Assignment 3

/*叶俊利 3130100672*/

1. Logarithmic Operation for Visibility Enhancement

First, read in the image and store the YUV matrix, and get the max value of Y.

Then, according to the equation

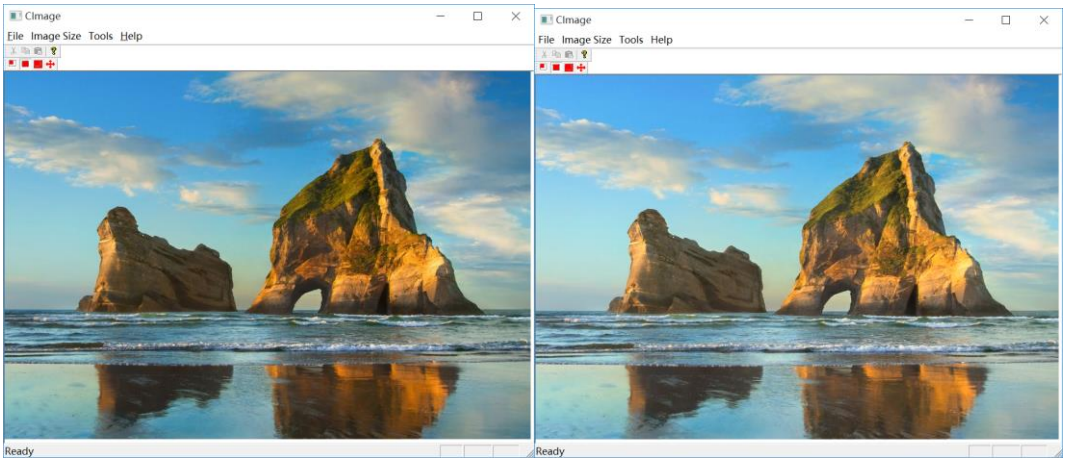
$$L_d = \frac{\log(L_w+1)}{\log(L_{max}+1)}$$

So we can process the image according to the equation.

The code is shown below.

```
pixel = imgOriginal.GetPixel(j, i);
r = GetRValue(pixel);
g = GetGValue(pixel);
b = GetBValue(pixel);
Y[i][j] = 0.3*r + 0.59*g + 0.11*b;
U[i][j] = 0.493*(b - Y[i][j]);
V[i][j] = 0.877*(r - Y[i][j]);
if (Y[i][j]>max)
        max = Y[i][j];
```

According to the this, we can get the result:



2. Histogram Equalization

First, we need to convert the image to YUV, get the Y histrogram of the image.
To store the grayscale histogram of the image, I use an array n[255] to store the number of pixel of each Y intensity.

The code to prepare the histogram is shown below:

```
byte r, g, b, avg;
pixel = imgOriginal.GetPixel(j, i);
r = GetRValue(pixel);
g = GetGValue(pixel);
b = GetBValue(pixel);
Y[i][j] = 0.3*r + 0.59*g + 0.11*b;
U[i][j] = 0.493*(b - Y[i][j]);
V[i][j] = 0.877*(r - Y[i][j]);
n[(int)Y[i][j]]++;
```

Second, we need to determine the function T. To represent the function, I use a array, transform[255], to represent the rule of transforming. For example, if transform[150]=160, it means that all pixel of Y value 150 should be converted to 160.

To calculate the array transform[256], we need the follow equation:

$$s_k = T(r_k) = \frac{1}{n}\sum_{i=0}^{k} n_i$$

Thus I write the follow code:

```
int accumulate = 0;
int tranform[256];
for (int i = 0; i < 256; i++){
        accumulate += n[i];
        double p = (double)accumulate / h / w;
        tranform[i] = round1(p * 255);
}
```

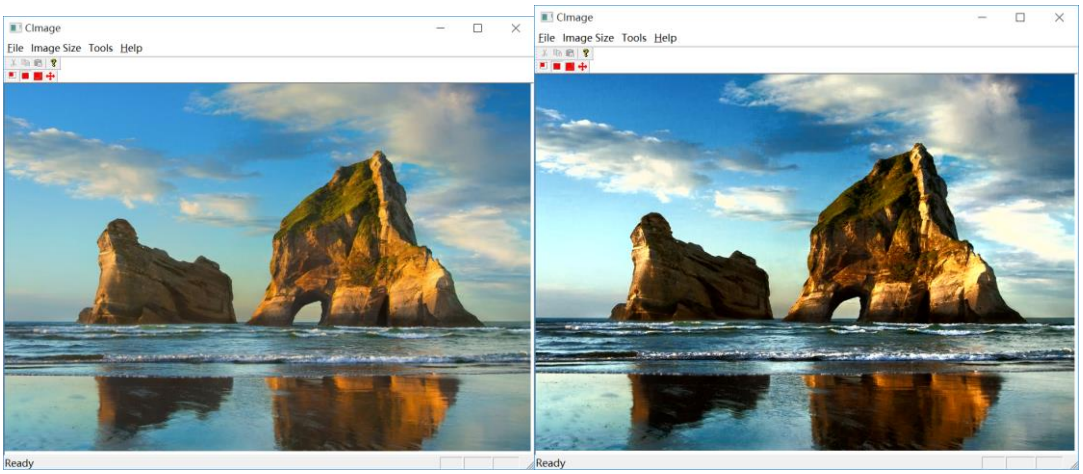
At last, we just need to convert the image according to the array transform[256]:

```
for (int i = 0; i < h; i++)
for (int j = 0; j < w; j++){
        Y[i][j] = tranform[(int)Y[i][j]];
        double r = Y[i][j] + 1.14 * V[i][j];
        double b = Y[i][j] + 2.03 * U[i][j];
```

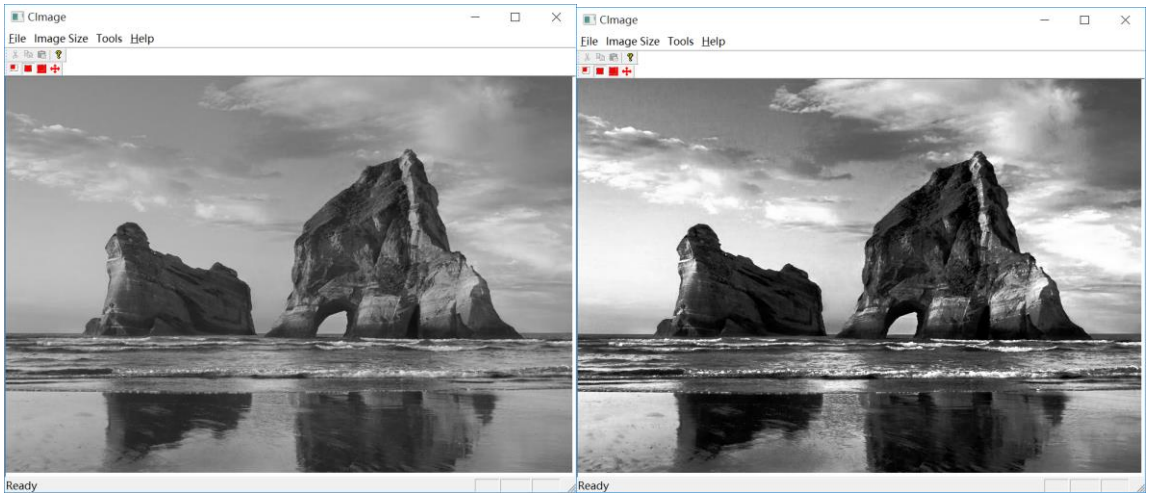
```
            double g = Y[i][j] - 0.39*U[i][j] - 0.58* V[i][j];
            r = r>255 ? 255 : r;
            g = g>255 ? 255 : g;
            b = b>255 ? 255 : b;
            r = r<0 ? 0 : r;
            g = g<0 ? 0 : g;
            b = b<0 ? 0 : b;

            imgOriginal.SetPixelRGB(j, i, r, g, b);
        }
```

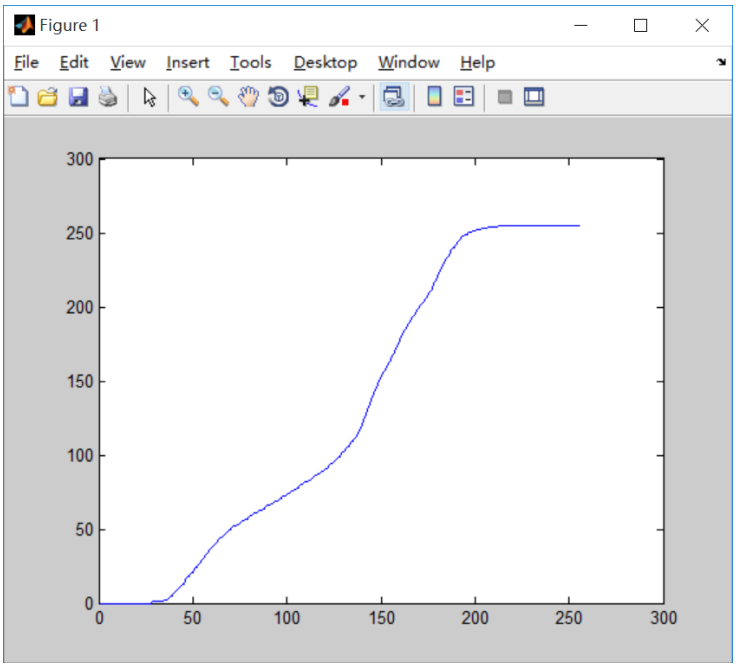
So we can get:



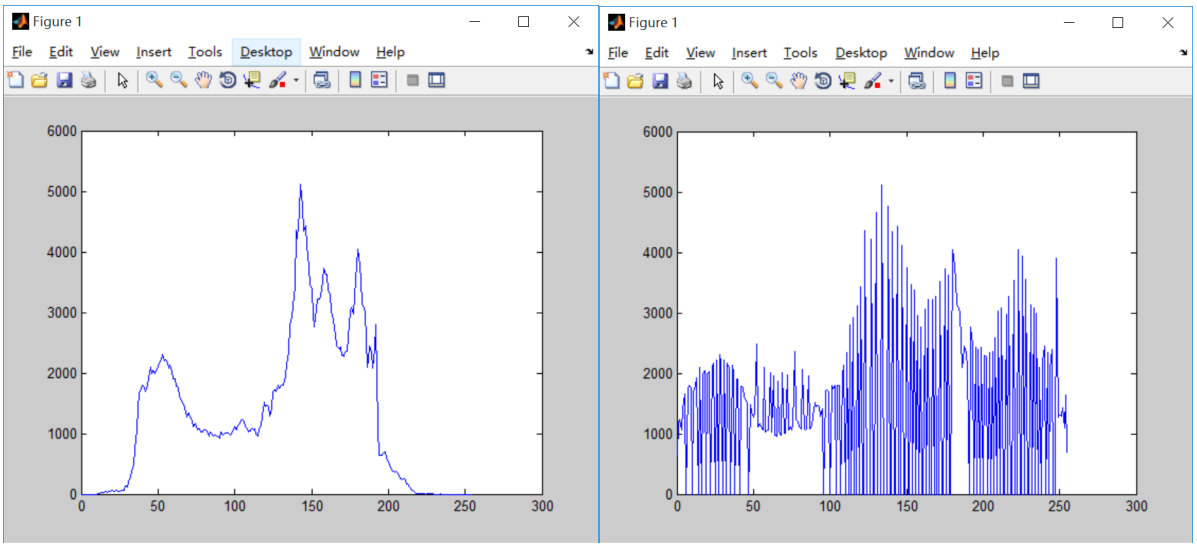
If the picture is black & white:



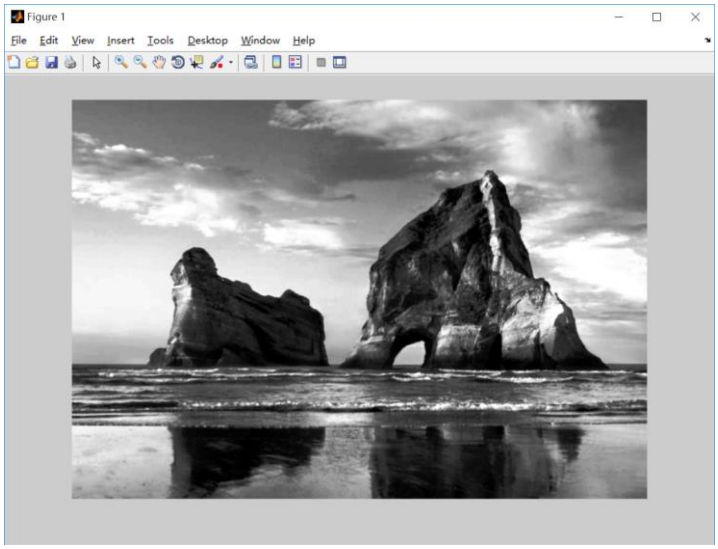
In this case, the transformation function transform[256] is:



And the following is the change of histogram:



To verify the result, the following image is from MATLAB:



Exactly the same with mine.