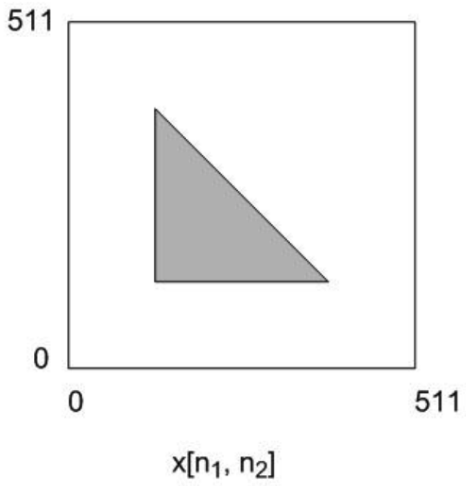
7.1 Assume that a bi-level input image $\mathbf{x[n_1, n_2]}$ of **512×512** as shown below, where the dark region has amplitude of **50** and the white background has amplitude of **200**.



An edge detector (consisting of two filters) is applied to $\mathbf{x[n_1, n_2]}$ which will produce two filtered images $\mathbf{G_1(x[n_1, n_2])}$ and $\mathbf{G_2(x[n_1, n_2])}$. These two images will be combined to form one gray level image $\mathbf{y[n_1, n_2]}$ using the absolute sum

$$\mathbf{y[n_1, n_2]=|G_1(x[n_1, n_2])| + |G_2(x[n_1, n_2])|}$$

A segmentation will then be performed on this image to produce a binary edge image. Assume that segmentation is based on the operator

$$z[n_1,n_2]=\begin{cases}\mathbf{255}\ (\textit{white}) & \textit{if} \quad y[n_1,n_2]\geq T\\ \mathbf{0}\ (\textit{black}) & \textit{if} \quad y[n_1,n_2]<T\end{cases}$$

7.1.1. Use Roberts edge detector on this image. Select an appropriate threshold values **T** such that **z[n1, n2]** will only show all the edges. Specify your threshold **T** and sketch your output image.

R1 Filter:

-1	0
0	1

R2 Filter:

0	-1
1	0

Vertical Edge:

R1

200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50

First Output:

$200 \bullet (-1) + 200 \bullet (0) + 200 \bullet (0) + 200 \bullet (1) = 0$

Second Output:

$200 \bullet (-1) + 50 \bullet (0) + 200 \bullet (0) + 50 \bullet (1) = -150$

Third Output:

$50 \bullet (-1) + 50 \bullet (0) + 50 \bullet (0) + 50 \bullet (1) = 0$

R2

200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50
200 200 200 50 50 50

First Output:

$200 \bullet (0) + 200 \bullet (-1) + 200 \bullet (1) + 200 \bullet (0) = 0$

Second Output:

$200 \bullet (0) + 50 \bullet (-1) + 200 \bullet (1) + 50 \bullet (0) = 150$

Third Output:

$50 \bullet (0) + 50 \bullet (-1) + 50 \bullet (1) + 50 \bullet (0) = 0$

Horizontal Edge:

R1

50 50 50 50 50 50
50 50 50 50 50 50
50 50 50 50 50 50
200 200 200 200 200 200
200 200 200 200 200 200
200 200 200 200 200 200

First Output:

$50 \bullet (-1) + 50 \bullet (0) + 50 \bullet (0) + 50 \bullet (1) = 0$

Second Output:

$50 \bullet (-1) + 50 \bullet (0) + 200 \bullet (0) + 200 \bullet (1) = 150$

Third Output:

$200 \bullet (-1) + 200 \bullet (0) + 200 \bullet (0) + 200 \bullet (1) = 0$

R2

50 50 50 50 50 50
50 50 50 50 50 50
50 50 50 50 50 50
200 200 200 200 200 200
200 200 200 200 200 200
200 200 200 200 200 200

First Output:

$50 \bullet (0) + 50 \bullet (-1) + 50 \bullet (1) + 50 \bullet (0) = 0$

Second Output:

$50 \bullet (0) + 50 \bullet (-1) + 200 \bullet (1) + 200 \bullet (0) = 150$

Third Output:

$200 \bullet (0) + 200 \bullet (-1) + 200 \bullet (1) + 200 \bullet (0) = 0$

Diagonal Edge:

R1

200 200 200 200 200 200
50 200 200 200 200 200
50 50 200 200 200 200
50 50 50 200 200 200
50 50 50 50 200 200
50 50 50 50 50 200
50 50 50 50 50 50
50 50 50 50 50 50

First Output:

$200 \bullet (-1) + 200 \bullet (0) + 200 \bullet (0) + 200 \bullet (1) = 0$

Second Output:

$200 \bullet (-1) + 200 \bullet (0) + 50 \bullet (0) + 200 \bullet (1) = 0$

Third Output:

$50 \bullet (-1) + 200 \bullet (0) + 50 \bullet (0) + 50 \bullet (1) = 0$

Fourth Output:

$50 \bullet (-1) + 50 \bullet (0) + 50 \bullet (0) + 50 \bullet (1) = 0$

R2

200 200 200 200 200 200
50 200 200 200 200 200
50 50 200 200 200 200
50 50 50 200 200 200
50 50 50 50 200 200
50 50 50 50 50 200
50 50 50 50 50 50
50 50 50 50 50 50

First Output:

$200 \bullet (0) + 200 \bullet (-1) + 200 \bullet (1) + 200 \bullet (0) = 0$

Second Output:

$200 \bullet (0) + 200 \bullet (-1) + 50 \bullet (1) + 200 \bullet (0) = -150$

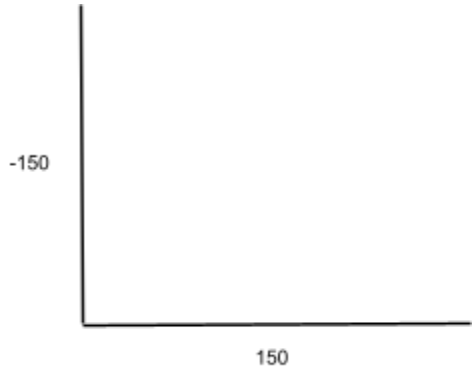
Third Output:

$50 \bullet (0) + 200 \bullet (-1) + 50 \bullet (1) + 50 \bullet (0) = -150$

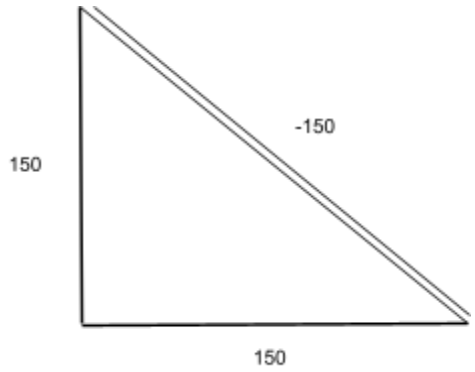
Fourth Output:

$50 \bullet (0) + 50 \bullet (-1) + 50 \bullet (1) + 50 \bullet (0) = 0$

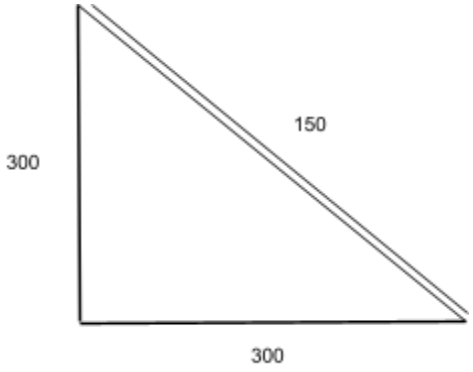
G_X



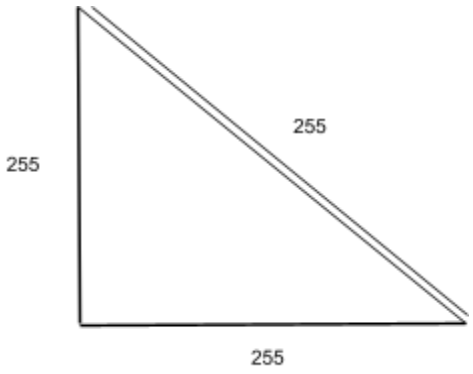
G_Y



$|G_X|+|G_Y|$ =



With Threshold of 100:



7.1.2. Repeat **7.1.1.** using Prewitt edge detector.

(**Note:** each actual edge may result in single, double, triple or more lines depending on different edge detector, please specify the lines you get.)

P1 Filter:

-1	-1	-1
0	0	0
1	1	1

P2 Filter:

-1	0	1
-1	0	1
-1	0	1

Vertical Edge:

P1

200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50

First Output:

$200\bullet(-1) + 200\bullet(-1) + 200\bullet(-1) + 200\bullet(1) + 200\bullet(1) + 200\bullet(1) = 0$

Second Output:

$200\bullet(-1) + 200\bullet(-1) + 50\bullet(-1) + 200\bullet(1) + 200\bullet(1) + 50\bullet(1) = 0$

Third Output:

$200\bullet(-1) + 50\bullet(-1) + 50\bullet(-1) + 200\bullet(1) + 50\bullet(1) + 50\bullet(1) = 0$

Fourth Output:

$50\bullet(-1) + 50\bullet(-1) + 50\bullet(-1) + 50\bullet(1) + 50\bullet(1) + 50\bullet(1) = 0$

P2

200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50
200	200	200	50	50	50

First Output:

$200\bullet(-1) + 200\bullet(1) + 200\bullet(-1) + 200\bullet(1) + 200\bullet(-1) + 200\bullet(1) = 0$

Second Output:

$200\bullet(-1) + 50\bullet(1) + 200\bullet(-1) + 50\bullet(1) + 200\bullet(-1) + 50\bullet(1) = -450$

Third Output:

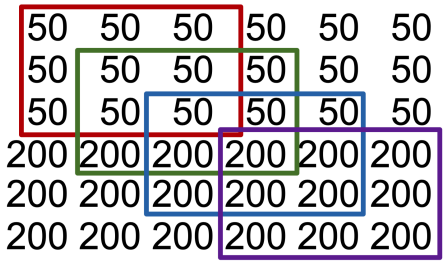
$200\bullet(-1) + 50\bullet(1) + 200\bullet(-1) + 50\bullet(1) + 200\bullet(-1) + 50\bullet(1) = -450$

Fourth Output:

$50\bullet(-1) + 50\bullet(-1) + 50\bullet(-1) + 50\bullet(1) + 50\bullet(1) + 50\bullet(1) = 0$

Horizontal Edge:

P1



First Output:

$50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (1) + 50 \bullet (1) + 50 \bullet (1) = 0$

Second Output:

$50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) = 450$

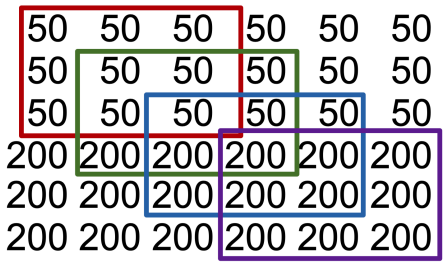
Third Output:

$50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) = 450$

Fourth Output:

$200 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) = 0$

P2



First Output:

$50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (1) + 50 \bullet (1) + 50 \bullet (1) = 0$

Second Output:

$50 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (-1) + 50 \bullet (1) + 50 \bullet (1) + 200 \bullet (1) = 0$

Third Output:

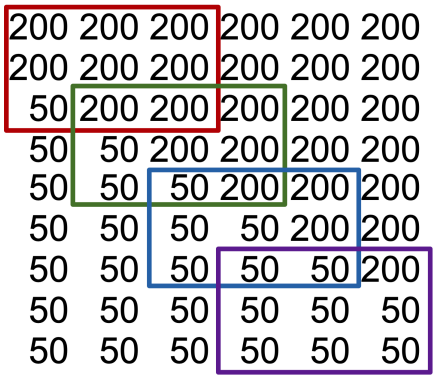
$50 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (-1) + 50 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) = 0$

Fourth Output:

$200 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) = 0$

Diagonal Edge:

P1



First Output:

$200 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (-1) + 50 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) =$ -150

Second Output:

$200 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (-1) + 50 \bullet (1) + 50 \bullet (1) + 200 \bullet (1) =$ -300

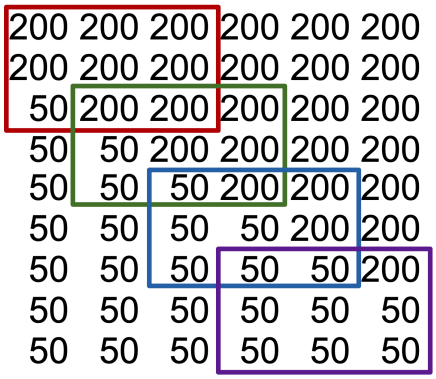
Third Output:

$50 \bullet (-1) + 200 \bullet (-1) + 200 \bullet (-1) + 50 \bullet (1) + 50 \bullet (1) + 50 \bullet (1) =$ -300

Fourth Output:

$50 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (-1) + 50 \bullet (1) + 50 \bullet (1) + 50 \bullet (1) =$ -150

P2



First Output:

$200 \bullet (-1) + 200 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) =$ 150

Second Output:

$200 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (1) + 200 \bullet (1) + 200 \bullet (1) =$ 300

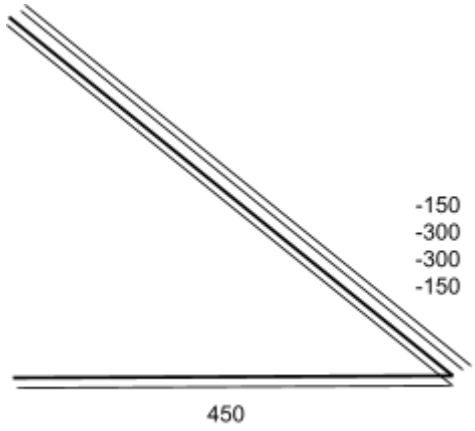
Third Output:

$50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (1) + 200 \bullet (1) + 50 \bullet (1) =$ 300

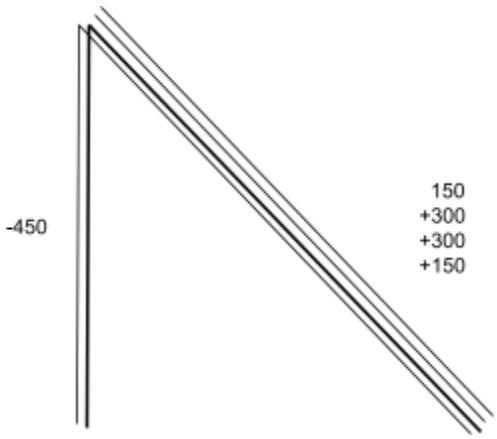
Fourth Output:

$50 \bullet (-1) + 50 \bullet (-1) + 50 \bullet (-1) + 200 \bullet (1) + 50 \bullet (1) + 50 \bullet (1) =$ 150

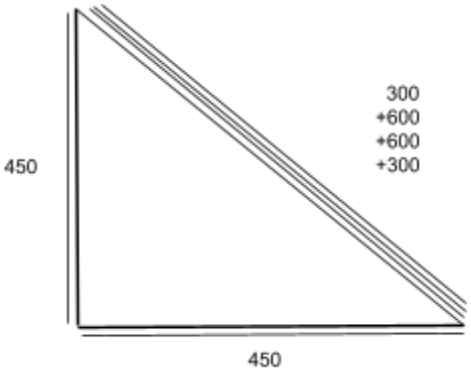
G_X



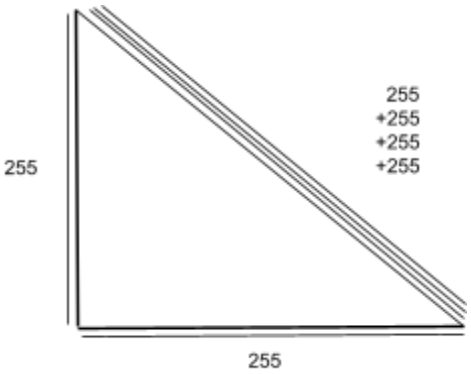
G_Y



$|G_X|+|G_Y|$ =



With Threshold of 100:



7.2 Based on the **imageprocessing.c** structure, write a small routine which can automatically calculate the global threshold value according to the iterative global threshold estimation algorithm we discussed in class.

Hint: you have to initialize a **T**; then read through the image several times to update the **T**; your iteration will stop when your newly updated $\mathbf{T_i}$ is not much different from the previous $\mathbf{T_{i-1}}$, i.e. $|\mathbf{T_i - T_{i-1}}| < \mathbf{a}$. You can let $\mathbf{a = 5}$ for example. You should try to let the program display the updated $\mathbf{T_i}$ at each iteration so you'll have an idea of whether it is running properly. Finally you should apply your final **T** to the image and obtained a binary output image and print out the result.

$$\mathbf{image\_out[n_1,n_2]} = \begin{cases} \mathbf{255\ (white)} & \mathbf{if\quad image\_in[n_1,n_2] \geq T} \\ \mathbf{0\ (black)} & \mathbf{if\quad image\_in[n_1,n_2] < T} \end{cases}$$

```
int Ti = 0;
int T = 128;
int a = 0;

int sum1, sum2, num1, num2;

for (i = 0; (i < 100) && !(abs(Ti - T) <= a); i++)
{
    Ti = T;
    sum1 = 0;
    sum2 = 0;
    num1 = 0;
    num2 = 0;
    for (j = 0; j < height; j++)
        for (k = 0; k < width; k++)
            if (image_in[j][k] < Ti)
            {
                sum1 += image_in[j][k];
                num1++;
            } else
            {
                sum2 += image_in[j][k];
                num2++;
            }
    T = (sum1 / num1 + sum2 / num2) / 2;
    cout << abs(Ti - T) << endl;
}
Ti = T;

for (j = 0; j < height; j++)
    for (k = 0; k < width; k++)
    {
        if (image_in[j][k] < Ti)
            image_out[j][k] = 0;
        else
            image_out[j][k] = 255;
    }
```

Intput

Output