**Code :**

**Project2.m (Main 主要執行) :**

close 'all';

clear 'all';

I = imread('img01.jpg'); %read file

J = imread('img02.jpg');

I\_G = rgb2gray(I); %transform RGB file into Gray-Scale file

figure(1),

subplot(2, 2, 1)

imshow(I\_G) %print original file

subplot(2, 2, 2)

imhist(I\_G) %print original file's histogram

subplot(2, 2, 3)

II = histeq(I\_G); %doing histogram equalization

imshow(II) %print equalized file

subplot(2, 2, 4)

imhist(II) %print equalized file's histogram

figure(2),

subplot(2, 2, 1)

imshow(J) %print original file

subplot(2, 2, 2)

imhist(J) %print original file's histogram

subplot(2, 2, 3)

JJ = histeq(J); %doing histogram equalization

imshow(JJ) %print equalized file

subplot(2, 2, 4)

imhist(JJ) %print equalized file's histogram

%/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sobel Operator\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

IS = rgb2gray(I); %transform RGB file into Gray-Scale file

C = double(IS); %turn the image array into double precission(Let computing more precise)

for i = 1 : size(C, 1) - 2 %from 1 to Xmax - 2

for j = 1 : size(C, 2) - 2 %from 1 to Ymax - 2

%Sobel mask for x-direction:

% j, j+1, j+2

%i [-1, -2, -1]

%i + 1 [ 0, 0, 0]

%i + 2 [+1, +2, +1]

Gx = ((C(i + 2, j) + 2 \* C(i + 2, j + 1) + C(i + 2, j + 2)) - (C(i, j) + 2 \* C(i, j + 1) + C(i, j + 2)));

%Sobel mask for y-direction:

% j, j+1, j+2

%i [-1, 0, +1]

%i + 1 [-2, 0, +2]

%i + 2 [-1, 0, +1]

Gy = ((C(i, j + 2) + 2 \* C(i + 1, j + 2) + C(i + 2, j + 2)) - (C(i, j) + 2 \* C(i + 1, j) + C(i + 2, j)));

IS(i, j) = abs(Gx) + abs(Gy);

end

end

figure(3),

subplot(2, 2, 1)

imshow(I); %print original file

subplot(2, 2, 2)

imshow(IS); %print sobeled file

JS = J;

C = double(JS);

for i = 1 : size(C, 1) - 2

for j = 1 : size(C, 2) - 2

%Sobel mask for x-direction:

% j, j+1, j+2

%i [-1, -2, -1]

%i + 1 [ 0, 0, 0]

%i + 2 [+1, +2, +1]

Gx = ((C(i + 2, j) + 2 \* C(i + 2, j + 1) + C(i + 2, j + 2)) - (C(i, j) + 2 \* C(i, j + 1) + C(i, j + 2)));

%Sobel mask for y-direction:

% j, j+1, j+2

%i [-1, 0, +1]

%i + 1 [-2, 0, +2]

%i + 2 [-1, 0, +1]

Gy = ((C(i, j + 2) + 2 \* C(i + 1, j + 2) + C(i + 2, j + 2)) - (C(i, j) + 2 \* C(i + 1, j) + C(i + 2, j)));

JS(i, j) = abs(Gx) + abs(Gy);

end

end

subplot(2, 2, 3)

imshow(J); %print original file

subplot(2, 2, 4)

imshow(JS); %print sobeled file

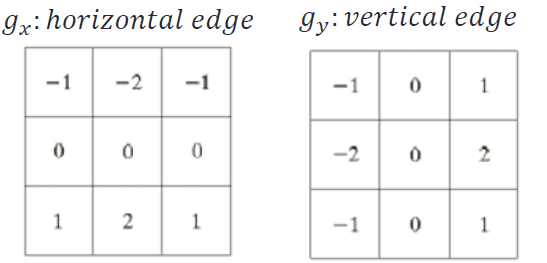
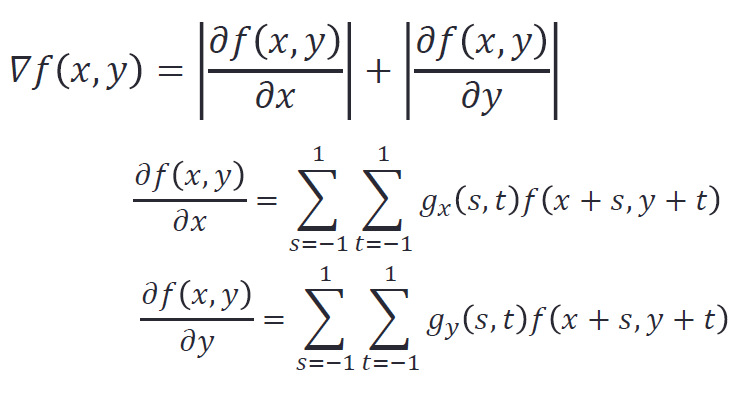
**程式碼解說 :**

Histogram Equalization :

使用matlab內建的function : “histeq()”就能進行Histogram Equalization，並且有用”imhist()”來印出分布情形，另外，處理RGB圖片要先將其轉為Gray-scale，才能使用”histeq()”來進行處理

Sobel operator :

首先利用”double()”來將array轉成double type以提高精準度，之後再用迴圈來進行計算，Gx、Gy的計算就是利用附圖的矩陣來做運算，最後再利用框起來的公式做運算，得到經過Sobel Operator運算後的圖片



**Result :**

