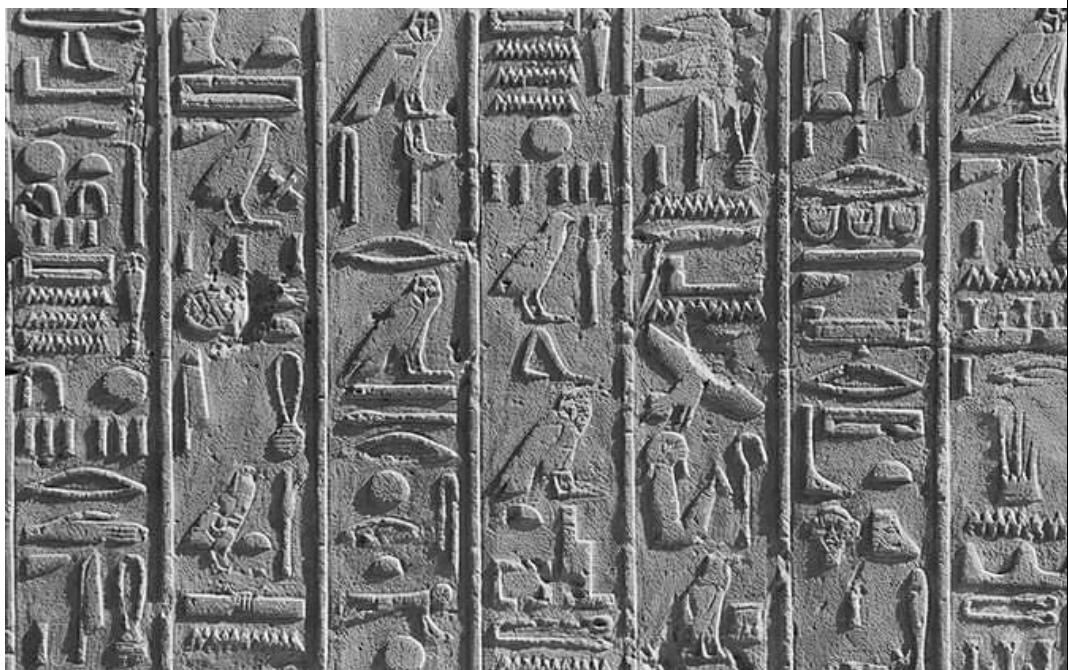
在本次的噪音表現中 **Laplacian < Sobel 45 = Sobel < Roberts = Prewitt**，但是面對不同噪音以及不同的後續處理方式也可能有不同結果，需要根據實際情況選擇適合的濾波器。

2.8

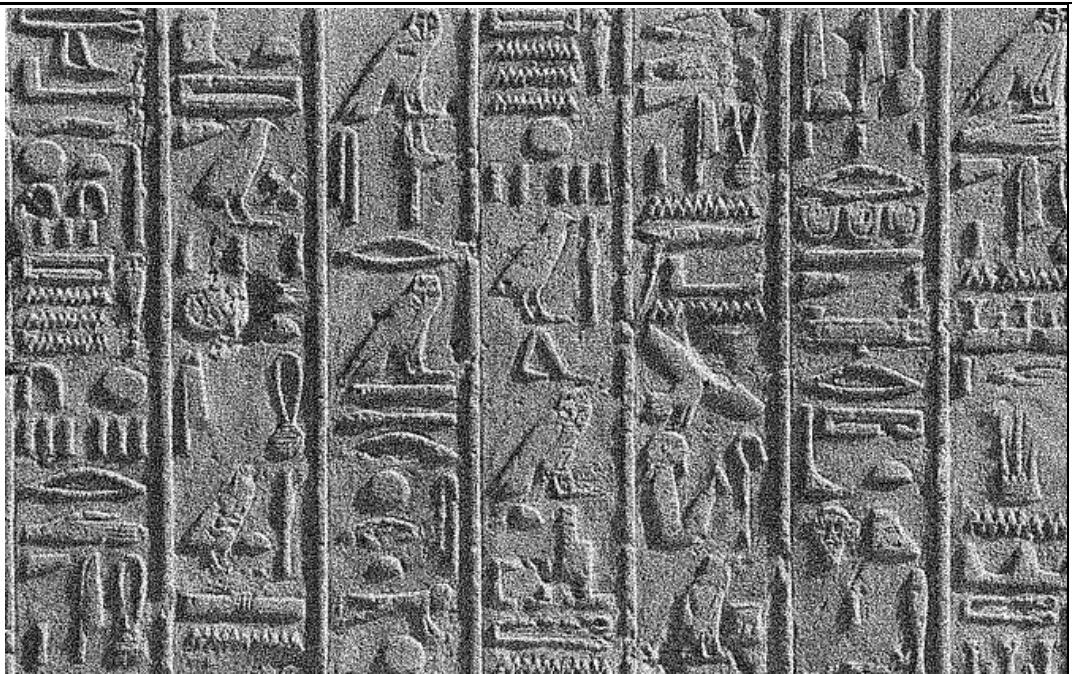
Figure



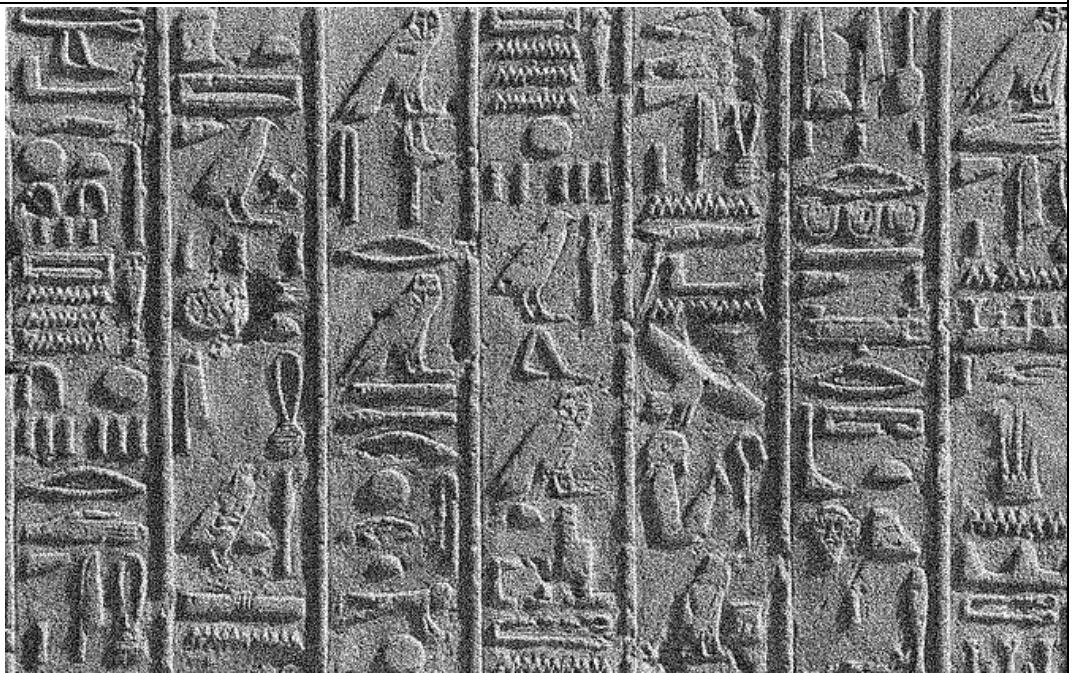
highboost_box_slate_.png



highboost_gau_slate_.png



highboost_box_slate_noise_.png



highboost_gau_slate_noise_.png

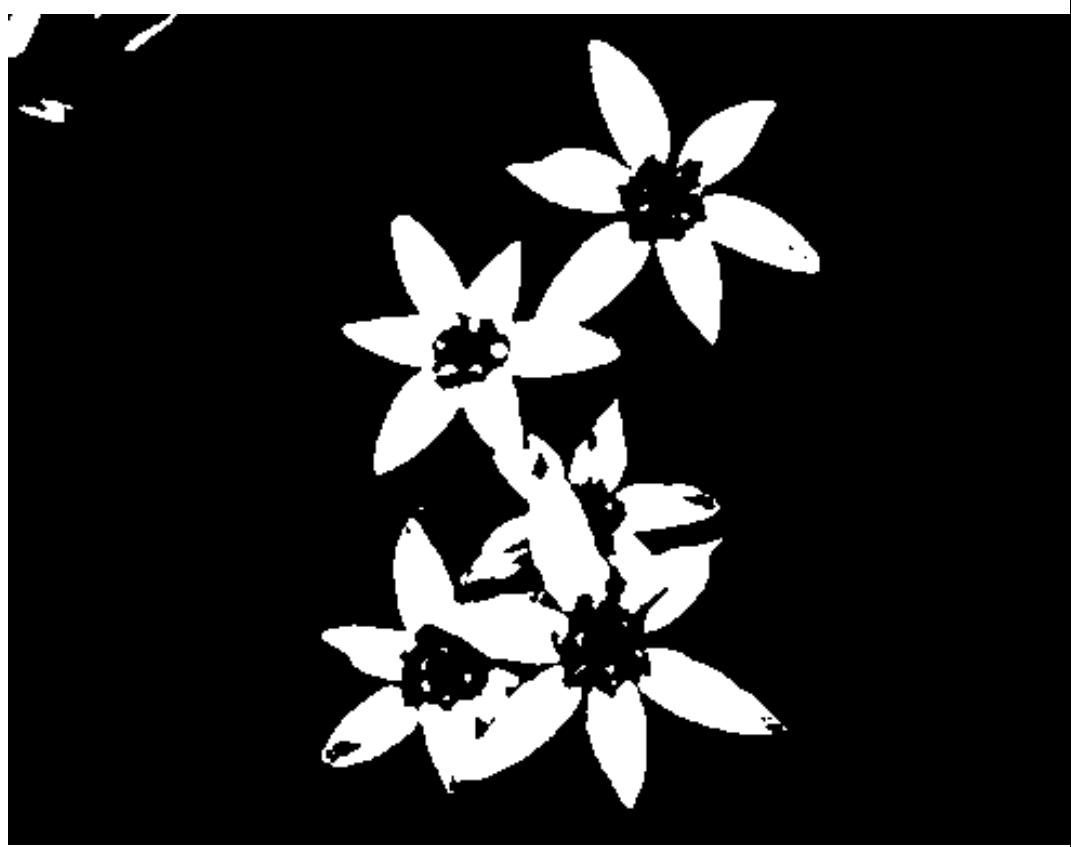
Discussion

High-boost filtering 可以定義為 $f_{hb}(x,y) = A^*f(x,y) - \bar{f}(x,y)$

$\bar{f}(x,y)$ 為 low pass filter，我在這邊分別設計了 box、gaussian 兩種平滑方式，filter size 都是用 3x3，A 的係數統一為 2，可以發現兩者差異並不大，主要在有噪音中 gaussian 的 high-boost 表現比起 box 在細節中要更好一些。

3.1

Figure



blur_thresholding_flower.png

Discussion

本題嘗試過 gaussian、box 兩種方式，兩者差異並不大最後採用 gaussian 的方式，可以發現左上角有殘留，因為在 thresholding 的設定上我嘗試過 128 會有許多背景沒有被去掉、220 會導致太多花瓣損失，而左上角的地方仍會有殘留，因此最後採用比較適中的 190 可以保留更多花瓣細節的前提下允許左上角的殘留。

3.2

Figure



logic.blur.flower.png

Discussion

本題對背景用 gaussian blur，並對前景使用 high-boost，可以發現主體的邊緣細節被更好的凸顯，而背景的模糊產生出單眼相機中的景深效果，在設置上 high-boost 的 A 設置為 2，在超出邊界時的處理採用 mirroring mask processing，而在項數值超過 255 時則設為 255，小於 0 則設為 0 。

。