Assignment 2

/*叶俊利 3130100672*/

1. Binarization (Global Thresholding)

Because

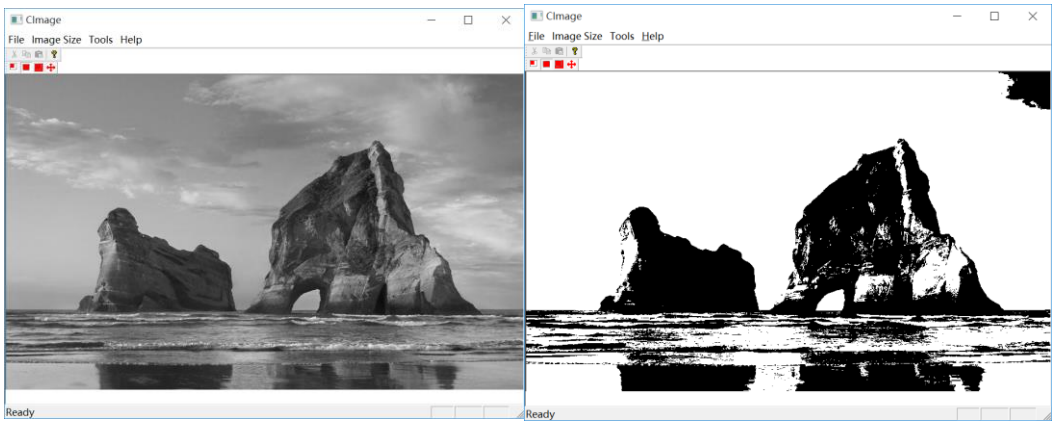
$$\sigma^2_{between}(T) = \frac{N_{Fgrd}}{N}(\mu_{Fgrd} - \mu)^2 + \frac{N_{Bgrd}}{N}(\mu_{Bgrd} - \mu)^2$$

So, I try every value of T from 0 to 255, to get the value when the variance is maximized.

The code of calculating the theshold is shown below:

```
for (int T = 0; T < 255; T++){
        int  countFG = 0, countBG = 0;
        int sumTotal = 0, sumFG = 0, sumBG = 0;
        double avgTotal, avgFG, avgBG;
        for (int i = 0; i < h; i++)
        for (int j = 0; j < w; j++){
                sumTotal += grayScale[i][j];
                if (grayScale[i][j]>T){
                        sumFG += grayScale[i][j];
                        countFG++;
                }
                else{
                        sumBG += grayScale[i][j];
                        countBG++;
                }
        }
        avgBG = (countBG > 0) ? sumBG / countBG : 0;
        avgFG = (countFG > 0) ? sumFG / countFG : 0;
        avgTotal = sumTotal / countTotal;
        double between = (double)countBG / countTotal * (avgBG - avgTotal)*(avgBG -
avgTotal) + (double)countFG / countTotal * (avgFG - avgTotal)*(avgFG - avgTotal);
        if (between > maxBetween){
                resT = T;
                maxBetween = between;
        }
```

According to the thres hold, we can get the binary result:



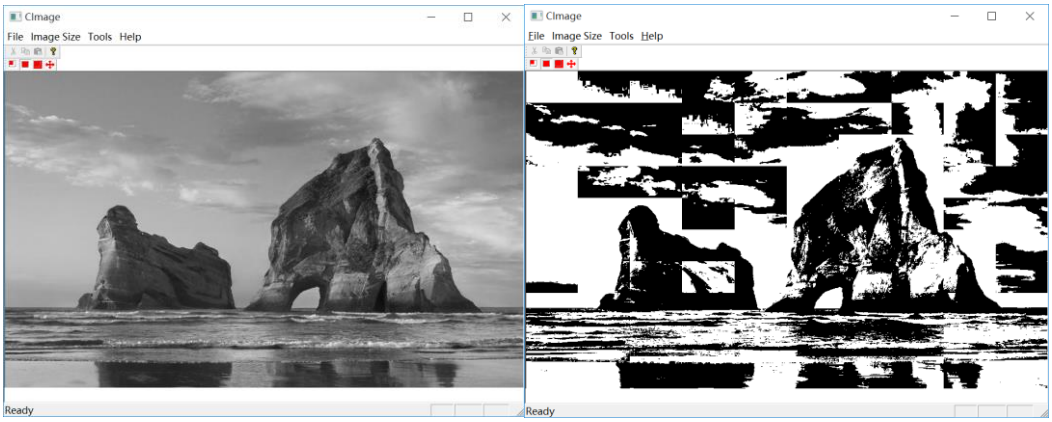
2. Binarization (Local Thresholding)

First, I divide the image in to 10 * 10 blocks.

Then, calculate the threshold of each block.

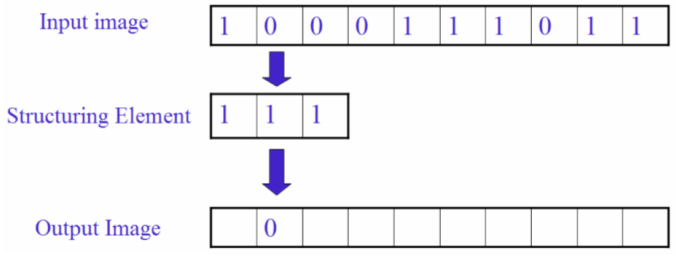
Third, get the binary result of that block.

So we can get:



3. Erosion

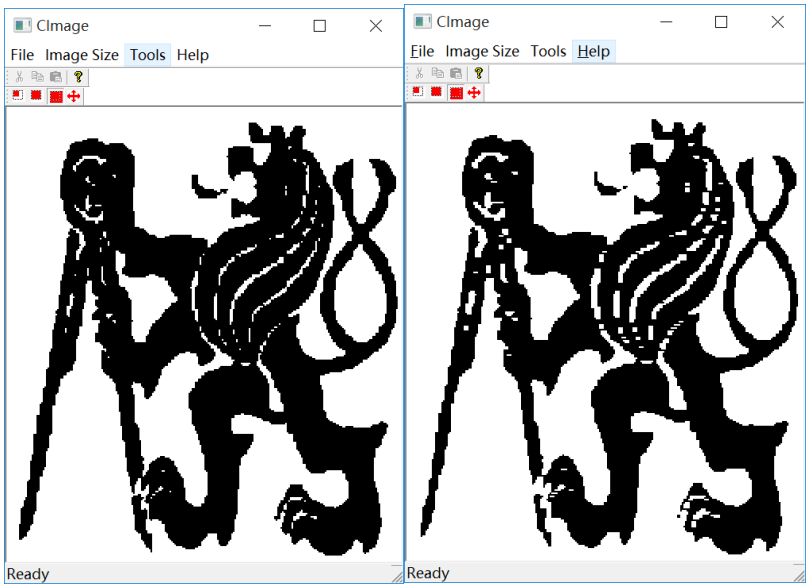
For every row of the image, I do the following operation.



The code is shown below:

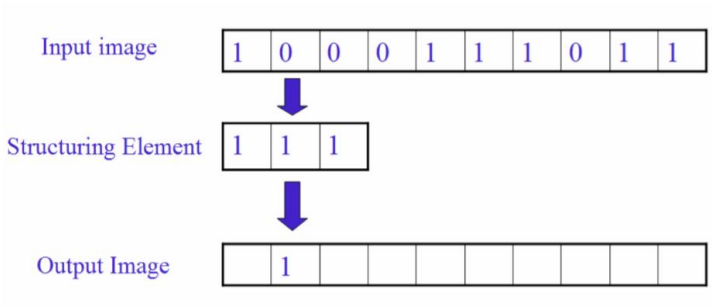
```
                for (int i = 0; i < h; i++)
                for (int j = 0; j < w; j++){
                        if (j - 1 < 0 && j + 1 >= w)
                                erosion[i][j] = grayScale[i][j];
                        else if(grayScale[i][j - 1] == 0 && grayScale[i][j] == 0 && grayScale[i][j + 1]
== 0)
                                erosion[i][j] = 0;
                        else
                                erosion[i][j] = 255;
                }
                for (int i = 0; i < h; i++)
                for (int j = 0; j < w; j++){
                        imgOriginal.SetPixelRGB(j, i, erosion[i][j], erosion[i][j], erosion[i][j]);
                }
```

So we can get:



4. Dilation

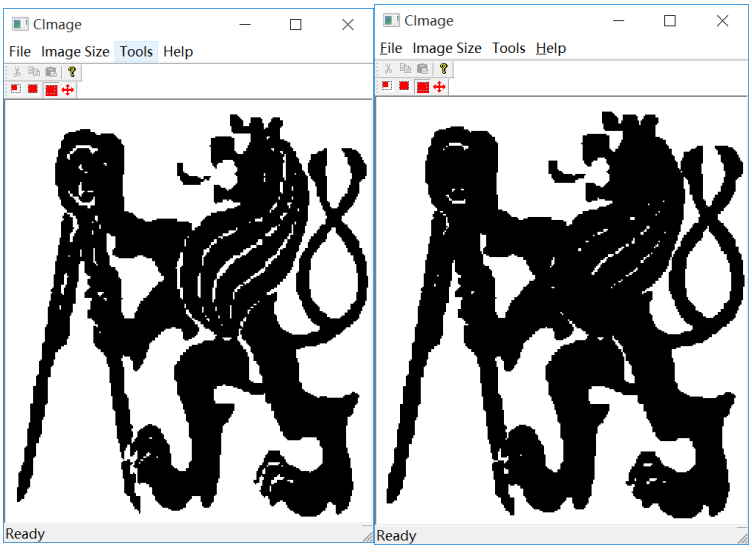
For every row of the image, I do the following operation.



The code is shown below:

```
                for (int i = 0; i < h; i++)
                for (int j = 0; j < w; j++){
                        if (j - 1 < 0 && j + 1 >= w)
                                dilation[i][j] = grayScale[i][j];
                        else if (grayScale[i][j - 1] == 0 || grayScale[i][j] == 0 || grayScale[i][j +
1] == 0)
                                dilation[i][j] = 0;
                        else
                                dilation[i][j] = 255;
                }
                for (int i = 0; i < h; i++)
                for (int j = 0; j < w; j++){
                        imgOriginal.SetPixelRGB(j, i, dilation[i][j], dilation[i][j], dilation[i][j]);
                }
```

So we can get:

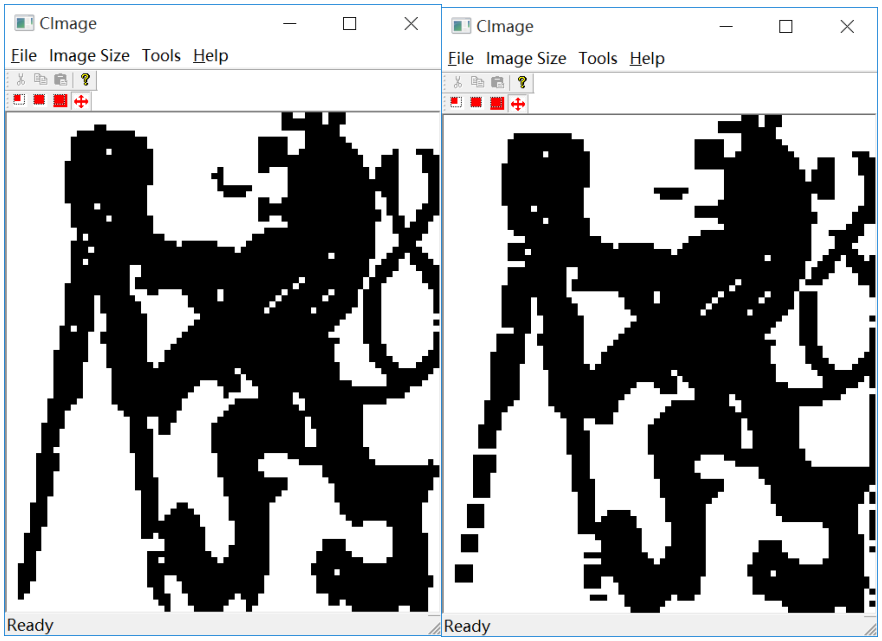


5. Openning

First, do Erosion.

Second, do Dilation.

So we can get:



6. Closing

First, do Dilation.

Second, do Erosion.

So we can get: