HW2

1 Exercise

1.1 Suppose that you have performed histogram equalization on a digital image. Will a second pass of histogram equalization (on the histogram-equalized image) produce exactly the same result as the first pass? Prove your answer.

Let consider an example:

$$\begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 2 & 0 & 2 \\ 0 & 5 & 7 & 0 \\ 2 & 2 & 5 & 2 \end{bmatrix}$$

r	0	1	2	3	4	5	6	7
n	5	2	6	0	0	2	0	1
p	5/16	1/8	3/8	0	0	1/8	0	1/16
C	5/16	7/16	13/16	13/16	13/16	15/16	15/16	1
s	2	3	6	6	6	7	7	7

The result image is:

$$\begin{bmatrix} 2 & 2 & 3 & 6 \\ 3 & 6 & 2 & 6 \\ 2 & 7 & 7 & 2 \\ 6 & 6 & 7 & 6 \end{bmatrix}$$

r	0	1	2	3	4	5	6	7
n	0	0	5	2	0	0	6	3
p	0	0	5/16	1/8	0	0	3/8	3/16
C	0	0	5/16	7/16	7/16	7/16	13/16	1
s	0	0	2	3	3	3	6	7

The second pass result is:

$$\begin{bmatrix} 2 & 2 & 3 & 6 \\ 3 & 6 & 2 & 6 \\ 2 & 7 & 7 & 2 \\ 6 & 6 & 7 & 6 \end{bmatrix}$$

So a second pass of histogram equalization on the histogram-equalized image produce exactly the same result as the first pass.

Prove:

To prove the conclusion, we should firstly prove $S_k=S_k', S_k=T(r_k), S_k'=T(S_k)$.

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} P_r(r_j)$$

That is to prove $r_k=S_k$. And $r_k \in [0, L-1]$, we only need to prove $\sum P_r(r_j)$

$\in [0,1]$. This is obviously true.

So we have proved the conclusion that a second pass of histogram equalization on the histogram-equalized image produce exactly the same result as the first pass.

1.2

Consider a 4×4 gray image and a 3×3 filter:

$$Image: \begin{bmatrix} 85 & 13 & 29 & 83 \\ 169 & 8 & 243 & 28 \\ 17 & 155 & 27 & 33 \\ 91 & 25 & 173 & 79 \end{bmatrix} \quad Filter: \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

1. Convolve the gray image with the given filter with zero-padding, and show your result (whose size should be 4×4).

$$\begin{bmatrix} -177 & -420 & -279 & -271 \\ -74 & -72 & -90 & 52 \\ 61 & 131 & 2 & 19 \\ 172 & 199 & 215 & 60 \end{bmatrix}$$

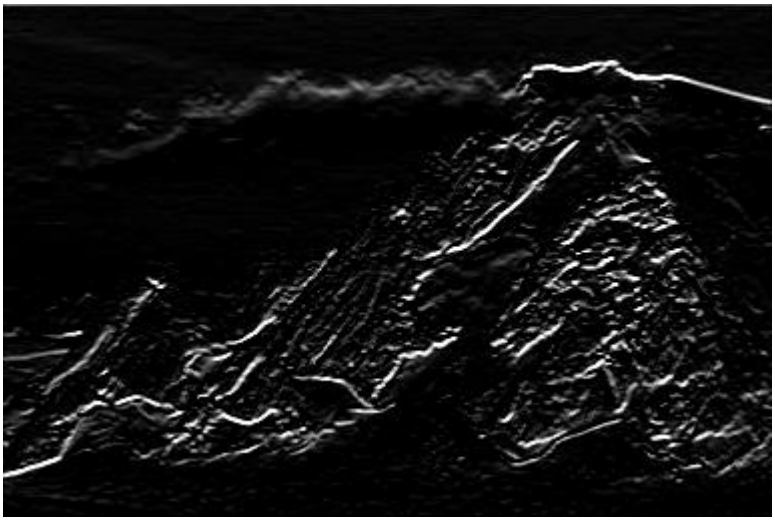
2. Discuss the meanings of positive values and negative values in your convolution result respectively.

   The positive values stand for one side of an edge and the negative values stand for the other side of the edge.

3. Describe some applications of the given filter based on your own knowledge.

The source picture:

The picture using the given filter:



The application of the given filter is to highlight the edges of things in the picture.

2 Programming

2.1 I make use of the openVC library to load the picture and store it.

```
#include <cv.h>
#include <highgui.h>
```

```
//读入原图
Mat img = imread("./81.png");
```

```
    waitKey(0);
    imwrite("./Highboost.png", highBoost);
```

```
    waitKey(0);
    imwrite("./Sharpen.png", filter2d(img, Lap, 3));
```

```
    waitKey(0);
    imwrite("./11*11.png", filter2d(img, ave3, 11));
```

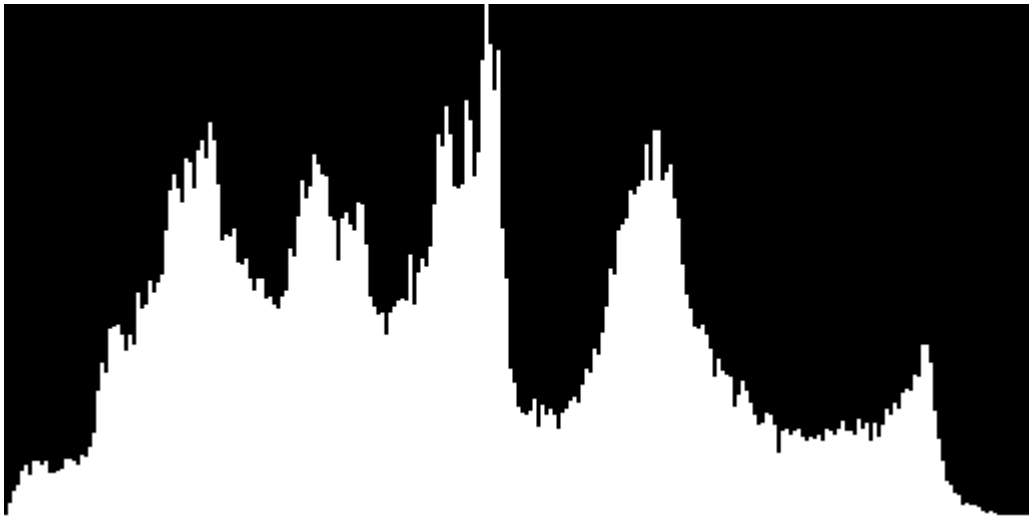
```
    waitKey(0);
    imwrite("./7*7.png", filter2d(img, ave2, 7));
```

```
    waitKey(0);
    imwrite("./3*3.png", filter2d(img, ave1, 3));
```

```
waitKey(10000000000);
imwrite("./Source.png", img1);
imwrite("./SourceHist.png", hist_img1);
imwrite("./Result.png", res);
imwrite("./ResultHist.png", hist_res);
```

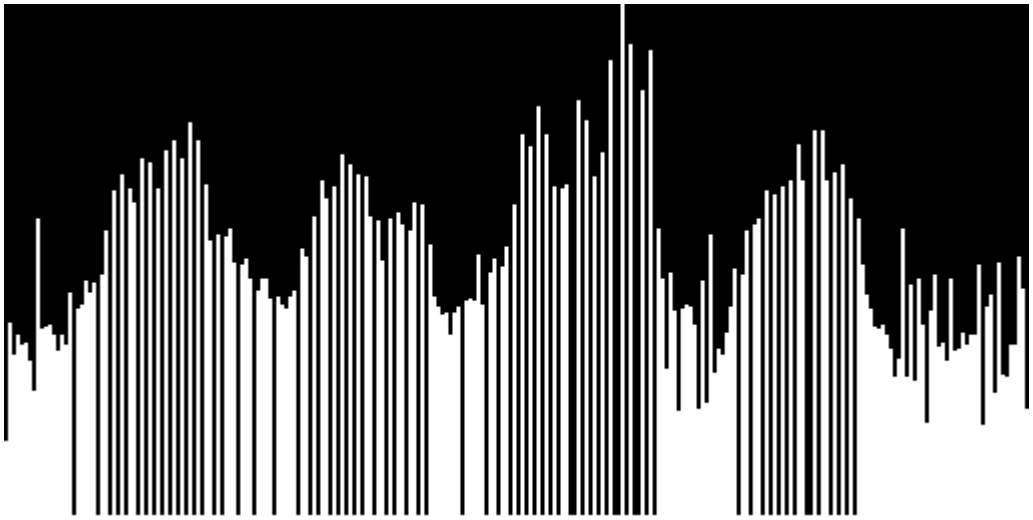
This is my original picture----81.png.

2.2 1. Source histogram:



2. Result:

Result histogram:



3. Analyzation:

The result histogram has a more balanced distribution of gray values than the original one. It has more values on the darker side and brighter side. So the result picture looks better than the source picture since it has more detail on the darker part and brighter part.

4. Implementation:

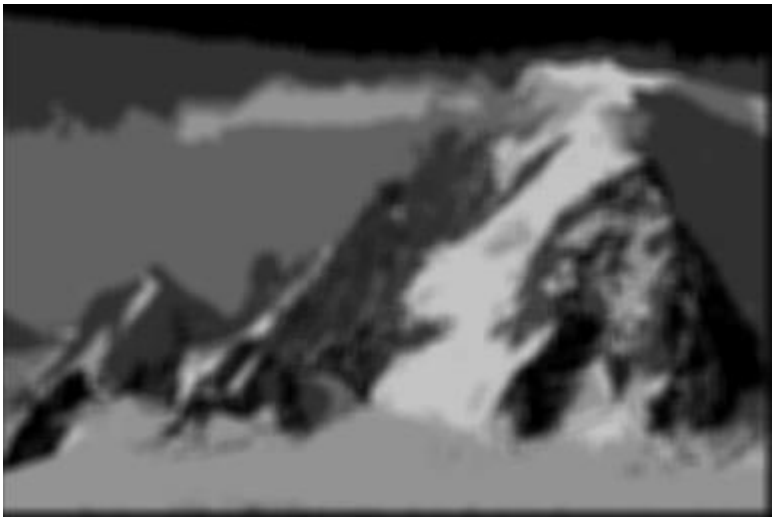
The algorithm has five parts. The first part is calculating the number of each gray value. You can scan the picture and count up the numbers. The second part is calculating the probability of each gray value. You can easily get it dividing the number of a gray value by the size of the picture. Then you can go to the third part----calculating the cumulative distribution. The cumulative value of a gray value is depending on its probability and the cumulative value of the former gray value.

The forth part is turning the cumulative value into an integer. Because the probabilities are floating numbers, the cumulative values are also floating numbers. You should transform them into integers between 0 and 255. The last part of the algorithm is to use the cumulative value of a pixel's old gray value as that pixel's new gray value.

2.3 1. Smoothing:

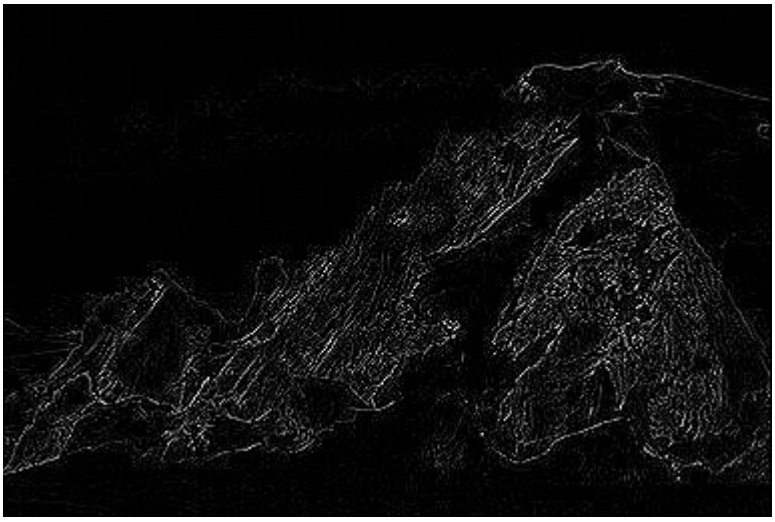
3*3:

7*7:



11*11:

2. Sharpening:



I choose the filter $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ to sharpen the picture.

The Laplacian filter is a differential filter. It mainly focuses on the slope of function. It can highlight these parts where gray values change rapidly and ignore the areas where gray values change slowly. As a result, it can sharpen the picture.

3. High-boosting:

The selected K is 1.2.

4. Implementation:

The algorithm can be divided into two parts. The first part is receiving the filter and calling the function calculate for each pixel. The second part is how to calculate the new gray value for each pixel using the filter. The first part is so easy since you only need to scan the whole picture and call a function. The second part is more complex because you should pay attention to the edges of the picture. You should judge whether the subscripts of a pixel are out of range. To have different results of a same picture, you can only change to filter and don’t need to change the function filter2d and the function calculate.